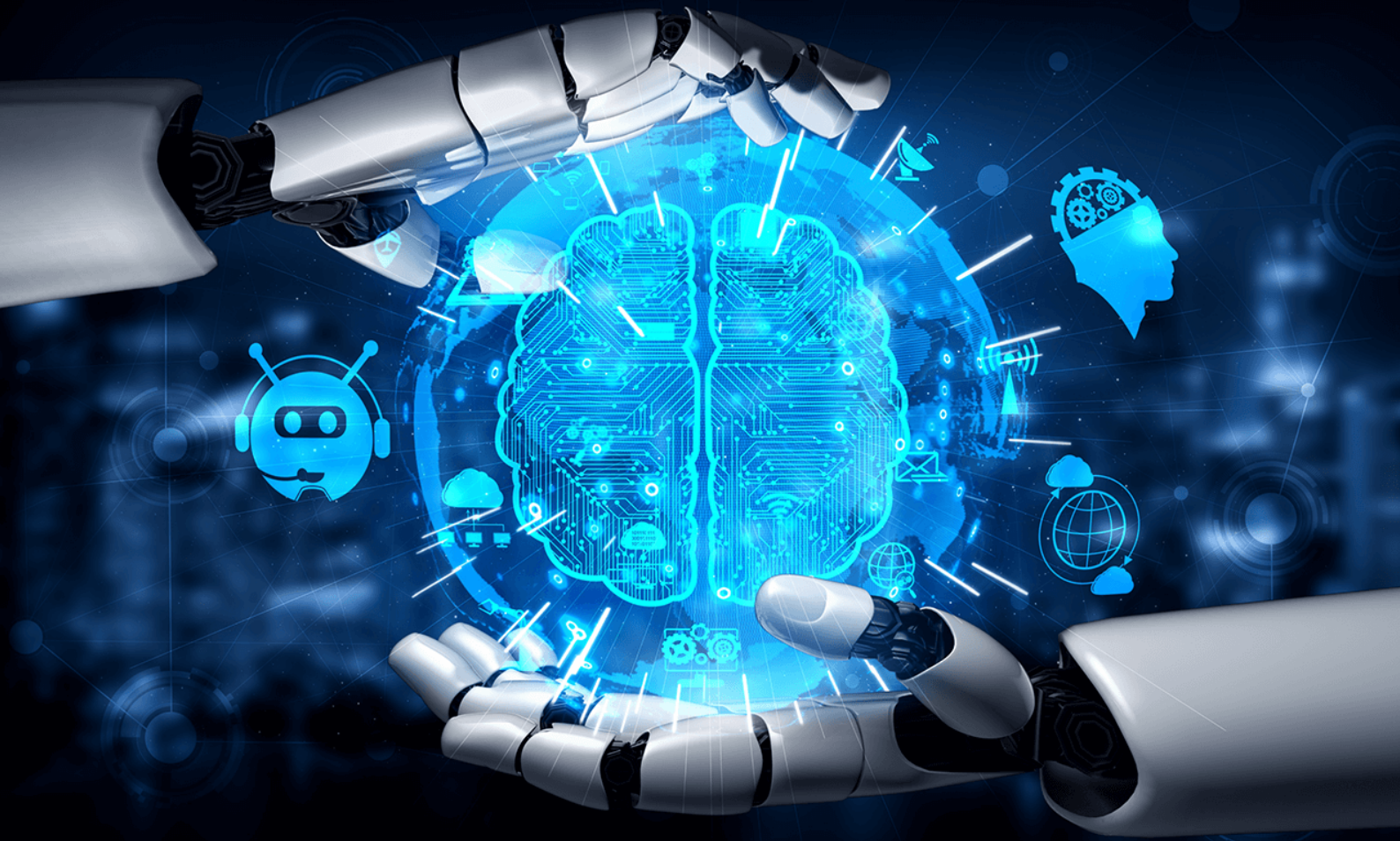


Dr. Ir. Agus Wibowo, M.Kom, M.Si, MM



KECERDASAN BISNIS (BI) DIDUKUNG AI :

Meningkatkan prediksi dan pembuatan keputusan
dengan ML (Machine Learning)



YAYASAN PRIMA AGUS TEKNIK



Dr. Ir. Agus Wibowo, M.Kom, M.Si, MM



KECERDASAN BISNIS (BI) DIDUKUNG AI :

Meningkatkan prediksi dan pembuatan keputusan
dengan ML (Machine Learning)



YAYASAN PRIMA AGUS TEKNIK

PENERBIT :

YAYASAN PRIMA AGUS TEKNIK
Jl. Majapahit No. 605 Semarang
Telp. (024) 6723456. Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

ISBN 978-634-7227-65-2 (PDF)



9

786347

227652

KECERDASAN BISNIS (BI) DIDUKUNG AI :
Meningkatkan prediksi dan pembuatan keputusan dengan ML
(Machine Learning)

Penulis :

Dr. Ir. Agus Wibowo, M.Kom, M.Si, MM

ISBN : 978-634-7227-65-2 (PDF)

Editor :

Dr. Joseph Teguh Santoso, S.Kom., M.Kom.

Penyunting :

Dr. Mars Caroline Wibowo. S.T., M.Mm.Tech

Desain Sampul dan Tata Letak :

Irdha Yuniarto, S.Ds., M.Kom

Penebit :

Yayasan Prima Agus Teknik Bekerja sama dengan
Universitas Sains & Teknologi Komputer (Universitas STEKOM)

Anggota IKAPI No: 279 / ALB / JTE / 2023

Redaksi :

Jl. Majapahit no 605 Semarang

Telp. 08122925000

Fax. 024-6710144

Email : penerbit_ypat@stekom.ac.id

Distributor Tunggal :

Universitas STEKOM

Jl. Majapahit no 605 Semarang

Telp. 08122925000

Fax. 024-6710144

Email : info@stekom.ac.id

Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara
apapun tanpa ijin dari penulis

KATA PENGANTAR

Dengan mengucapkan syukur kepada Tuhan Yang Maha Esa, penulis mempersembahkan buku berjudul "*Kecerdasan Bisnis (BI) Didukung AI*" sebagai wujud komitmen menghadirkan panduan lengkap bagi para profesional, akademisi, dan praktisi bisnis yang ingin memahami dan mengimplementasikan kecerdasan buatan dalam ranah Business Intelligence. Buku ini lahir dari pemahaman mendalam atas kebutuhan bisnis modern yang semakin bergantung pada data dan teknologi untuk meraih keunggulan kompetitif di era digital. Melalui buku ini, pembaca diajak untuk menyelami bagaimana AI mengubah metode BI konvensional menjadi lebih canggih, efektif, dan strategis.

Bab pertama membuka cakrawala dengan menjelaskan bagaimana AI telah mengubah lanskap BI secara signifikan. Pembaca diperkenalkan dengan berbagai kasus penggunaan AI mulai dari otomasi proses, kemudahan penggunaan, hingga peningkatan kemampuan peramalan dan pemanfaatan data tak terstruktur. Bab ini juga membahas pentingnya memahami intuisi dibalik AI dan pembelajaran mesin serta cara-cara memetakan ide penggunaan AI yang berpotensi besar memberikan dampak bisnis.

Bab kedua menekankan pentingnya menilai kelayakan proyek AI dengan mengutamakan aspek data, menggunakan kerangka kerja 4V untuk menilai kesiapan data, serta membahas keputusan strategis penting seperti memilih membuat atau membeli layanan AI. Pembahasan juga mencakup desain arsitektur dasar sistem AI, pertimbangan etis yang tak kalah penting, serta cara menyusun peta jalan prioritas penggunaan AI yang realistis dan berdampak.

Bab ketiga mendalami dunia pembelajaran mesin, membahas proses pembelajaran terawasi, berbagai algoritma populer, teknik pembelajaran mendalam, serta cara evaluasi model yang tepat. Bab ini juga mengingatkan tentang jebakan umum yang harus dihindari untuk mencapai implementasi pembelajaran mesin yang sukses.

Bab keempat membekali pembaca dengan pemahaman tentang pembuatan prototipe sebagai langkah esensial dalam menguji dan mengembangkan solusi AI yang aplikatif di bidang kecerdasan bisnis, lengkap dengan toolkit khusus yang direkomendasikan.

Bab kelima hingga kedelapan menyajikan analisis bertingkat bertenaga AI. Bab kelima mengupas analisis deskriptif berbasis AI dengan studi kasus pengkuarian dan peringkasan data menggunakan bahasa alami. Bab keenam menjelaskan analisis diagnostik untuk menghasilkan wawasan otomatis yang membantu identifikasi penyebab masalah. Bab ketujuh fokus pada analisis prediktif yang mengotomatiskan klasifikasi tugas, peningkatan prediksi KPI, dan deteksi anomali otomatis. Bab kedelapan menghadirkan analisis preskriptif dengan rekomendasi tindakan terbaik yang dihasilkan AI, meningkatkan ketepatan keputusan bisnis.

Pada bab kesembilan, pembaca diajak memanfaatkan data tak terstruktur seperti teks, dokumen, dan gambar menggunakan teknologi AI dalam mendapatkan wawasan baru, yang selama ini sulit diperoleh dengan metode BI tradisional.

Bab kesepuluh membahas langkah-langkah membangun dasbor analisis pelanggan yang interaktif dan bertenaga AI. Bab ini membahas pernyataan masalah, solusi yang diusulkan, persiapan data, hingga kesimpulan yang mengarah pada dasbor yang efektif untuk pengambilan keputusan pemasaran dan layanan pelanggan.

Bab kesebelas menutup buku dengan panduan praktis dari prototipe ke produksi, menyoroti perbedaan antara penemuan solusi dan pengiriman produk jadi, kriteria keberhasilan, serta langkah-langkah strategis untuk memastikan solusi AI dapat diimplementasikan dengan sukses dan memberikan dampak bisnis yang optimal.

Penulis mengucapkan terima kasih yang tulus kepada para ahli, praktisi, rekan penulis, institusi, serta semua stakeholder yang telah memberikan dukungan, masukan, serta inspirasi dalam penyusunan buku ini. Kontribusi mereka sangat berharga dalam mewujudkan buku ini menjadi karya yang bermanfaat luas dalam pengembangan kecerdasan bisnis berbasis AI.

Semoga buku ini menjadi sumber inspirasi, pengetahuan, dan panduan praktis yang dapat memperkuat kemampuan pembaca dalam mengintegrasikan AI ke dalam Business Intelligence guna menghadapi tantangan dan meraih peluang dalam dunia bisnis masa kini dan mendatang. Kritik dan saran yang membangun sangat kami nantikan demi peningkatan kualitas karya selanjutnya. Selamat membaca dan semoga buku ini memberikan manfaat besar dalam perjalanan transformasi bisnis Anda.

Selamat & Semangat Membaca...!!!

Semarang, November 2025

Penulis

Dr. Agus Wibowo, M.Kom, M.Si, MM

DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI.....	iv
BAB 1 MENCIPTAKAN NILAI BISNIS DENGAN AI	1
1.1 AI Mengubah Lanskap BI	1
1.2 Kasus Penggunaan AI Umum Untuk BI.....	4
1.3 Otomatisasi & Kemudahan Penggunaan.....	5
1.4 Peningkatan Peramalan & Prediksi.....	8
1.5 Memanfaatkan Data Tak Terstruktur	9
1.6 Memahami Intuisi Untuk AI & Pembelajaran Mesin.....	10
1.7 Memetakan Ide Kasus Penggunaan AI Untuk Dampak Bisnis	13
1.8 Ringkasan.....	16
BAB 2 MENILAI KELAYAKAN UNTUK PROYEK AI	17
2.1 Mengutamakan Data	17
2.2 Menilai Kesiapan Data Dengan Kerangka Kerja 4V	18
2.3 Keputusan Membuat Atau Membeli Layanan AI	21
2.4 Arsitektur Dasar Sistem AI	25
2.5 Pertimbangan Etis.....	30
2.6 Membuat Peta Jalan Kasus Penggunaan Yang Diprioritaskan.....	32
2.7 Kesimpulan	35
BAB 3 DASAR-DASAR PEMBELAJARAN MESIN	37
3.1 Proses Pembelajaran Mesin Terawasi	37
3.2 Algoritma Pembelajaran Mesin Populer	40
3.3 Pembelajaran Mendalam	44
3.4 Evaluasi Model Pembelajaran Mesin	46
3.5 Jebakan Umum Pembelajaran Mesin	52
3.6 Kesimpulan	56
BAB 4 PEMBUATAN PROTOTIPE	58
4.1 Pengertian Prototipe Dan Peran Penting	58
4.2 Pembuatan Prototipe Dalam Kecerdasan Bisnis	59
4.3 Toolkit Prototipe AI Untuk Buku Ini	61
BAB 5 ANALISIS DESKRIPTIF BERTENAGA AI	76
5.1 Kasus Penggunaan 1: Mengkueri Data Dengan Bahasa Alami	76
5.2 Kasus Penggunaan 2: Meringkas Data Dengan Bahasa Alami.....	87
5.3 Ringkasan.....	93
BAB 6 ANALISIS DIAGNOSTIK BERTENAGA AI	94
6.1 Kasus Penggunaan 3: Wawasan Otomatis.....	94

6.2	Ringkasan.....	110
BAB 7	ANALISIS PREDIKTIF BERTENAGA AI	111
7.1	Prasyarat.....	111
7.2	Kasus Penggunaan: Mengotomatiskan Tugas Klasifikasi.....	112
7.3	Kasus Penggunaan: Meningkatkan Prediksi KPI	145
7.4	Kasus Penggunaan: Deteksi Anomali Otomatis.....	167
7.5	Ringkasan.....	186
BAB 8	ANALISIS PRESKRIPTIF BERTENAGA AI	187
8.1	Kasus Penggunaan: Rekomendasi Tindakan Terbaik Berikutnya.....	187
8.2	Ringkasan.....	210
BAB 9	MEMANFAATKAN DATA TAK TERSTRUKTUR DENGAN AI	211
9.1	Kasus Penggunaan: Mendapatkan Wawasan Dari Data Teks	211
9.2	Kasus Penggunaan: Mengurai Dokumen Dengan AI	228
9.3	Kasus Penggunaan: Mendeteksi Objek Dalam Gambar	244
9.4	Kesimpulan	251
BAB 10	MEMBANGUN DASBOR ANALISIS PELANGGAN BERTENAGA AI	252
10.1	Pernyataan Masalah	252
10.2	Ikhtisar Solusi.....	252
10.3	Mempersiapkan Set Data	253
10.4	Kesimpulan	280
BAB 11	LANGKAH BERIKUTNYA: DARI PROTOTIPE KE PRODUKSI.....	282
11.1	Penemuan Vs. Pengiriman.....	282
11.2	Kriteria Keberhasilan Pengiriman Produk AI	283
11.3	Kesimpulan	290
DAFTAR PUSTAKA	292

BAB 1

MENCIPTAKAN NILAI BISNIS DENGAN AI

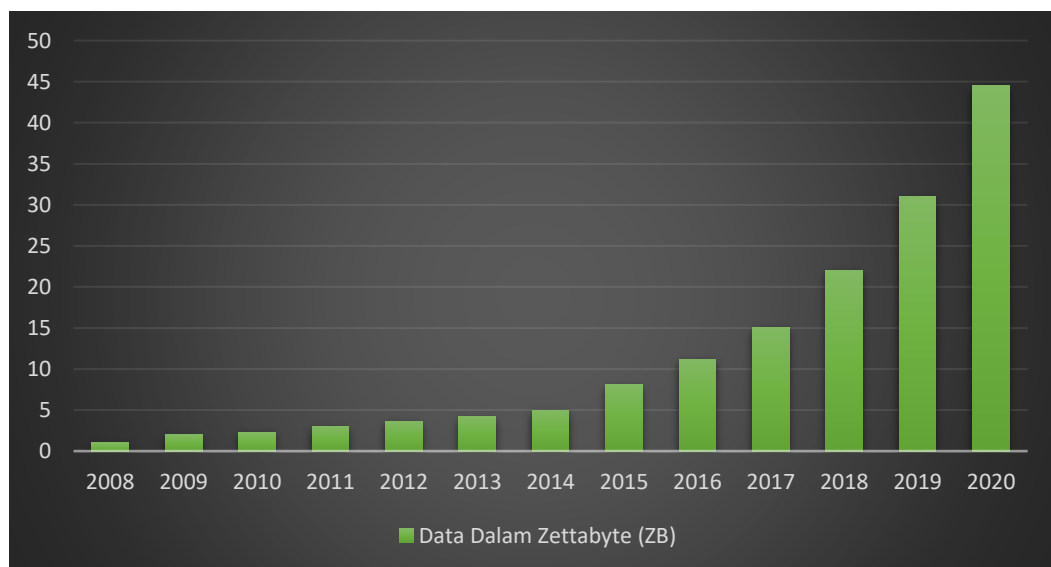
1.1 AI MENGUBAH LANSKAP BI

Selama 30 tahun terakhir, Kecerdasan Bisnis perlahan namun pasti telah menjadi kekuatan pendorong di balik budaya berbasis data di berbagai perusahaan. Perhatian telah bergeser ke arah ilmu data, pembelajaran mesin, dan kecerdasan buatan. Bagaimana ini bisa terjadi? Dan apa artinya ini bagi organisasi BI Anda?

Ketika kita menengok kembali ke awal era pertama sistem BI di tahun 1970an, kita melihat sistem teknis yang digunakan oleh para pakar TI untuk mendapatkan wawasan dari kumpulan data kecil (untuk skala saat ini). Menganalisis data merupakan hal baru sehingga wawasan paling mendasar pun tampak sangat menakjubkan.

Era kedua BI dimulai pada pertengahan tahun 2000an dan didominasi oleh analitik swalayan. Alat dan teknologi baru semakin memudahkan audiens non-teknis untuk memilah dan membagi data, menghasilkan visualisasi, dan mendapatkan wawasan dari sumber data yang semakin besar.

Sebagian besar perusahaan besar saat ini masih terjebak dalam fase kedua BI ini. Mengapa demikian? Salah satu alasannya adalah karena banyak upaya teknologi dalam beberapa tahun terakhir difokuskan pada upaya untuk secara teknis menangani pertumbuhan eksponensial data dasar yang seharusnya diproses dan menghasilkan wawasan oleh sistem BI. Kedua, peningkatan data seperti yang ditunjukkan pada Gambar 1.1 memicu semakin berkurangnya jumlah orang yang melek data dan mahir menangani kumpulan data berdimensi tinggi dengan perangkat yang tepat (yang, dalam hal ini, bukan Excel). Data tumbuh pada tingkat tahunan gabungan sebesar 40 persen, mencapai hampir 45 ZB pada tahun 2020.

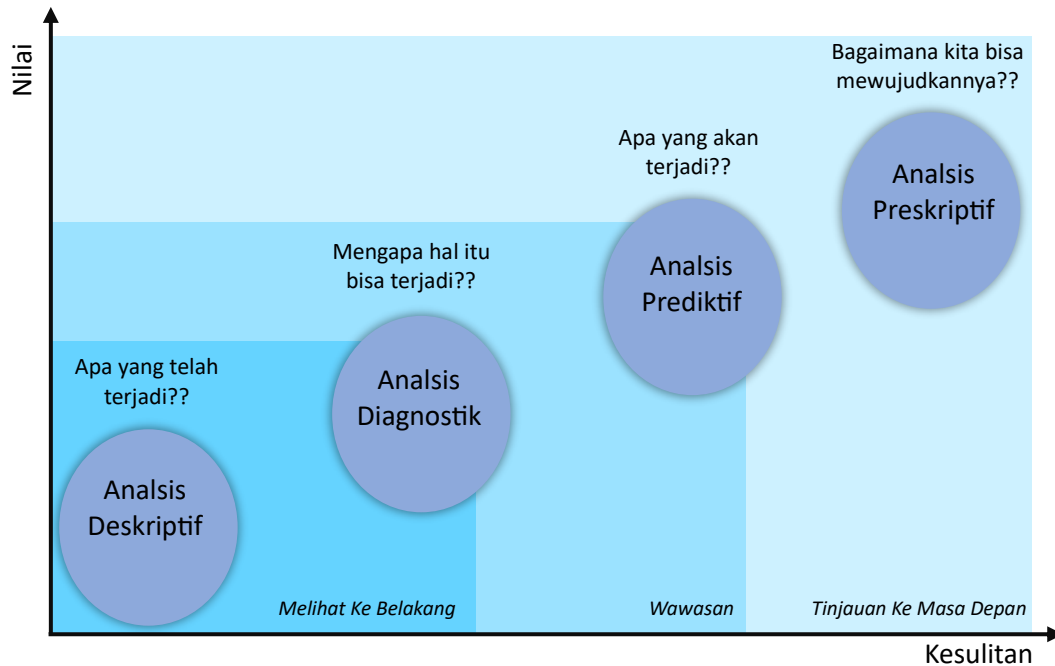


Dibandingkan dengan pasar konsumen, aplikasi AI masih kurang terlayani di ranah BI profesional. Hal ini kemungkinan disebabkan oleh fakta bahwa talenta AI dan BI berada di tim yang berbeda dalam sebuah organisasi, dan jika mereka bertemu, mereka kesulitan berkomunikasi secara efektif satu sama lain. Hal ini terutama disebabkan oleh fakta bahwa kedua tim biasanya berbicara bahasa yang berbeda dan memiliki prioritas yang berbeda. Artinya, para pakar BI biasanya tidak banyak berbicara tentang pelatihan dan pengujian data, dan ilmuwan data jarang membahas paket SSIS dan rutinitas ETL. Namun, kebutuhan adopsi AI dalam BI pasti akan meningkat karena tren yang sedang berlangsung berikut ini:

- **Kebutuhan untuk mendapatkan jawaban cepat dari data:** Agar tetap kompetitif, organisasi menuntut wawasan berbasis data untuk membantu mereka berkembang. Analisis Data kewalahan dengan pertanyaan untuk mengeksplorasi metrik ini atau itu atau memeriksa kumpulan data ini atau itu. Pada saat yang sama, kebutuhan pengguna bisnis untuk mendapatkan jawaban cepat dan mudah dari data meningkat. Jika mereka dapat bertanya kepada Google atau Alexa tentang harga saham perusahaan tertentu saat ini, mengapa mereka tidak dapat bertanya kepada sistem BI profesional mereka tentang angka penjualan kemarin?
- **Demokratisasi wawasan:** Pengguna bisnis telah terbiasa mendapatkan wawasan dari data dengan solusi BI swalayan. Namun, data itu sendiri seringkali terlalu besar dan terlalu kompleks untuk diserahkan kepada bisnis untuk analitik swalayan murni. Dimensionalitas yang tinggi dan terkadang ukuran data yang sangat besar menyulitkan, bahkan mungkin mustahil, bagi pengguna non-teknis untuk menganalisis data dengan alat yang familiar di komputer lokal mereka. Untuk terus mendemokratisasi wawasan di seluruh organisasi, dibutuhkan sistem BI yang mudah digunakan dan menampilkan wawasan secara otomatis kepada pengguna akhir.
- **Aksesibilitas layanan ML:** Seiring meningkatnya penggunaan AI di dalam organisasi, ekspektasi akan peramalan atau prediksi yang lebih baik pun meningkat. Hal ini bahkan lebih berlaku untuk BI; platform Low-Code atau No-Code semakin memudahkan penyediaan teknologi pembelajaran mesin bagi non-ilmuwan data dan memberikan tekanan pada tim BI untuk memasukkan wawasan prediktif ke dalam laporan mereka. Kemajuan yang sama dalam ilmu data juga diperkirakan akan terjadi di bidang BI, cepat atau lambat.

Untuk mendapatkan pemahaman yang lebih baik tentang bagaimana tim Business Intelligence dapat memanfaatkan AI, mari kita tinjau secara singkat model wawasan analitis yang diterbitkan oleh Gartner¹ seperti berikut pada Gambar 1.2:





Gambar 1.2. Jenis wawasan dan metodologi analitik

Fungsi inti setiap infrastruktur BI atau pelaporan adalah memberikan tinjauan ke belakang dan wawasan menggunakan analitik deskriptif dan diagnostik pada data historis. Kedua metode ini sangat penting untuk semua proses analitik lanjutan yang mengikutinya.

Pertama, sebuah organisasi perlu memahami apa yang terjadi di masa lalu dan apa yang mendorong peristiwa ini dari perspektif data. Ini biasanya disebut pelaporan dasar dengan beberapa fitur wawasan. Tingkat kesulitan dan kompleksitas teknisnya relatif rendah, tetapi nilai intrinsik informasi ini juga rendah.

Perhatikan contoh berikut: Dengan sistem pelaporan BI, Anda dapat dengan mudah memeriksa berapa banyak pelanggan yang berhenti berlangganan minggu lalu (deskriptif). Anda bahkan dapat memahami segmen pelanggan mana yang berhenti berlangganan dan tindakan apa yang mereka ambil atau tidak ambil sebelum mereka berhenti berlangganan (diagnostik).

Meskipun informasi ini sendiri cukup berguna, nilainya bagi organisasi masih terbatas. Wawasan itu sendiri tidak akan mencegah terjadinya churn di masa mendatang. Oleh karena itu, langkah logis selanjutnya dari sistem BI dan pelaporan kami adalah bertanya: Berapa banyak (dan yang mana) pelanggan yang akan berhenti berlangganan di masa mendatang, misalnya minggu depan?

Kami sekarang mengeksplorasi peristiwa masa depan dan memasuki bidang analitik prediktif. Karena kami meninggalkan ranah data historis, kompleksitasnya meningkat. Kami tidak dapat lagi memberikan wawasan dalam istilah biner benar atau salah. Sebaliknya, kami sekarang berbicara tentang probabilitas terjadinya peristiwa tertentu. Tentu saja, hal ini menyebabkan peningkatan kompleksitas dan kesulitan teknis, tetapi Anda diharapkan dapat mengenali nilai tambahnya di sini.



Mengetahui pelanggan mana yang kemungkinan akan berhenti berlangganan di masa mendatang memberi kita beberapa pilihan yang baik. Berikut beberapa contoh: Kami dapat mencoba menerapkan tindakan pencegahan yang diharapkan dapat mempertahankan pelanggan atau kami dapat mengantisipasi perubahan ini dengan perencanaan yang lebih baik.

Akhirnya, setelah kita mengetahui apa yang mungkin terjadi di masa mendatang, kita harus memperhatikan tindakan apa yang kita ambil untuk mencegah atau mengantisipasi prediksi kita. Selamat datang di fase analitik preskriptif. Berikut contoh bagaimana hal ini dapat terjadi: Kita memprediksi churn pelanggan untuk setiap ID Pengguna di sistem CRM kita. Model preskriptif akan menyarankan serangkaian tindakan yang harus diambil untuk setiap pelanggan guna mengurangi kemungkinan churn, seperti mengirim email atau menerapkan diskon harga tertentu.

Saat kita mengamati organisasi dengan ribuan pelanggan atau lebih, menjadi jelas bahwa untuk mengoptimalkan tugas-tugas ini dari perspektif makro, kita perlu mengandalkan otomatisasi pada tingkat mikro. Mustahil untuk memproses semua keputusan mini ini secara manual. Sebaliknya, keputusan yang bersifat universal berdasarkan firasat siapa pun yang duduk di depan komputer kemungkinan besar tidak akan mencapai hasil terbaik. Dan di sinilah AI dan BI bekerja sama dengan sempurna.

Pertimbangkan AI yang merekomendasikan kemungkinan churn bersama dengan saran tindakan terbaik berikutnya untuk setiap basis pelanggan. Kini, kita dapat memadukan informasi ini dengan metrik BI klasik seperti pendapatan historis pelanggan atau loyalitas pelanggan, yang memungkinkan kita membuat keputusan tentang tindakan-tindakan yang memiliki dampak bisnis tertinggi dan peluang keberhasilan terbaik. Hubungan antara AI dan BI dapat diringkas dengan baik dalam rumus berikut:

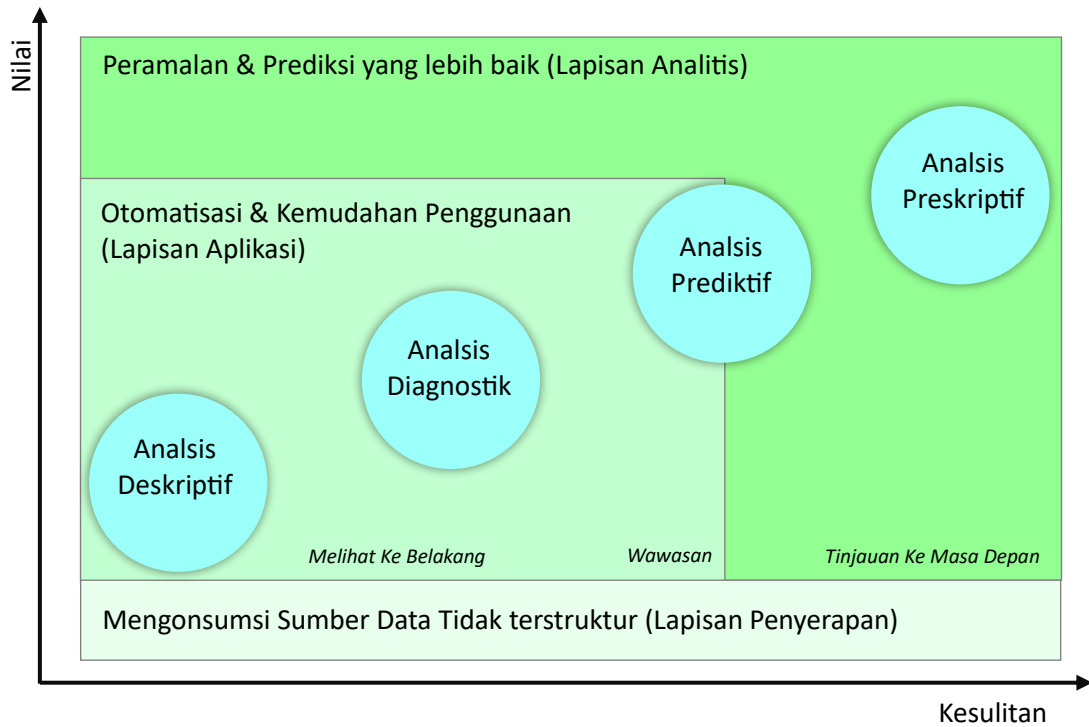
$$\text{Kecerdasan Buatan} + \text{Kecerdasan Bisnis} = \text{Kecerdasan Keputusan}$$

Aplikasi BI bertenaga AI yang paling efektif hanya ada jika kita memadukan pengambilan keputusan otomatis dan manusia. Kita akan membahas hal ini lebih lanjut di bab 2. Sekarang, mari kita lihat secara konkret bagaimana AI dapat secara sistematis membantu kita meningkatkan BI.

1.2 KASUS PENGGUNAAN AI UMUM UNTUK BI

AI biasanya dapat menambah nilai pada BI dalam tiga cara: (1) Mengotomatiskan wawasan dan membuat proses analisis lebih ramah pengguna, (2) menghitung prakiraan dan prediksi yang lebih baik, dan (3) memungkinkan sistem BI untuk mendorong wawasan bahkan dari sumber data yang tidak terstruktur. Gambar 1.3 memberikan gambaran umum tingkat tinggi tentang bagaimana area aplikasi ini dipetakan ke berbagai metode analisis.



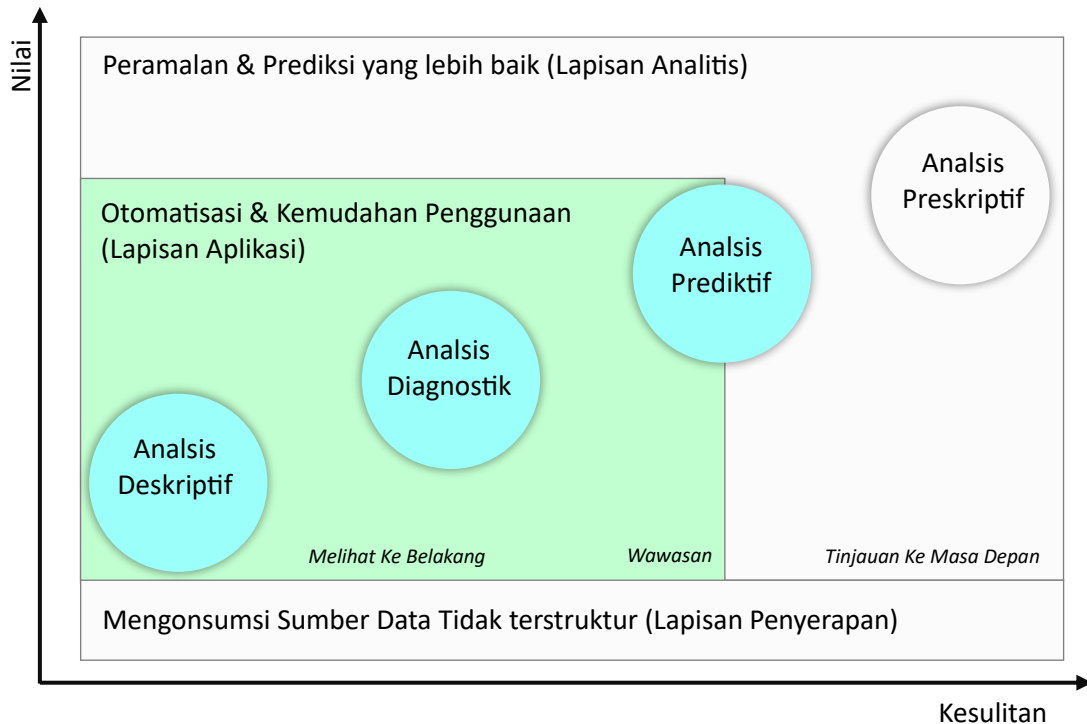


Gambar 1.3. Memetakan kapabilitas AI ke metode analitis
Mari kita telusuri area-area ini lebih detail.

1.3 OTOMATISASI & KEMUDAHAN PENGGUNAAN

Menjadikan alat BI itu sendiri lebih cerdas dan mudah digunakan akan membuatnya semakin mudah diakses oleh pengguna non-teknis, sehingga beban kerja para analis teralihkan. Kemudahan penggunaan ini biasanya didorong oleh otomatisasi yang terjadi di balik layar. Algoritme cerdas memungkinkan pengguna untuk menyisir tumpukan data dalam hitungan detik dan memunculkan pola atau wawasan menarik bagi pengguna bisnis atau analis. Seperti yang ditunjukkan Gambar 1.4, rutinitas ini bekerja dengan baik khususnya untuk tahap analitis deskriptif dan diagnostik, di mana korelasi menarik atau observasi yang tidak biasa dalam data ditemukan. Namun, otomatisasi dan kemudahan penggunaan juga memengaruhi tahap analitik prediktif, misalnya dengan semakin memudahkan pengguna untuk melatih dan menerapkan model ML kustom.





Gambar 1.4. BI Bertenaga AI: Otomatisasi & Kemudahan Penggunaan (Lapisan Aplikasi)

Ada satu hal penting yang perlu diperhatikan: Kemampuan AI pada tahap ini biasanya sudah terintegrasi ke dalam lapisan aplikasi, yaitu perangkat lunak BI Anda. Jadi, biasanya AI bukanlah sesuatu yang bisa Anda "tambahkan" begitu saja ke platform BI hanya dengan beberapa baris kode Python (berbeda dengan prediksi bertenaga AI dan membuka data tak terstruktur, yang akan kita bahas nanti di bab 6, 7, dan 8). Jika Anda menggunakan platform BI modern seperti Microsoft PowerBI atau Tableau, Anda akan menemukan fitur-fitur yang didukung AI ini di dalam perangkat tersebut, terkadang tersembunyi atau berjalan begitu mulus sehingga Anda bahkan tidak menyadari adanya AI di sini.

Berikut beberapa indikator di mana AI biasanya bekerja di balik layar untuk mempermudah pekerjaan Anda sebagai analis:

- **Anda dapat berinteraksi dengan data menggunakan Bahasa Alami**

Dengan menggunakan teknologi Proses Bahasa Alami yang didukung AI, mesin jauh lebih baik dalam menafsirkan dan memproses masukan tekstual dari pengguna. Misalnya, Anda ingin mengetahui hasil penjualan bulan lalu atau bagaimana penjualan di AS tahun lalu dibandingkan tahun ini. Anda dapat mengetikkan kueri berikut:

Bagaimana penjualan saya di Texas dalam 5 tahun terakhir? atau Penjualan Rp di Texas tahun lalu vs Penjualan Rp di Texas tahun ini.

Tidak perlu kode atau bahasa kueri yang rumit. Lapisan masukan seperti Tanya Jawab ini membuat BI jauh lebih mudah diakses oleh pengguna non-teknis dan juga lebih nyaman bagi analis yang tidak dapat mengantisipasi setiap pertanyaan yang mungkin

diajukan pengguna bisnis dengan laporan yang telah disiapkan. Sebagian besar pengguna akan cukup familiar dengan pendekatan ini karena mirip dengan menggunakan mesin pencari seperti Google. Terlepas dari apakah alat Tanya Jawab sudah terintegrasi dengan alat BI Anda atau tidak, tidak semua implementasi ini bekerja dengan baik. Faktanya, ada kompleksitas besar yang perlu dipecahkan di balik layar agar fitur-fitur ini bekerja dengan andal di lingkungan kerja produksi. Analisis harus melacak jenis pertanyaan yang diajukan pengguna bisnis dan memvalidasi apakah keluaran yang dihasilkan benar. Sinonim dan istilah khusus domain perlu didefinisikan untuk memastikan sistem dapat menginterpretasikan perintah pengguna dengan benar. Dan seperti semua sistem TI, hal-hal ini membutuhkan pemeliharaan yang konstan. Harapannya adalah sistem akan membaik dan upaya manual yang diperlukan di latar belakang akan berkurang seiring waktu.

- **Merangkum Hasil Analisis**

Meskipun bagan tampak cukup jelas, merangkum wawasan utama dalam satu atau dua baris merupakan praktik yang baik, untuk mengurangi risiko salah tafsir. Namun, siapa yang benar-benar suka menulis deskripsi yang tampaknya sudah jelas di bawah plot dalam laporan atau presentasi? Kebanyakan orang tidak, dan di situlah AI dapat membantu. Pemrosesan Bahasa Alami (NLP) yang didukung AI tidak hanya dapat membantu Anda menginterpretasikan masukan bahasa alami, tetapi juga dapat menghasilkan teks ringkasan untuk Anda berdasarkan beberapa data. Teks yang dihasilkan secara otomatis ini akan mencakup karakteristik deskriptif tentang data serta perubahan atau rentetan yang penting.

Berikut contoh keterangan plot yang dihasilkan secara otomatis dari Microsoft PowerBI:

Penjualan Rp untuk Texas meningkat selama 5 tahun terakhir yang tercatat dan mengalami periode pertumbuhan Penjualan terpanjang antara tahun 2010 dan 2014.

Seperti yang Anda lihat, cuplikan teks kecil yang dihasilkan AI ini dapat sangat memudahkan Anda sebagai analis dan menghemat banyak waktu dalam mengomunikasikan wawasan kepada pemangku kepentingan lainnya.

- **Penemuan pola otomatis dalam data**

Kita telah melihat bagaimana kemampuan Bahasa Alami dapat membantu Anda mendapatkan wawasan deskriptif dari data Anda secara efisien. Langkah logis berikutnya adalah mencari tahu mengapa pengamatan tertentu terjadi di masa lalu, seperti: Mengapa tepatnya penjualan di Texas meningkat begitu pesat? Dengan analitik diagnostik, Anda biasanya perlu menyisir kumpulan data Anda untuk mengeksplorasi perubahan yang signifikan dalam distribusi data yang mendasarinya. Dalam contoh ini, Anda mungkin ingin mencari tahu apakah suatu produk atau peristiwa tertentu mendorong perubahan secara keseluruhan. Proses ini dapat dengan cepat menjadi



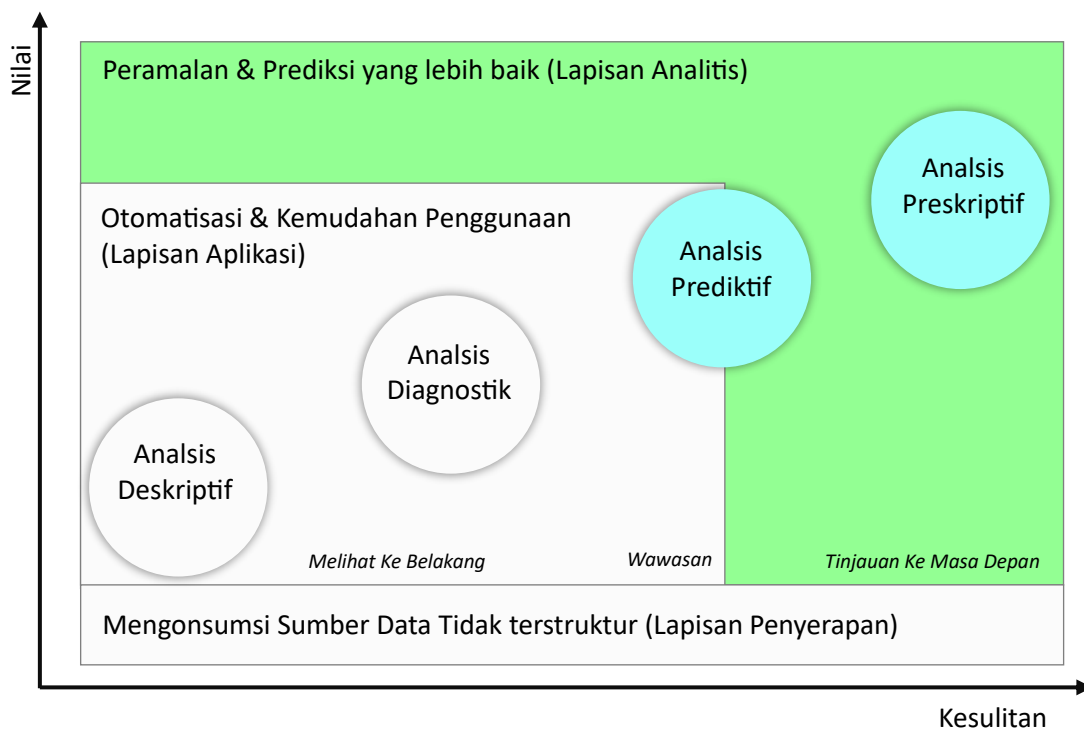
membosankan dan rumit. AI dapat membantu Anda mengurangi *Time to Insights* (TTS). Algoritma sangat baik dalam mengenali pola dasar dalam data dan menampilkannya. Misalnya, dengan perangkat berbasis AI seperti pohon dekomposisi atau analisis key influencer, Anda dapat dengan cepat mengetahui karakteristik mana dalam data Anda yang menyebabkan efek yang diamati secara keseluruhan dengan cepat.

Di Bab 4 dan 5, kita akan melihat tiga contoh konkret tentang bagaimana Anda dapat menggunakan kapabilitas berbasis AI di Power BI untuk mempermudah pekerjaan Anda sebagai analis data atau pengguna bisnis.

1.4 PENINGKATAN PERAMALAN & PREDIKSI

Meskipun analitik deskriptif dan diagnostik telah menjadi inti dari setiap sistem BI, keinginan yang melekat selalu ada, tidak hanya untuk memahami masa lalu tetapi juga untuk meramalkan masa depan. Seperti yang Anda lihat pada Gambar 1.5, kapabilitas yang disempurnakan dengan AI dapat mendukung pengguna akhir untuk menerapkan metode analitik prediktif dan preskriptif yang canggih untuk peramalan dan prediksi yang lebih baik berdasarkan data historis.

Hal ini akan menambah kompleksitas karena kita meninggalkan ranah data biner dari masa lalu dan memperkenalkan tebakan probabilistik tentang masa depan, yang secara alami mengandung banyak ketidakpastian. Pada saat yang sama, nilai prospektif meningkat: Jika kita hendak memprediksi masa depan, kita dapat membuat keputusan yang jauh lebih baik saat ini.



Gambar 1.5. BI Bertenaga AI: Peramalan & Prediksi yang Lebih Baik (Lapisan Analitis)



Nah, jika Anda pernah mendengar tentang metode statistik seperti regresi atau *Auto-Regressive Integrated Moving Average* (ARIMA) sebelumnya dan bertanya-tanya apa keunggulan AI, perhatikan dua aspek berikut:

- **AI dapat menghasilkan prakiraan yang lebih baik dengan lebih banyak data dan lebih sedikit pengawasan manusia**

AI pada intinya memanfaatkan teknik-teknik lama seperti regresi linier, tetapi pada saat yang sama menerapkannya ke ruang multidimensi yang dikombinasikan dengan algoritma stokastik seperti penurunan gradien untuk menemukan solusi optimal dalam ruang yang sangat kompleks dalam waktu yang sangat singkat tanpa memerlukan pengawasan manusia yang besar. Algoritma khusus untuk prediksi deret waktu dirancang untuk mengenali pola dalam data deret waktu yang lebih besar. AI mencoba mengoptimalkan prakiraan berdasarkan pemilihan fitur dan meminimalkan fungsi kerugian. Hal ini dapat menghasilkan prediksi yang lebih baik atau lebih akurat menggunakan horizon waktu yang pendek, atau mencoba memprediksi lebih akurat dalam jangka waktu yang lebih lama. Model non-linier yang lebih kompleks dapat menghasilkan hasil prediksi yang lebih terperinci dan pada akhirnya lebih baik.

- **AI dapat menghitung prediksi dalam skala besar untuk pengambilan keputusan yang optimal.** Memprediksi jumlah total pelanggan selama kuartal berikutnya memang bagus. Namun, yang lebih baik lagi adalah menghitung kemungkinan churn untuk setiap pelanggan dalam basis data Anda berdasarkan data terbaru. Dengan informasi ini, kita tidak hanya dapat mengetahui pelanggan mana yang kemungkinan akan churn bulan depan, tetapi kita juga dapat mengoptimalkan pengambilan keputusan, seperti: Dari semua pelanggan yang akan churn bulan depan, mana yang harus ditargetkan dengan kampanye pemasaran? Menggabungkan pembelajaran mesin dengan BI menciptakan proposisi nilai yang sangat besar bagi sebuah organisasi. Dan dengan kemajuan teknik baru seperti AutoML dan AI-as-a-Service, yang akan kita bahas lebih lanjut di Bab 3, organisasi dapat mengurangi hambatan karena kurangnya ilmuwan data atau praktisi ML untuk memanfaatkan potensi AI ini.

Kemampuan AI untuk peramalan yang ditingkatkan atau prediksi yang lebih baik dapat ditemukan baik sebagai bagian integral dari perangkat lunak BI yang ada (lapisan aplikasi), atau dapat diterapkan secara independen langsung pada tingkat basis data (lapisan analitis). Hal ini membuatnya selalu tersedia, apa pun alat BI yang Anda gunakan. Kita akan membahas cara menerapkan teknik ini di Bab 6 dan 7.

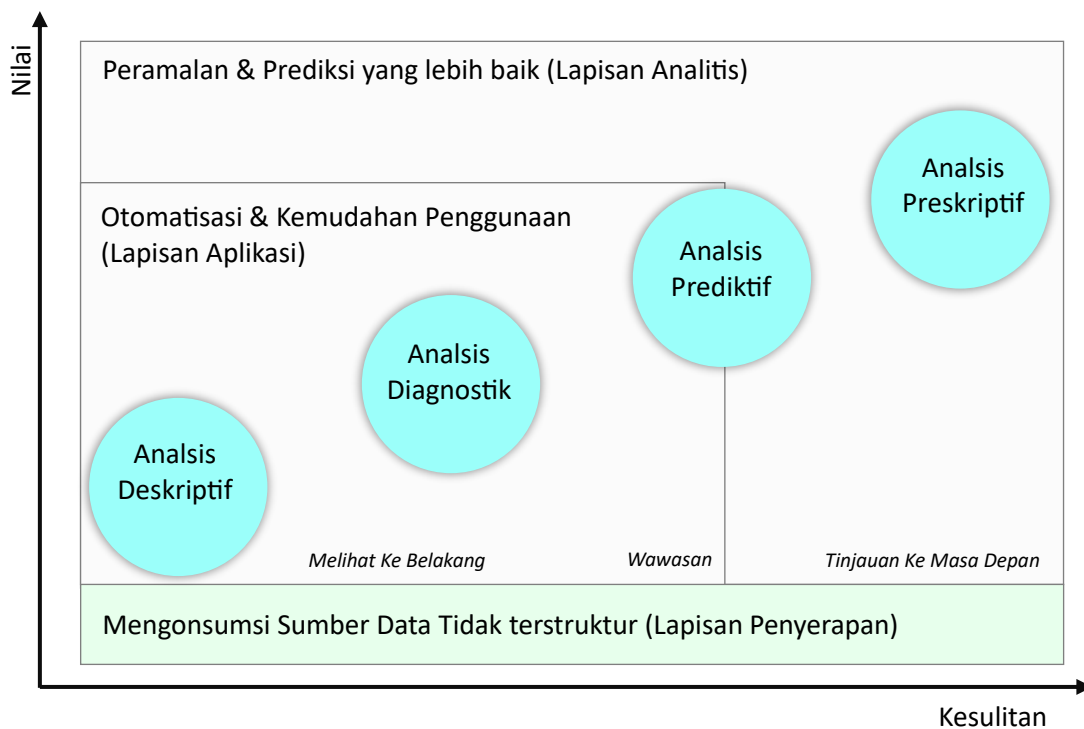
1.5 MEMANFAATKAN DATA TAK TERSTRUKTUR

Sistem BI biasanya bekerja dengan data tabular dari (sebaiknya) basis data relasional seperti gudang data perusahaan. Namun, dengan meningkatnya digitalisasi di semua saluran, kita melihat peningkatan dramatis dalam penggunaan data tak terstruktur dalam bentuk teks, gambar, atau berkas audio. Secara historis, bentuk-bentuk ini sulit dianalisis dalam skala besar oleh pengguna BI. AI hadir untuk mengubah hal itu.



AI dapat meningkatkan keluasan dan kedalaman data yang tersedia dan dapat dibaca mesin dengan menggunakan teknologi seperti visi komputer atau pemrosesan bahasa alami untuk mengakses sumber data baru yang sebelumnya belum dimanfaatkan. Data tak terstruktur seperti berkas teks mentah, dokumen PDF, gambar, berkas audio, dll. dapat diubah menjadi format terstruktur yang sesuai dengan skema tertentu, misalnya dalam bentuk tabel atau berkas CSV, dan kemudian dapat dikonsumsi dan dianalisis melalui sistem BI.

Karena ini adalah sesuatu yang terjadi pada tingkat penyerapan data, proses ini pada akhirnya akan memengaruhi semua tahapan platform BI (lihat Gambar 1.6). Dengan menggabungkan berkas-berkas ini ke dalam analisis kami, kami dapat memperoleh lebih banyak informasi yang berpotensi menghasilkan prediksi yang lebih baik atau pemahaman yang lebih baik tentang faktor-faktor pendorong utama. Bab 8 akan memandu Anda melalui beberapa contoh bagaimana hal ini bekerja dalam praktik.



Gambar 1.6. BI Bertenaga AI: Membuka Data Tak Terstruktur di Lapisan Ingesti

1.6 MEMAHAMI INTUISI UNTUK AI & PEMBELAJARAN MESIN

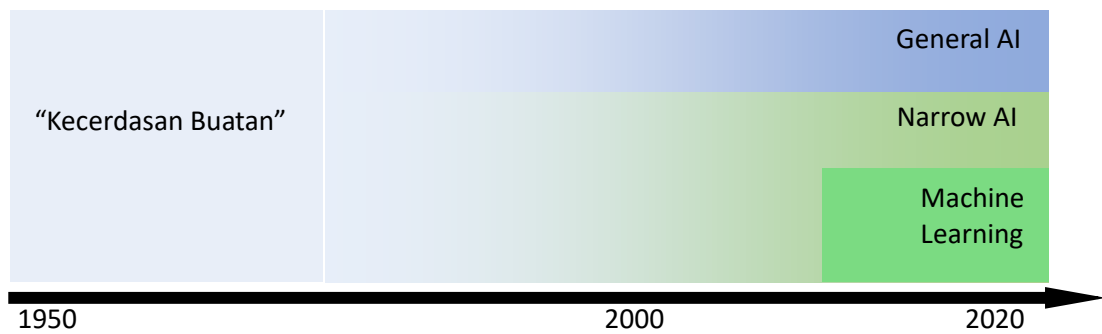
Kita telah banyak membahas bagaimana AI dapat digunakan dengan BI. Namun, untuk benar-benar membangun produk atau layanan bertenaga AI, kita perlu menggali lebih dalam dan memahami apa itu AI dan apa yang mampu (dan tidak mampu) dicapainya. Jadi, apa sebenarnya kecerdasan buatan itu? Jika Anda bertanya kepada 10 orang, Anda mungkin akan mendapatkan 11 jawaban. Untuk buku ini, penting untuk memiliki pemahaman umum tentang arti sebenarnya dari istilah ini.

Pertama-tama, mari kita akui bahwa istilah "Kecerdasan Buatan" bukanlah hal baru. Faktanya, istilah ini berasal dari laboratorium penelitian militer pada tahun 1950an. Sejak saat



itu, para peneliti telah mencoba berbagai pendekatan untuk mencapai tujuan agar komputer atau mesin dapat mereplikasi kecerdasan manusia.

Seperti yang ditunjukkan Gambar 1.7, dua bidang AI yang luas telah muncul sejak awal kemunculannya: AI Umum dan AI Sempit. AI Umum, atau AI Kuat, mengacu pada teknologi yang bertujuan untuk memecahkan masalah apa pun yang belum pernah dilihat atau dihadapi sistem sebelumnya, mirip dengan cara kerja otak manusia.



Gambar 1.7. Perkembangan Kecerdasan Buatan

AI Umum tetap menjadi topik penelitian yang hangat, namun masih cukup jauh: para peneliti masih belum yakin apakah AI Umum akan pernah tercapai. AI Sempit, di sisi lain, mengacu pada solusi yang cukup spesifik yang mampu memecahkan satu masalah yang terdefinisi dengan baik yang telah dirancang dan dilatih untuknya. AI Sempit telah mendorong semua terobosan AI yang telah kita lihat belakangan ini, baik dalam bidang penelitian maupun praktis atau bisnis.

Inti dari AI Sempit adalah satu pendekatan yang menonjol dalam hal dampak bisnis dan kemajuan pengembangan: pembelajaran mesin (ML). Bahkan, setiap kali saya membahas AI dalam buku ini, kita melihat solusi yang dimungkinkan melalui pembelajaran mesin. Itulah sebabnya saya akan menggunakan AI dan pembelajaran mesin secara bergantian dalam buku ini dan menganggap AI sebagai istilah yang cukup luas dengan makna yang cukup literal: AI adalah alat untuk membangun entitas (yang tampak) cerdas yang mampu menyelesaikan tugas-tugas spesifik, terutama melalui pembelajaran mesin. Setelah hubungan antara AI dan ML diharapkan menjadi sedikit lebih jelas, mari kita bahas apa sebenarnya pembelajaran mesin itu.

Pembelajaran mesin adalah paradigma pemrograman yang bertujuan menemukan pola dalam data untuk tujuan tertentu. Biasanya memiliki dua fase: Pembelajaran (pelatihan) dan inferensi (juga disebut pengujian atau prediksi). Ide inti di balik ML adalah kita menggunakan data historis untuk menemukan pola di dalamnya guna menyelesaikan tugas tertentu, seperti, menempatkan observasi ke dalam berbagai kategori, menilai probabilitas, atau menemukan kesamaan di antara berbagai item. Contoh kasus penggunaan umum untuk pembelajaran mesin adalah menganalisis data transaksi pelanggan historis untuk menghitung probabilitas individual dari kemungkinan churn pelanggan. Dengan inferensi, tujuan kita adalah menghitung prediksi untuk titik data baru berdasarkan semua yang kita pelajari dari



data historis tersebut. Untuk meningkatkan pemahaman Anda tentang pembelajaran mesin, mari kita bahas komponen inti definisi ML kita.

- Pembelajaran mesin adalah paradigma pemrograman:
Perangkat lunak tradisional dibangun dengan mengodekan aturan untuk menulis program tertentu. Jika Anda mengembangkan sistem dukungan pelanggan, Anda akan memikirkan semua logika yang seharusnya terjadi setelah pelanggan mengajukan tiket dukungan, misalnya memberi tahu agen dukungan melalui email. Anda akan mendokumentasikan semua aturan, memasukkannya ke dalam program, dan mengirimkan perangkat lunak. Namun, pembelajaran mesin membalikkan paradigma ini. Alih-alih mengodekan aturan secara permanen ke dalam sistem, Anda menyajikan contoh input dan output yang diinginkan (label) secukupnya, lalu membiarkan algoritma pembelajaran mesin menghasilkan aturan yang telah ditetapkan untuk Anda. Meskipun pengaturan ini tidak efektif untuk membangun sistem dukungan pelanggan, pengaturan ini sangat efektif untuk skenario tertentu di mana aturannya tidak diketahui atau sangat sulit dijelaskan. Misalnya, jika Anda ingin memprioritaskan tiket dukungan pelanggan berdasarkan berbagai fitur seperti teks tiket, jenis pelanggan, tanggal pembuatan tiket, dll., algoritma pembelajaran mesin dapat menghasilkan model prioritas untuk Anda hanya dengan melihat bagaimana tiket sebelumnya diprioritaskan. Alih-alih membuat logika if-then-else yang rumit, algoritma pembelajaran mesin akan menyelesaikannya dengan waktu komputasi dan sumber daya komputasi tertentu.
- Penemuan pola dalam data: Untuk menemukan pola yang berguna dalam data, tiga konsep penting bekerja sama: Model, algoritma, dan pelatihan. Model pembelajaran mesin adalah seperangkat aturan atau fungsi matematika yang akan menghitung nilai keluaran berdasarkan masukan data tertentu. Anggap saja sebagai tumpukan besar pernyataan if-then-else yang terbobot. Algoritma pembelajaran mesin menggambarkan proses komputasi yang harus diikuti mesin untuk mencapai model ini. Dan istilah pelatihan berarti melakukan iterasi berkali-kali pada kumpulan data yang ada untuk menemukan model terbaik yang memungkinkan untuk kumpulan data tertentu ini, yang menghasilkan kesalahan prediksi yang rendah dan generalisasi yang baik pada masukan data baru yang belum pernah dilihat sebelumnya, sehingga model tersebut dapat digunakan untuk tujuan tertentu.
- Tujuan khusus: Tugas pembelajaran mesin biasanya dikategorikan berdasarkan masalah yang ingin dipecahkan. Area utamanya adalah pembelajaran terawasi dan tak terawasi. Meskipun ini bukan buku tentang dasar-dasar pembelajaran mesin, kami akan membahasnya lebih detail di Bab 2.

Jika kita mempertimbangkan semua komponennya, tugas praktisi pembelajaran mesin dalam situasi dunia nyata adalah mengumpulkan data sebanyak mungkin tentang situasi yang diminati, memilih dan menyempurnakan algoritma untuk membuat model situasi tersebut, lalu melatih model tersebut agar cukup akurat dan bermanfaat.



Salah satu kesalahpahaman terbesar tentang AI dan pembelajaran mesin yang sering dimiliki para pemimpin bisnis adalah kenyataan bahwa AI dan pembelajaran mesin sangat sulit diimplementasikan. Meskipun merancang dan memelihara sistem pembelajaran mesin yang sangat spesifik dan berkinerja tinggi merupakan tugas yang sangat rumit, kita juga harus mengakui bahwa AI telah menjadi komoditas dan dikomersialkan sehingga bahkan non-pakar ML pun dapat membangun solusi ML yang berkinerja baik menggunakan pustaka kode yang ada dalam kode mereka atau menggunakan solusi tanpa kode/sedikit kode.

Di Bab 3, Anda akan mempelajari lebih lanjut tentang teknik-teknik ini sehingga Anda dapat mengimplementasikan solusi ML sendiri tanpa bantuan ilmuwan data atau insinyur pembelajaran mesin. Sebagai penutup bagian ini, ingatlah bahwa AI sebagai sebuah istilah bisa menakutkan dan mengintimidasi bagi orang-orang yang tidak benar-benar memahaminya. Kenyataannya, kita masih jauh dari Terminator dan AI pada umumnya.

Jika Anda ingin solusi AI diterima dan diadopsi secara lebih luas di dalam organisasi Anda, Anda perlu mengomunikasikan apa itu AI dengan bahasa yang ramah dan non-teknis. Membayangkan AI sebagai otomatisasi atau mampu menerapkan keputusan yang lebih baik berdasarkan pembelajaran sebelumnya akan membuat Anda cukup nyaman untuk menemukan kasus penggunaan yang berpotensi baik dan berbagi semangat tersebut dengan rekan kerja.

1.7 MEMETAKAN IDE KASUS PENGGUNAAN AI UNTUK DAMPAK BISNIS

Setelah Anda mempelajari lebih lanjut tentang AI dan bagaimana penerapannya pada BI, Anda mungkin sudah memiliki beberapa ide untuk menerapkan AI pada kasus penggunaan Anda sendiri. Untuk mengetahui kasus penggunaan mana yang paling potensial dan layak dikembangkan, kita akan melihat kerangka kerja pemetaan cerita yang dapat Anda gunakan untuk tujuan ini. Kerangka kerja ini terinspirasi oleh teknik manajemen proyek agile dan akan membantu Anda menyusun proses berpikir.

Ide inti dari kerangka kerja pemetaan cerita AI ini adalah untuk membandingkan implementasi suatu proses saat ini dengan bagaimana implementasi proses yang didukung AI akan terlihat. Teknik ini akan memberi Anda gambaran umum tingkat tinggi dan menyeluruh tentang apa yang akan berbeda, hal-hal apa yang perlu Anda ubah, dan yang terpenting, membantu Anda menyusun proses berpikir.

Pembuatan storyboard sangatlah mudah. Ambil selembar kertas kosong dan bagi menjadi tabel dengan empat kolom dan dua baris. Empat kotak di atas akan memetakan proses Anda saat ini dan kotak di bawah akan menggambarkan implementasi yang diantisipasi di masa mendatang. Beri nama kolom dari kiri ke kanan: Pengaturan, tindakan, hasil, dan hasil. Gambar 1.8 menunjukkan tampilan kertas Anda:



	<i>Aturan</i>	<i>Tindakan</i>	<i>Keluaran</i>	<i>Hasil</i>
Implementasi Saat Ini				
Implementasi Masa Depan				

Gambar 1.8. Templat Storyboard

Untuk membuat storyboard, Anda perlu mengisi kolom dari kiri ke kanan. Anda mulai dengan baris pertama, yang menguraikan bagaimana implementasi terkini dari suatu proses berjalan berdasarkan dimensi berikut:

- "Pengaturan" menjelaskan bagaimana proses dimulai, dan mencantumkan asumsi, sumber daya, atau kriteria awal Anda.
- "Tindakan" memuat semua tugas dan item tindakan yang dieksekusi oleh atau pada sumber daya yang diuraikan dalam pengaturan.
- "Keluaran" menjelaskan artefak aktual dari proses tersebut. Apa sebenarnya yang dihasilkan, dibuat, atau dimodifikasi?
- "Hasil" memuat dampak hasil terhadap bisnis dan/atau langkah selanjutnya untuk hasil tersebut. Misalnya, menampilkan laporan di dasbor merupakan hasil, tetapi tidak memiliki dampak apa pun. Dampak adalah apa yang terjadi berdasarkan informasi yang ditampilkan di dasbor dan siapa yang melakukannya.

Pada langkah berikutnya, Anda akan melakukan hal yang sama untuk implementasi yang diantisipasi di masa mendatang. Pada akhirnya, Anda akan mendapatkan perbandingan langsung antara pendekatan lama dan baru, yang akan memberikan Anda kejelasan lebih lanjut tentang bagaimana hal-hal akan berubah dan dampak apa yang mungkin ditimbulkan oleh perubahan tersebut.

Untuk memberikan konteks lebih lanjut tentang cara kerja latihan ini, Gambar 1.9 menunjukkan contoh storyboard untuk kasus penggunaan churn pelanggan kami.



Implementasi Saat ini	<i>Aturan</i>	<i>Tindakan</i>	<i>Keluaran</i>	<i>Hasil</i>
Implementasi Masa Depan				

Gambar 1.9. Contoh storyboard

Mari kita telusuri contoh storyboard kita. Kita mulai dari pojok kiri atas, dengan memaparkan pengaturan proses yang ada saat ini.

Saat ini, churn pelanggan dideteksi oleh staf penjualan yang mendapatkan umpan balik dari pelanggan yang sudah ada saat berbicara dengan mereka dalam rapat rutin, atau oleh staf dukungan pelanggan yang menerima umpan balik dari pelanggan bahwa beberapa hal tidak berjalan sesuai harapan mereka atau mereka menghadapi masalah lain.

Pada langkah selanjutnya, dukungan pelanggan atau bagian penjualan mencoba menyelesaikan masalah secara langsung dengan pelanggan, misalnya dengan memberikan bantuan orientasi. Hasil utama dari proses ini adalah dukungan pelanggan (semoga) menyelesaikan permasalahan dan kendala yang ada bagi pelanggan. Permasalahan tersebut dapat dilaporkan ke tingkat manajemen atau sistem manajemen keluhan. Dengan demikian, pelanggan diharapkan akan tetap menggunakan layanan yang ada setelah masalah terselesaikan.

Mari kita bandingkan hal ini dengan bagaimana implementasi berbasis AI akan terlihat, dimulai dari pojok kiri bawah dan berlanjut ke kanan. Dalam pengaturan kami, kami akan mengumpulkan data historis tentang bagaimana pelanggan menggunakan berbagai produk dan layanan, serta menandai pelanggan yang berhenti berlangganan dan yang tidak. Kami juga akan melibatkan staf penjualan dan layanan pelanggan untuk berbagi keahlian mereka dengan analis.

Tindakan kami selanjutnya adalah menganalisis data historis untuk menentukan apakah faktor-faktor pendorong utama churn pelanggan dapat diidentifikasi dalam kumpulan data. Jika demikian, kami akan mengembangkan model prediktif yang akan menghitung risiko churn individual untuk setiap pelanggan dalam basis data kami serta memberikan wawasan tentang kemungkinan terjadinya churn. Sebagai hasilnya, skor risiko churn dan alasan churn ini akan ditampilkan kepada bisnis, misalnya, melalui laporan di sistem CRM atau BI, di mana informasi ini dapat digabungkan dengan metrik lain, misalnya pendapatan pelanggan.

Dengan informasi ini, dukungan pelanggan kini dapat secara proaktif menghubungi pelanggan dengan risiko churn tinggi dan melihat apakah mereka dapat menyelesaikan



masalah atau menghilangkan hambatan sebelum pelanggan benar-benar menandai tiket dukungan atau berhenti berlangganan tanpa membuka tiket sama sekali. Hasilnya, tingkat churn pelanggan secara keseluruhan akan berkurang seiring berjalannya waktu karena organisasi dapat mengatasi alasan churn pelanggan dengan lebih baik dalam skala besar.

Dengan kedua peta cerita proses yang sudah ada dan yang baru, Anda akan merasa lebih percaya diri untuk menjelaskan seperti apa solusi AI yang memungkinkan, manfaat apa yang bisa dibawanya, dan apakah pendekatan baru tersebut masuk akal untuk menggantikan atau menggabungkannya dengan proses yang sudah ada.

Sebagai kesimpulan, ide papan cerita adalah menyediakan satu halaman sederhana untuk setiap kasus penggunaan yang secara intuitif membandingkan perbedaan dan manfaat dari solusi yang sudah ada dan yang baru. Ini akan membantu Anda menyusun proses berpikir dan menjadi titik awal yang solid dalam memprioritaskan kasus penggunaan AI.

1.8 RINGKASAN

Dalam bab ini, kita mempelajari bagaimana AI mengubah lanskap BI yang didorong oleh kebutuhan pengguna bisnis untuk mendapatkan jawaban yang lebih cepat dari data, meningkatnya permintaan akan wawasan yang demokratis, dan ketersediaan alat pembelajaran mesin yang dikomoditiasi secara keseluruhan yang lebih tinggi. Kami mengeksplorasi bagaimana AI dapat mendukung BI melalui otomatisasi dan kegunaan yang lebih baik, peramalan yang lebih baik, dan akses ke sumber data baru, sehingga memberdayakan orang untuk membuat keputusan yang lebih baik. Sekarang Anda seharusnya sudah memiliki pemahaman dasar tentang cara kerja AI dan ML serta kemampuan teknologi ini saat ini. Anda juga mempelajari kerangka kerja yang dapat Anda gunakan untuk menyusun proses berpikir dan menyusun ide untuk kasus penggunaan ML. Di bab selanjutnya, kita akan membahas lebih dalam tentang bagaimana sistem AI dirancang dan faktor-faktor apa saja yang perlu Anda pertimbangkan sebelum menerapkan teknologi ini dalam layanan BI perusahaan Anda.



BAB 2

MENILAI KELAYAKAN UNTUK PROYEK AI

2.1 MENGUTAMAKAN DATA

Proyek AI membutuhkan pola pikir yang berbeda dari proyek BI klasik. Sebagian besar proyek BI dilakukan dengan cara yang relatif mudah, seringkali mengikuti model waterfall tradisional: tentukan metrik yang ingin Anda tampilkan, rancang model data, integrasikan data, dan pastikan metrik tersebut berfungsi (cukup intensif). Ulangi jika perlu. Selesai.

Perbedaan utama dalam proyek AI adalah bahkan dalam kondisi ideal hasilnya sangat tidak pasti. Kita tidak tahu apakah model AI akan bekerja dengan data kita dan cukup baik untuk memberikan nilai hingga kita melihatnya beraksi.

Oleh karena itu, proyek AI biasanya memerlukan beberapa siklus iterasi yang lebih pendek dalam kerangka kerja proyek yang mirip agile seperti CRISP-DM. Cara umum untuk mendekati proyek AI dan memberikan ROI adalah dengan terlebih dahulu mengembangkan solusi minimum yang layak atau prototipe, dan baru setelah memvalidasi prototipe tersebut, meluncurkan proyek skala besar. Kita akan membahas topik ini secara lebih rinci di Bab 3 berikutnya.

Namun demikian, ada beberapa hal yang dapat Anda lakukan bahkan sebelum membangun prototipe: Pikirkan masalah apa yang ingin Anda selesaikan dan data apa yang akan Anda gunakan untuk melakukannya. Pada tahap ini, Anda tidak perlu khawatir tentang kerumitan teknis seperti basis data atau format data. Sebaliknya, saya sarankan Anda melihat data dari sudut pandang yang lebih luas dan menggunakan kata-kata sederhana untuk menggambarkan apa yang ingin Anda capai. Dalam contoh kita sebelumnya, mungkin terlihat seperti ini:

"Kami ingin menggunakan data historis terutama dari sistem CRM dan ERP untuk memprediksi churn pelanggan untuk bulan berikutnya."

Pertanyaan penting Anda tentang kasus penggunaan ini mungkin meliputi:

- Data apa yang Anda butuhkan untuk kasus penggunaan Anda?
- Apakah Anda memiliki data ini? Jika ya, apa sebenarnya yang Anda miliki (tabel, berkas media, teks, dll.)?
- Apakah Anda memiliki akses ke data ini?
- Siapa yang dapat mengaksesnya?
- Apakah perusahaan Anda memilikinya, atau Anda harus membelinya?
- Apakah Anda memiliki wewenang hukum untuk menggunakan data tersebut?

Semua pertanyaan ini masih belum ada hubungannya dengan kekhawatiran seperti "ini tidak ada di gudang data" atau "kami tidak dapat melakukan staging file PNG".

Jika Anda tidak dapat menjawab pertanyaan-pertanyaan penting tentang data ini, disarankan untuk tidak melanjutkan penilaian data lebih lanjut. Sebaliknya, akan lebih baik



untuk mengembangkan ide-ide untuk menjawab pertanyaan-pertanyaan ini. Bagaimana Anda bisa mendapatkan datanya? Dengan siapa Anda perlu berbicara? Dan seterusnya.

Namun, mari kita asumsikan Anda dapat menjawab pertanyaan-pertanyaan penting ini dan datanya tampaknya tersedia (setidaknya secara teori). Dalam hal ini, ada baiknya mengeksplorasi beberapa fitur data secara lebih rinci sebelum membuat prototipe Anda. Tidak ada standar emas untuk melakukan hal ini secara sistematis. Namun, salah satu kerangka kerja yang populer adalah kerangka kerja 4V, yang saya sukai karena kesederhanaannya. Kita akan mengeksplorasi kerangka kerja ini secara lebih rinci di bagian selanjutnya.

2.2 MENILAI KESIAPAN DATA DENGAN KERANGKA KERJA 4V

Kerangka kerja 4V membantu Anda lebih memahami karakteristik data tanpa terlalu teknis dalam empat dimensi: Volume, Variasi, Kecepatan, dan Kebenaran. Pendekatan ini akan memungkinkan Anda untuk mempertimbangkan kasus penggunaan mana yang lebih atau kurang realistis pada tahap tertentu. Sekali lagi, Anda sebaiknya hanya melakukan analisis ini setelah menjawab pertanyaan yang lebih mendasar, seperti: Apakah Anda benar-benar memiliki data tersebut, dan apakah Anda diizinkan untuk menggunakannya?

Volume menunjukkan berapa banyak data yang tersedia pada waktu tertentu. Anda dapat menginterpretasikannya setidaknya dalam dua cara: Pertama, jumlah observasi (contoh, baris), dan kedua, kedalaman informasi yang terkandung dalam data (atribut, kolom).

Berbagai jenis pembelajaran mesin membutuhkan jumlah data yang berbeda-beda, sehingga sulit untuk memperkirakan secara wajar berapa jumlah data yang "cukup". Namun, untuk memberikan gambaran kasar, lihat Tabel dibawah ini untuk persyaratan data untuk berbagai teknik pembelajaran mesin.

Kemampuan ML	Volume aturan praktis yang dibutuhkan
Regresi	~ 100 contoh
Klasifikasi	~ 1.000 contoh per kategori untuk dimensionalitas fitur sederhana
(10)	
Gambar klasifikasi/deteksi	~ 100 contoh per kategori untuk kategori yang jelas berbeda (kucing vs. anjing) ~ 1000 - 10.000 contoh untuk kategori yang lebih mirip (misalnya, ras anjing)
Teks / Bahasa Alami	~ 500 contoh per kelas untuk tugas seperti deteksi sentimen atau entitas

Dalam kebanyakan kasus, model menjadi lebih baik seiring bertambahnya data. Oleh karena itu, volume data yang lebih besar hampir selalu diinginkan. Namun, volume saja tidak serta merta menambah nilai.

Variasi: Sebagian besar data di organisasi Anda bahkan tidak disimpan dalam bentuk tabel. Teks mentah, gambar, atau berkas log, misalnya, termasuk dalam kategori yang sering



disebut sebagai data tak terstruktur. Data tak terstruktur berarti data tersebut tidak memiliki skema andal yang sesuai dengan tujuan Anda. Meskipun data terstruktur paling cocok untuk model pembelajaran mesin, data tak terstruktur mungkin memerlukan prapemrosesan ekstensif atau layanan AI untuk mengubahnya menjadi data terstruktur (seperti yang akan kita bahas di bab-bab selanjutnya). Namun, keragaman juga dapat diinterpretasikan dalam konteks data tabel dan mengacu pada seberapa baik data Anda merepresentasikan dunia nyata, termasuk kasus-kasus khusus, atau seberapa representatif data Anda secara umum. Misalnya, apakah Anda memiliki observasi pada semua label kelas dalam kumpulan data Anda?

Bergantung pada bagaimana Anda memandang istilah ini, keragaman mungkin merupakan sesuatu yang perlu dituju atau dihindari. Apa pun pilihannya, Anda perlu memilih pendekatan yang koheren dan sesuai dengan organisasi Anda, serta kerangka kerja evaluasi untuk menilai berbagai kasus penggunaan dengan cara yang sama.

Kecepatan adalah laju produksi dan aliran data melalui sistem Anda. Apakah Anda berurusan dengan data batch yang mungkin diperbarui sekali sehari? Atau apakah Anda berurusan dengan data streaming yang terus diperbarui? Kecepatan data memengaruhi dua hal: persyaratan teknologi dan pergeseran data. Data streaming cenderung memberikan tuntutan yang lebih tinggi pada infrastruktur Anda.

Data berkecepatan tinggi perlu dipantau lebih ketat untuk mengetahui varians data. Data berkecepatan tinggi yang tetap konsisten dapat membantu melatih sistem pembelajaran mesin dengan cepat karena membantu mengakumulasi volume. Namun, jika data ini berubah dengan cepat seiring waktu karena kecepatannya, akan sulit untuk mempertahankan pandangan yang konsisten tentang kebenaran dasar di setiap titik waktu.

Kebenaran: Seberapa akurat data sebagai representasi dari dunia nyata? Apakah data tersebut menunjukkan inkonsistensi, ketidaklengkapan, atau ambiguitas? Kebenaran berkaitan dengan apakah data Anda cukup baik untuk tujuan penggunaannya.

Dan dalam kasus pembelajaran mesin, hal pertama dan terpenting yang perlu diperhatikan adalah: apakah data tersebut mengandung kebenaran dasar, yaitu, apakah Anda memiliki label data? Jika data Anda tidak mengandung label, satu-satunya cara untuk mendapatkannya seiring waktu adalah dengan membeli layanan pelabelan atau menunggu proses bisnis menghasilkan data berlabel sebagai keluaran.

Menggabungkan 4V untuk Menilai Kesiapan Data

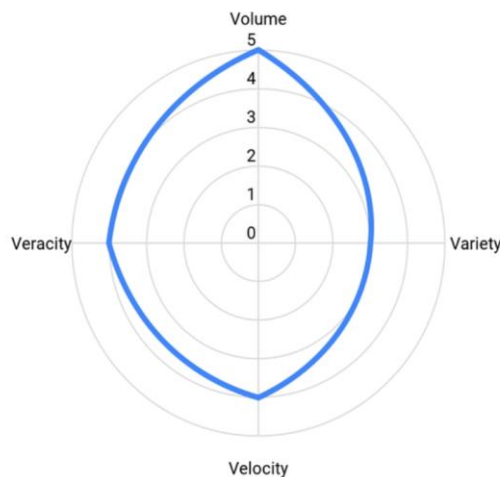
Dengan melihat secara kritis semua 4V secara bersamaan, Anda dapat mulai melihat beberapa kekuatan dan kelemahan relatif data Anda untuk kasus penggunaan tertentu. Tabel di bawah berikut merangkum beberapa pertanyaan kunci yang perlu Anda ajukan untuk setiap kategori dan skala yang dapat Anda gunakan untuk mengevaluasinya.

Dimensi	Pertanyaan	Skor yang mungkin dari... sampai ...	
Volume	Berapa banyak data yang Anda miliki?	Jumlah Kecil (1)	Jumlah Besar (5)
	Berapa banyak data yang akan diproduksi?		



Variasi	Apakah data tersebut mengandung cukup variasi untuk menangkap kejadian langka sekalipun? (1) Apakah data tersebut mengandung begitu banyak variasi sehingga mengandung terlalu banyak gangguan dan memerlukan pembersihan data yang besar?	Variasi yang Tidak Diinginkan Data yang kurang representatif (5)	Variasi yang Diinginkan (5) Data yang sangat representatif
Kecepatan (Velocity)	Seberapa sering sumber data yang relevan diproduksi atau diperbarui? Apakah sumber data diperbarui cukup sering untuk melatih ulang model cukup cepat guna mengurangi risiko pergeseran data?	Kecepatan rendah (1) Risiko penyimpangan data yang tinggi (1)	Kecepatan tinggi (5) Risiko penyimpangan data rendah (5)
Kebenaran (Veracity)	Seberapa akurat datanya? Seberapa lengkap datanya? Seberapa konsisten datanya? Apakah Anda memiliki label?	Kualitas data buruk (1) Tanpa label (1) Tidak cocok untuk digunakan (1)	Kualitas data tinggi (5) Semua contoh dengan benar diberi label (5) Cocok untuk digunakan (5)

Dengan skor-skor ini, Anda dapat membuat grafik netto seperti yang ditunjukkan pada Gambar di bawah untuk contoh kasus churn. Grafik tersebut dapat diinterpretasikan sebagai berikut:



Gambar 2.1. Placeholder keterangan gambar

Contoh: Bagan bersih untuk kasus penggunaan churn pelanggan

- Volume mendapat skor 5 dari 5: Kami memiliki basis pelanggan yang besar dengan banyak atribut terkait perilaku pembelian dan kami rasa ini akan cukup untuk melatih model pembelajaran mesin.
- Variasi mendapat skor 3 dari 5: Meskipun semua data kami berbentuk tabel dan seharusnya mudah dipahami untuk model pembelajaran mesin, kami perkirakan hanya



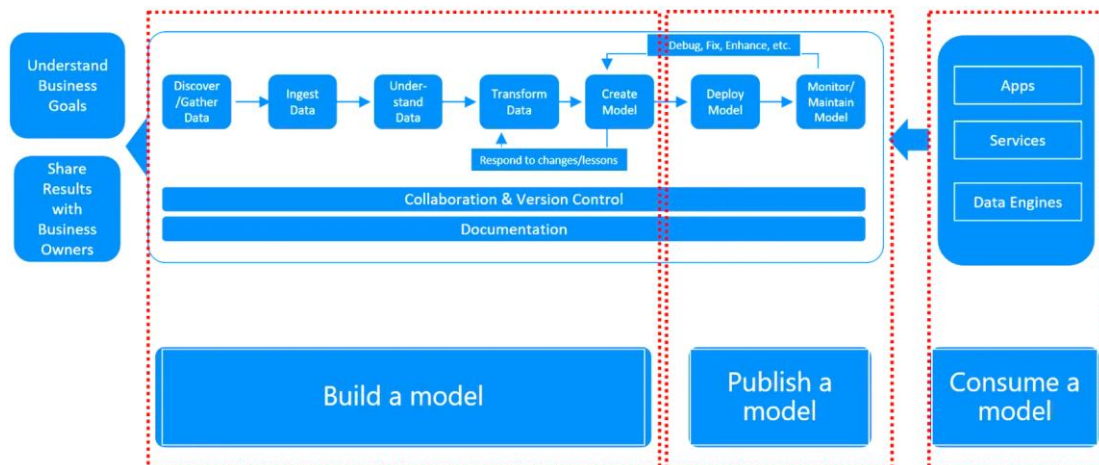
akan memiliki sedikit contoh pelanggan jangka panjang dan bernilai tinggi. Di sisi lain, mungkin ada banyak pelanggan baru yang informasinya tidak tersedia seperti preferensi pembelian karena data ini mungkin tidak ada di sistem CRM atau tersimpan di tempat lain.

- Velocity mendapat skor 4 dari 5: Data di sistem CRM dan ERP kami seharusnya cukup mutakhir dan diperbarui setidaknya setiap hari. Kami berasumsi bahwa hal ini cukup sering dilakukan untuk menghindari penyimpanan data.
- Veracity mendapat skor 4 dari 5: Kami dapat memperoleh label yang sebenarnya (Churned) dari data itu sendiri, sehingga tidak diperlukan layanan pelabelan. Lebih lanjut, kami berharap data dalam sistem CRM sebagian besar akurat karena adanya proses tata kelola data.

Bagan bersih ini akan menunjukkan kasus penggunaan yang layak dari perspektif data pada tahap ini dan memungkinkan kami untuk membandingkan pendekatan ini dengan kasus penggunaan lainnya.

2.3 KEPUTUSAN MEMBUAT ATAU MEMBELI LAYANAN AI

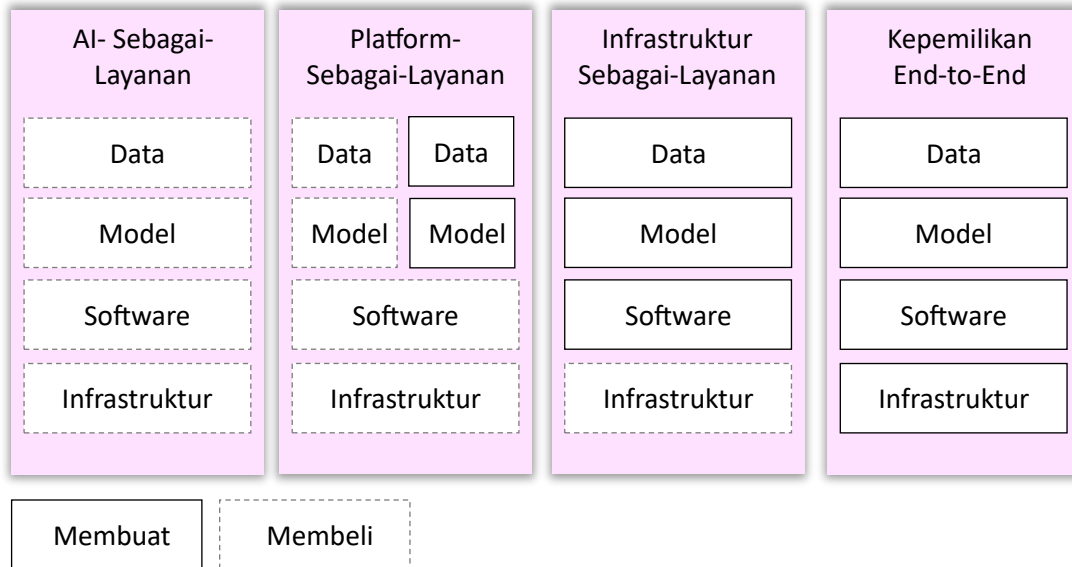
Membangun solusi AI adalah proses yang memakan waktu dan membutuhkan pemeliharaan berkelanjutan yang signifikan. Gambar di bawah menunjukkan contoh proses dengan semua langkah yang diperlukan untuk membangun solusi AI menyeluruh sendiri.



Gambar 2.2. Siklus hidup menguraikan tahapan utama yang biasanya dijalankan oleh proyek.

Ketika Anda memikirkan hal ini dan kompleksitas yang terlibat, terkadang solusi AI terbaik adalah tidak mengembangkan solusi AI sama sekali. Hal baiknya adalah Anda tidak harus melakukan semua langkah ini sendiri. Salah satu pilihannya mungkin mengembangkan beberapa komponen saja sendiri, pilihan lain adalah mengandalkan sepenuhnya pada layanan AI siap pakai. Pada akhirnya, seperti keputusan bisnis lainnya, solusi AI adalah pilihan membuat atau membeli.

Untuk memberikan gambaran umum tentang kemungkinan tahapan pola make-or-buy terkait AI, mari kita lihat berbagai tahapan yang dapat Anda pilih sebagai sebuah bisnis dan meninjau secara singkat kelebihan dan kekurangannya.



Gambar 2.3. Placeholder keterangan gambar

AI-sebagai-Layanan

Dengan AI-sebagai-Layanan (AlaaS), Anda menyewa layanan AI yang dikelola sepenuhnya, yang biasanya Anda bayar per penggunaan (misalnya, panggilan API). Misalnya, Anda mengirim gambar ke API dan menerima kembali dokumen JSON dengan label yang dikenali dalam gambar tersebut.

Keuntungannya adalah Anda tidak perlu mengembangkan atau memelihara apa pun. Anda hanya membayar untuk apa yang Anda gunakan, dan seringkali ada paket gratis sehingga Anda dapat mencoba layanan AI sebelum mengeluarkan uang untuk melihat seberapa baik kinerjanya dengan data Anda. Selain itu, penawaran AlaaS tidak memerlukan data pelatihan dari Anda.

Penyedia telah melatih layanan AI pada kasus penggunaan tertentu, seperti pengenalan wajah atau deteksi sentimen, dan Anda dapat langsung masuk ke mode inferensi. Karena terdapat beragam layanan AI untuk berbagai kasus penggunaan umum, relatif mudah untuk beralih di antara berbagai layanan, sehingga Anda dapat mencoba berbagai layanan tanpa risiko. Terakhir, layanan AI tidak memerlukan keahlian atau pengetahuan ML. Jika antarmuka pengguna grafis (GUI) disediakan, pengguna bisnis dapat menggunakan layanan ini tanpa pelatihan intensif.

Kelemahan terbesar dari layanan AlaaS adalah biasanya berupa kotak hitam. Artinya, Anda tidak tahu apa yang terjadi di dalam layanan. Anda hanya mendapatkan input dan output, tidak lebih. Selain itu, Anda tidak dapat memodifikasi AI sendiri. Jadi, jika Anda ingin mengkhususkan diri pada data Anda sendiri atau menggunakan label yang berbeda dari yang disediakan oleh layanan, biasanya tidak banyak yang dapat Anda lakukan. Karena alasan ini,



layanan AI siap pakai seperti Azure Cognitive Services, Google Cloud Vision, Amazon Rekognition, dll. biasanya berfungsi dengan baik untuk kasus penggunaan yang sangat umum (misalnya, mendeteksi sentimen dalam ulasan pengguna), tetapi layanan tersebut cepat memburuk untuk kasus penggunaan yang lebih spesifik.

Misalnya, Anda biasanya tidak dapat menggunakan layanan AI untuk mengekstrak hal-hal spesifik seperti nama produk dari data teks. Terakhir, Anda perlu memercayai perusahaan yang menyediakan layanan tersebut. Menghapus data untuk setiap kueri API mungkin tidak menguntungkan mereka, sehingga Anda umumnya kurang fleksibel dan kurang memiliki kendali atas data Anda.

Platform-sebagai-Layanan

Platform-sebagai-Layanan (PaaS) adalah solusi yang tepat bagi sebagian besar perusahaan. Dengan konsep PaaS, Anda mendapatkan akses ke platform pembelajaran mesin terkelola tempat Anda dapat melatih model Anda sendiri atau mengakses model yang sudah terlatih melalui marketplace. Biasanya, Anda membayar per lisensi, per penggunaan, atau kombinasi keduanya.

Jika Anda ingin membangun model dari awal, sebagian besar platform ML mendukung Anda dengan alat seperti AutoML. AutoML, atau pembelajaran mesin otomatis, menggunakan pembelajaran mesin itu sendiri untuk menemukan model yang paling sesuai untuk kumpulan data tertentu, sehingga bahkan praktisi yang kurang berpengalaman pun dapat dengan cepat membuat model pembelajaran mesin awal yang baik hanya dengan sekali klik. Kita akan membahas pendekatan ini nanti di buku ini. Tentu saja, dengan konsep seperti AutoML, Anda perlu membawa data pelatihan Anda sendiri.

Jika Anda tidak memiliki atau hanya memiliki sedikit data pelatihan, sebagian besar platform ML menyediakan marketplace tempat Anda dapat membeli model AI siap pakai untuk berbagai kasus penggunaan. Ini disebut model yang sudah terlatih. Dalam beberapa kasus, Anda dapat menyesuaikan model ini dengan data Anda, yang membutuhkan data pelatihan jauh lebih sedikit daripada melatih model dari awal. Misalnya, alih-alih membutuhkan ribuan gambar untuk membangun model visi komputer dari awal, penyempurnaan hanya membutuhkan sekitar selusin gambar dari setiap kategori untuk menghasilkan hasil yang baik, tergantung pada kasus penggunaannya.

Banyak penawaran PaaS juga mendukung Anda dengan fitur yang lebih canggih seperti layanan pelabelan khusus atau dukungan untuk menerapkan dan memantau model Anda. Dalam kebanyakan kasus, platform ML berjalan di cloud (contohnya termasuk Azure ML Studio, Amazon Sagemaker, Google Vertex AI), tetapi ada juga banyak vendor yang mendukung lingkungan on-premise, seperti Datarobot atau H2O.

Berikut adalah keuntungan dan kerugian utama PaaS:

1. Kelebihan:

- Kemudahan penggunaan: Dalam kebanyakan kasus, Anda tidak perlu khawatir tentang infrastruktur karena disediakan oleh vendor PaaS.
- Kecepatan: Sangat menghemat waktu karena tidak perlu melakukan pengaturan dan pemeliharaan infrastruktur. Onboarding lebih cepat untuk non-pakar ML.



2. Kekurangan:

- Mahal dalam skala besar: Jika Anda perlu melatih banyak model atau kasus penggunaan yang berbeda, biayanya bisa cepat membengkak.
- Vendor lock-in: Anda terkunci pada satu platform untuk melatih dan menerapkan model Anda. Bagaimana jika Anda perlu beralih?

Saya rasa bagi sebagian besar perusahaan yang ingin bereksperimen dengan pembelajaran mesin, ini adalah cara yang tepat untuk memulai. PaaS memberi Anda infrastruktur yang terawat baik dan mudah digunakan, yang tetap cukup fleksibel untuk beradaptasi dengan kebutuhan unik Anda. Anda dapat bereksperimen dengan cepat tanpa harus berurusan dengan terlalu banyak biaya overhead.

Infrastruktur sebagai Layanan

Dalam skenario Infrastruktur sebagai Layanan (IaaS), Anda menyewa layanan penyimpanan, komputasi, dan jaringan dari penyedia cloud dengan sistem bayar per penggunaan. Anda dapat menerapkan kerangka kerja pembelajaran mesin apa pun yang Anda inginkan. Pendekatan IaaS masuk akal bagi perusahaan yang bersedia berinvestasi besar dalam pembelajaran mesin dan talenta perangkat lunak, tetapi tidak ingin terbebani dengan perangkat keras infrastruktur. Pendekatan ini biasanya direkomendasikan ketika organisasi Anda telah mencapai tingkat kematangan tertentu dalam mengembangkan dan memelihara model pembelajaran mesin.

Pro dan kontra IaaS secara singkat:

1. Kelebihan:

- Fleksibilitas tinggi: Mudah ditingkatkan dan diturunkan skalanya. Hal ini memungkinkan penyesuaian sumber daya Anda dengan cepat sesuai kebutuhan bisnis.
- Menghindari ketergantungan: Jika Anda hanya menggunakan infrastruktur, Anda dapat beralih penyedia dengan relatif mudah jika perlu untuk mengurangi biaya atau meningkatkan kualitas layanan.
- Fokus pada ilmu data: Dalam skenario IaaS, Anda memiliki kendali penuh atas data dan perangkat lunak Anda. Oleh karena itu, Anda dapat fokus pada pengembangan algoritma pembelajaran mesin Anda tanpa terganggu oleh masalah terkait infrastruktur sekaligus memiliki kemampuan kustomisasi penuh.

2. Kekurangan:

- Dibutuhkan pakar cloud: Mengelola infrastruktur cloud membutuhkan keahlian. Jika Anda tidak memiliki bakat, ini akan menjadi hambatan yang krusial.
- Dibutuhkan pakar ML: Karena Anda tidak bergantung pada platform ML yang sudah jadi, Anda perlu melakukan integrasi perangkat lunak sendiri, yang memakan waktu dan membutuhkan keahlian.

IaaS sangat cocok untuk perusahaan yang ingin berfokus pada ilmu data dan pengembangan perangkat lunak sekaligus memiliki kemampuan untuk meningkatkan dan menurunkan skala dengan cepat.



Kepemilikan End-to-End

Dengan kepemilikan sistem end-to-end, Anda bertanggung jawab atas segala hal mulai dari perangkat keras infrastruktur, jaringan, kerangka kerja perangkat lunak, basis data, dll. Fakta menarik: Banyak perusahaan tanpa sadar memulai dari sini karena mereka sudah menjalankan infrastruktur on-prem dan meminta ilmuwan data mereka melakukan berbagai hal di notebook Jupyter di komputer mereka. Pendekatan ini akan segera menemui hambatan jika Anda mencoba menerapkannya dalam skala besar tanpa manajemen platform yang tepat.

Kepemilikan menyeluruh biasanya merupakan skenario yang paling kompleks, dan juga yang paling intensif sumber daya jika Anda melakukannya untuk lebih dari beberapa kasus penggunaan. Untuk membenarkan biaya overhead, perusahaan Anda membutuhkan setidaknya satu kasus penggunaan AI yang sangat kuat yang akan menentukan keberhasilan atau kegagalan bisnis (misalnya, mengemudi otonom di Tesla). Kepemilikan menyeluruh wajib jika Anda ingin membuat POC awal tetapi tidak diizinkan untuk memindahkan data ke cloud atau kebijakan perusahaan Anda tidak mengizinkan Anda mengadopsi platform ML.

Saat memilih pendekatan infrastruktur AI, ingatlah bahwa tidak semua perusahaan perlu memiliki seluruh proses dari awal. Tingkat yang Anda butuhkan bergantung pada kasus penggunaan, anggaran, dan yang terpenting, bakat yang tersedia. Solusi pembelajaran mesin bukanlah proses satu kali, melainkan membutuhkan pengembangan dan pemeliharaan berkelanjutan.

Pikirkan kembali bagian mana yang ingin Anda fokuskan dan bagaimana Anda akan mendapatkan manfaat darinya. Dalam contoh analisis churn pelanggan kita, kemungkinan besar kita akan memilih platform ML terkelola karena kita perlu membangun model kita sendiri untuk kasus penggunaan ini (tidak ada AI yang dapat melakukan analisis churn pelanggan sebagai layanan tanpa pelatihan) dan data dari sistem sumber kita kemungkinan dapat dengan mudah diunggah ke platform ML.

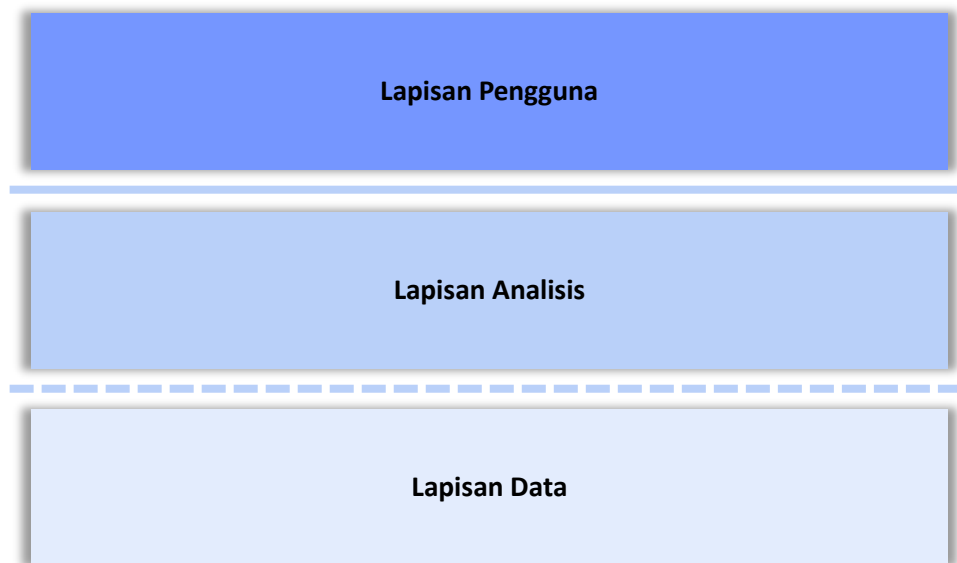
2.4 ARSITEKTUR DASAR SISTEM AI

Setelah kita membahas aspek kelayakan kasus penggunaan AI dalam hal pengaturan data dan infrastruktur, mari kita bahas secara singkat arsitektur umum sistem AI modern. Saya berjanji kita tidak akan membahas terlalu teknis di sini. Namun, jika Anda memahami bagaimana kasus penggunaan AI dibangun, Anda akan berada dalam posisi yang lebih baik untuk menilai kelayakan teknis dari kasus penggunaan AI tertentu.

Pada tingkat teratas, kita biasanya dapat mengidentifikasi tiga lapisan tempat solusi AI dibangun: Lapisan data, lapisan analitik, dan lapisan pengguna, seperti yang ditunjukkan pada Gambar di bawah ini.

- Lapisan Pengguna
- Lapisan Analisis
- Lapisan Data





Gambar 2.4. Placeholder keterangan gambar

Kerangka kerja tingkat tinggi ini akan cukup bagi kita untuk memahami

1. masalah apa yang ingin kita selesaikan untuk pengguna dan bagaimana hasilnya (Lapisan Pengguna),
2. kapabilitas ML apa yang akan kita gunakan untuk ini (Lapisan Analisis),
3. dan data apa yang kita butuhkan dengan pemrosesan yang mana (Lapisan Data).

Arsitektur teknis yang mendasarinya tentu saja jauh lebih kompleks, tetapi abstraksi tingkat tinggi ini akan sangat membantu Anda.

Nantinya di buku ini, Anda akan menemukan arsitektur tingkat tinggi ini untuk kasus penggunaan yang akan kita bahas, sehingga Anda dapat dengan cepat melihat cara kerja solusinya. Izinkan saya menjelaskan secara singkat setiap lapisan, artinya, diikuti dengan dua contoh arsitektur.

Lapisan Pengguna

Lapisan pertama adalah lapisan pengguna dan biasanya merupakan langkah yang baik untuk memulai dari sini. Di sinilah kita menentukan masalah apa yang ingin kita selesaikan dan, yang lebih penting, seperti apa solusi kita nantinya. Apakah kita akan meng-host

API yang dapat diakses pengguna? Atau apakah hasil akhir kita berupa dasbor yang menampilkan prakiraan spesifik di BI Anda? Kejelasan hasil akhir akan memberikan cakupan yang dibutuhkan untuk kasus penggunaan AI kita, meskipun hasil yang diinginkan dapat berubah selama proses pengembangan. Berikut beberapa elemen populer yang dapat digunakan di lapisan pengguna:

- API
- Aplikasi Web
- Integrasi Dasbor/BI
- File (Excel, CSV, ...)
- Konektor Lama (SAP, perangkat lunak perusahaan khusus, ...)



Lapisan Data

Setelah pendahuluan bab ini, mari kita lihat data yang ingin kita gunakan untuk mencapai hasil yang diusulkan di lapisan pengguna sebelum kita membahas detail teknis analisisnya. Lapisan data berisi komponen yang menjelaskan sumber atau sistem sumber data yang ingin kita gunakan. Beberapa contoh komponennya adalah:

- Basis Data
- File Media
- Gudang Data
- API
- File (CSV)
- Input Pengguna
- Konektor Lama (SAP, perangkat lunak perusahaan khusus, ...)
- ...

Selain itu, kita dapat menggunakan lapisan data untuk menjelaskan secara garis besar apa yang ingin kita lakukan dengan data tersebut sebelum menerapkan layanan ML.

- Langkah-langkahnya meliputi:
- Melabeli data
- Menggabungkan data
- Mengagregasi data
- Membentuk ulang data
- Mensimulasikan data
- ...

Seberapa jauh hal ini akan berjalan bergantung pada kasus penggunaan Anda. Untuk beberapa kasus penggunaan, penalaran tentang sumber data dan manipulasinya akan menjadi bagian utama, sementara untuk kasus lainnya, fokus mungkin lebih bergeser ke lapisan Pengguna atau Analisis.

Lapisan Analisis

Setelah kita menentukan cakupan masalah dan data yang kita miliki, saatnya beralih ke bagian tengah arsitektur kita, tempat keajaiban terjadi dengan kapabilitas ML kita. Perlu dicatat bahwa menggabungkan kapabilitas ML tidak selalu diperlukan di sini. Sebagai gantinya, Anda juga dapat mempertimbangkan untuk mengisi lapisan tengah sepenuhnya tanpa ML untuk mendapatkan baseline pertama. Adakah aturan bisnis atau heuristik yang dapat Anda terapkan? Apakah ada model lain yang sudah ada? Jika ada, integrasikan dan gabungkan sebagai "arsitektur dasar" Anda. Setelah Anda membuat ini, Anda dapat mulai mengganti komponen dasar Anda dengan fungsi yang didukung ML. Dengan cara ini, Anda akan memiliki indikasi yang jelas tentang di mana pendekatan yang didukung ML Anda kemungkinan akan memberikan nilai tambah.

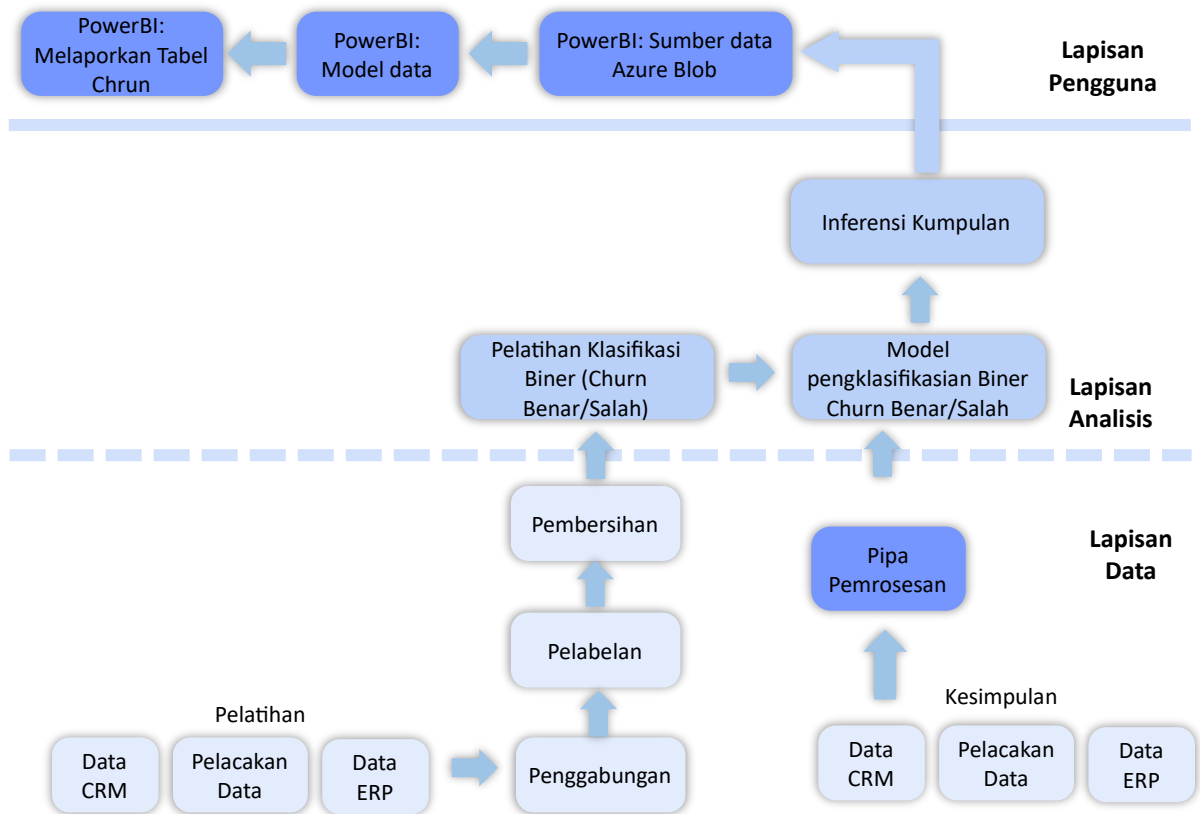
Perlu diketahui bahwa arsitektur Anda tidak selalu satu arah, yang berarti data dapat mengalir bolak-balik di antara berbagai lapisan. Anda akan memahaminya jika kita melihat sekilas dua contoh:



Contoh arsitektur pertama ditunjukkan pada Gambar di bawah dan mengilustrasikan kemungkinan pengaturan untuk kasus penggunaan churn pelanggan.

Mari kita bahas arsitektur ini dalam urutan terbalik, dimulai dengan lapisan pengguna. Dari kotak di sudut kiri atas "Power BI: Tabel Laporan Churn", kita dapat melihat bahwa hasil akhirnya adalah laporan di Microsoft Power BI yang menampilkan tabel churn yang diprediksi. Untuk melakukan ini, kita akan menyediakan model data (berbagai tabel dan relasinya) yang dapat dilihat pengguna di PowerBI (oleh karena itu, ini masih berada di lapisan pengguna).

Power BI mengakses data dari penyimpanan Azure Blob. Beralih ke lapisan analisis: data dari penyimpanan Azure Blob berasal dari pekerjaan inferensi batch yang menggunakan pengklasifikasi biner untuk membuat label churn benar/salah untuk data yang telah melalui alur pra-pemrosesan dari sistem CRM, ERP, dan pelacakan web.



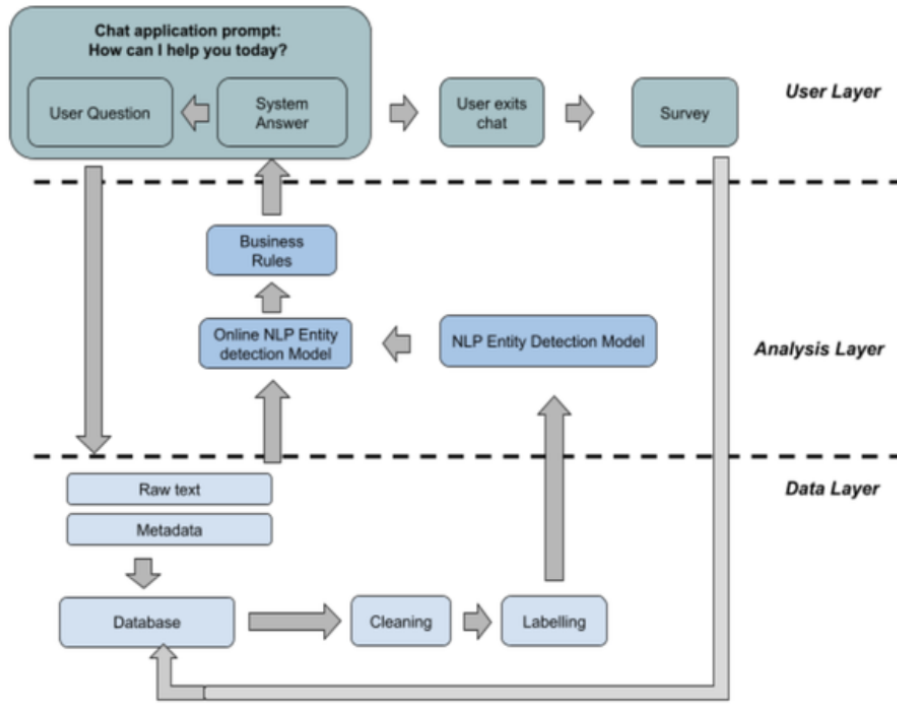
Gambar 2.5. Placeholder keterangan gambar

Untuk mendapatkan model tersebut, kami menggunakan sumber data yang sama untuk pelatihan, menggabungkan, memberi label, dan membersihkannya, lalu menggunakan data ini untuk melatih pengklasifikasi biner yang melakukan prediksi. Tanpa perlu menulis satu baris kode pun atau bahkan menyentuh satu sumber data pun, kita dapat menggunakan arsitektur ini untuk membahasnya dengan para pemangku kepentingan bisnis dan mengajukan pertanyaan seperti:

- Apakah solusi ini memang seperti yang diharapkan?
- Seberapa rumitkah solusi ini menurut kita?

- Apa produk atau prototipe minimum yang layak yang dapat kita luncurkan?
- Apakah itu sesuai dengan pemetaan cerita yang telah kita buat?

Permintaan aplikasi obrolan: Apa yang bisa saya bantu hari ini?



Gambar 2.6. Placeholder keterangan gambar

Mari kita lihat arsitektur lainnya.

Gambar di bawah menunjukkan contoh kasus penggunaan chatbot di mana data bergerak bolak-balik di antara berbagai lapisan. Di kiri atas, Anda dapat melihat bahwa interaksi pengguna terakhir seharusnya berupa aplikasi obrolan di situs web, yang biasanya memulai percakapan dengan perintah seperti "Ada yang bisa saya bantu hari ini?". Pengguna merespons dengan sebuah pertanyaan, yang kami simpan dalam basis data beserta beberapa metadata (stempel waktu, informasi pengguna, jika diizinkan). Sejalan dengan penyimpanan data, kami memasukkan teks pengguna ke dalam model NLP waktu nyata untuk pengenalan entitas yang mengekstrak entitas dari masukan pengguna. Entitas ini dapat berupa topik umum seperti "penagihan" atau "masalah teknis" atau entitas yang lebih detail seperti nama produk.

Dalam kasus chatbot yang sangat sederhana ini, entitas digunakan untuk memicu aturan bisnis. Ini bisa berupa respons dengan jawaban yang telah ditulis sebelumnya terkait entitas yang disebutkan oleh pelanggan atau mengajukan pertanyaan klarifikasi. Pengguna merespons dengan masukan lain dan proses berlanjut hingga pengguna meninggalkan obrolan. Ketika pengguna meninggalkan obrolan, mereka diundang ke survei dan dimintai umpan balik. Umpan balik ini disimpan dalam basis data kami dan digabungkan dengan riwayat obrolan sebelumnya yang telah kami kumpulkan.



Kami membersihkan data ini dan memberi label, yaitu, kami memberi label pada entitas dalam percakapan historis. Kami kemudian dapat menggunakan label ini untuk melatih ulang model pengenalan entitas NLP dan meningkatkan solusi yang ada. Seperti yang Anda lihat dari kasus penggunaan ini, arsitekturnya dapat bersifat dua arah dan data dapat mengalir bolak-balik di antara berbagai lapisan ini.

Kerangka kerja tingkat tinggi ini telah banyak membantu saya dalam menentukan cakupan proyek AI dan mendiskusikan ide-ide kasus penggunaan dengan para pemangku kepentingan teknis dan non-teknis. Saya harap kerangka kerja ini juga akan membantu Anda saat Anda memikirkan kasus penggunaan AI Anda berikutnya.

2.5 PERTIMBANGAN ETIS

Topik terakhir yang akan kita bahas untuk menilai kelayakan kasus penggunaan AI kita adalah konsep etika. Etika AI adalah bidang yang berkembang pesat, dan bahkan jika Anda secara intuitif tidak akan memikirkannya ("Saya tidak ingin menyakiti siapa pun!" Saya mendengar Anda berkata), tetap penting untuk mempertimbangkannya, setidaknya secara prinsip.

Di bagian ini, saya ingin memberikan kerangka kerja ringkas untuk membantu Anda mengidentifikasi area mana kasus penggunaan AI tertentu mungkin kritis atau kurang kritis. Mari kita bahas ini segera: Ketika kita berbicara tentang etika AI, kita berbicara tentang layanan AI yang merupakan alat yang sangat terspesialisasi, bukan entitas yang sadar diri yang memikirkan konsekuensi tindakan mereka (dan dengan demikian terbebas dari tanggung jawab atas diri mereka sendiri).

Oleh karena itu, alih-alih menilai layanan AI itu sendiri, kita perlu menilai kasus penggunaan atau aplikasi yang diadaptasi oleh teknologi AI. Dan orang-orang yang bertanggung jawablah yang pada akhirnya memikul tanggung jawab dan harus dimintai pertanggungjawaban.

Dalam hampir setiap kasus penggunaan AI, terdapat potensi pertimbangan etika yang harus ditangani baik sebelum maupun sesudah solusi diimplementasikan. Dalam beberapa kasus, masalah ini begitu serius sehingga kasus penggunaan tersebut tidak dapat dilanjutkan. Dalam beberapa kasus, Anda perlu melakukan perubahan pada kasus penggunaan agar aman secara moral. Dan ada kasus penggunaan di mana risiko diskriminasi terhadap orang dalam skala besar sangat rendah.

Intinya adalah sebagai pemimpin bisnis atau seseorang yang mencoba mengembangkan solusi AI, Anda bertanggung jawab untuk menjalani proses pemikiran ini sebelum mencoba prototipe pertama Anda di lapangan. "Kritis" dalam hal ini memiliki dua dimensi:

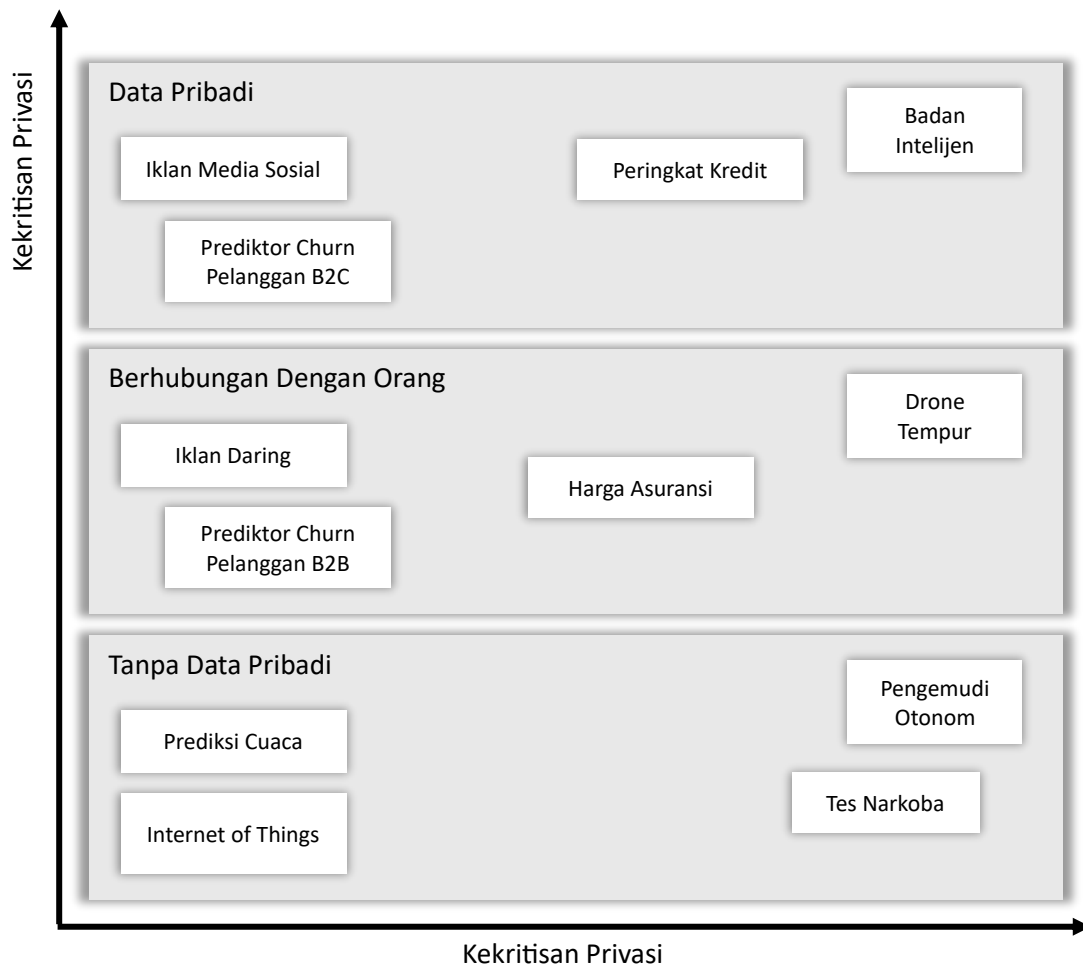
Dimensi pertama, "Kritisitas Etis", berkisar dari aplikasi yang tidak berdampak pada manusia lain hingga aplikasi yang melibatkan hidup dan mati satu atau lebih manusia.

Dimensi kedua, "Kritisitas Privasi", berkisar dari data mesin non-personal hingga data yang berkaitan dengan pribadi, hingga data pribadi yang sangat sensitif berdasarkan kategori privasi.



Kerangka kerja ini ditunjukkan pada Gambar di bawah. Beberapa aplikasi disajikan dan diklasifikasikan di sini sebagai contoh. Aplikasi di pojok kiri bawah diagram ini tampaknya paling tidak berbahaya, karena data tanpa referensi pribadi sudah cukup dan hasilnya hanya berdampak kecil atau bahkan tidak berdampak sama sekali terhadap orang lain. Banyak aplikasi *Internet of Things* (IoT) termasuk dalam kategori ini, misalnya.

Bayangkan kendali mutu di pabrik, di mana sistem berbasis AI menentukan apakah suatu benda kerja memenuhi kriteria mutu. Ketika menangani barang-barang yang berisiko kecil bagi manusia (misalnya papan ketik komputer), persyaratan mutu produsen untuk AI dapat ditegakkan tanpa pertimbangan etika yang ekstensif.



Gambar 2.7. Sumber: AI Ethics, Springer (Zwingmann, Gärtner)

Namun, ketika produk memiliki dampak signifikan terhadap kehidupan manusia, kasus penggunaannya sangat penting secara etika, meskipun tidak ada data pribadi yang digunakan. Ambil contoh pengujian obat medis. Pada suatu titik dalam eksperimen medis, data sangat anonim sehingga mustahil untuk mengaitkan titik data dengan individu tertentu.

Bayangkan sebuah kasus penggunaan di mana Anda mengevaluasi pengembangan layanan AI baru yang memprediksi keberhasilan suatu obat pada kelompok pasien tertentu, alih-alih melakukan analisis statistik manual. Meskipun data yang dibutuhkan untuk ini tidak



mencakup data pribadi, keseluruhan kasus penggunaan ini sangat penting secara etika, karena hasil akhirnya (yang dihasilkan menggunakan AI) dapat menentukan hidup dan mati seseorang.

Mari kita lihat sisi ekstrem lainnya di kanan atas diagram. Ini adalah area berisiko tertinggi yang memerlukan pertimbangan etika paling cermat. Area ini mencakup kasus penggunaan yang secara langsung menyesuaikan atau membangun data pribadi dan berdampak langsung pada kehidupan orang-orang dalam skala besar. Layanan intelijen termasuk dalam kategori ini, tetapi begitu pula peringkat kredit.

Jika algoritma Anda bermasalah dalam produksi dan Anda melayani jutaan pelanggan, Anda mencegah sejumlah besar orang mengakses sumber daya keuangan atau layanan lain seperti perumahan, belanja, dll. yang bergantung pada data peringkat tersebut. Mengambil contoh kasus penggunaan churn pelanggan kita, kita dapat menempatkan kasus penggunaan ini di bagian tengah kiri diagram jika merupakan prediktor churn B2B. Dalam kasus ini, kami akan menghitung probabilitas churn untuk akun (perusahaan), alih-alih pelanggan individu (individu).

Hal ini menghasilkan data yang relevan dengan individu, tetapi tidak secepat, misalnya, contoh churn B2C di mana kami menghitung skor churn pada tingkat individu. Bagaimanapun, dampak prediktor churn kami terhadap individu relatif kecil. Dalam kasus terburuk, beberapa individu akan menerima tawaran personal yang berkinerja buruk (atau bahkan tidak ada sama sekali) dengan insentif untuk tetap bertahan.

Saat mengembangkan kasus penggunaan AI, Anda harus menyadari keduanya jumlah data pribadi yang dibutuhkan untuk mewujudkan kasus penggunaan tersebut, dan kedua, dampak solusi Anda terhadap individu. Kedua kategori ini akan membantu Anda memprioritaskan kasus penggunaan mana yang membutuhkan keterlibatan etis yang lebih mendalam dan mana yang tampaknya kurang penting.

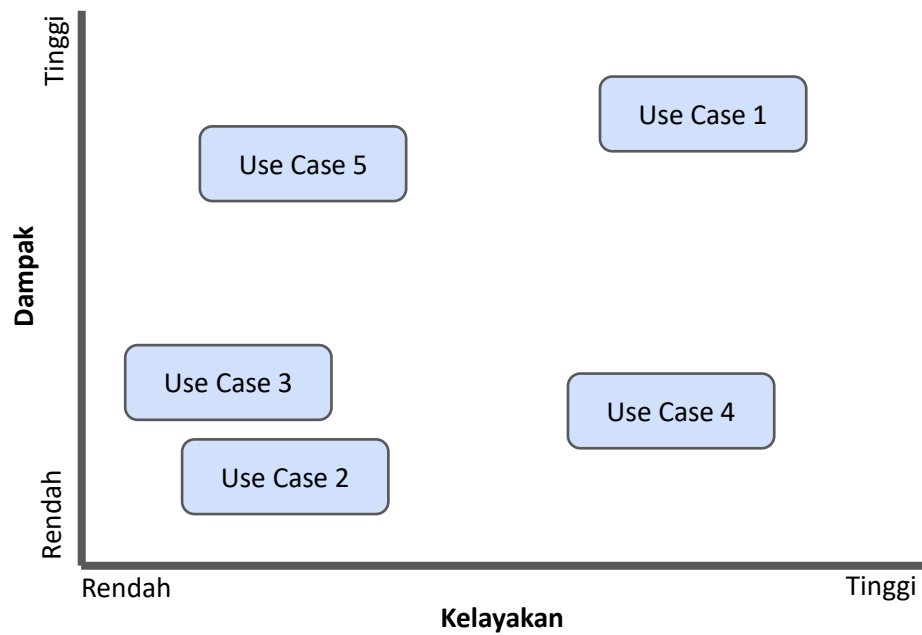
Baik Anda menggunakan pendekatan ini atau yang lain: Saya sangat menyarankan Anda untuk mengembangkan dan menerapkan kerangka kerja etis yang konsisten untuk mengevaluasi kasus penggunaan pembelajaran mesin. Kerangka kerja ini tidak harus super rumit dan menghasilkan hasil dengan akurasi dua digit. Kerangka kerja ini dapat bersifat luas dan sangat disesuaikan dengan kebutuhan Anda. Yang terpenting, kerangka kerja ini harus konsisten dan komprehensif. Baik pelanggan maupun bisnis Anda akan berterima kasih karenanya.

2.6 MEMBUAT PETA JALAN KASUS PENGGUNAAN YANG DIPRIORITASKAN

Di akhir bab ini, Anda diharapkan dapat menilai kelayakan kasus penggunaan Anda berdasarkan tiga tema: data, infrastruktur/arsitektur, dan etika. Ini akan memberi Anda dimensi kedua yang Anda butuhkan untuk melengkapi skor dampak yang telah kita tentukan dengan latihan pemetaan cerita dari Bab 1.

Sekarang kita dapat membuat diagram sederhana untuk setiap kasus penggunaan yang menunjukkan dampak dan kelayakan, seperti yang ditunjukkan pada Gambar di bawah:



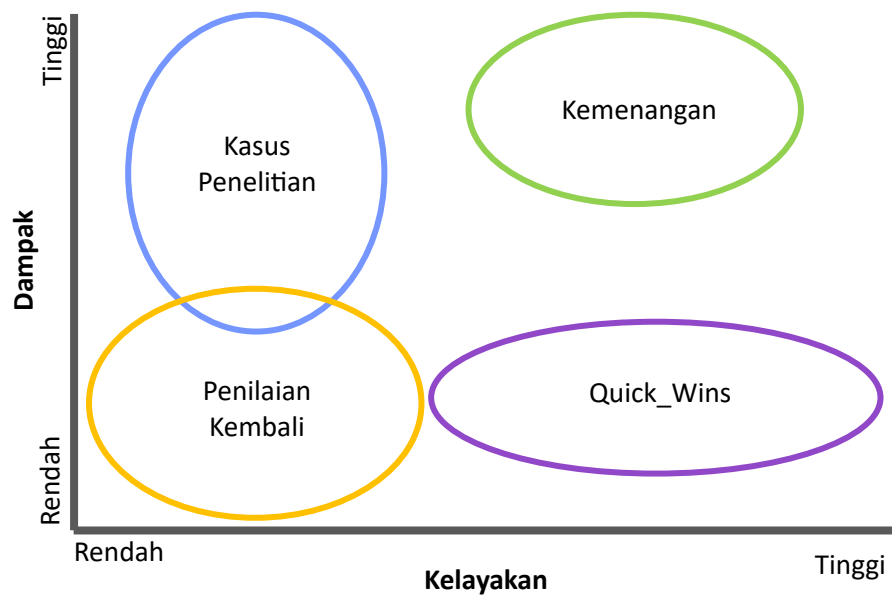


Gambar 2.8. Placeholder keterangan gambar

Seperti yang mungkin Anda perhatikan, kami tidak menggunakan metrik penilaian formal dalam bab ini untuk menghitung nilai numerik aktual untuk setiap kasus penggunaan. Anda dapat melakukannya, dan pendekatan yang lebih canggih pun dapat melakukannya, tetapi saran saya adalah memulai dengan sederhana dan membandingkan semua ide atau kasus penggunaan Anda dalam skala relatif satu sama lain.

Kasus penggunaan mana yang kemungkinan besar akan memberikan dampak terbesar, mana yang paling kecil. Kasus penggunaan mana yang menurut Anda akan relatif mudah diimplementasikan dan mana yang akan lebih menantang? Mengikuti sistem ini akan memberi Anda empat jenis kasus penggunaan untuk dipetakan ke diagram Anda, seperti yang ditunjukkan pada Gambar di bawah.





Gambar 2.9. Placeholder keterangan gambar

Area-area kasus penggunaan ini adalah sebagai berikut:

- **Champions:** Kasus penggunaan yang memiliki dampak dan kelayakan tinggi. Ini harus menjadi prioritas utama Anda dan mendorong peta jalan ML Anda.
- **Quick Wins:** Kasus penggunaan memiliki kelayakan tinggi yang serupa dengan champion, tetapi dampak bisnisnya lebih rendah. Karena kasus penggunaan ini seringkali kurang kompleks, mereka lebih mudah diimplementasikan. Namun, mereka dapat menjadi contoh yang baik dan menunjukkan nilai pembelajaran mesin dan aplikasi AI. Anggap ini sebagai area di mana layanan AI siap pakai dapat digunakan, misalnya.
- **Kasus Penelitian:** Ini adalah area yang memiliki dampak sangat tinggi, tetapi umumnya tidak layak dalam jangka pendek. Entah Anda tidak memiliki datanya, Anda belum diizinkan untuk menggunakannya (belum), atau ada batasan teknis tertentu. Namun, karena kasus penggunaan ini berpotensi memiliki dampak yang sangat besar pada bisnis Anda, jangan abaikan mereka, tetaplah pantau mereka.
- **Tinjau kembali nanti:** Kasus penggunaan ini berpotensi menghabiskan sumber daya yang sangat besar tanpa memberikan banyak nilai bisnis. Hal-hal tersebut sebaiknya dikesampingkan dan dievaluasi ulang nanti. Kasus penggunaan dengan dampak dan kelayakan yang rendah dapat berubah menjadi kemenangan cepat seiring perubahan teknologi. Misalnya, beberapa waktu lalu, membuat peringkat prioritas otomatis untuk tiket dukungan pelanggan sangatlah rumit. Namun, dengan perkembangan baru dalam layanan AI, hal ini tiba-tiba dapat dilakukan dalam waktu singkat, menjadikannya kemenangan cepat yang berpotensi menarik bagi bisnis Anda.

Bagaimana kita akan mengevaluasi kasus penggunaan churn kita? Sulit untuk mengatakannya, karena kita belum memiliki banyak informasi. Berhentilah sejenak dan pikirkan: Di klaster mana Anda akan menempatkannya?



Berdasarkan apa yang kita ketahui sejauh ini, saya akan menilai kelayakan teknisnya relatif tinggi, yang menjadikan kasus penggunaan Churn kandidat yang baik untuk Kemenangan Cepat atau Juara. Dari alur cerita di bab sebelumnya dan juga dari pengalaman saya dengan kasus penggunaan churn lainnya, saya memperkirakan dampaknya akan tinggi jika dilakukan dengan benar. Oleh karena itu, saya akan lebih condong ke Juara. Apakah Anda setuju dengan hal itu?

Sebagai kesimpulan, berikut adalah beberapa panduan dan praktik terbaik untuk menerjemahkan diagram Anda menjadi peta jalan kasus penggunaan yang diprioritaskan:

Gabungkan Champion dan Quick-Wins

Menetapkan tujuan yang ambisius memang baik, tetapi akan lebih baik lagi jika mencapainya secara berkala. Sekali lagi, proyek AI seringkali bersifat terbuka dan Anda tidak pernah tahu apakah semuanya akan berjalan sesuai harapan. Oleh karena itu, selalu baik untuk memiliki beberapa kasus penggunaan dalam alur kerja Anda yang memiliki waktu pengembangan yang relatif singkat dan probabilitas implementasi yang tinggi, meskipun dampaknya tidak terlalu besar.

Identifikasi Sumber Data Umum

Jika Anda harus memilih di antara berbagai kasus penggunaan, pilihlah kasus yang menggunakan sumber data yang sama daripada menggabungkan terlalu banyak domain data yang berbeda sekaligus. Misalnya, saya lebih suka menjalankan 3 kasus penggunaan berdasarkan data CRM daripada melihat data CRM dan sensor dari fasilitas produksi. Dengan setiap sumber data baru, kemungkinan besar Anda akan mengidentifikasi area masalah baru yang tidak Anda duga sebelumnya. Setelah Anda benar-benar memahami suatu sumber data, Anda harus mencoba memanfaatkannya sebaik mungkin.

Bangun Visi yang Menarik

Apakah Anda merasa perusahaan Anda memiliki "kasus penggunaan yang luar biasa" tetapi masih jauh dari layak secara teknis? Apakah Anda melihat keunggulan kompetitif bagi perusahaan Anda di bidang AI, tetapi belum dapat memanfaatkannya? Masukkan keunggulan tersebut ke dalam peta jalan Anda, tetapi tegaskan bahwa perjalanan masih panjang. Menyusun cerita yang menarik dan mengejar tujuan yang ambisius akan memberi Anda narasi yang kuat untuk menyelaraskan kasus penggunaan lain (yang kurang kompleks dan kurang berdampak) hingga Anda akhirnya mencapainya. Seperti kata pepatah, Roma tidak dibangun dalam sehari. Tetapi Anda bisa mulai sekarang dengan Sirkus Maximus.

2.7 KESIMPULAN

Dalam bab ini, Anda mempelajari cara mengevaluasi kelayakan kasus penggunaan AI dengan mempertimbangkan data, infrastruktur/arsitektur, dan data. Anda harus mengetahui cara melakukan penilaian data awal menggunakan kerangka kerja 4V dan apa saja pilihan Anda saat memilih infrastruktur ML seperti AI-as-a-Service atau Platform-as-a-Service. Kita juga membahas arsitektur tingkat tinggi yang dapat Anda gunakan untuk mendeskripsikan kasus penggunaan AI Anda. Dan saya harap Anda tidak pernah terjebak dalam tujuh jebakan pembelajaran mesin. Memang banyak, tetapi Anda telah meletakkan fondasi yang sempurna



untuk mengembangkan prototipe bertenaga AI yang sukses secara mandiri. Sebelum kita membahas penerapan praktis kasus penggunaan AI dalam berbagai skenario BI, bab selanjutnya akan memperkenalkan Anda pada perangkat prototipe untuk buku ini.



BAB 3

DASAR-DASAR PEMBELAJARAN MESIN

3.1 PROSES PEMBELAJARAN MESIN TERAWASI

Pembelajaran mesin terawasi berarti Anda melatih model pembelajaran mesin berdasarkan data historis yang kebenarannya dasarnya diketahui. Misalnya, jika Anda ingin memprediksi harga properti, dalam skenario pembelajaran terawasi, Anda memerlukan data harga rumah historis dan beberapa informasi lain yang menjelaskan rumah-rumah tersebut (misalnya, ukuran rumah, kamar tidur, lokasi, dll.). Pembelajaran mesin terawasi berbeda dengan pembelajaran mesin tanpa pengawasan, di mana kebenarannya dasarnya tidak diketahui dan komputer harus mengelompokkan titik data ke dalam kategori yang serupa. Pengelompokan adalah contoh populer dari pembelajaran tanpa pengawasan.

Sebagian besar masalah pembelajaran mesin perusahaan termasuk dalam ranah pembelajaran terawasi. Saya akan memandu Anda melalui proses dasar pembelajaran mesin terawasi dengan menunjukkan langkah-langkah kuncinya sehingga Anda dapat menerapkannya nanti saat kita membangun model pembelajaran mesin pertama kita mulai dari Bab 7.

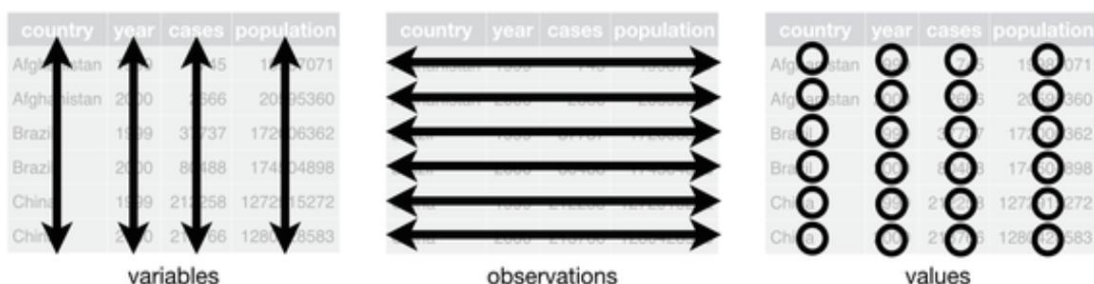
Langkah 1: Mengumpulkan Data Historis

Agar algoritma pembelajaran mesin dapat belajar, ia harus memiliki data historis. Oleh karena itu, Anda perlu mengumpulkan data dan menyajikannya dalam bentuk yang dapat diproses komputer secara efisien.

Kebanyakan algoritma mengharapkan data Anda "rapi". Data yang rapi memiliki tiga karakteristik:

- Setiap observasi berada di barisnya sendiri,
- setiap variabel berada di kolomnya sendiri, dan
- setiap pengukuran merupakan sel.

Gambar 3.1 menunjukkan contoh bagaimana data yang rapi terlihat. Merapikan data Anda mungkin merupakan proses yang paling rumit di seluruh alur kerja ML, tergantung dari mana Anda mendapatkan datanya.



Gambar 3.1. Data Rapi, Sumber: R for Data Science (O'Reilly)

Langkah 2: Identifikasi Fitur dan Label

Algoritme Anda mencoba membangun model yang akan memprediksi keluaran y dengan beberapa masukan x .

Mari kita bahas konsep-konsep ini secara singkat:

Label (juga disebut target, atau keluaran) adalah variabel yang ingin Anda prediksi berdasarkan variabel lain, yang disebut fitur (atribut, masukan) dalam jargon pembelajaran mesin. Misalnya, jika Anda ingin memprediksi harga rumah, harga rumah akan menjadi label Anda dan variabel seperti kamar tidur, ukuran, lokasi akan menjadi fitur Anda. Dalam pembelajaran terawasi, Anda memiliki data historis yang berisi fitur dan label untuk sejumlah observasi tertentu, yang juga disebut contoh pelatihan.

Jika label Anda numerik, proses ini disebut masalah regresi. Jika label Anda kategoris, proses ini disebut masalah klasifikasi. Algoritma yang berbeda digunakan untuk masalah klasifikasi atau regresi. Di bagian selanjutnya ("Algoritma Pembelajaran Mesin Populer"), saya akan memperkenalkan tiga algoritma populer yang cukup memadai untuk keperluan pembuatan prototipe kita.

Langkah 3: Membagi Data Anda dalam Set Pelatihan dan Uji

Setelah data Anda tertata rapi dan fitur serta label telah ditentukan, biasanya Anda perlu membagi set data Anda menjadi set pelatihan dan set uji. Set pelatihan adalah bagian dari set data historis yang akan digunakan untuk melatih model pembelajaran mesin (biasanya antara 70% - 80% data).

Set uji (atau holdout) adalah bagian dari data Anda yang akan digunakan untuk evaluasi akhir model pembelajaran mesin Anda. Dalam sebagian besar skenario Pembelajaran Mesin Otomatis (AutoML), Anda tidak perlu melakukan pemisahan ini secara manual, karena akan ditangani secara otomatis oleh alat AutoML. Namun, penting untuk memahami mengapa Anda harus melakukan pemisahan ini dan mengapa Anda hanya dapat mengevaluasi model Anda sekali pada set uji. Alasan melakukan pemisahan ini adalah Anda ingin memastikan bahwa model tidak hanya berkinerja baik pada data yang diketahuinya, tetapi juga pada data baru yang belum pernah dilihat sebelumnya dari distribusi yang sama. Itulah mengapa penting untuk menyisihkan beberapa data untuk tujuan evaluasi akhir.

Performa model Anda pada set pengujian akan menjadi indikator performa Anda untuk data baru yang belum pernah dilihat sebelumnya.

Langkah 4: Menemukan Model Terbaik Menggunakan Berbagai Algoritma

Pada langkah ini, Anda akan mencoba menemukan model yang paling mewakili data Anda dan yang memiliki daya prediksi tertinggi. Anda melakukannya dengan mencoba berbagai algoritma pembelajaran mesin dengan parameter yang berbeda.

Sekali lagi, mari kita bahas konsep-konsep ini secara singkat:

Model adalah hasil akhir dari proses pelatihan pembelajaran mesin Anda. Model ini merupakan fungsi kompleks (yang dapat berubah-ubah) yang menghitung nilai keluaran untuk setiap set nilai masukan yang diberikan.

Mari kita pertimbangkan contoh yang sangat sederhana. Model Anda bisa seperti ini:



$$y = 200x + 1000$$

Ini akan menjadi persamaan model regresi linier sederhana. Dengan nilai input x apa pun (misalnya ukuran rumah), rumus ini akan menghitung harga akhir dengan mengalikan ukuran dengan 200, lalu menambahkan 1.000. Tentu saja, model pembelajaran mesin biasanya jauh lebih kompleks, tetapi idenya tetap sama.

Jadi, bagaimana komputer, dengan beberapa data input dan output historis, dapat menghasilkan rumus seperti itu? Dua komponen diperlukan untuk ini: Pertama, kita perlu menyediakan algoritma untuk komputer, seperti dalam kasus ini regresi linier. Komputer akan mengetahui bahwa outputnya harus terlihat seperti persamaan regresi:

$$y = b_1x + b_0$$

Komputer sekarang akan menggunakan data pelatihan historis untuk menemukan parameter terbaik b_1 dan b_0 untuk permasalahan ini. Proses estimasi parameter inilah yang disebut "pembelajaran" atau "pelatihan" dalam pembelajaran mesin. Ada berbagai cara untuk mencapai dan mempercepat proses pembelajaran ini untuk kumpulan data besar, tetapi hal ini di luar cakupan buku ini dan juga tidak diperlukan untuk tujuan kita. Satu-satunya hal yang perlu Anda ketahui adalah Anda perlu mendefinisikan algoritma pembelajaran mesin untuk melatih model pada data pelatihan. Di sinilah para ahli seperti Ilmuwan Data atau pakar Pembelajaran Mesin dapat menguasainya. Dalam kasus kita, kita akan lebih mengandalkan solusi AutoML untuk menemukan model terbaik.

Langkah 5: Mengevaluasi Model Akhir

Setelah proses pelatihan selesai, akan ada satu model yang menunjukkan kinerja terbaik pada set pengujian. Kriteria evaluasi yang berbeda digunakan untuk berbagai permasalahan pembelajaran mesin, yang akan kita bahas lebih lanjut di bagian "Evaluasi Model Pembelajaran Mesin". Penting untuk memahami cara kerja metrik ini karena Anda perlu menentukan metrik ini ke layanan AutoML untuk menemukan model terbaik. Sekali lagi, jangan khawatir sebagian besar layanan akan memberikan saran metrik evaluasi awal yang baik.

Langkah 6: Penerapan

Setelah Anda memutuskan suatu model, biasanya Anda ingin menerapkannya di suatu tempat agar pengguna atau aplikasi dapat menggunakannya. Istilah teknis untuk bagian proses pembelajaran mesin ini adalah inferensi, prediksi, atau penilaian. Pada tahap ini, model Anda tidak lagi belajar, tetapi hanya menghitung keluaran saat menerima nilai masukan baru. Cara kerjanya bergantung pada pengaturan Anda.

Seringkali model dihosting sebagai API HTTP yang mengambil beberapa data masukan dan mengembalikan prediksi (prediksi daring). Sebagai alternatif, model dapat digunakan untuk menilai banyak data sekaligus, yang juga disebut prediksi batch.



Langkah 7: Pemeliharaan

Meskipun proses pengembangan model ML telah selesai setelah penerapan, sebenarnya prosesnya tidak pernah berakhir. Seiring perubahan pola data, model perlu dilatih ulang dan kita perlu melihat apakah performa awal model pembelajaran mesin dapat dipertahankan tinggi seiring waktu. Kita tidak perlu menangani hal-hal ini selama fase pembuatan prototipe, tetapi penting untuk dicatat bahwa model pembelajaran mesin membutuhkan banyak pemeliharaan setelah penerapan, yang harus Anda pertimbangkan dalam analisis kelayakan Anda. Kita akan membahas topik ini lebih lanjut di Bab 11.

Setelah Anda memahami secara umum proses pembelajaran mesin terawasi, mari kita lihat algoritma pembelajaran mesin paling populer yang digunakan untuk melatih model untuk masalah klasifikasi dan regresi.

3.2 ALGORITMA PEMBELAJARAN MESIN POPULER

Di bagian ini, saya akan memperkenalkan Anda pada tiga keluarga algoritma pembelajaran mesin yang populer, yaitu Regresi Linier, Pohon Keputusan, dan metode Ensemble. Jika Anda mengetahui ketiga kelas ini, Anda mungkin dapat menyelesaikan sekitar 90% dari semua masalah pembelajaran mesin terawasi dalam bisnis.

Tabel 3.1 memberikan ikhtisar tentang kelas-kelas algoritma ini, tetapi kita akan membahasnya lebih detail.

Tabel 3.1. Perbandingan pembelajaran mesin populer

Kelas Algoritma	Digunakan untuk	Performance	Interpretasi
Regresi Linier	Masalah Regresi	Rendah - Tinggi	Tinggi
Pohon Keputusan	Masalah Regresi dan klasifikasi	Rata-rata	Tinggi
Metode Ensemble (Bagging & Boosting)		Tinggi	Rendah

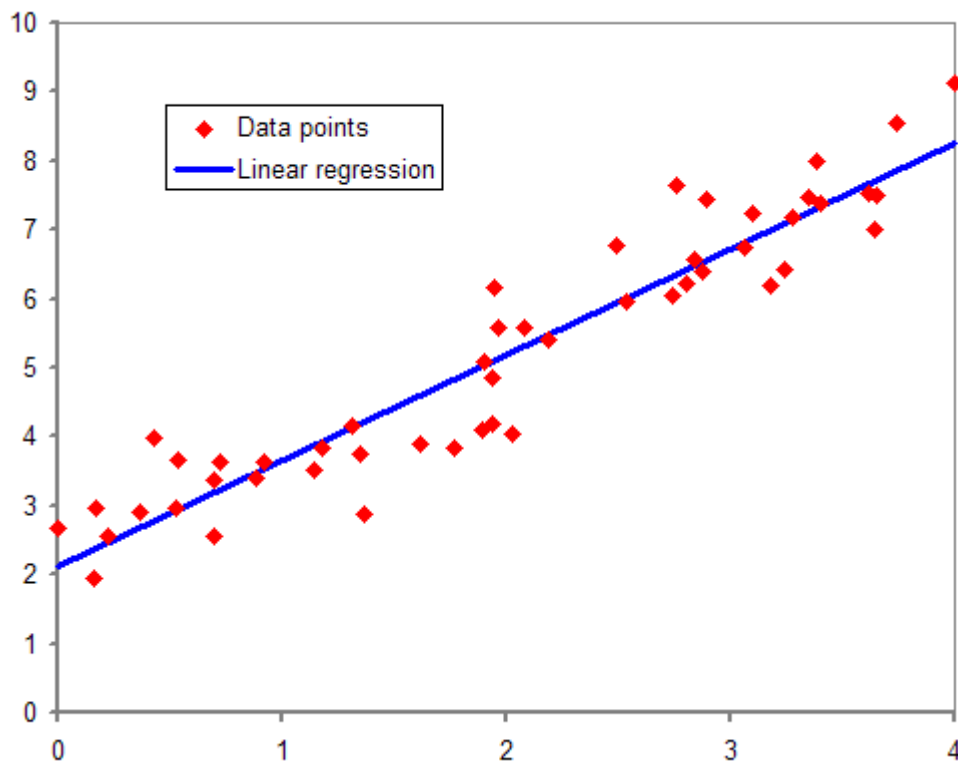
Regresi Linier

Regresi linier adalah salah satu algoritma terpenting untuk dipahami dalam Pembelajaran Mesin karena mendukung begitu banyak konsep yang berbeda. Regresi logistik, pohon regresi, analisis deret waktu, dan bahkan jaringan saraf tiruan menggunakan regresi linier sebagai intinya.

Cara kerja regresi linier mudah dijelaskan, tetapi sulit dikuasai: Dengan beberapa data masukan numerik, algoritma regresi akan menyesuaikan fungsi linier yang menghitung (memprediksi) data keluaran numerik yang sesuai.

Dua asumsi utama penting untuk regresi: 1) Fitur-fitur yang berbeda harus independen satu sama lain (korelasi rendah antar fitur) dan 2) hubungan antar fitur terhadap variabel hasil harus linier (misalnya, ketika satu variabel meningkat, variabel keluaran juga harus meningkat atau menurun dengan pola yang tetap).





Gambar 3.2. Regresi Linier Sederhana

Gambar 3.2 menunjukkan contoh regresi linier sederhana di mana variabel input pada sumbu x akan memprediksi variabel pada sumbu y. Titik-titik dalam plot adalah titik data aktual (label) yang diamati untuk setiap nilai x dalam kumpulan data historis. Prediksi dari regresi akan menjadi nilai y pada garis lurus untuk setiap nilai x. Tentu saja, ini hanyalah contoh dasar untuk tujuan ilustrasi. Regresi linier juga berfungsi dengan lebih dari satu variabel input dan juga dapat memodelkan berbagai bentuk selain garis lurus (Regresi Polinomial).

Seperti yang Anda lihat pada baris pertama Tabel 3.1, kinerja model Regresi Linier dapat berkisar dari sangat rendah hingga sangat tinggi. Hal ini karena kinerja algoritma regresi linier sangat bergantung pada data dan seberapa baik data tersebut memenuhi asumsi regresi linier. Jika variabel input benar-benar independen dan memiliki hubungan linier dengan variabel output, regresi dapat mengungguli jaringan saraf tiruan apa pun. Namun, regresi bukanlah pilihan yang baik jika data Anda mengandung pola non-linier. Misalnya, jika suatu nilai tertentu terpenuhi, tiba-tiba hubungan dengan variabel target berubah. Aturan "jika-maka-lain" semacam ini dapat dimodelkan dengan lebih baik menggunakan algoritma lain, seperti pohon keputusan.

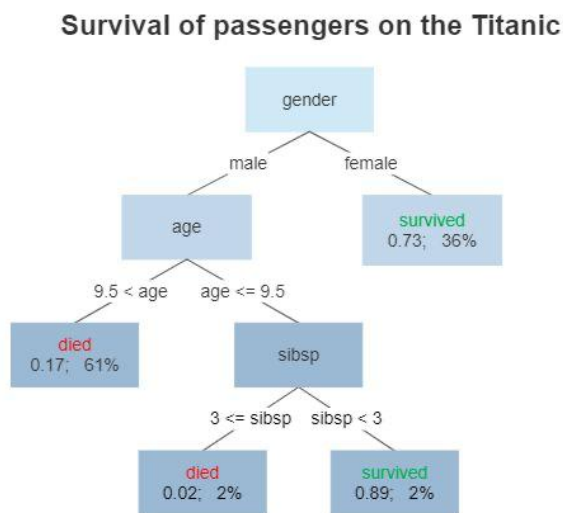
Pohon Keputusan

Berbeda dengan model regresi, model berbasis pohon juga dapat bekerja dengan data kategorikal dan numerik (pohon regresi) seperti yang Anda lihat pada Tabel 3.1. Cara kerja pohon keputusan adalah dengan membagi data pada tingkat yang berbeda, variabel demi variabel hingga irisan data menjadi sangat kecil sehingga Anda dapat membuat prediksi. Bayangkan pohon keputusan sebagai urutan hierarki aturan "jika-maka".



Pohon keputusan biasanya merupakan algoritma serba guna yang baik yang dapat digunakan pada hampir semua dataset tabular sebagai model dasar pertama. Pohon keputusan dapat dengan mudah dibagikan dan dijelaskan bahkan kepada pemangku kepentingan non-teknis. Di sisi lain, pohon keputusan tunggal seringkali tidak memberikan hasil terbaik dalam semua kasus karena algoritmanya "rakus". Artinya, sistem akan melakukan pemisahan terlebih dahulu pada data yang menunjukkan perbedaan tertinggi. Proses ini tidak ideal dalam semua situasi.

Gambar 3.3 menunjukkan contoh pohon keputusan yang memprediksi kelangsungan hidup penumpang Titanic, sebuah kumpulan data penelitian yang populer. Anda membaca pohon tersebut dari atas ke bawah, di mana setiap simpul mewakili langkah pengambilan keputusan.



Gambar 3.3. Pohon Keputusan

Karena pohon keputusan ini "serakah", ia akan melakukan pemisahan pertama berdasarkan jenis kelamin. Jika Anda memilih seorang penumpang perempuan secara acak dari dataset Titanic, peluang orang ini selamat adalah 73%. Lumayan untuk tebakan pertama. Untuk pria, pohon keputusan ini terlihat sedikit lebih rumit. Prediksinya akan "meninggal" atau "bertahan hidup" dengan mempertimbangkan beberapa faktor seperti usia orang tersebut dan jumlah saudara kandung atau pasangan ("sibsp").

Kami tidak akan membahas lebih detail di sini. Yang penting bagi Anda adalah memahami cara kerja pohon keputusan secara konseptual, karena kami akan mengembangkannya dalam konsep berikutnya: Pembelajaran Ensemble.

Metode Ensemble

Jika Regresi Linier dan Pohon Keputusan cukup lugas dan mudah dijelaskan cara kerjanya, metode Ensemble berada di ujung spektrum yang berlawanan. Seperti yang dapat Anda lihat pada Tabel 3.1, metode Ensemble umumnya memberikan daya prediksi yang tinggi,



tetapi pada saat yang sama tidak mudah diinterpretasikan dalam cara pengambilan keputusannya.

Baik regresi linier maupun pohon keputusan umumnya dianggap sebagai pembelajar yang 'lemah', yang berarti keduanya biasanya mengalami kesulitan jika hubungan data menjadi lebih kompleks. Misalnya, regresi tidak dapat bekerja dengan data non-linier dan pohon keputusan hanya dapat melakukan pemisahan sekali untuk setiap variabel dalam set data Anda.

Untuk mengatasi masalah ini, metode ensemble memungkinkan Anda menggabungkan beberapa pembelajar lemah menjadi pembelajar kuat yang menunjukkan kinerja kuat juga pada set data yang kompleks. Dua metode pembelajaran ensemble yang umum adalah bagging dan boosting yang dibedakan berdasarkan cara menggabungkan pembelajar lemah.

Bagging (dari agregasi bootstrap) mencoba menggabungkan beberapa pembelajar lemah dengan melatih berbagai model secara individual, lalu menggabungkannya menggunakan proses perata-rataan. Jika Anda melihat Gambar 3.4, Anda dapat melihat skema cara kerja Bagging. Di lapisan tengah, Anda dapat melihat empat model berbeda yang semuanya dilatih pada set data pelatihan yang sama. Ini bisa berupa empat pohon keputusan, misalnya, dengan parameter yang berbeda untuk kedalaman pohon atau ukuran simpul minimum.

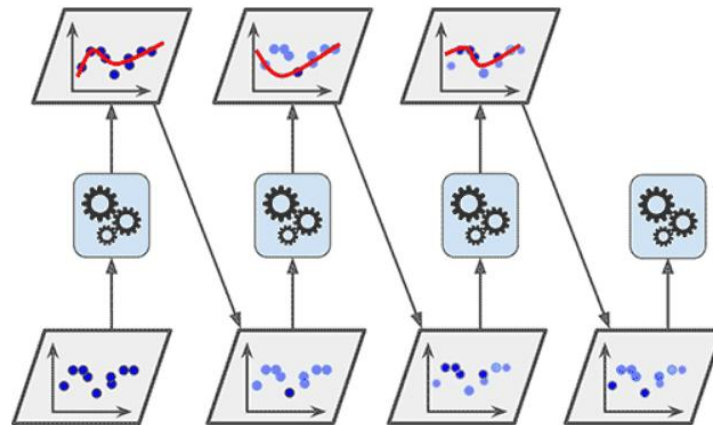
Untuk setiap titik data, masing-masing dari keempat model ini akan membuat prediksi individual (lapisan atas). Prediksi akhir akan berupa agregasi dari keempat hasil ini, misalnya suara mayoritas dalam kasus contoh klasifikasi atau nilai rata-rata untuk masalah regresi. Metode ansambel bagging yang populer disebut Hutan Acak di mana pohon keputusan dibagi menjadi beberapa sub-pohon agar korelasi di antara mereka lebih sedikit.



Gambar 3.4. Bagging

Di sisi lain, Boosting mengandalkan ide penambahan model secara berurutan ke dalam suatu ensemble, dengan masing-masing mengoreksi kesalahan model sebelumnya. Teknik ini sangat populer dan ampuh. Dua metode boosting yang populer adalah AdaBoost (*Adaptive Boosting*) dan Gradient Boosting.

Gambar 3.5 menunjukkan gambaran umum konseptual tentang cara kerja Bagging:



Gambar 3.5. Boosting

Proses pelatihan dimulai di kiri bawah, tempat model pertama (misalnya pohon keputusan) dilatih dan memberikan prediksi pertama dengan kesalahan yang relatif tinggi (kiri atas). Selanjutnya, pohon keputusan lain dipelajari yang mencoba mengoreksi kesalahan yang dibuat oleh model sebelumnya (model kedua di baris bawah). Proses ini akan diulang beberapa kali hingga kesalahan semakin mengecil dan model yang berkinerja baik ditemukan. Pada Gambar 3.5, terdapat tiga iterasi yang menghasilkan model akhir di kanan bawah.

Seperti yang mungkin Anda lihat dari penjelasan ini, metode ensemble jauh lebih sulit diinterpretasikan daripada regresi linier atau pohon keputusan. Namun, dalam kebanyakan kasus, metode ini memberikan kinerja yang lebih baik. Pendekatan mana yang harus Anda ambil bergantung pada tujuan Anda.

Jika kemampuan menjelaskan tidak sepenting akurasi untuk proyek Anda, memilih pembelajar yang kuat seperti Adaptive Boosting biasanya merupakan pilihan yang lebih baik. Di sisi lain, jika Anda ingin model pembelajaran mesin Anda dipahami dan diinterpretasikan dengan baik oleh pengguna atau pemangku kepentingan lainnya, memilih model yang sedikit kurang akurat seperti pohon keputusan atau regresi mungkin merupakan pilihan yang lebih baik. Jika ragu, pilih model yang lebih sederhana jika memberikan kinerja yang serupa.

3.3 PEMBELAJARAN MENDALAM

Algoritme yang telah kita bahas sejauh ini bekerja dengan sangat baik pada data tabular terstruktur seperti yang akan Anda temukan di spreadsheet atau gudang data. Untuk tipe data lain, seperti data gambar atau data teks, Anda akan memerlukan algoritma yang berbeda. Bidang ini sering juga disebut sebagai "Pembelajaran Mendalam".

Kami tidak akan membahas terlalu detail di sini. Layanan AI yang nantinya akan kami gunakan dari layanan Azure Cognitive didasarkan pada teknik Pembelajaran Mendalam, tetapi kami tidak akan mengembangkannya sendiri. Untuk tujuan kita, cukup memiliki gambaran umum tentang apa arti konsep pembelajaran mendalam ini dan cara kerjanya.

Pembelajaran mendalam mengacu pada berbagai pendekatan dan teknik dalam pembelajaran mesin yang dikhususkan untuk bekerja dengan kumpulan data berdimensi



tinggi, yaitu data dengan banyak fitur. Tidak ada definisi formal di mana pembelajaran mesin "dangkal" berakhir dan di mana "pembelajaran mendalam" dimulai, tetapi sebagian besar bergantung pada tipe datanya.

Perhatikan contoh berikut: Bahkan jika Anda membangun prediktor penjualan berdimensi sangat tinggi, Anda mungkin tidak akan membuat lebih dari beberapa lusin, bahkan mungkin ratusan fitur dari data CRM tabular Anda.

Sekarang bayangkan Anda menganalisis gambar untuk deteksi objek. Bahkan untuk gambar yang agak kecil dengan resolusi 300 x 300 piksel, Anda akan mendapatkan 90.000 dimensi (satu dimensi untuk setiap piksel). Dengan gambar RGB berwarna, jumlah ini menjadi tiga kali lipat.

Anda dapat melihat bahwa masih ada perbedaan yang cukup besar antara tabel berdimensi tinggi yang mungkin hanya berisi 1.000 kolom (dimensi) dan representasi gambar kecil dengan 90.000 dimensi untuk setiap gambar. Dan itulah mengapa pembelajaran mendalam sering dipertimbangkan untuk data non-tabular seperti gambar, video, dan berkas teks tak terstruktur. Dua kategori luas telah muncul dalam pembelajaran mendalam: Visi Komputer (menangani data gambar atau video) dan Pemrosesan Bahasa Alami (menangani data teks).

Pemrosesan Bahasa Alami

Anda mungkin sudah pernah mendengar tentang layanan AI yang mampu menafsirkan bahasa manusia, melakukan terjemahan lebih akurat daripada sebelumnya, atau bahkan menghasilkan teks baru sepenuhnya otomatis. Model bahasa besar yang telah dilatih pada miliaran contoh teks manusia, dikombinasikan dengan teknologi terobosan bernama Transformers, telah melambungkan bidang NLP dari domain penelitian kecil menjadi salah satu area terpanas dalam pengembangan AI.

Kami tidak akan melatih model NLP sendiri dalam buku ini, tetapi Anda akan diperkenalkan dengan penggunaan model bahasa mutakhir untuk menganalisis data teks dalam skala besar.

Visi Komputer

Visi komputer adalah teknologi yang memungkinkan mesin untuk "melihat" dan menginterpretasikan berkas gambar dan video. Teknologi yang disebut Jaringan Saraf Konvolusional (CNN) telah menghasilkan terobosan luar biasa di bidang ini dalam beberapa tahun terakhir. Aplikasinya beragam, mulai dari mengidentifikasi entitas komoditas dalam gambar seperti mobil, sepeda, rambu jalan, hewan, dan manusia, hingga pengenalan wajah dan mengekstraksi struktur teks dari gambar. Bidang ini beragam dengan banyak hal yang terjadi. Di Bab 9, kita akan menggunakan layanan AI Visi Komputer untuk mendeteksi dan menghitung mobil dalam berkas gambar.

Pembelajaran Penguatan

Kategori pembelajaran mendalam terakhir yang ingin saya soroti di sini adalah bidang yang disebut Pembelajaran Penguatan. Kita akan melihat layanan Pembelajaran Penguatan beraksi di Bab 8: Analisis Preskriptif Bertenaga AI.



Pembelajaran Penguatan bekerja secara berbeda dari pembelajaran terawasi atau tanpa pengawasan dan entah bagaimana merupakan bidang yang terpisah. Alasannya terletak pada cara model pembelajaran penguatan dipelajari. Alih-alih mengandalkan data historis, pendekatan pembelajaran penguatan membutuhkan aliran data baru yang konstan, yang memungkinkan model mempelajari strategi terbaik berdasarkan kondisi sistem dan kebijakan terkini yang memungkinkan model tersebut bertindak.

Pembelajaran penguatan telah mencapai terobosan besar dengan kemampuannya mengalahkan pemain manusia kelas dunia dalam permainan seperti *Go* dan *Starcraft*. Namun, seperti yang akan Anda ketahui nanti, pembelajaran penguatan juga merupakan cara yang hebat untuk mempersonalisasi pengalaman pengguna dan mempelajari seiring waktu apa yang diinginkan pengguna dan cara berinteraksi dengan mereka.

3.4 EVALUASI MODEL PEMBELAJARAN MESIN

Dengan pengetahuan sejauh ini, Anda seharusnya cukup percaya diri untuk membaca buku ini dan menerapkan model pembelajaran mesin dalam tahap pembuatan prototipe ke kasus penggunaan di dunia nyata. Namun, masih ada satu komponen yang hilang. Yaitu bagaimana menilai seberapa baik model Anda sebenarnya bekerja.

Kemampuan untuk mengukur kinerja model pembelajaran mesin sangat penting, tidak hanya selama pembuatan prototipe tetapi juga di tahap produksi selanjutnya. Metrik evaluasi yang kita pilih tidak boleh berbeda di antara keduanya, hanya nilai metriknya saja. Jadi, bagaimana kita mengevaluasi kinerja model pembelajaran mesin? Selamat datang di dunia metrik evaluasi! Metrik evaluasi dihitung karena dua alasan:

1. Metrik evaluasi digunakan untuk membandingkan berbagai model guna menemukan model prediktif terbaik.
2. Metrik evaluasi digunakan secara berkelanjutan untuk mengukur kinerja model yang sebenarnya pada data baru yang belum pernah dilihat sebelumnya.

Ide dasar di balik evaluasi model selalu sama: Kita akan membandingkan nilai yang diprediksi model kita dengan nilai yang seharusnya diprediksi model berdasarkan kebenaran dasar kita, misalnya, label dalam kumpulan data kita.

Apa yang terdengar sederhana secara intuitif, sebenarnya mengandung kompleksitas yang cukup tinggi. Bahkan, Anda bisa menulis buku tentang metrik evaluasi saja. Dalam kasus kita, kita akan membahasnya secara singkat dan hanya berfokus pada konsep terpenting yang kemungkinan besar akan Anda temui saat bekerja dengan algoritma pembelajaran mesin. Terdapat berbagai metrik evaluasi untuk masalah klasifikasi dan regresi.

Mengevaluasi Model Regresi

Ingatlah bahwa model regresi memprediksi variabel numerik kontinu seperti pendapatan, kuantitas, ukuran, dll. Metrik yang paling populer untuk mengevaluasi model regresi adalah *Root Mean Squared Error (RMSE)*. RMSE adalah akar kuadrat dari rata-rata kuadrat kesalahan dalam nilai prediksi regresi (\hat{y}). RMSE dapat didefinisikan sebagai berikut:



$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}}$$

RMSE mengukur akurasi keseluruhan model dan dapat digunakan untuk membandingkan berbagai model regresi. Semakin kecil nilainya, semakin baik kinerja model tersebut. RMSE ditampilkan dalam skala asli sebagai nilai prediksi. Misalnya, jika model Anda memprediksi harga rumah dalam dolar AS dan Anda melihat RMSE sebesar 256.6, ini berarti "prediksi model salah rata-rata sebesar 256.6 rupiah".

Metrik populer lainnya yang akan Anda lihat dalam konteks evaluasi regresi adalah koefisien determinasi, yang juga disebut statistik R-kuadrat. R-kuadrat berkisar antara 0 hingga 1 dan mengukur proporsi variasi data yang diperhitungkan dalam model. Nilai ini berguna ketika Anda ingin menilai seberapa baik model tersebut sesuai dengan data. Rumus untuk R-kuadrat adalah:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

Anda dapat mengartikannya sebagai satu dikurangi varians terjelaskan model dibagi dengan total varians variabel target. Umumnya, semakin tinggi R-kuadrat, semakin baik. Nilai R-kuadrat 1 berarti model dapat menjelaskan semua varians dalam data.

Meskipun statistik ringkasan merupakan cara yang bagus untuk membandingkan berbagai model satu sama lain dalam skala besar, masih sulit bagi kita untuk mengetahui apakah regresi kita berfungsi seperti yang diharapkan hanya berdasarkan metrik ini. Oleh karena itu, pendekatan yang baik dalam pemodelan regresi adalah dengan melihat distribusi kesalahan regresi, yang juga disebut residual model kita. Distribusi residual memberi kita umpan balik visual yang baik tentang kinerja model regresi. Diagram residual memplot nilai prediksi terhadap residual sebagai diagram sebar sederhana. Idealnya, plot residual harus terlihat seperti pada Gambar di bawah untuk mendukung asumsi berikut:

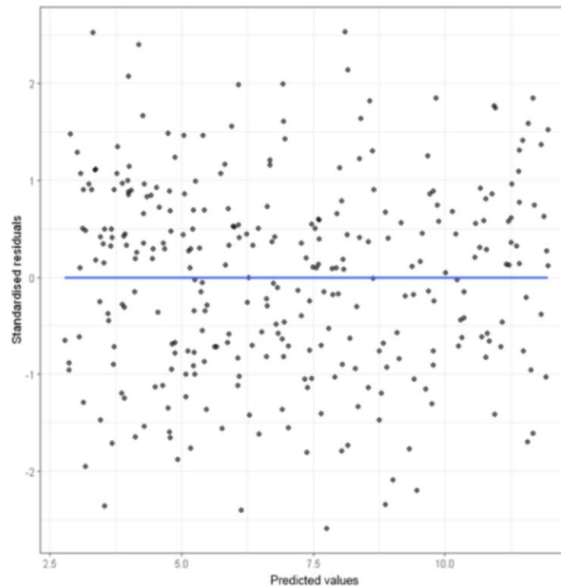
Linearitas

Titik-titik dalam plot residual harus tersebar secara acak tanpa memperlihatkan terlalu banyak "kelengkungan".

Heteroskedastisitas

"Penyebaran" titik-titik di antara nilai-nilai prediksi harus kurang lebih sama untuk semua nilai prediksi.





Gambar 3.6. Analisis Residu

Plot residu akan menunjukkan bagian data mana yang menunjukkan model regresi bekerja dengan baik dan bagian mana yang tidak. Tergantung pada kasus penggunaan Anda, Anda dapat memutuskan apakah hal ini bermasalah atau masih dapat diterima dengan mempertimbangkan semua faktor lainnya. Melihat statistik ringkasan saja tidak akan memberikan wawasan ini. Memantau residu selalu merupakan ide yang baik saat mengevaluasi model regresi.

Mengevaluasi Model Klasifikasi

Bagaimana kita mengevaluasi kinerja model prediktif jika nilai prediksi bukan numerik tetapi kategoris? Kita tidak bisa hanya menerapkan metrik yang kita pelajari dari model regresi. Untuk memahami alasannya, mari kita lihat beberapa data pada Tabel 3.2.

Tabel 3.2 Contoh klasifikasi output

Target	Diprediksi	Keluaran Probabilistik
1	1	0.954
1	0	0.456
0	0	0.012
0	1	0.567
0	0	0.234
...

Tabel menunjukkan label sebenarnya dari masalah klasifikasi (kolom "Target") serta keluaran yang sesuai ("Prediksi" dan "Keluaran Probabilistik") dari model klasifikasi.

Jika Anda melihat empat baris pertama, ada empat hal yang mungkin terjadi di sini:

- Label sebenarnya adalah `1`, dan prediksi kita adalah `1` (prediksi benar)
- Label sebenarnya adalah `0`, dan prediksi kita adalah `0` (prediksi benar)



- Label sebenarnya adalah `1`, dan prediksi kita bukan `1` (prediksi salah)
- Label sebenarnya adalah `0`, dan prediksi kita bukan `0` (prediksi salah)

Keempat hasil ini juga akan terjadi jika kita memiliki lebih dari dua kategori. Kita cukup mengamati keempat hasil ini untuk setiap kategori dalam kumpulan data kita.

Cara paling populer untuk mengukur kinerja model klasifikasi adalah dengan menampilkan keempat hasil ini dalam apa yang disebut tabel kebingungan atau matriks kebingungan seperti yang terlihat pada Gambar di bawah.

Tabel 3.3 Matriks Kebingungan

		Kelas yang diprediksi	
		Negatif	Positif
Aktual	Negatif	2.388 (True Negatif/TN)	558 (False Positif/FP)
	Positif	415 (False Negatif/FN)	2.954 (True Positif/TP)

Tabel kebingungan ini memberi tahu kita bahwa untuk contoh masalah klasifikasi kita, terdapat `2388` observasi dengan label aktual `0` dan model kita melabelinya dengan benar. Observasi ini disebut Negatif Sejati, karena model memprediksi label kelas Negatif dengan tepat. Demikian pula, `2954` observasi diidentifikasi oleh model kita sebagai "Positif Sejati", dengan hasil aktual dan prediksi berlabel kelas Positif (dalam hal ini angka `1`). Namun, model kita juga membuat dua jenis kesalahan. Jenis kesalahan pertama adalah salah memprediksi kelas Negatif menjadi Positif (558 kasus), yang juga dikenal sebagai "kesalahan Tipe I". Jenis kesalahan kedua adalah salah memprediksi 415 kasus negatif yang sebenarnya positif. Kesalahan-kesalahan ini juga dikenal sebagai "Kesalahan Tipe II". Para ahli statistik kurang kreatif dalam menamai hal-hal ini.

Sekarang, ada berbagai metrik yang dapat kita hitung dari tabel kebingungan ini. Metrik paling populer yang akan sering Anda lihat dan dengar adalah akurasi. Akurasi menggambarkan seberapa sering model kita benar, berdasarkan semua prediksi. Kita dapat dengan mudah menghitung akurasi model dengan membagi jumlah prediksi yang benar dengan jumlah semua prediksi:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

Untuk matriks kebingungan di atas, akurasinya adalah $(2388 + 2954) / (2954 + 2388 + 558 + 415) = 0,8459$, yang berarti prediksi model kami benar dalam 84,59% dari semua kasus.

Meskipun akurasi merupakan metrik yang banyak digunakan dan sangat populer, terdapat satu peringatan penting: Akurasi hanya berfungsi untuk masalah klasifikasi yang seimbang.

Bayangkan contoh berikut:



Kami ingin memprediksi penipuan kartu kredit, yang hanya terjadi pada 0,1% dari semua transaksi kartu kredit. Jika model kami secara konsisten memprediksi `Tanpa Penipuan`, hasilnya akan mencapai skor akurasi yang mencengangkan, yaitu `99,9%`, karena prediksi tersebut akan benar dalam sebagian besar kasus.

Meskipun model ini memiliki metrik akurasi yang sangat baik, jelas bahwa model ini sama sekali tidak berguna. Itulah sebabnya matriks kebingungan memungkinkan kami menghitung lebih banyak metrik yang lebih seimbang untuk mengidentifikasi label kelas positif atau negatif dengan tepat. Metrik pertama yang bisa kita lihat disebut Presisi. Presisi mengukur proporsi positif sejati (label benar yang diprediksi ternyata benar).

Presisi, atau juga disebut nilai prediksi positif (PPV), didefinisikan sebagai:

$$PPV = \frac{TP}{TP + FP}$$

Dalam contoh kita, presisinya adalah $2954 / (2954 + 558) = 0,8411$. Nilai ini dapat diartikan bahwa ketika model memberi label titik data sebagai positif (`1`), data tersebut benar dalam 84% dari semua kasus. Kita akan memilih metrik ini jika biaya positif palsu sangat tinggi dan biaya negatif palsu agak rendah, misalnya, ketika kita ingin memutuskan apakah kita akan menampilkan iklan mahal kepada pengguna daring.

Metrik lain yang melihat kelas positif tetapi memiliki fokus yang sedikit berbeda adalah Recall, terkadang juga disebut "Sensitivitas" atau rasio positif sejati (TPR). Metrik ini memberikan persentase semua kelas positif yang diklasifikasikan dengan benar sebagai positif. Recall didefinisikan sebagai:

$$TPR = \frac{TP}{TP + FN}$$

Dalam contoh kita, nilainya adalah $2954 / (2954 + 415) = 0,8768$. Nilai ini berarti model dapat mengidentifikasi 87,68% dari semua kelas positif dalam dataset. Kita akan memilih metrik ini untuk mengoptimalkan model kita dalam menemukan kelas positif dalam dataset kita (misalnya, deteksi penipuan). Menggunakan recall sebagai metrik evaluasi masuk akal ketika risiko kehilangan kelas positif ini (yaitu, Negatif Palsu) sangat tinggi.

Jika Anda tidak ingin condong ke Presisi atau recall dan akurasi tampaknya tidak cukup baik dalam mengatasi masalah Anda, ada metrik lain yang umum digunakan yang disebut `F₁ – Score`. Meskipun namanya teknis, metrik ini relatif intuitif. F₁ – Score hanyalah rata-rata harmonis dari presisi dan recall.

Dengan demikian, kita dapat dengan mudah menghitung F₁ – Score jika kita sudah mengetahui presisi dan recall:

$$F_1 = 2 \times \frac{PPV \times TPR}{PPV + TPR} = \frac{2TP}{2TP + FP + FN}$$



Dalam kasus kita, F_1 – Score adalah $2 * (0,8411 * 0,8768) / (0,8411 + 0,8768) = 0,8586$.

Skor F1 yang lebih tinggi menunjukkan model yang berkinerja lebih baik, tetapi apa sebenarnya arti skor tersebut? Skor F1 memang tidak mudah dijelaskan dalam satu kalimat seperti metrik di atas, tetapi biasanya merupakan metrik yang sangat baik untuk menilai kualitas suatu model. Karena Skor F1 menggunakan harmonik presisi dan perolehan, alih-alih rata-rata aritmatika, skornya akan cenderung lebih rendah jika salah satu dari keduanya sangat buruk. Bayangkan contoh ekstrem di mana presisi adalah 0 dan perolehan adalah 1. Rata-rata aritmatika akan menghasilkan skor 0,5 yang secara kasar diterjemahkan menjadi "Jika salah satu bagus, tetapi yang lainnya buruk, maka rata-ratanya lumayan.". Namun, rata-rata harmoniknya akan menjadi 0 dalam kasus ini. Alih-alih merata-ratakan angka tinggi dan rendah, Skor F1 akan memberikan peringatan jika perolehan kembali atau presisi sangat rendah. Hal ini jauh lebih mendekati apa yang Anda harapkan dari metrik kinerja gabungan.

Mengevaluasi Model Multi-Klasifikasi

Metrik yang kami gunakan untuk masalah klasifikasi biner juga berlaku untuk masalah klasifikasi multi-kelas (lebih dari 2 kelas untuk diprediksi). Sebagai contoh, perhatikan nilai target dan nilai prediksi berikut:

```
Targets = [1, 3, 2, 0, 2, 2]
predicted = [1, 2, 2, 0, 2, 0]
```

Satu-satunya perbedaan adalah Anda akan menghitung metrik evaluasi dari atas untuk setiap kelas (0, 1, 2, 3) dan membandingkannya dengan semua kelas lainnya. Cara Anda melakukannya dapat mengikuti pendekatan yang berbeda, biasanya disebut sebagai nilai "mikro" dan "makro".

Skor "Mikro" biasanya menghitung metrik secara global dengan menghitung semua total positif benar, negatif salah, dan positif salah secara bersamaan, lalu membaginya satu sama lain.

Skor 'Makro' di sisi lain menghitung Presisi, Perolehan, dll. untuk setiap label terlebih dahulu, lalu menghitung rata-rata semuanya. Pendekatan ini menghasilkan hasil yang serupa, tetapi tetap berbeda. Misalnya, seri dummy kami untuk 4 label klasifikasi di atas memiliki Skor F1 'mikro' sebesar 0,667 dan Skor F1 makro sebesar 0,583. Jika Anda suka, cobalah dan hitung angka-angkanya secara manual untuk meningkatkan pemahaman Anda tentang metrik ini.

Tidak ada metrik mikro atau makro yang lebih baik atau lebih buruk. Ini hanyalah cara perhitungan yang berbeda dan statistik ringkasan yang sama berdasarkan pendekatan yang berbeda. Nuansa ini akan menjadi lebih penting setelah Anda mengerjakan masalah multikelas yang lebih kompleks.

Masih banyak lagi metrik kinerja yang perlu dipertimbangkan, tetapi hal-hal ini berada di luar cakupan buku ini. Misalnya, sejauh ini kita belum membahas metrik kinerja yang mempertimbangkan keluaran probabilistik model, seperti AUC (area di bawah Kurva ROC). Tidak ada solusi instan dalam hal metrik evaluasi. Penting bagi Anda untuk setidaknya memahami secara mendalam apa yang ingin ditangkap oleh metrik-metrik ini dan



mengidentifikasi metrik mana yang praktis untuk kasus penggunaan Anda. Dan: Kinerja mana yang cukup baik untuk menghentikan pelatihan model pembelajaran mesin yang semakin kompleks dan (tampaknya) akurat? Kita akan membahasnya lebih lanjut di bagian selanjutnya.

3.5 JEBAKAN UMUM PEMBELAJARAN MESIN

Pembelajaran mesin adalah alat yang ampuh untuk memecahkan banyak masalah modern, tetapi seperti alat lainnya, pembelajaran mesin mudah disalahgunakan dan dapat menyebabkan hasil yang buruk. Berikut adalah beberapa kesalahan pemula yang ingin Anda hindari.

Jebakan 1: Menggunakan Pembelajaran Mesin Saat Anda Tidak Membutuhkannya

Jika Anda memiliki palu, semuanya tampak seperti paku. Hal terburuk yang dapat Anda lakukan adalah menggunakan pengetahuan baru Anda tentang pembelajaran mesin dan mencari masalah bisnis yang tampaknya cocok. Sebaliknya, lihatlah dari sudut pandang yang berlawanan: Setelah Anda mengidentifikasi masalah bisnis yang relevan, pertimbangkan apakah pembelajaran mesin dapat membantu menyelesaikannya. Dan bahkan setelah itu, penting untuk memulai dengan garis dasar yang sederhana terlebih dahulu.

Garis dasar ini penting untuk mendapatkan tolok ukur kinerja yang dibutuhkan oleh algoritma pembelajaran mesin untuk mengunggulinya. Jika Anda belum pernah melakukan analisis churn sebelumnya, Anda sebaiknya tidak memulai dengan pendekatan berbasis ML, melainkan menggunakan heuristik sederhana atau aturan yang dikodekan secara kaku. Pendekatan ini memiliki kompleksitas yang sangat rendah dan mudah dipahami oleh para pemangku kepentingan bisnis. Beralihlah ke model ML segera setelah Anda menemukan bahwa model tersebut berkinerja jauh lebih baik daripada solusi dasar Anda.

Jebakan 2: Terlalu Serakah

Jangan mencoba bersikap terlalu serakah dan memaksimalkan metrik kinerja set pelatihan Anda. Hal ini dapat menyebabkan model Anda berkinerja buruk pada data baru yang belum pernah dilihat sebelumnya. Hal ini mungkin terdengar berlawanan dengan intuisi, tetapi Anda harus lebih konservatif dalam pelatihan, terutama pada set data yang lebih kecil, karena hal ini dapat menghasilkan generalisasi model yang lebih baik ke data yang belum pernah dilihat. Konsep ini juga dikenal sebagai "Pisau Cukur Occam" dan berarti bahwa ketika ragu, solusi sederhana lebih baik daripada solusi yang kompleks.

Jebakan 3: Membangun Model yang Terlalu Kompleks

Membangun model yang terlalu kompleks sejalan dengan poin sebelumnya: untuk mencapai akurasi tinggi, orang yang tidak berpengalaman cenderung membuat model yang terlalu kompleks seperti jaringan saraf. Dalam banyak kasus, model yang kurang akurat dengan kompleksitas yang lebih rendah lebih disukai daripada model yang sangat kompleks dan sangat akurat.

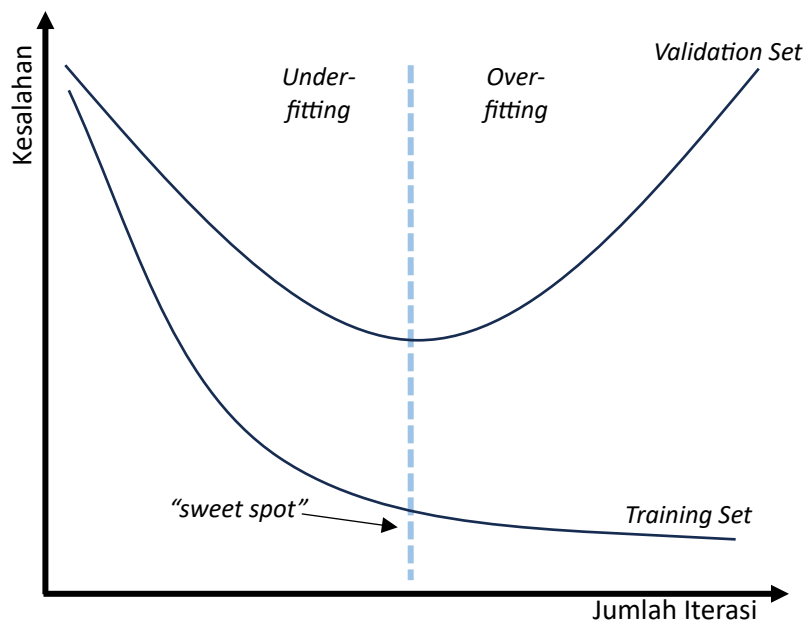
Model yang kompleks memiliki dua kelemahan:

- Pertama, model-model tersebut sulit dipelihara dan di-debug dalam produksi. Jika terjadi kesalahan atau prediksi model Anda meleset (yang pasti akan terjadi dari waktu



ke waktu), model yang sangat kompleks lebih sulit diperbaiki daripada model yang lebih sederhana.

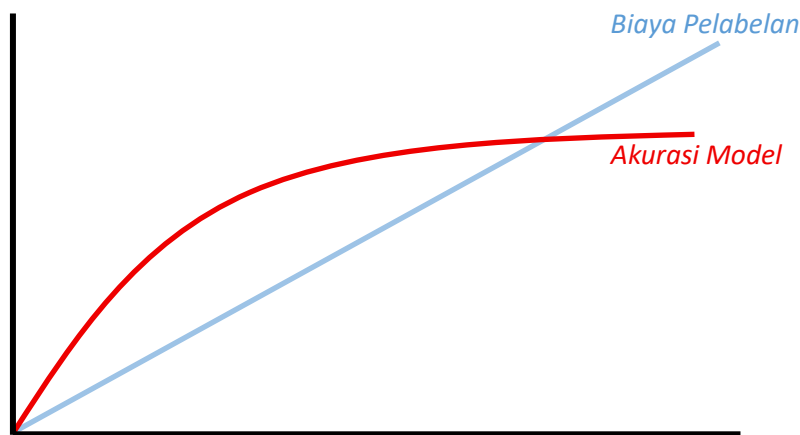
- Kedua, model yang kompleks dapat menyebabkan fenomena yang disebut overfitting. Overfitting terjadi ketika model Anda berkinerja sangat baik pada data pelatihan tetapi buruk pada data baru. Analisis prediktif bertujuan untuk menemukan titik optimal (lihat Gambar 3-7) di mana kesalahan prediksi dalam pelatihan (data historis) dan pengujian (data baru) kira-kira sama. Untuk menghindari overfitting, Anda harus selalu memantau kinerja model Anda pada data baru serta kompleksitasnya.



Gambar 3.7. Kesalahan Pelatihan vs. Kesalahan Pengujian

Jebakan 4: Tidak Berhenti Ketika Anda Memiliki Data yang Cukup

Jarang sekali Anda memiliki data pembelajaran mesin yang berlabel sempurna di organisasi Anda. Seperti yang telah kita pelajari, biasanya semakin banyak contoh (observasi) yang kita miliki, semakin baik. Namun, pengalaman menunjukkan bahwa algoritma pembelajaran mesin mencapai titik jenuh pada titik tertentu di mana contoh pelatihan tambahan tidak lagi meningkatkan akurasi secara signifikan. Efek ini diilustrasikan pada Gambar 3.8.



Gambar 3.8. Akurasi Model vs. Biaya Pelabelan

Ketika Anda mencapai titik ini, menghabiskan lebih banyak uang untuk pelabelan data tidak lagi efektif dari segi biaya. Oleh karena itu, penting untuk memiliki tujuan yang jelas. Seberapa akurat model Anda agar dapat digunakan? Ambang batasnya dapat berkisar dari "apa pun yang lebih baik daripada baseline" hingga "diperlukan akurasi 99,99", seperti dalam kasus medis. Mengetahui tujuan Anda dan apa yang ingin Anda capai dapat membantu Anda menghindari biaya tinggi.

Jebakan 5: Terjebak dalam Kutukan Dimensionalitas

Meskipun prinsip "semakin banyak data, semakin baik" berlaku untuk observasi (baris), prinsip ini sebenarnya dapat kontraproduktif untuk fitur (kolom). Bayangkan contoh berikut. Anda ingin memprediksi harga rumah di AS berdasarkan variabel ukuran, kamar tidur, dan lokasi (kode pos). Ukuran dan kamar tidur dapat dimodelkan sebagai dua fitur individual (kolom), yang hanya meningkatkan dimensionalitas dataset Anda sebanyak dua kali lipat seperti yang terlihat pada contoh di Gambar 3.9.

Titik data di berbagai dimensi. Sumber <https://aiaspirant.com> kutukan dimensionalitas

Gambar 3.9. Titik data di berbagai dimensi. Sumber: <https://aiaspirant.com/curse-of-dimensionality/>

Namun, setelah Anda mulai menggunakan kode pos, sebagian besar algoritma pembelajaran mesin mengharuskan Anda mengodekan variabel kategoris ini ke dalam fitur satu kolom yang lebarnya sama persis dengan jumlah nilai unik. Hal ini meningkatkan dimensionalitas data tidak hanya dengan urutan 1, tetapi juga dengan urutan 41.692 (kemungkinan kode pos AS), karena data tersebut akan berupa kolom tunggal yang berisi nilai "0" atau "1" untuk setiap contoh.

Sebagian besar algoritma pembelajaran mesin akan kesulitan menemukan pola aktual dalam data karena datanya terlalu jarang. Tidak ada aturan mutlak untuk menghindari kutukan dimensionalitas, tetapi berikut beberapa tipsnya:

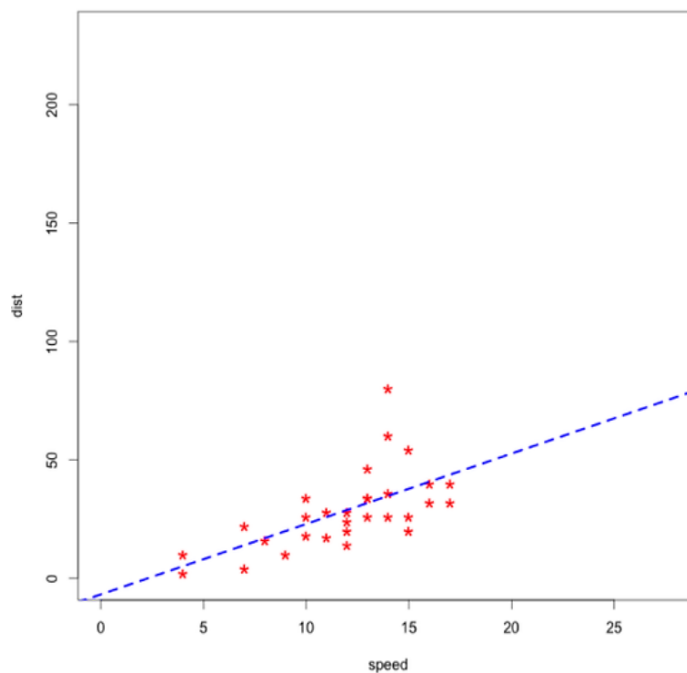
- Berhati-hatilah saat menambahkan fitur baru ke dataset Anda dan jangan ragu untuk menghapus fitur yang berlebihan atau tidak relevan. Hal ini terkadang sulit, karena Anda mungkin memerlukan keahlian domain yang solid untuk langkah ini.



- Cobalah untuk mengurangi jumlah relasi antar atribut dengan mengodekan dependensi ini sebagai satu atribut. Misalnya, alih-alih memiliki dua atribut "harga" dan "ukuran", Anda dapat menghitung atribut baru "harga per kaki persegi". Semakin sedikit dependensi yang Anda miliki antar variabel dalam dataset Anda, semakin mudah bagi algoritma pembelajaran mesin Anda untuk memahaminya.

Jebakan 6: Mengabaikan Outlier

Outlier adalah titik data yang jauh di atas nilai rata-rata dataset Anda. Misalnya, bayangkan sebuah dataset yang berisi gaji dan kekayaan bersih seseorang. Jika Anda memplot data tersebut, hasilnya mungkin akan terlihat seperti Gambar 3.10.

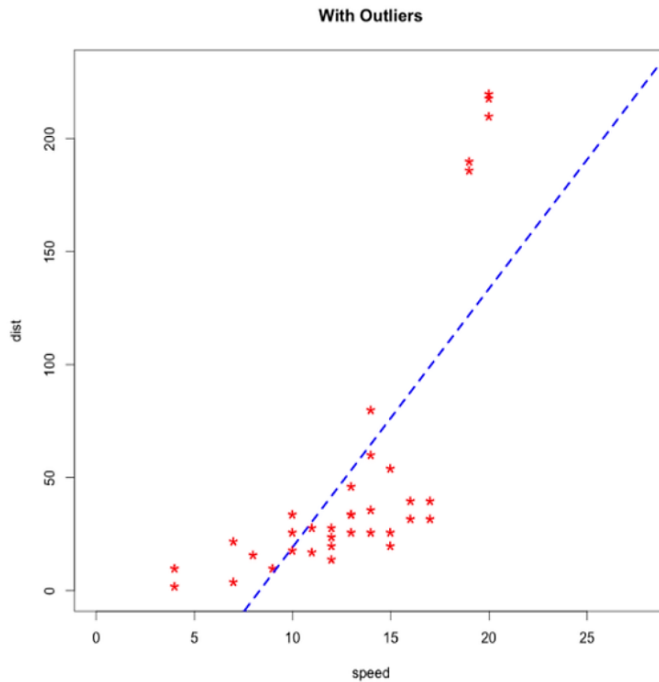


Gambar 3.10. Contoh data tanpa outlier

Anda dapat melihat adanya korelasi yang kuat antara kedua variabel, karena sebagian besar titik berada tidak jauh dari diagonal.

Namun, jika Anda memplot titik tambahan di atas data Anda, Anda akan melihat signifikansi sebenarnya dari nilai ini. Gambar 3.11 menunjukkan kumpulan data yang sama, tetapi dengan nilai tambahan (titik merah) yang jauh lebih tinggi daripada yang lain. Ini mewakili seseorang yang menyelesaikan transaksi senilai Rp 100 miliar di tahun pertama kerjanya:





Gambar 3.11. Contoh data dengan outlier

Dampak outlier bisa sangat besar bagi banyak algoritma, terutama yang bekerja dengan tugas regresi. Ini berarti Anda harus sangat memperhatikan pendeteksian outlier dalam set data Anda.

Jebakan 7: Menganggap Infrastruktur Cloud Begitu Saja

Dalam buku ini, saya berani berasumsi bahwa Anda memiliki akses ke komputasi cloud untuk pelatihan model dan inferensi model. Namun, jujur saja, di banyak perusahaan hal itu mungkin belum terjadi (belum). Meskipun adopsi komputasi cloud berkembang pesat, sebagian besar perusahaan masih menggunakan solusi on-premise. Ada banyak alasan untuk ini, tetapi salah satu alasan penting adalah ketakutan kehilangan kendali atas data.

Saya sangat menyarankan Anda mencoba komputasi cloud (AI-as-a-Service atau platform ML) setidaknya untuk pembuatan prototipe dengan data yang tidak penting, seperti yang dijelaskan di bab berikutnya. Meskipun ini mungkin belum menjadi praktik yang dipraktikkan di organisasi Anda, ini akan memberi Anda permulaan yang cepat dan menghubungkan Anda dengan alur kerja dan alat ML yang mutakhir. Ini juga akan memungkinkan Anda berdiskusi dengan tim Anda tentang bagaimana komputasi awan dapat menjadi investasi yang menguntungkan strategi AI/ML Anda secara keseluruhan.

3.6 KESIMPULAN

Dalam bab ini, Anda mempelajari konsep dasar pembelajaran mesin terawasi dan mendapatkan gambaran tentang algoritma mana yang penting untuk pembelajaran mesin. Kami juga membahas istilah-istilah seperti Pembelajaran Mendalam, Visi Komputer, atau Pemrosesan Bahasa Alami.

Tentu saja, masih banyak lagi yang perlu dipelajari tentang pembelajaran mesin, tetapi halaman-halaman ini akan memberi Anda semua yang Anda butuhkan untuk membangun



prototipe bertenaga ML Anda sendiri dengan bantuan alat-alat dalam buku ini. Saya harap Anda merasa sedikit lebih percaya diri untuk melakukannya sekarang. Jika belum, jangan khawatir!

Nantinya, saat kita mendekati kasus penggunaan praktis, konsep-konsep dari bab ini mungkin akan terasa jauh lebih mudah dipahami oleh Anda karena terhubung dengan beberapa praktik. Jika Anda ingin mendalami pembelajaran mesin lebih dalam, saya merekomendasikan buku-buku berikut:

- Pola Desain Pembelajaran Mesin oleh Valliappa Lakshmanan, Sara Robinson, dan Michael Munn (O'Reilly)
- Pengantar Pembelajaran Mesin dengan Python oleh Andreas C. Müller dan Sarah Guido (O'Reilly)
- Membangun Aplikasi Berbasis Pembelajaran Mesin oleh Emmanuel Ameisen

Saya juga menyarankan untuk menandai bab ini dan kembali ke sini kapan pun Anda ingin meninjau suatu bagian.



BAB 4

PEMBUATAN PROTOTIPE

4.1 PENGERTIAN PROTOTIPE DAN PERAN PENTING

Mari kita hadapi kenyataan pahit: sebagian besar proyek pembelajaran mesin gagal. Dan itu bukan karena kekurangan dana atau kekurangan talenta (meskipun itu juga merupakan masalah umum). Alasan utama kegagalan proyek pembelajaran mesin adalah ketidakpastian yang luar biasa: Persyaratan, cakupan solusi, penerimaan pengguna, infrastruktur, pertimbangan hukum, dan yang terpenting, kualitas hasil, semuanya merupakan hal-hal yang sangat sulit diprediksi sebelum inisiatif baru. Terutama dalam hal hasil, Anda tidak pernah benar-benar tahu apakah data Anda memiliki cukup sinyal sampai Anda melalui proses pengumpulan, persiapan, dan pembersihan data, serta membangun model yang sebenarnya. Banyak perusahaan memulai proyek AI/ML pertama mereka dengan sangat antusias, hingga mereka menyadari bahwa proyek tersebut ternyata sia-sia. Berbulan-bulan kerja dan ribuan dolar dihabiskan hanya untuk mencapai hasil sederhana: Tidak berhasil.

Prototipe adalah cara untuk meminimalkan risiko dan mengevaluasi dampak serta kelayakan kasus penggunaan ML Anda sebelum meningkatkan skalanya. Dengan pembuatan prototipe, proyek ML Anda akan lebih cepat, lebih murah, dan menghasilkan ROI yang lebih tinggi. Dalam buku ini, prototipe adalah produk yang belum selesai dengan tujuan sederhana: validasi. Konsep seperti *Proof of Concept* (POC) atau *Minimum Viable Product* (MVP) memiliki tujuan yang sama, tetapi konsep-konsep ini memiliki latar belakang yang berbeda dan melibatkan lebih dari sekadar konsep validasi.

Ide validasi sangat penting karena Anda tidak perlu membangun prototipe jika tidak memiliki apa pun untuk divalidasi. Dalam konteks proyek AI atau pembelajaran mesin, ada dua dimensi yang ingin Anda validasi untuk kasus penggunaan Anda: Dampak dan Kelayakan. Itulah mengapa sangat penting bagi Anda untuk memiliki gambaran kasar tentang kategori-kategori ini sebelum Anda melanjutkan dan membangun sesuatu. Berikut adalah beberapa contoh umum hal-hal yang dapat Anda validasi dalam konteks pembelajaran mesin dan AI:

- Apakah solusi Anda memberikan nilai bagi pengguna?
- Apakah solusi Anda digunakan sebagaimana mestinya?
- Apakah solusi Anda diterima oleh pengguna?
- Apakah data Anda mengandung sinyal yang cukup untuk membangun model yang bermanfaat?
- Apakah layanan AI bekerja cukup baik dengan data Anda untuk memberikan nilai yang memadai?

Agar menjadi alat validasi yang efektif, prototipe pembelajaran mesin harus menyeluruh. Membuat model yang akurat itu bagus. Membuat model yang bermanfaat itu lebih baik. Dan satu-satunya cara untuk mendapatkan informasi tersebut adalah dengan menghadirkan solusi pembelajaran mesin Anda di hadapan pengguna nyata. Anda dapat membangun prediktor



churn pelanggan terbaik di dunia, tetapi jika bagian pemasaran dan penjualan belum siap untuk menerapkan prediksi tersebut, proyek Anda akan gagal. Itulah mengapa Anda membutuhkan umpan balik pengguna yang cepat.

Saat Anda membuat prototipe solusi bertenaga ML, jangan berkompromi dengan data. Jangan gunakan data yang tidak akan tersedia dalam produksi. Jangan lakukan pembersihan manual yang tidak berskala. Anda dapat memperbaiki model, Anda dapat memperbaiki UI. Tetapi Anda tidak dapat memperbaiki data. Perlakukanlah seperti Anda memperlakukannya dalam skenario produksi.

Sebuah prototipe biasanya harus memiliki cakupan yang terdefinisi dengan baik:

- Tujuan yang jelas: misalnya, "Bisakah kita membangun model yang lebih baik sebagai baseline B kita dalam memprediksi Y dengan beberapa data X?"
- Batas waktu yang ketat: misalnya, "Kami akan membangun model terbaik yang kami bisa dalam waktu maksimal dua minggu"
- Kriteria penerimaan: misalnya, "Prototipe kami lulus uji penerimaan pengguna jika umpan balik F terjadi."

Dalam hal teknologi, sebuah prototipe seharusnya menyerahkan kepada Anda untuk memutuskan tumpukan teknologi mana yang terbaik untuk mencapai tujuan dalam jangka waktu tersebut.

Prototipe memberi Anda wawasan berharga tentang area masalah potensial (misalnya, masalah kualitas data) sekaligus melibatkan para pemangku kepentingan bisnis sejak awal dan mengelola ekspektasi mereka.

4.2 PEMBUATAN PROTOTIPE DALAM KECERDASAN BISNIS

Selama beberapa dekade, proses pengembangan tipikal untuk sistem kecerdasan bisnis adalah metode waterfall klasik: proyek direncanakan sebagai proses linier yang dimulai dengan pengumpulan persyaratan, kemudian beralih ke desain, diikuti oleh rekayasa data, pengujian, dan penerapan laporan atau metrik yang diinginkan oleh bisnis.

Bahkan tanpa AI atau pembelajaran mesin, metode waterfall mencapai batasnya. Hal ini karena beberapa faktor bekerja sama: Persyaratan bisnis menjadi semakin tidak jelas seiring dengan semakin kompleksnya bisnis itu sendiri. Selain itu, teknologinya juga semakin kompleks: Sepuluh tahun yang lalu, Anda harus mengintegrasikan beberapa sumber data, tetapi kini bahkan perusahaan kecil pun harus mengelola lusinan sistem yang berbeda.

Ketika rencana proyek Anda untuk inisiatif BI skala besar sudah siap, seringkali rencana tersebut sudah terlampaui oleh kenyataan. Dengan AI/ML dan ketidakpastian yang menyertainya, segalanya tidak akan menjadi jauh lebih mudah. Untuk mewujudkan solusi berbasis AI yang sukses sebagai bagian dari sistem BI, Anda juga memerlukan prototipe di perangkat Anda. Bagaimana caranya?

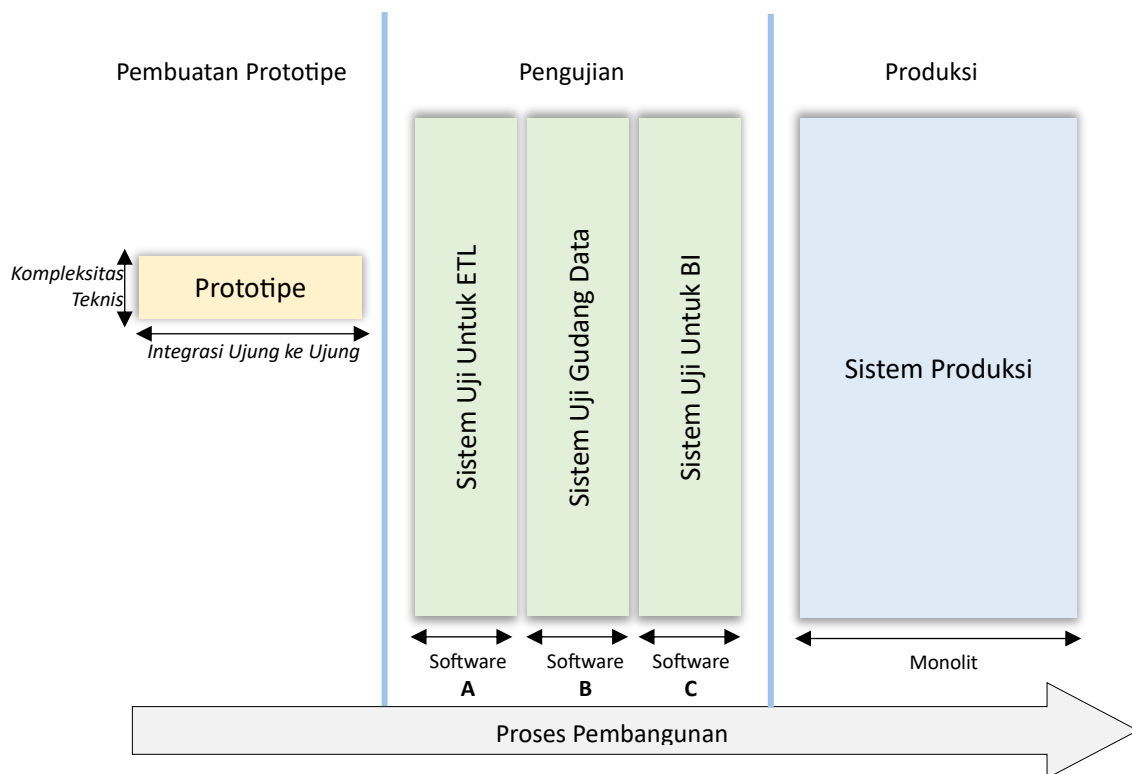
Laporan dalam sistem produksi BI diharapkan secara default andal dan tersedia, serta informasi yang dikandungnya akurat. Jadi, sistem produksi Anda bukanlah pilihan terbaik untuk menguji sesuatu di antara pengguna.

Apa lagi yang kita miliki?



Perusahaan biasanya bekerja dengan sistem pengujian untuk mencoba fitur-fitur baru sebelum memasukkannya ke dalam produksi. Namun, masalah dengan sistem pengujian adalah bahwa sistem tersebut biasanya melibatkan upaya dan tantangan teknis yang sama dengan sistem produksi. Biasanya, seluruh lingkungan produksi direplikasi untuk tujuan pengujian: Gudang data uji, antarmuka pengguna (BI) uji, area uji untuk proses ETL, dll., karena Anda ingin memastikan bahwa apa yang berfungsi pada sistem uji juga berfungsi dalam produksi.

Sesuai definisi kami, sebuah prototipe dibuat sebelum pengujian, seperti yang ditunjukkan pada Gambar di bawah. Untuk prototipe, Anda memerlukan integrasi vertikal yang mendalam, mulai dari penyerapan data melalui layanan analitis hingga antarmuka pengguna, dengan tetap menjaga kompleksitas teknis secara keseluruhan tetap rendah.



Gambar 4.1. Dalam kebanyakan kasus, terutama dengan sistem BI monolitik, tumpukan produksi dan pengujian Anda tidak cocok untuk pembuatan prototipe hanya dalam beberapa hari atau minggu.

Untuk tujuan kita, kita membutuhkan sesuatu yang lebih sederhana dan ringan. Apa pun tumpukan teknologi yang Anda gunakan untuk ini, Anda harus memastikan bahwa:

Tumpukan teknologi mendukung ketiga lapisan arsitektur kasus penggunaan AI (data, analitik, dan antarmuka pengguna) untuk memastikan bahwa data tersedia dan sesuai untuk digunakan, model Anda berkinerja cukup baik pada data Anda, dan bahwa pengguna menerima dan melihat nilai dalam solusi Anda.

Tumpukan Anda harus terbuka dan menyediakan konektivitas tinggi untuk memungkinkan integrasi di masa mendatang ke dalam pengujian dan bahkan mungkin sistem



produksi. Aspek ini penting karena, jika berhasil, Anda ingin memungkinkan transisi yang lancar dari fase pembuatan prototipe ke fase pengujian. Jika Anda membuat prototipe di mesin lokal, akan sulit untuk mentransfer alur kerja dari komputer Anda ke server jarak jauh dan mengharapkannya berjalan semulus itu.

Alat seperti Docker akan membantu Anda mempermudah transisi, menambahkan hambatan teknis tambahan yang tidak lagi berkaitan dengan hipotesis validasi awal Anda. Oleh karena itu, platform prototipe yang sempurna untuk kasus penggunaan BI berbasis AI menawarkan konektivitas tinggi, banyak layanan integrasi yang tersedia, dan biaya investasi awal yang rendah. Platform cloud telah terbukti memenuhi persyaratan ini dengan cukup baik.

Selain itu, Anda sebaiknya menggunakan platform yang memberikan akses mudah ke berbagai layanan AI/ML yang telah dibahas di Bab 2. Jika Anda memiliki data yang tersedia, pembuatan prototipe model ML sebaiknya tidak lebih dari 2-4 minggu, terkadang bahkan hanya beberapa hari. Gunakan Auto ML untuk tugas regresi/klasifikasi, AI-as-a-Service untuk layanan komoditas seperti pengenalan karakter, dan model yang telah dilatih sebelumnya untuk aplikasi Pembelajaran Mendalam khusus daripada melatihnya dari awal.

Untuk menghindari pemborosan waktu dan uang yang tidak perlu pada proyek pembelajaran mesin Anda berikutnya, buat prototipe terlebih dahulu sebelum memulai sepenuhnya.

4.3 TOOLKIT PROTOTIPE AI UNTUK BUKU INI

Kita akan menggunakan Microsoft Azure sebagai platform untuk pembuatan prototipe dalam buku ini. Ada banyak alat lain seperti AWS atau Google Cloud Platform yang dapat Anda gunakan. Platform-platform ini memiliki fitur yang sangat mirip, dan apa yang kita lakukan dengan Azure pada dasarnya dapat Anda lakukan dengan platform lain mana pun.

Saya memilih Azure karena dua alasan berikut:

- Banyak bisnis menggunakan tumpukan Microsoft, dan Azure akan menjadi platform cloud yang paling familiar bagi mereka.
- Azure terintegrasi dengan lancar dengan Power BI, alat BI yang sangat populer di banyak perusahaan.

Anda mungkin bertanya-tanya bagaimana platform cloud lengkap seperti Azure bisa cukup ringan untuk pembuatan prototipe? Alasan utamanya adalah kita tidak menggunakan seluruh platform Azure, tetapi hanya beberapa bagiannya yang disediakan "sebagai layanan", sehingga kita tidak perlu khawatir tentang detail teknis di baliknya.

Layanan utama yang akan kita gunakan adalah Azure Machine Learning Studio, Azure Blob Storage, Azure Compute, dan serangkaian Azure Cognitive Services, penawaran AI-sebagai-Layanan dari Microsoft. Hal baiknya adalah, meskipun kami masih membuat prototipe, infrastrukturnya juga akan mampu menangani skenario produksi. Kami perlu mengubah proses, pemeliharaan, integrasi, dan manajemen solusi, tetapi fondasi teknisnya akan tetap sama. Kami akan membahas hal ini lebih detail di Bab 11.

Saya akan mencoba untuk tidak terlalu bergantung pada fitur yang hanya berfungsi antara Azure dan PowerBI, tetapi tetap menyediakannya agar Anda dapat menghubungkan



model AI Anda sendiri ke BI Anda sendiri. Selain itu, sebagian besar integrasi Azure ke Power BI memerlukan lisensi Pro, yang tidak ingin saya sebutkan.

Mari kita mulai dengan pengaturan untuk Microsoft Azure.

Bekerja dengan Microsoft Azure

Bagian berikut akan memberikan pengantar singkat tentang Microsoft Azure dan memungkinkan Anda mengaktifkan semua alat yang kami butuhkan untuk kasus penggunaan yang dimulai di Bab 7. Meskipun Anda mungkin tidak langsung membutuhkan semua layanan ini, disarankan agar Anda mengaturnya terlebih dahulu agar Anda dapat fokus pada kasus penggunaan yang sebenarnya nanti, alih-alih berurusan dengan beban teknis.

Daftar Microsoft Azure

Untuk membuat dan menggunakan layanan AI mulai Bab 6 dan seterusnya, Anda memerlukan akun Microsoft Azure. Jika Anda baru menggunakan platform ini, Azure memberi Anda akses gratis ke semua layanan yang kami bahas dalam buku ini selama 12 bulan pertama (saat artikel ini ditulis). Anda juga mendapatkan kredit gratis sebesar Rp 2.000.000 untuk layanan yang ditagih berdasarkan penggunaan, seperti sumber daya komputasi atau penerapan model AI. Anda dapat melihat harga gratisnya di sini.

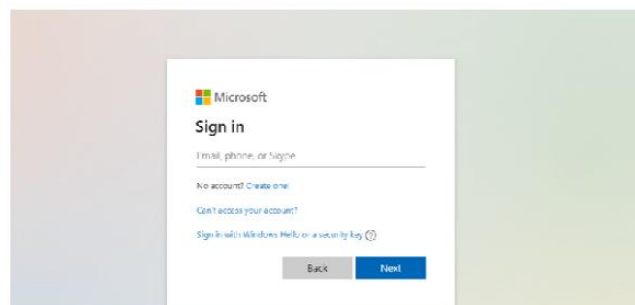
Untuk mendaftar, kunjungi <https://azure.microsoft.com/free/> dan mulai proses pendaftaran.

CATATAN

Jika Anda sudah memiliki akun Azure, Anda dapat melanjutkan ke bagian berikutnya "Membuat Sumber Daya Komputasi Azure". Perlu diketahui bahwa Anda mungkin akan ditagih untuk layanan ini jika Anda tidak sedang dalam uji coba gratis. Jika Anda tidak yakin, hubungi administrator Azure Anda sebelum melanjutkan.

Untuk mendaftar, Anda akan diminta membuat akun Microsoft. Sebaiknya buat akun baru untuk uji coba (dan manfaatkan bonus gratisnya!). Jika Anda memiliki akun perusahaan, Anda dapat mencoba melanjutkannya, tetapi mungkin ada batasan dari perusahaan Anda. Oleh karena itu, mungkin sebaiknya Anda memulai dari awal, bereksperimen sedikit, dan setelah semuanya beres, lanjutkan dengan akun perusahaan Anda.

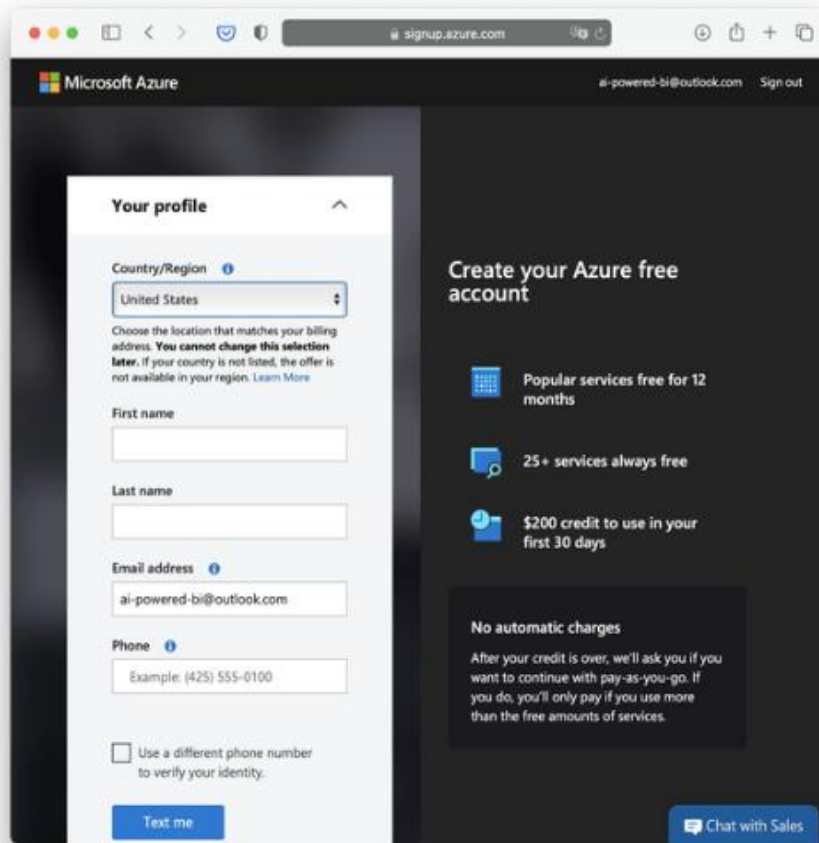
Di halaman berikutnya, pilih "Buat akun!"



Gambar 4.2. Placeholder keterangan gambar



Selanjutnya, masukkan alamat email Anda dan klik "Berikutnya". Buat kata sandi dan lanjutkan. Setelah mengonfirmasi bahwa kami bukan robot, Anda akan melihat layar berikut, seperti yang ditunjukkan pada Gambar di bawah:

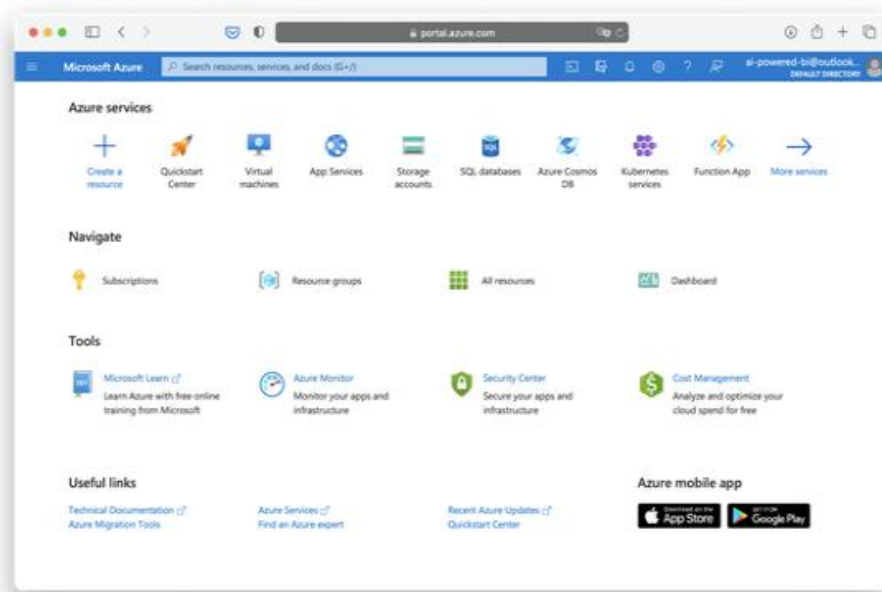


Gambar 4.3. Placeholder keterangan gambar

Anda harus mengonfirmasi akun Anda dengan nomor telepon yang valid. Masukkan informasi Anda dan klik "Kirim SMS" atau "Telepon". Masukkan kode verifikasi dan klik "Verifikasi kode". Setelah verifikasi berhasil, periksa syarat dan ketentuan, lalu klik "Berikutnya".

Kita hampir selesai! Perintah terakhir akan meminta Anda untuk memberikan informasi kartu kredit Anda. Mengapa demikian? Sama halnya dengan semua platform cloud utama. Setelah kredit gratis Anda habis atau periode gratis berakhir, Anda akan dikenakan biaya untuk penggunaan layanan ini. Namun, kelebihan Microsoft untuk pemula adalah Anda tidak akan ditagih secara otomatis. Ketika periode gratis Anda berakhir, Anda harus memilih penagihan bayar sesuai pemakaian.

Setelah berhasil mendaftar, Anda akan melihat layar berikut di portal Azure, seperti yang ditunjukkan pada Gambar di bawah:



Gambar 4.4. Placeholder keterangan gambar

Selamat datang di Microsoft Azure! Anda dapat mengakses beranda ini kapan saja dengan mengunjungi portal.azure.com.

Mari kita jelajahi beranda portal Azure dengan cepat:

1. Bilah menu. Anda dapat membuka menu portal di sebelah kiri atau menavigasi ke pengaturan akun Anda di kanan atas.
2. Layanan Azure: Layanan yang direkomendasikan untuk Anda. Tampilannya mungkin berbeda untuk Anda, terutama jika Anda baru mendaftar.
3. Buat sumber daya: Ini akan menjadi salah satu tombol yang paling umum digunakan. Anda menggunakannya untuk membuat sumber daya baru seperti layanan di Azure.
4. Navigasi dan semua item menu lainnya: Membawa Anda ke berbagai pengaturan. Anda tidak memerlukannya untuk saat ini.

Ketika Anda melihat layar ini, Anda telah siap dengan pengaturan untuk Microsoft Azure. Jika Anda mengalami masalah dalam menyiapkan akun Azure, saya merekomendasikan sumber daya berikut:

<https://azure.microsoft.com/en-us/resources/videos/sign-up-for-microsoft-azure/>

<https://azurelessons.com/create-azure-free-account/>

Buat Ruang Kerja Azure Machine Learning Studio

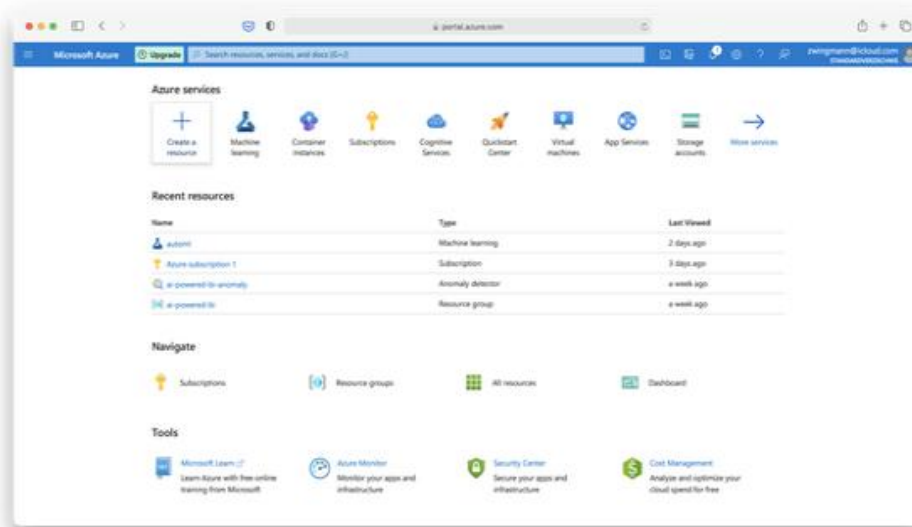
Azure Machine Learning Studio akan menjadi meja kerja kita di Microsoft Azure untuk membangun dan menerapkan model pembelajaran mesin kustom serta mengelola set data. Hal pertama yang perlu kita lakukan adalah membuat ruang kerja Azure Machine Learning.

Ruang kerja ini merupakan sumber daya dasar di akun Azure kita yang berisi semua eksperimen, pelatihan, dan model pembelajaran mesin yang telah Anda terapkan. Sumber daya ruang kerja ini terkait dengan langganan Azure Anda. Meskipun Anda dapat membuat dan mengelola layanan ini secara terprogram, kita akan menggunakan portal Azure untuk



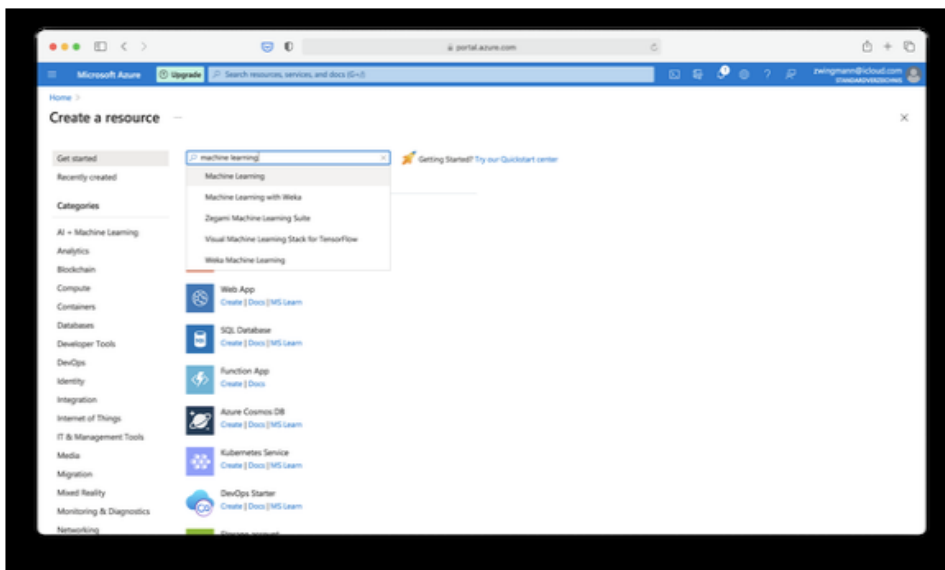
menavigasi prosesnya. Ingat, semua yang Anda lakukan di sini dengan mouse dan keyboard juga dapat dilakukan melalui skrip otomatis jika Anda membutuhkannya nanti.

Masuk ke akun Microsoft yang Anda gunakan untuk langganan Azure dan kunjungi portal.azure.com. Klik tombol "Buat Sumber Daya", seperti yang ditunjukkan pada Gambar di bawah.



Gambar 4.5. Placeholder keterangan gambar

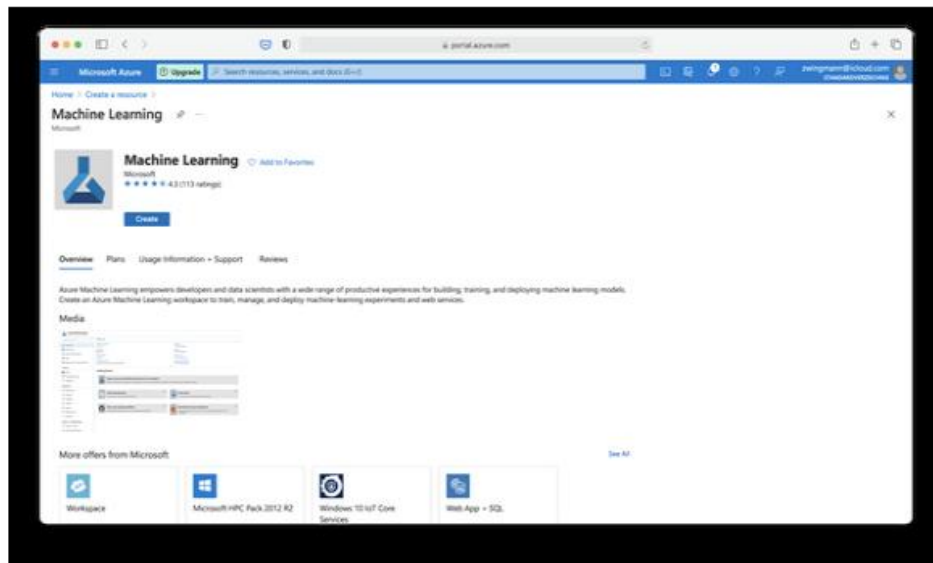
Sekarang, gunakan bilah pencarian untuk menemukan "Machine Learning" dan pilih.



Gambar 4.6. Placeholder keterangan gambar

Di bagian Pembelajaran Mesin, klik Buat.





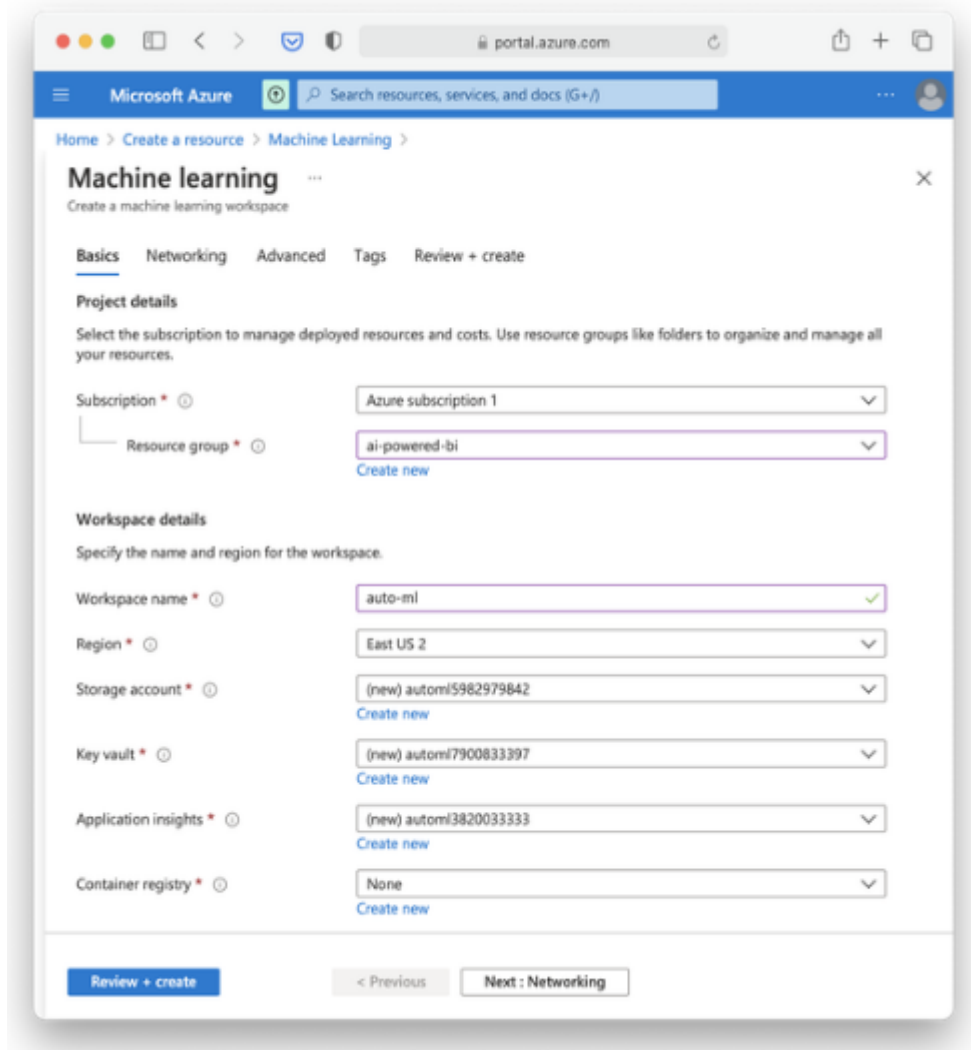
Gambar 4.7. Placeholder keterangan gambar

Anda perlu memberikan beberapa informasi untuk ruang kerja baru Anda:

- Nama Ruang Kerja: Pilih nama unik untuk ruang kerja Anda yang membedakannya dari ruang kerja lain yang Anda buat. Nama proyek biasanya merupakan kandidat yang baik untuk nama ruang kerja. Dalam contoh ini, kami menggunakan automl. Nama tersebut harus unik dalam grup sumber daya yang dipilih.
- Langganan: Pilih langganan Azure yang ingin Anda gunakan.
- Grup sumber daya: Masukkan nama untuk membuat grup sumber daya atau pilih yang sudah ada. Grup sumber daya menggabungkan berbagai sumber daya dalam langganan Azure Anda. Ini bisa berupa departemen Anda misalnya. Dalam contoh saya, saya menggunakan `ai-powered-bi`. Setiap kali Anda melihat referensi ini di suatu tempat, Anda perlu menggantinya dengan nama grup sumber daya Anda sendiri.
- Lokasi: Pilih lokasi fisik yang paling dekat dengan pengguna dan sumber daya data Anda. Berhati-hatilah saat mentransfer data ke luar wilayah geografis yang dilindungi seperti Uni Eropa. Sebagian besar layanan biasanya tersedia pertama kali di wilayah AS.

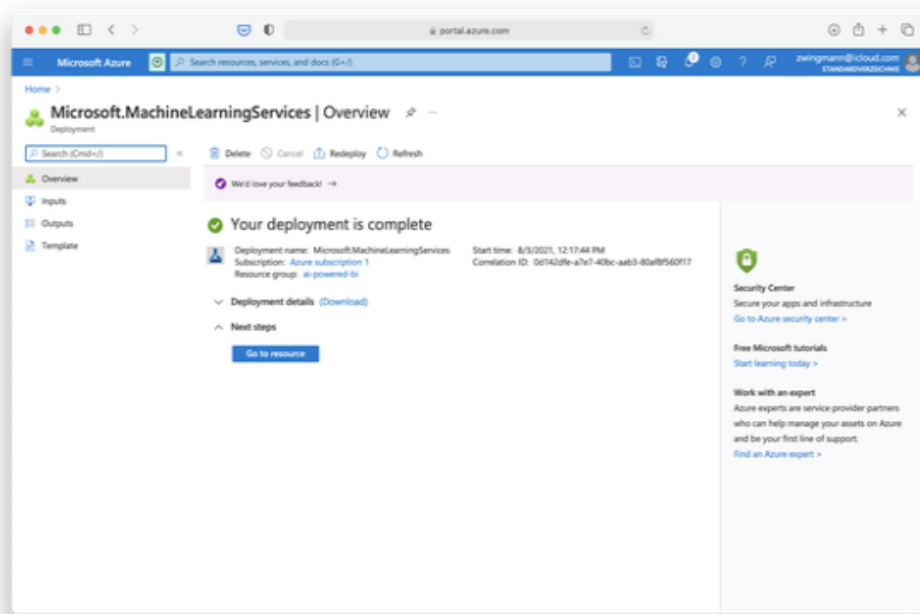
Setelah Anda memasukkan semuanya, tekan "Tinjau + buat". Setelah validasi awal berlalu, tekan buat lagi.





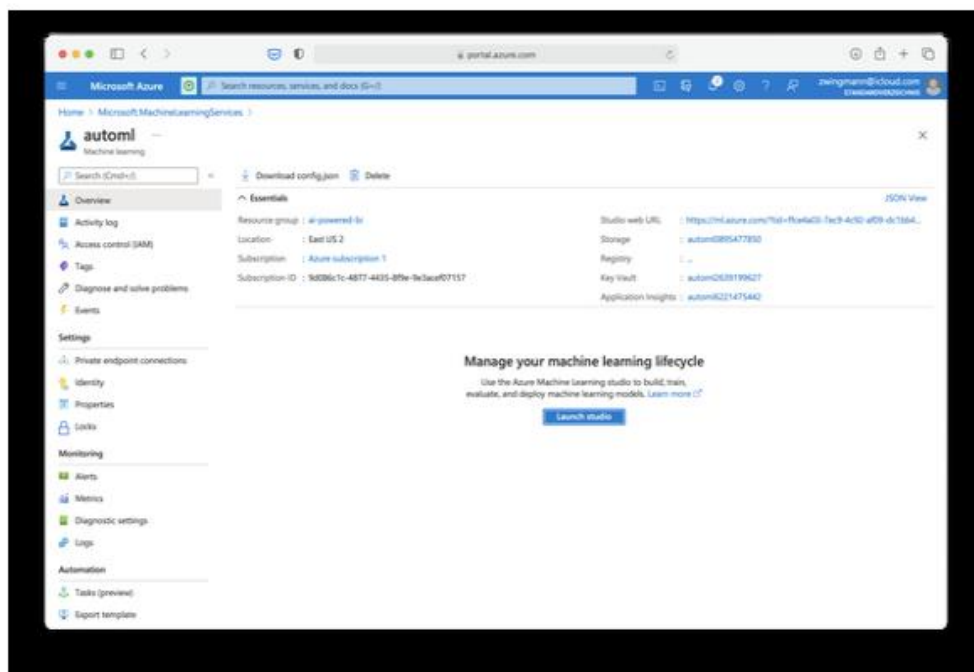
Gambar 4.8. Placeholder keterangan gambar

Membuat ruang kerja Anda di cloud Azure dapat memakan waktu beberapa menit. Setelah proses selesai, Anda akan melihat pesan sukses seperti pada Gambar di bawah.



Gambar 4.9. Placeholder keterangan gambar

Untuk melihat ruang kerja baru Anda, klik "Buka sumber daya". Ini akan membawa Anda ke halaman sumber daya Pembelajaran Mesin seperti pada Gambar di bawah:

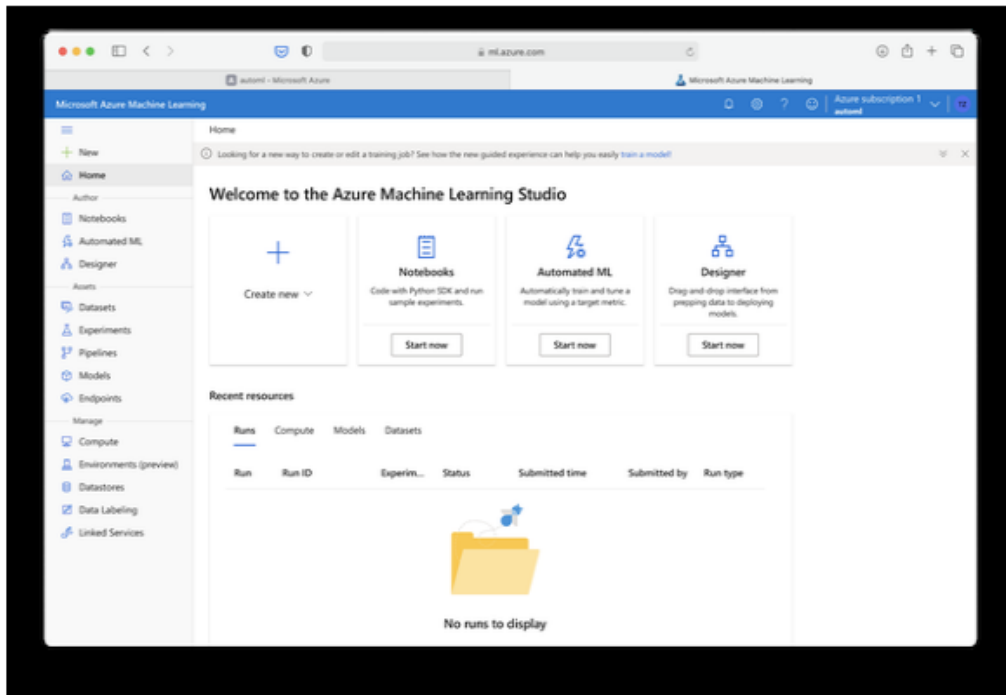


Gambar 4.10. Placeholder keterangan gambar

Kita sekarang telah memilih ruang kerja yang telah dibuat sebelumnya. Verifikasi Langganan dan grup sumber daya Anda di sini. Jika kita ingin melatih dan menerapkan model pembelajaran mesin melalui kode secara terprogram, kita sekarang juga dapat membuka



editor kode favorit kita dan melanjutkan dari sana. Namun karena kita ingin melanjutkan perjalanan tanpa kode, kita klik "Luncurkan Studio" untuk melatih dan menerapkan model Pembelajaran Mesin tanpa menulis satu baris kode pun. Ini akan membawa kita ke layar selamat datang studio Pembelajaran Mesin. Anda juga dapat mengaksesnya langsung melalui ml.azure.com.



Gambar 4.11. Placeholder keterangan gambar

Studio Pembelajaran Mesin adalah antarmuka web yang mencakup berbagai alat pembelajaran mesin untuk menjalankan skenario ilmu data bagi praktisi ilmu data dari semua tingkat keahlian. Studio ini didukung di semua peramban web modern (Tidak, ini tidak termasuk Internet Explorer).

Layar selamat datang terbagi menjadi tiga area utama:

- Di sebelah kiri, Anda akan menemukan bilah menu dengan akses ke semua layanan di dalam ML Studio.
- Di bagian atas, Anda akan menemukan beberapa layanan yang disarankan dan Anda dapat langsung membuat lebih banyak layanan dengan mengklik buat baru.
- Di bagian bawah, Anda akan melihat ikhtisar layanan atau sumber daya yang baru diluncurkan di dalam ML Studio (yang seharusnya masih kosong saat ini)

Membuat Sumber Daya Komputasi Azure

Sejauh ini kita telah membuat langganan Azure dan ruang kerja untuk Studio Pembelajaran Mesin. Anda dapat menganggap hal-hal ini masih sebagai "overhead". Sejauh ini, kita belum meluncurkan layanan atau sumber daya apa pun yang benar-benar berfungsi. Hal itu akan segera berubah.



Dalam langkah-langkah berikut, kita akan membuat apa yang disebut "sumber daya komputasi". Bayangkan sumber daya komputasi sebagai mesin virtual (atau kluster mesin virtual) tempat beban kerja aktual pekerjaan kita berjalan. Pada platform cloud modern seperti Microsoft Azure, sumber daya komputasi dapat dibuat dan dihapus hanya dalam beberapa klik (atau perintah dari baris perintah) dan akan diprovisi atau dimatikan dalam hitungan detik. Anda dapat memilih dari berbagai pengaturan mesin mulai dari komputer kecil yang murah hingga mesin berat dengan dukungan GPU.

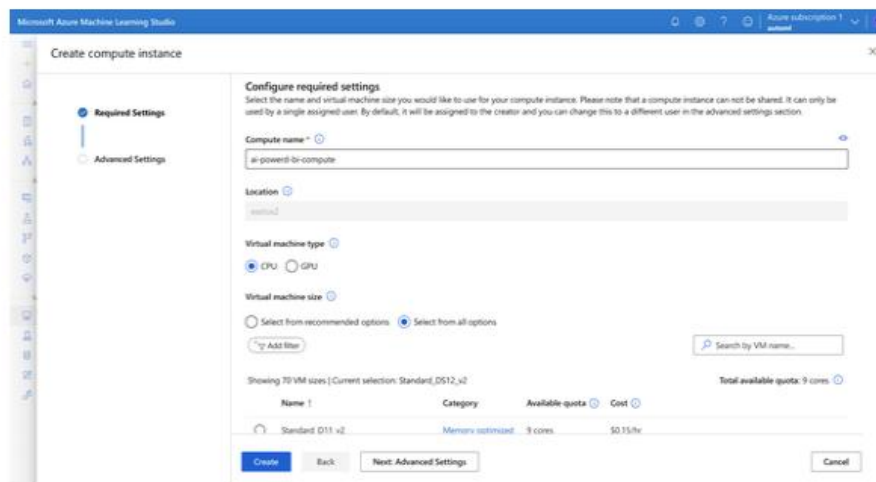
CATATAN

Setelah Anda membuat sumber daya komputasi dan sumber daya ini online, Anda akan dikenakan biaya sesuai harga yang tercantum dalam daftar sumber daya komputasi (misalnya, Rp 6.000/jam). Jika Anda masih menggunakan kredit Azure gratis, biaya ini akan dipotong. Jika tidak, kartu kredit Anda akan dikenakan biaya sesuai ketentuan. Kami hanya membutuhkan instans komputasi ini selama kurang lebih satu jam untuk menjalankan simulasi. Setelah Anda menyelesaikan bab ini, pastikan untuk menghapus sumber daya Anda. Untuk melakukannya, buka Kelola → Komputasi dan pilih sumber daya komputasi Anda dari daftar. Pilih "hentikan" untuk menjeda layanan komputasi atau "hapus" untuk menghapusnya sepenuhnya.

Kita akan membuat sumber daya komputasi yang dapat digunakan untuk semua contoh dalam buku ini. Dalam praktiknya, Anda mungkin ingin menyiapkan sumber daya komputasi yang berbeda untuk proyek yang berbeda, juga untuk mendapatkan transparansi yang lebih baik tentang proyek mana yang mengeluarkan biaya yang mana. Namun, dalam contoh prototipe kita, satu sumber daya saja sudah cukup, dan Anda dapat dengan mudah melacak apakah sumber daya tersebut berjalan atau tidak sehingga tidak menghabiskan anggaran uji coba gratis Anda.

Kita dapat membuat sumber daya komputasi langsung di Azure ML Studio. Klik "Compute" di bilah navigasi kiri. Kemudian klik "Create new compute" dan layar seperti pada Gambar di bawah akan muncul. Untuk memilih jenis mesin, pilih "Select from all options".

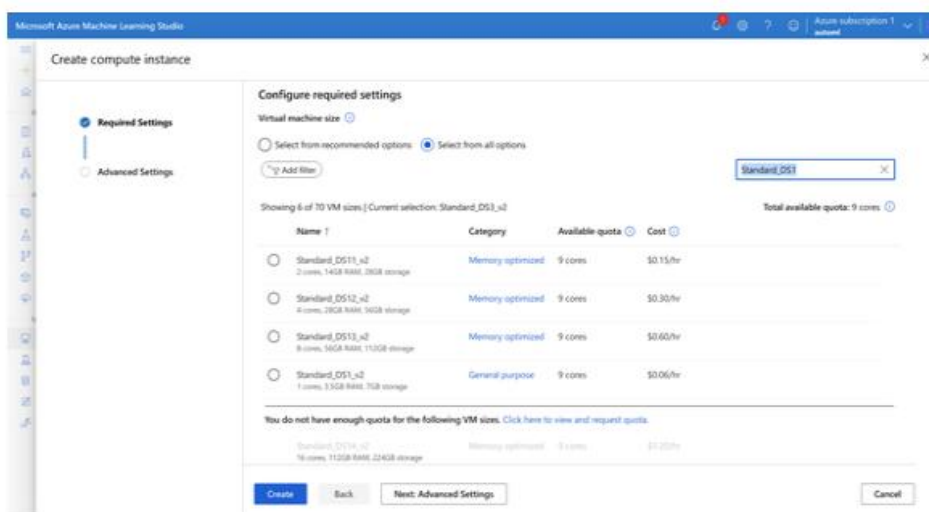




Gambar 4.12. Membuat instans komputasi baru

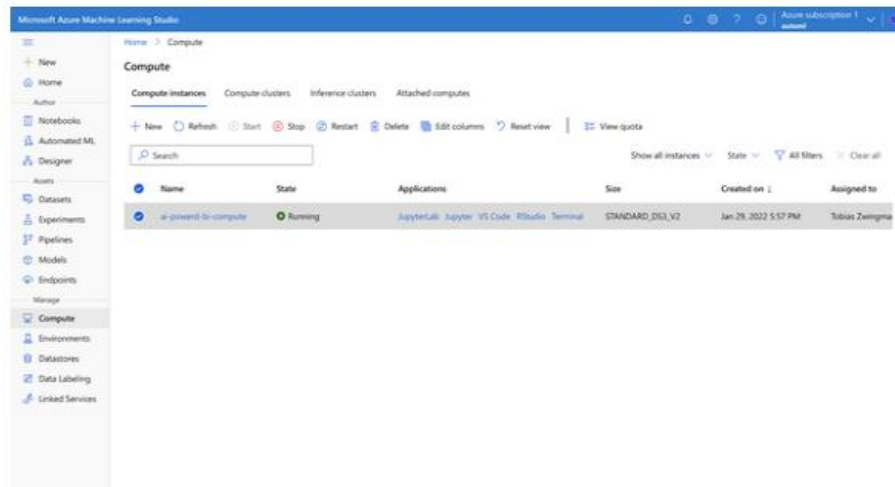
Di sini Anda dapat melihat semua jenis mesin yang tersedia. Semakin tinggi performa yang Anda alokasikan, semakin cepat pelatihan pembelajaran mesin biasanya. Sumber daya termurah yang tersedia adalah mesin "Standard_DS1_v2" dengan 1 inti, RAM 3,5 GB, dan penyimpanan 7 GB yang biayanya sekitar Rp 600 per jam di wilayah AS Timur, seperti yang ditunjukkan pada Gambar 7.10.

Ketik "Standard_DS1" di kolom "Cari berdasarkan nama VM" dan pilih mesin dengan harga terendah, seperti yang terlihat pada Gambar di bawah. Anda sebenarnya dapat memilih jenis mesin apa pun di sini, pastikan saja harganya terjangkau, karena saat ini kita tidak membutuhkan performa tinggi.



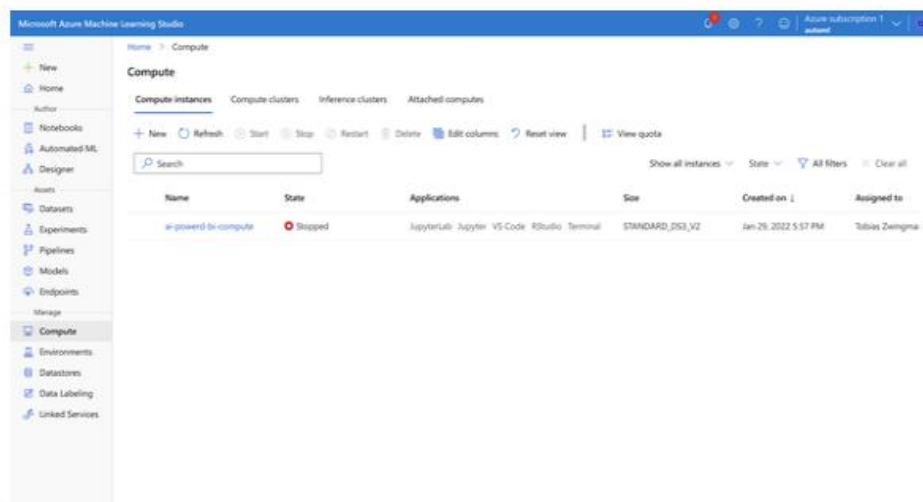
Gambar 4.13. Pilih mesin Standard_DS1

Klik "Buat". Anda akan dibawa kembali ke halaman ikhtisar sumber daya komputasi Anda. Tunggu hingga sumber daya dibuat, yang biasanya membutuhkan waktu antara 2 dan 5 menit. Sumber daya komputasi akan dimulai secara otomatis setelah diprovisi.



Gambar 4.14. Placeholder keterangan gambar

Karena kita tidak akan membutuhkan sumber daya ini hingga Bab 4, pilih sumber daya tersebut dan klik "Hentikan" untuk mencegahnya terbebani. Pada akhirnya, halaman ringkasan sumber daya komputasi Anda akan terlihat seperti Gambar di bawah. Seharusnya ada sumber daya komputasi yang telah dibuat tetapi dihentikan.

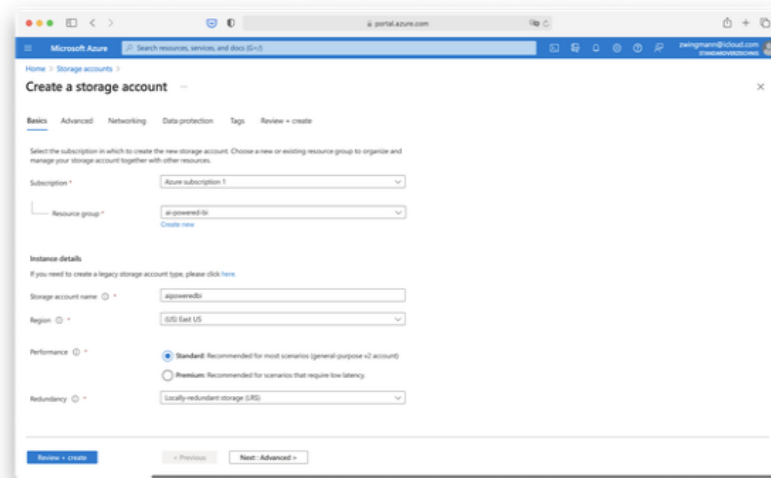


Gambar 4.15. Placeholder keterangan gambar

Membuat Azure Blob Storage

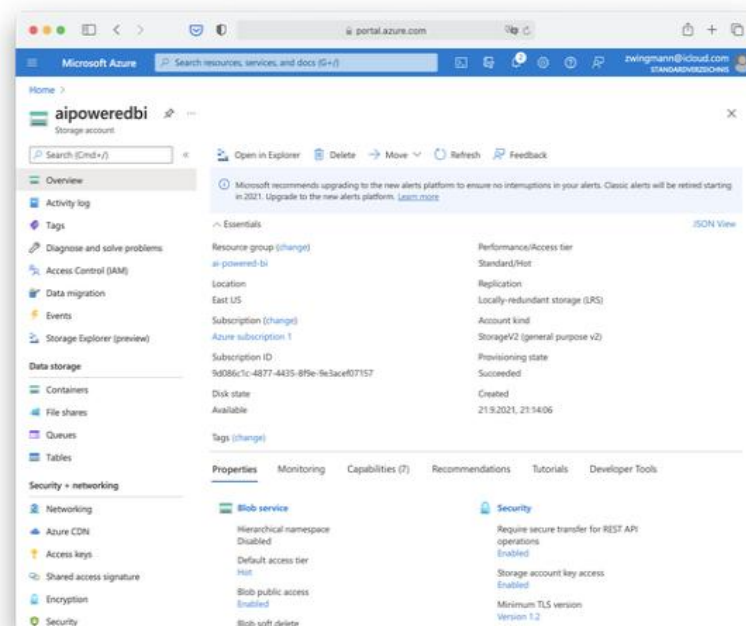
Blok penyusun terakhir yang kita perlukan untuk pengaturan prototipe kita adalah tempat untuk menyimpan dan mengunggah berkas seperti tabel CSV atau gambar. Tempat umum untuk berkas "biner" ini adalah yang disebut penyimpanan Blob, di mana blob merupakan singkatan dari objek besar biner - pada dasarnya penyimpanan berkas untuk hampir semua hal dengan skala yang hampir tak terbatas. Kita akan menggunakan penyimpanan ini terutama untuk menulis keluaran dari model AI kita atau melakukan staging berkas gambar untuk kasus penggunaan di Bab 7.





Gambar 4.16. Membuat akun penyimpanan

Setelah penerapan selesai, klik "Buka sumber daya". Anda akan melihat antarmuka seperti yang ditunjukkan pada Gambar di bawah.

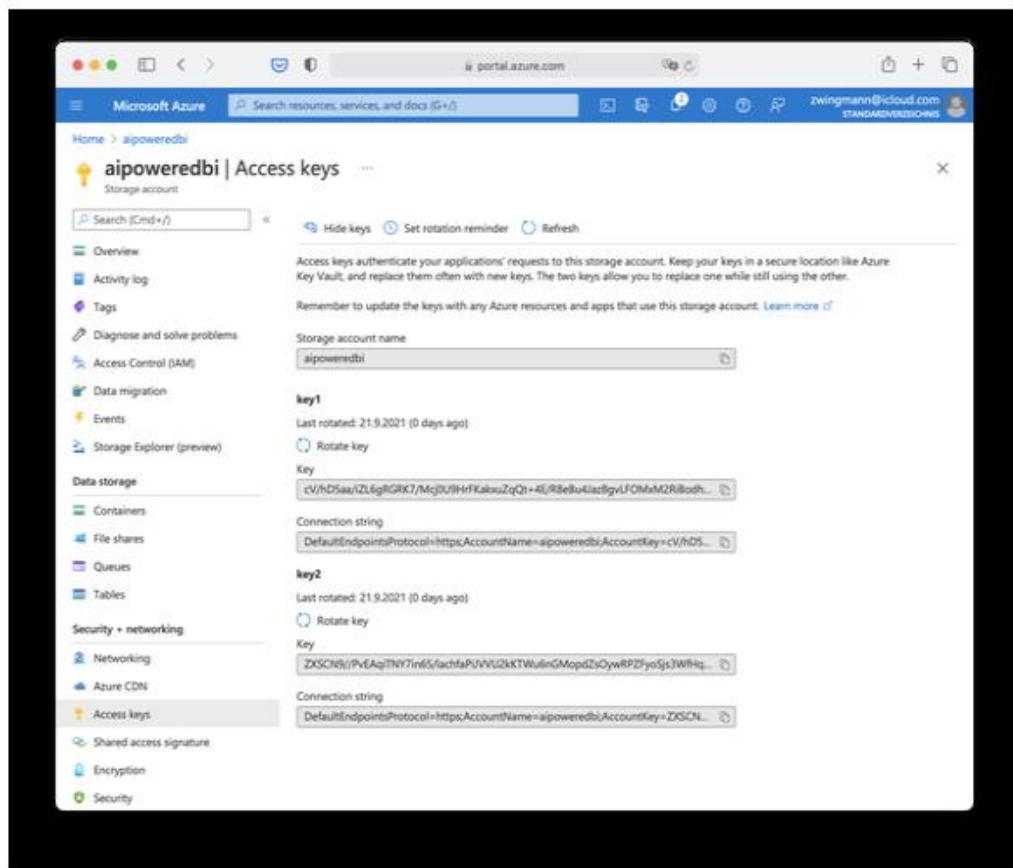


Gambar 4.17. Antarmuka akun penyimpanan Azure

Untuk membuat penyimpanan blob di Azure, kunjungi portal.azure.com dan cari "akun penyimpanan". Klik "Buat" dan buat akun penyimpanan baru di wilayah yang sama dengan lokasi sumber daya studio pembelajaran mesin Anda. Beri nama unik pada akun ini, pilih kinerja standar dan penyimpanan redundan lokal seperti yang ditunjukkan pada Gambar di bawah. Karena ini hanya data uji, kita tidak perlu membayar lebih untuk standar ketersediaan

data yang lebih tinggi. Sekali lagi, kita hanya akan menyimpan data di sini untuk sementara dan biayanya akan lebih dari cukup untuk ditanggung oleh anggaran uji coba gratis Anda.

Arahkan ke "Kunci Akses" seperti yang ditunjukkan pada Gambar di bawah. Anda akan memerlukan kunci akses ini jika ingin mengakses objek di penyimpanan tersebut secara terprogram.



Gambar 4.18. Kunci akses untuk akun penyimpanan Azure

Setelah kita memiliki akun penyimpanan dan kunci untuk memanipulasi data, kita perlu membuat kontainer, semacam folder file yang membantu kita mengelola file.

Arahkan ke "Kontainer" di sisi kanan menu dan klik "+ Kontainer" di bagian atas. Buat kontainer bernama "tabel". Atur tingkat akses ke "Kontainer" agar kita dapat mengakses file nanti melalui alat eksternal seperti Power BI.

Hati-hati, jika Anda menangani data sensitif atau produksi, tentu saja Anda harus memilih "Pribadi" di sini untuk memastikan bahwa otorisasi diperlukan sebelum data dapat diakses. Voilà, kontainer baru akan muncul dalam daftar dan sekarang siap untuk menyimpan beberapa berkas untuk kita.

Bekerja dengan Microsoft Power BI

Untuk lapisan pengguna, kita akan mengandalkan Microsoft Power BI sebagai alat untuk menampilkan laporan, dasbor, dan menampilkan hasil dari model AI kita. Power BI hadir dalam dua versi: Power BI Desktop dan Power BI Service.



Power BI Desktop saat ini hanya tersedia untuk Windows. Ini adalah aplikasi offline yang terinstal di komputer Anda dan memiliki semua fitur yang Anda harapkan dari Power BI. Jika Anda berlangganan Microsoft Office, Power BI kemungkinan besar merupakan bagian darinya. Anda dapat mengunduh Power BI di sini:

<https://powerbi.microsoft.com/id-id/downloads/>

Power BI Service adalah versi online Power BI yang berjalan di semua platform, termasuk Mac. Saat ini, versi ini tidak memiliki semua fitur seperti versi desktop, tetapi fitur-fitur baru ditambahkan secara berkala. Saya tidak dapat menjamin bahwa semua hal yang dibahas dalam buku ini akan berfungsi pada Power BI Service, tetapi patut dicoba jika Anda tidak dapat atau tidak ingin menginstal Power BI di komputer Anda. Anda dapat memulai layanan Power BI di sini:

<https://powerbi.microsoft.com/en-us/getting-started-with-power-bi/>

Jika Anda belum pernah menggunakan Power BI sebelumnya, mungkin ada baiknya menonton kursus pengantar Power BI berdurasi 45 menit dari Microsoft ini, yang memberikan Anda pengantar tingkat tinggi tentang cara kerja alat ini:

Pengantar Power BI

Jika Anda tidak suka menggunakan Power BI atau ingin menggunakan BI Anda sendiri, Anda seharusnya dapat meniru semua yang akan kita bahas dalam studi kasus (kecuali Bab 4 & 5) di alat BI lainnya, asalkan mendukung eksekusi kode skrip Python atau R.



BAB 5

ANALISIS DESKRIPTIF BERTENAGA AI

5.1 KASUS PENGGUNAAN 1: MENGGUERI DATA DENGAN BAHASA ALAMI

Proses berpikir analitis yang umum dilakukan oleh pengguna bisnis sering kali dimulai dengan pertanyaan sederhana seperti:

- Bagaimana penjualan saya bulan lalu?
- Apakah penjualan kita tahun ini lebih banyak daripada tahun lalu?
- Apa produk terlaris?

Pertanyaan-pertanyaan ini sebagian besar bersifat deskriptif: Pengguna perlu memahami status quo terlebih dahulu sebelum mereka dapat mendalami analisis atau bahkan mengantisipasi kejadian di masa mendatang.

Cara "klasik" untuk menyelesaikan masalah ini adalah dengan analisis yang menghasilkan sejumlah laporan statis untuk pertanyaan-pertanyaan yang paling umum. Cara lain untuk menjawab masalah ini adalah dengan BI swalayan. Alih-alih membuat semua laporan dan visualisasi penting terlebih dahulu, pengguna bisnis dapat memilah dan membagi data serta membuat visualisasi yang mereka butuhkan sendiri.

Namun, ada dua masalah dengan pendekatan swalayan:

Pertama, bagi analisis bisnis atau perancang BI, sangat sulit untuk mengantisipasi semua pertanyaan paling umum dengan laporan atau visualisasi statis. Masalahnya, tergantung pada masalah yang Anda lihat dan orang yang Anda tanyai, area minat yang tepat bisa sangat berbeda. Dalam lingkungan perusahaan, hal ini biasanya mengarah pada rapat panjang yang melelahkan dengan berbagai pemangku kepentingan bisnis yang membahas plot mana yang akan ditampilkan di dasbor.

Dan kedua, mustahil untuk mengaktifkan setiap alur berpikir yang memungkinkan melalui drill-down yang dapat disesuaikan atau dimensi tambahan. Satu topik sering kali mengarah ke topik lain dan laporan tingkat tinggi seringkali hanya awal dari segudang pertanyaan lainnya. Faktor-faktor ini menyebabkan dasbor yang rumit hanya untuk menampilkan analitik deskriptif dari masa lalu. Selain itu, tidak semua pengguna bisnis memahami cara drill-down atau memilah data. Menggabungkan berbagai ukuran dan dimensi ke dalam laporan khusus masih menimbulkan terlalu banyak hambatan bagi konsumen non-teknis yang bukan analisis data terlatih.

Yang dilakukan pengguna bisnis adalah mengajukan pertanyaan. Di sinilah mereka biasanya akan meminta bantuan analisis data, memecahkan masalah hambatan teknis dengan melibatkan dukungan manusia. Tentu saja, pendekatan ini tidak mudah diterapkan karena di sebagian besar lingkungan perusahaan, biasanya terdapat lebih banyak pengguna bisnis daripada analisis data. Dan meskipun Anda mampu memiliki banyak pengetahuan analitis internal, Anda tentu tidak ingin membuat analisis Anda sibuk dengan tugas-tugas yang repetitif dan mudah diselesaikan.



Sekarang mari kita gambarkan skenario untuk contoh kasus penggunaan kita.

Pernyataan Masalah

Kami sedang mempelajari data penjualan di Power BI. Manajemen penjualan telah menghubungi tim BI karena mereka menyadari tren pendapatan yang positif dan ingin mendapatkan pemahaman yang lebih mendalam tentang apa yang sedang terjadi. Sebagai analis BI, kami ditugaskan untuk memberikan dukungan kepada staf penjualan dan membantu mereka mendapatkan wawasan terkait perkembangan pendapatan di bidang minat mereka. Serupa dengan BI swalayan, kami ingin menyediakan "bantuan untuk swalayan" dan memberi staf penjualan kemungkinan untuk berinteraksi dengan kumpulan data secara lancar dan tanpa hambatan teknis.

Ikhtisar Solusi

Cara yang lebih mudah untuk mendapatkan jawaban dari data secara intuitif adalah dengan membiarkan pengguna mengajukan pertanyaan terhadap data tersebut menggunakan bahasa alami dan mendapatkan jawaban otomatis tanpa bantuan manusia, atau dengan sedikit bantuan manusia. Kebanyakan orang tidak menyadari bahwa mereka telah terlatih dalam metode ini selama bertahun-tahun, atau selama mereka telah menggunakan mesin pencari internet seperti Google. Misalnya, jika pengguna bisnis ingin membandingkan angka penjualan untuk negara tertentu dengan angka dari tahun lalu, mereka mungkin ingin mengajukan kueri seperti:

"Tunjukkan penjualan di AS tahun lalu vs tahun ini."

Faktanya, fitur Bahasa Alami (Natural Language) yang didukung AI memungkinkan pengguna melakukan hal itu. Yang terjadi di balik layar adalah model Bahasa Alami (Natural Language) yang didukung AI menafsirkan masukan pengguna dan mencoba memetakannya ke data yang tersedia dalam kumpulan data, lalu menghasilkan visualisasi yang sesuai. Pengalamannya harus cepat, interaktif, dan lancar.

Kami menggunakan Power BI sebagai alat BI kami dan kami akan memanfaatkan kapabilitas AI bawaan Power BI untuk tujuan kami. Alat BI lain seperti Tableau menawarkan fungsionalitas serupa, tetapi detailnya mungkin berbeda. Bagaimanapun, memilih antarmuka BI yang baik untuk tugas ini adalah kuncinya, karena Anda tidak dapat menambahkan kapabilitas NLP ke perangkat lunak BI jika tidak ada dukungan bawaan untuk ini (atau add-on yang menawarkannya).

Di Power BI, fitur NLP disebut visual Q&A. Visual Tanya Jawab memungkinkan Anda menggunakan bahasa alami untuk mengeksplorasi data dengan kata-kata Anda sendiri. Tanya Jawab bersifat interaktif, dan bahkan bisa menyenangkan. Visual ini tampaknya cocok untuk rumusan masalah kita.

Visual Tanya Jawab dapat diseret ke laporan seperti visual lainnya sehingga pengguna dapat melakukan kueri khusus terhadap data. Alat Tanya Jawab juga dapat ditambahkan sebagai tombol di pita, terpisah dari laporan, dan juga dapat diintegrasikan ke dalam dasbor. Visual Tanya Jawab gratis dan tersedia di Power BI desktop dan layanan Power BI. Perangkat



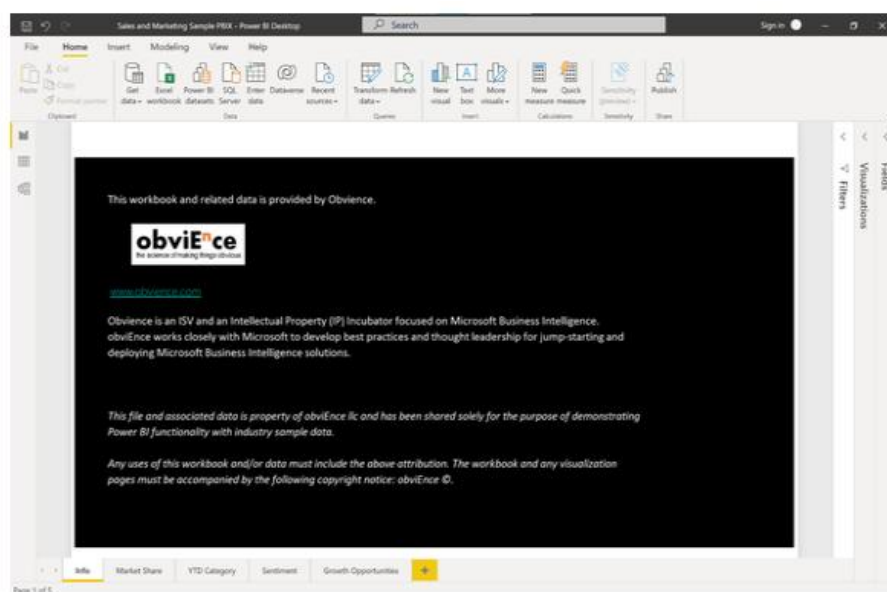
seluler didukung dengan aplikasi Power BI di iOS dan Android. Alat Tanya Jawab berfungsi dengan sumber data berikut: Data yang diimpor, koneksi langsung ke Azure Analysis Services dan SQL Server Analysis Services (melalui gateway) serta dataset Power BI.

Panduan Power BI

Untuk mengikuti contoh berikut, Anda perlu mengunduh file PBIX contoh Penjualan & Pemasaran terlebih dahulu. Dataset tersebut berisi berbagai data penjualan dan pemasaran dari perusahaan manufaktur fiktif. Dataset ini telah diisi sebelumnya dengan beberapa laporan untuk melacak pangsa pasar perusahaan, volume produk, angka penjualan, dan skor sentimen. Untuk studi kasus ini, kami berasumsi bahwa semua produsen dalam dataset ini tergabung dalam satu grup holding sehingga kami dapat menganalisis total pendapatan sebagai metrik yang menarik. Sebagai gantinya, kami tidak terlalu tertarik pada pangsa pasar masing-masing produsen seperti yang disorot dalam skenario awal. Anda dapat mempelajari lebih lanjut tentang dataset ini di situs web.

Mari kita mulai memuat dataset ke Power BI. Anda dapat menggunakan Power BI Desktop atau Power BI Service untuk mengikuti langkah-langkahnya. Di bagian kiri atas Power BI, pilih File > Buka laporan > Telusuri laporan. Kemudian, pilih file PBIX contoh Sales & Marketing.

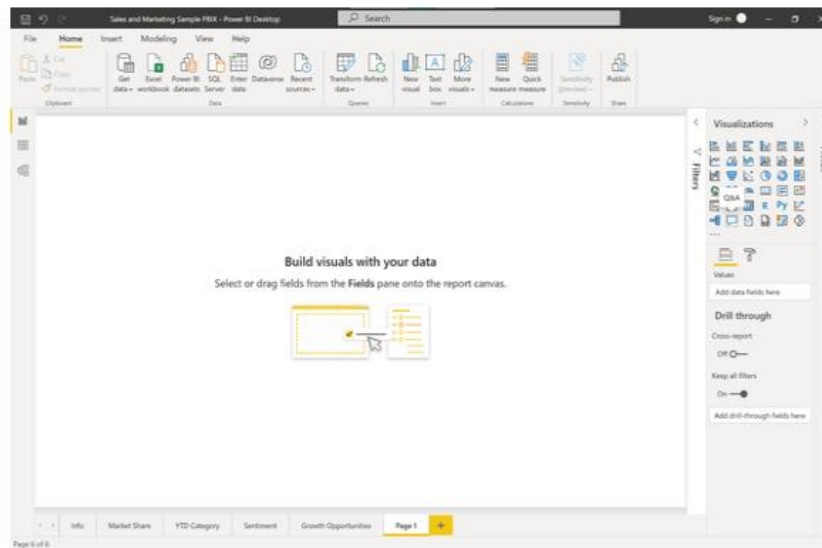
Anda akan melihat layar intro file laporan ini, seperti yang ditunjukkan pada Gambar 5.1.



Gambar 5.1. Contoh Laporan Power BI

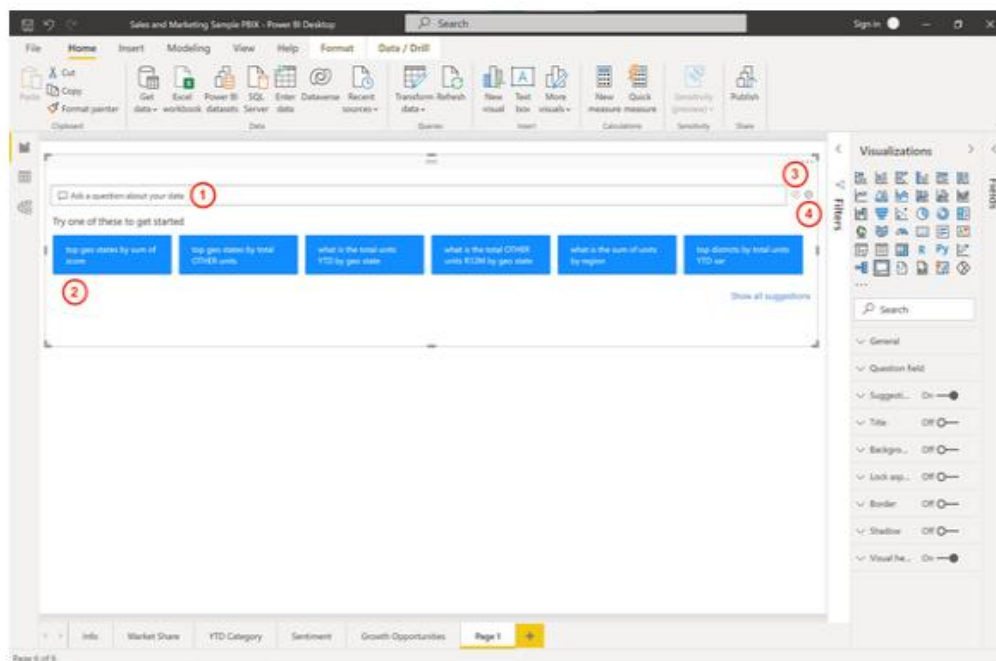
Untuk saat ini, kita tidak akan membahas laporan dan dasbor yang sudah jadi. Sebagai gantinya, kita akan mulai dari awal untuk mengeksplorasi angka pendapatan dan penjualan.

Pertama-tama, kita perlu menambahkan halaman kosong baru ke laporan dengan mengklik tanda plus di bagian bawah layar. Pilih ikon visual Tanya Jawab dari panel visualisasi (Gambar 5.2) dan seret ke kanvas atau cukup klik dua kali ikon tersebut.



Gambar 5.2. Visual Tanya Jawab

Seret batas agar visual memenuhi seluruh lebar laporan Anda. Layer laporan Anda sekarang akan terlihat seperti Gambar 5.3.



Gambar 5.3. Layer Default Visual Tanya Jawab

Sekarang, mari kita jelajahi visual ini lebih detail. Visual Tanya Jawab terdiri dari empat komponen inti:

- Kotak teks (1): di sinilah pengguna mengetik pertanyaan atau kueri mereka dan di mana mereka akan melihat fitur pelengkapan otomatis dan saran otomatis.
- Daftar pertanyaan yang disarankan (2): Daftar yang telah diisi sebelumnya ini akan berisi beberapa contoh pertanyaan yang dapat dijalankan pengguna dengan sekali klik.

- Ikon konversi (3): Ini akan mengonversi output dari alat Tanya Jawab menjadi visual Power BI standar.
- Ikon roda gigi Tanya Jawab (4): Ini akan membuka menu pengaturan yang memungkinkan desainer untuk mengonfigurasi mesin bahasa alami yang mendasarinya.

Mari kita coba visual ini dengan memilih pertanyaan pertama yang disarankan. Klik contoh pertanyaan pertama:

"status geografis teratas berdasarkan jumlah skor"

Power BI akan merespons dengan visual yang tampaknya paling sesuai untuk analisis ini. Dalam hal ini, Power BI menampilkan visual peta yang menunjukkan negara bagian teratas. Kita dapat mengubahnya menjadi hampir semua visual dengan menjadikannya bagian dari kueri kita. Coba:

"negara bagian geografis teratas berdasarkan jumlah skor sebagai diagram batang"

Kueri ini akan menampilkan diagram batang horizontal, bukan peta.

Sekarang, mari kita lihat kotak teks dan jelajahi bagaimana pengguna memasukkan pertanyaan di sini. Alat Tanya Jawab dapat menjawab berbagai pertanyaan, termasuk tetapi tidak terbatas pada pertanyaan yang tercantum dalam Tabel 5.1.

Tabel 5.1. Contoh Perintah untuk Daya Bi Q&A & Visual

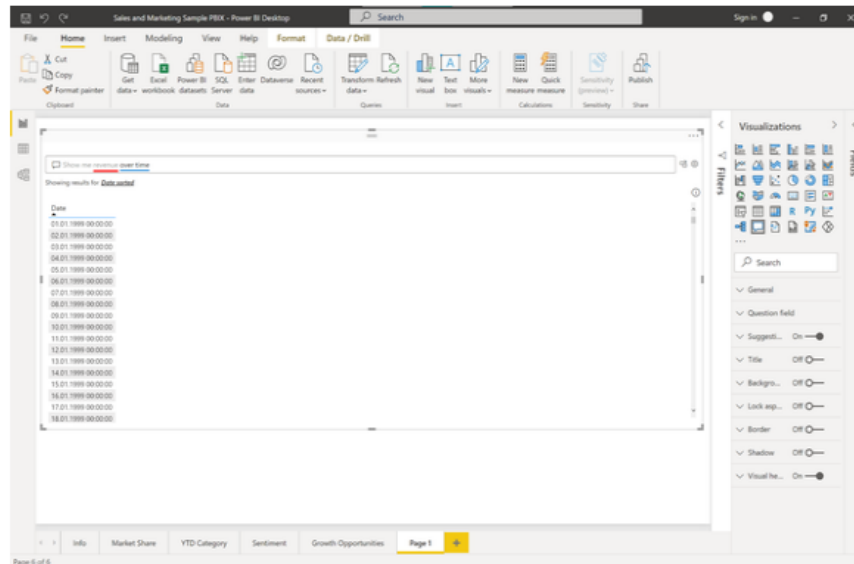
Jenis	Contoh
Ajukan pertanyaan alami	Penjualan mana yang memiliki pendapatan tertinggi?
Pemfilteran tanggal relatif	Penjualan pada tahun lalu
Filter berdasarkan variabel	Penjualan di AS
Filter berdasarkan kondisi	Penjualan dimana kategori produknya adalah Kategori A atau Kategori B
Tampilkan visual tertentu	Penjualan berdasarkan produk dalam bentuk diagram lingkaran
Tampilkan agregasi	Penjualan rata-rata berdasarkan produk
Penyortiran	10 negara teratas berdasarkan penjualan diurutkan berdasarkan kode negara
Perbandingan	Tanggal berdasarkan total penjualan vs total biaya
Waktu	Penjualan seiring berjalannya waktu

Karena kita tertarik dengan tren pendapatan, kita ingin membuat diagram garis sederhana yang menunjukkan total pendapatan dari waktu ke waktu, idealnya dirinci berdasarkan tahun. Untuk mencapai hal ini, mari kita ajukan kueri sederhana ke Power BI:

"Tampilkan pendapatan dari waktu ke waktu"



Tunggu, hasil yang kita lihat pada Gambar 5.4 tidak sesuai dengan yang kita harapkan. Outputnya hanya menampilkan beberapa tanggal dari tahun 1999. Apa yang terjadi di sini?



Gambar 5.4. Menampilkan pendapatan dari waktu ke waktu

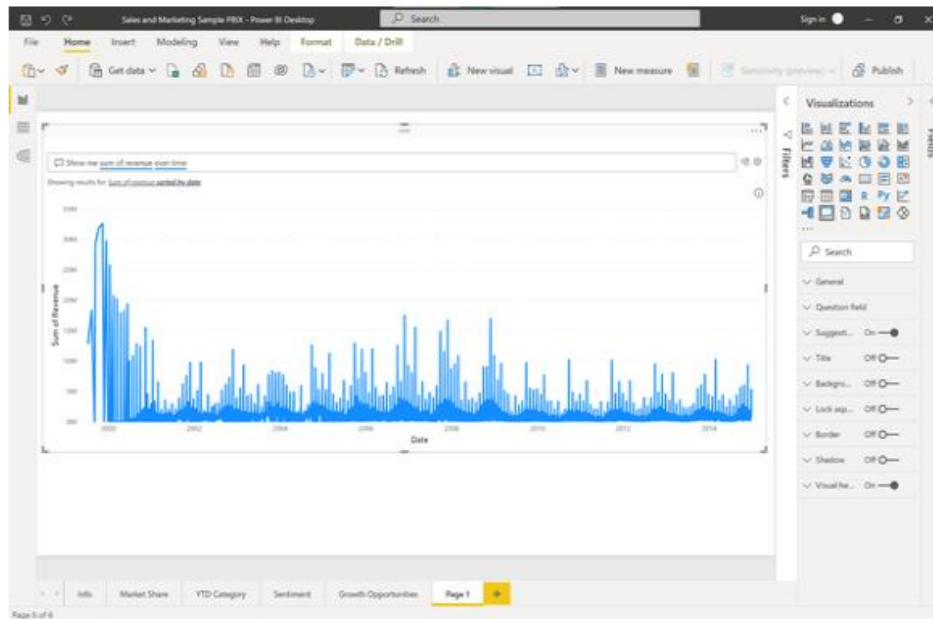
Kita dapat mengetahui lebih lanjut dengan melihat kotak area teks tempat Power BI membantu kita mengidentifikasi frasa bermasalah dalam kueri kita: Garis merah di bawah "pendapatan" menunjukkan bahwa Power BI tidak dapat menginterpretasikan variabel ini. Faktanya, garis bawah di sini mengikuti kode warna yang berbeda:

- Garis bawah biru pekat menunjukkan bahwa sistem berhasil mencocokkan kata dengan kolom atau nilai dalam model data.
- Garis bawah oranye menunjukkan bahwa ekspresi tersebut dicocokkan dengan tingkat keyakinan rendah. Hal ini terjadi jika ekspresi ambigu, misalnya karena ada beberapa kolom yang berisi ekspresi seperti "penjualan".
- Garis bawah merah berarti alat Tanya Jawab tidak dapat mencocokkan kata tersebut dengan apa pun dalam model data.

Jika Anda mengklik kata yang digarisbawahi, Anda akan melihat beberapa saran tentang bagaimana Power BI akan menyelesaikan konflik tersebut. Dalam kasus ekspresi "pendapatan", Power BI akan memberikan saran berikut:

"Maksud Anda: Tunjukkan jumlah pendapatan dari waktu ke waktu".

Seperti yang Anda lihat, Anda harus spesifik dengan kueri jika mengandalkan nilai default. Kita akan memperbaikinya nanti, tetapi untuk saat ini, mari kita terima saran tersebut dan lihat apa yang terjadi pada Gambar 5.5.



Gambar 5.5. Menampilkan jumlah pendapatan dari waktu ke waktu

Hal ini lebih sesuai dengan yang kita harapkan. Power BI menampilkan diagram garis yang mencantumkan total pendapatan dari waktu ke waktu, dibagi berdasarkan hari. Meskipun ini tidak buruk, sulit bagi kita untuk melihat tren besarnya.

Jadi, mari kita lebih spesifik dan bertanya:

"Tunjukkan jumlah pendapatan dari waktu ke waktu berdasarkan tahun"

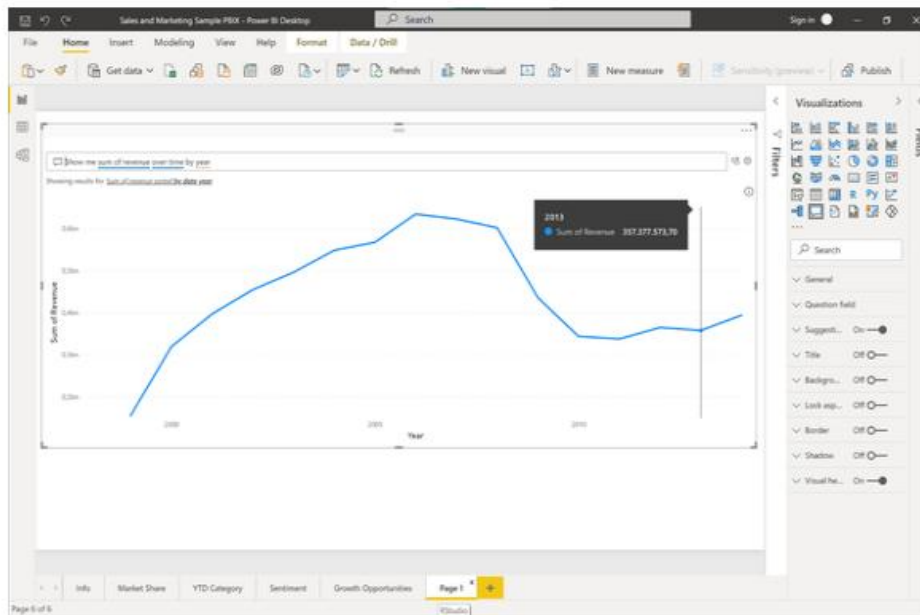
Itu dia. Power BI menampilkan diagram garis yang bersih di mana kita dapat langsung melihat bahwa pendapatan melonjak pada tahun 2006 dan kemudian mengalami tren negatif (lihat Gambar 4.6). Sejak tahun 2010, tren tersebut tampaknya kembali positif.

Kita ingin mempertahankan visual ini sebagai visual tingkat tinggi untuk konsumen, jadi mari kita simpan ini ke visual statis dengan menggunakan ikon "Ubah visual Tanya Jawab ini menjadi visual standar". Power BI akan mengonversi visual Tanya Jawab menjadi diagram garis biasa.

Sekarang saatnya untuk membuat laporan kita terlihat sedikit lebih bagus.

Perkecil visual pendapatan secara horizontal untuk menambah ruang di bawahnya. Tambahkan judul di atasnya yang bertuliskan "Analisis Pendapatan Swalayan".

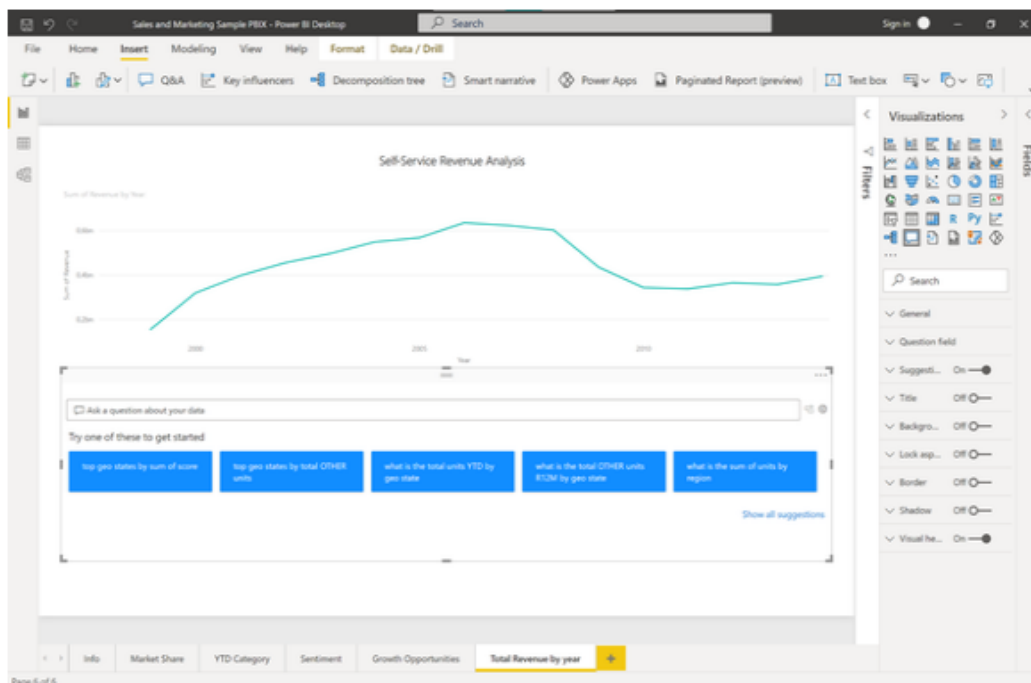




Gambar 5.6. Menampilkan jumlah pendapatan dari waktu ke waktu per tahun

Ubah nama lembar laporan ini dari "Halaman 1" menjadi "Total Pendapatan per Tahun".

Seret dan lepas visual Tanya Jawab lain dari panel visualisasi sehingga muncul di bawah diagram garis total pendapatan. Laporan Anda sekarang akan terlihat seperti Gambar 5.7:



Gambar 5.7. Tata letak laporan Tanya Jawab

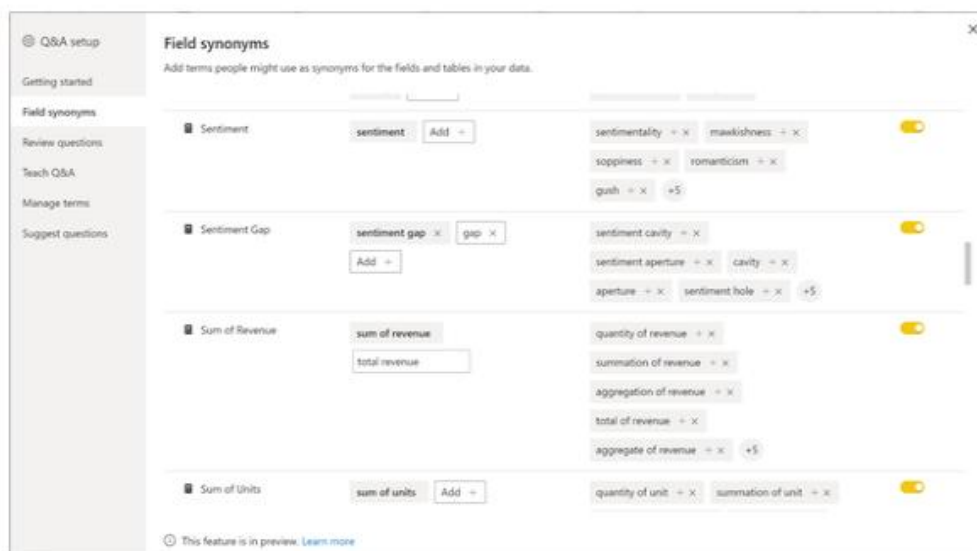
Idenya adalah kami menyediakan ikhtisar tingkat tinggi yang terstandarisasi tentang tren terkini atau metrik bisnis kepada pengguna bisnis. Di bawah ini, kami memungkinkan mereka untuk mengajukan pertanyaan khusus pada data. Untuk melakukannya, kami akan mengedit

visual Tanya Jawab untuk menyarankan pertanyaan yang lebih bermakna dan relevan dengan pendapatan.

Klik ikon roda gigi di sebelah kanan kotak teks untuk mengedit pengaturan visual Tanya Jawab. Anda akan melihat empat opsi di sini: Sinonim bidang, tinjau pertanyaan, ajarkan Tanya Jawab, dan sarankan pertanyaan.

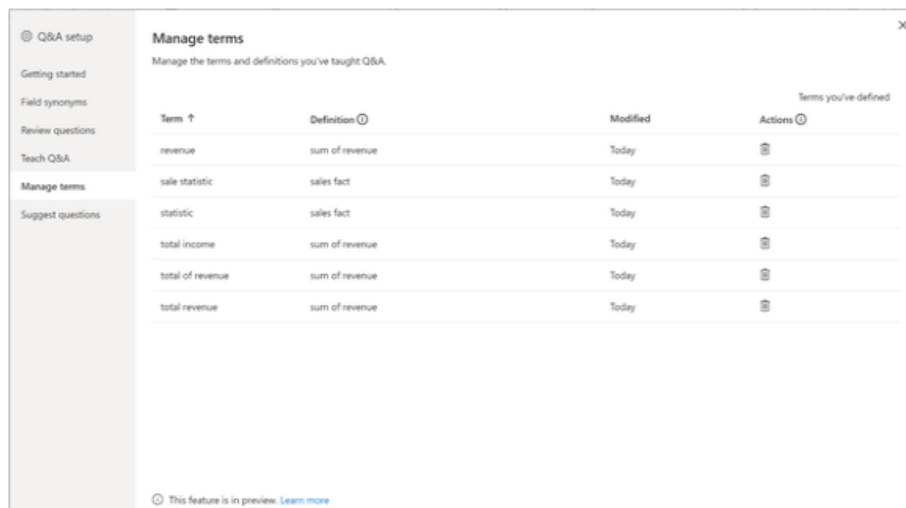
Buka bagian sinonim bidang terlebih dahulu. Power BI akan menampilkan daftar dimensi data Anda dan memungkinkan Anda menambahkan sinonim ke berbagai bidang dalam istilah bisnis yang digunakan pengguna Anda. Untuk tujuan ini, buka daftar SalesFact. Anda akan melihat semua ukuran dalam daftar ini beserta beberapa saran otomatis untuk sinonim dan tombol alih untuk menentukan apakah bidang ini harus disertakan dalam widget Tanya Jawab atau tidak.

Gulir ke bawah ke Jumlah Pendapatan. Tekan tombol "Tambah" dan masukkan "total pendapatan" di sini sebagai sinonim. Tambahkan juga "total pendapatan", "pendapatan", "total pendapatan", dan "total pendapatan" ke sinonim. Kita bisa membahas kolom lainnya juga, tetapi untuk saat ini, mari kita akhiri latihan ini.



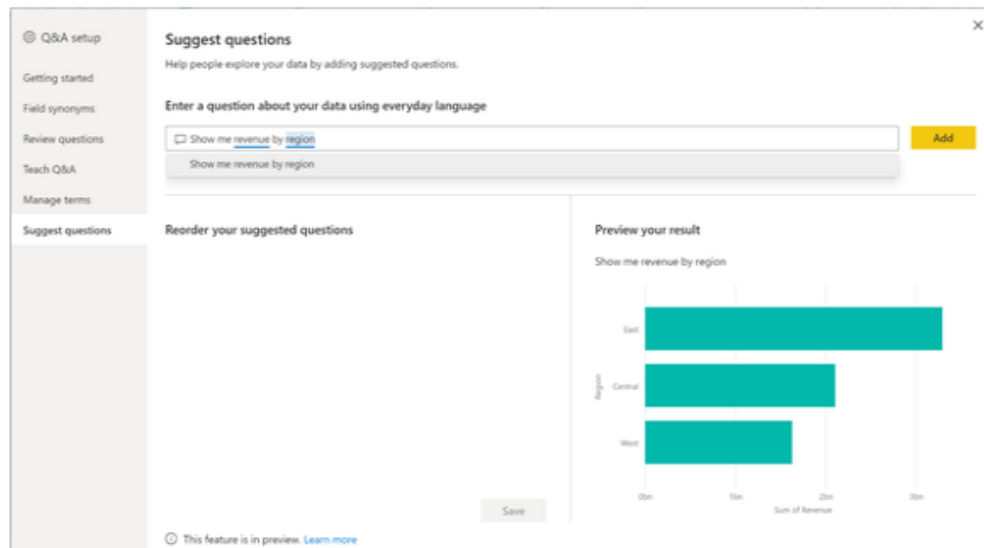
Gambar 5.8. Sinonim kolom Tanya Jawab

Klik "Kelola istilah" di menu sebelah kiri dan Anda akan melihat daftar semua istilah dan definisi yang ditambahkan ke visual Tanya Jawab. Anda dapat melihat contohnya pada Gambar 5.9. Dengan cara ini, Anda dapat dengan mudah melacak setiap perubahan yang Anda buat dan menghapus istilah yang tidak sesuai.



Gambar 5.9. Istilah pengelolaan Tanya Jawab

Kunjungi "Sarankan pertanyaan" di menu sebelah kiri sekarang. Bagian ini memungkinkan kita untuk mengedit pertanyaan yang disarankan kepada pengguna secara default, yang dapat Anda lihat pada Gambar 5.10. Mari kita mulai dengan rincian pendapatan berdasarkan wilayah. Ketik: "Tampilkan pendapatan berdasarkan wilayah" Ini akan menghasilkan plot di jendela pratinjau, seperti yang ditunjukkan pada Gambar 5.10.



Gambar 5.10. Saran Pertanyaan dari T&J

Baiklah, mari kita tambahkan saran tersebut ke saran kita dengan mengklik tombol "Tambah". Pengguna bisnis biasanya sangat pandai belajar melalui contoh. Untuk memberi pengguna lebih banyak intuisi tentang cara menggunakan alat T&J, mari kita tambahkan variasi pada pertanyaan sebelumnya. Nantinya, pengguna bisnis akan merasa nyaman mengganti hal-hal seperti wilayah dengan "produk" atau apa pun yang ada di pikiran mereka. Tambahkan saran berikut:

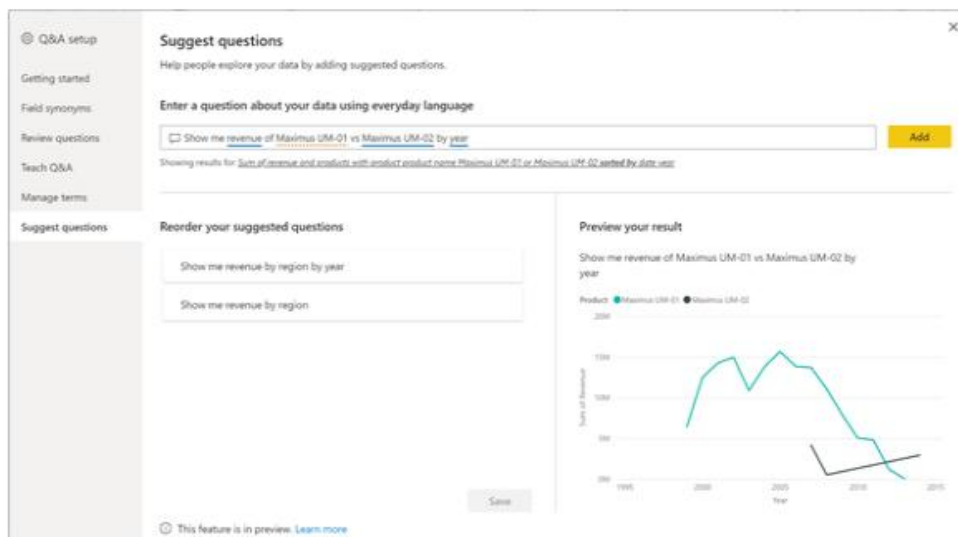
"Tampilkan pendapatan menurut wilayah menurut tahun"

Ini akan menghasilkan tiga diagram garis dan mengarah ke ide untuk membandingkan titik data. Jadi, mari kita tambahkan contoh bagaimana pengguna bisnis dapat membandingkan kinerja penjualan antar produk yang berbeda. Mari kita tambahkan pertanyaan yang sedikit lebih rumit ini:

"Tunjukkan pendapatan Maximus UM-01 vs. Maximus UM-02 per tahun"

Ini akan menghasilkan grafik yang sangat informatif, yang menunjukkan bahwa Maximus UM-01 mengalami penurunan pendapatan sejak puncaknya di tahun 2005, dan produk baru Maximus UM-02 tampaknya perlahan melejit setelah peluncuran pertamanya di tahun 2007.

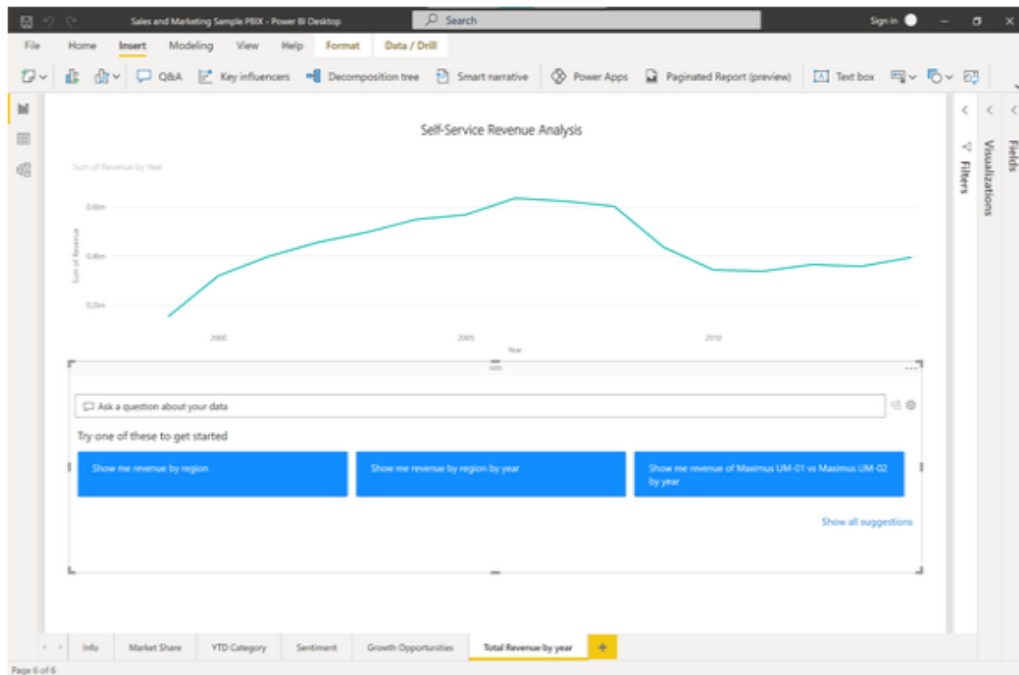
Pada tahun 2012, produk baru ini menyumbang lebih banyak pendapatan daripada pendahulunya, seperti yang dapat Anda lihat pada Gambar 5.11.



Gambar 5.11. Perbandingan pendapatan Tanya Jawab

Untuk mengatur daftar pertanyaan yang disarankan, Anda dapat menyeret pertanyaan ke dalam urutan yang diinginkan. Mari kita letakkan kueri yang paling sederhana di posisi pertama dan yang paling rumit di posisi terakhir. Simpan saran. Tutup jendela pengaturan Tanya Jawab.

Dengan melihat kembali laporan kita, kita sekarang dapat melihat tiga saran yang baru saja kita buat muncul di visual Tanya Jawab. Pengguna bisnis dapat memulai eksplorasi data mereka hanya dengan mengklik saran, lalu mengubah kueri sesuai keinginan, misalnya dengan mengganti wilayah dengan "status geografis". Anda dapat melihat contoh terakhir pada Gambar 5.12.



Gambar 5.12. Laporan akhir dengan visual Tanya Jawab

Pengaturan dasar ini sudah cukup untuk mengirimkan prototipe awal laporan ini kepada sejumlah pengguna bisnis beta. Anda berkesempatan untuk mengumpulkan umpan balik yang berharga, melihat apakah umpan balik ini diterima, dan juga mencari tahu area yang perlu ditingkatkan.

Dalam pengaturan alat Tanya Jawab, Anda dapat menemukan opsi untuk meninjau pertanyaan yang diajukan dalam 28 hari terakhir dan dengan demikian melihat dengan tepat bagaimana orang-orang berencana menggunakan laporan Anda dan sinonim, saran, atau kolom mana yang mungkin masih kurang.

Ini adalah tempat yang tepat untuk memulai iterasi produk Anda dan mengembangkannya sedemikian rupa sehingga Anda dapat menskalakannya ke lebih banyak pengguna. Jika Anda ingin membagikan laporan ini, Anda memerlukan lisensi Power BI Pro. Jika Anda ingin membaca lebih lanjut tentang cara kerja alat Tanya Jawab dan administrasinya secara detail, Anda mungkin akan menemukan sumber daya berikut cukup bermanfaat dari dokumentasi:

- Praktik Terbaik
- Pengantar Alat Tanya Jawab
- Mengedit Skema Linguistik Tanya Jawab dan Menambahkan Frasa di Power BI Desktop

5.2 KASUS PENGGUNAAN 2: MERINGKAS DATA DENGAN BAHASA ALAMI

Analisis bisnis sering kali harus menyusun cerita data untuk mengomunikasikan wawasan kepada para pemangku kepentingan bisnis. Meskipun sebagian besar waktu mereka biasanya dihabiskan untuk membuat visual yang menarik, proses memberi anotasi dengan kesimpulan bisa jadi rumit dan membosankan. Terlalu sering, analisis melewatkan langkah menambahkan

komentar verbal ke visual mereka karena grafik sudah berbicara sendiri, bukan? Tidak, faktanya, banyak grafik dapat diinterpretasikan dengan cara yang salah, terutama ketika menyangkut visualisasi yang lebih kompleks.

Mari kita bayangkan skenario berikut.

Pernyataan Masalah

Berdasarkan visual yang kita buat pada Kasus Penggunaan 1, kita sekarang ingin menambahkan keterangan untuk membantu pemangku kepentingan non-teknis dalam menginterpretasikan data. Kami ingin menyediakan teks untuk memperjelas wawasan, tetapi kami tidak ingin menghabiskan banyak waktu untuk menuliskannya secara manual.

Ikhtisar Solusi

Kemampuan Bahasa Alami tidak hanya memungkinkan kami menganalisis dan menafsirkan bahasa manusia dengan mesin, tetapi juga menghasilkan teks berdasarkan masukan yang telah ditentukan.

Dalam kasus kami, kami dapat menggunakan teknologi AI untuk membuat teks atau anotasi untuk plot tertentu dan data yang mendasarinya. Anotasi tersebut harus membahas poin-poin utama, menunjukkan tren, dan memungkinkan pengeditan bahasa dan format yang disesuaikan dengan audiens tertentu. Di Power BI, fitur ini disebut Narasi Cerdas dan tersedia di Power BI Desktop dan layanan Power BI untuk desainer & pengembang. Fitur ini tersedia untuk semua jenis lisensi dan tidak memerlukan lisensi Pro atau Premium.

Kita akan mengeksplorasi bagaimana kita dapat menggunakan keterangan yang dihasilkan secara otomatis ini untuk menghasilkan kisah data untuk visual yang kita buat di Kasus Penggunaan 1.

Panduan Power BI

Buka file Sales & Marketing PBIX dari Kasus Penggunaan 1. Buat halaman laporan baru dan beri nama "Kisah".

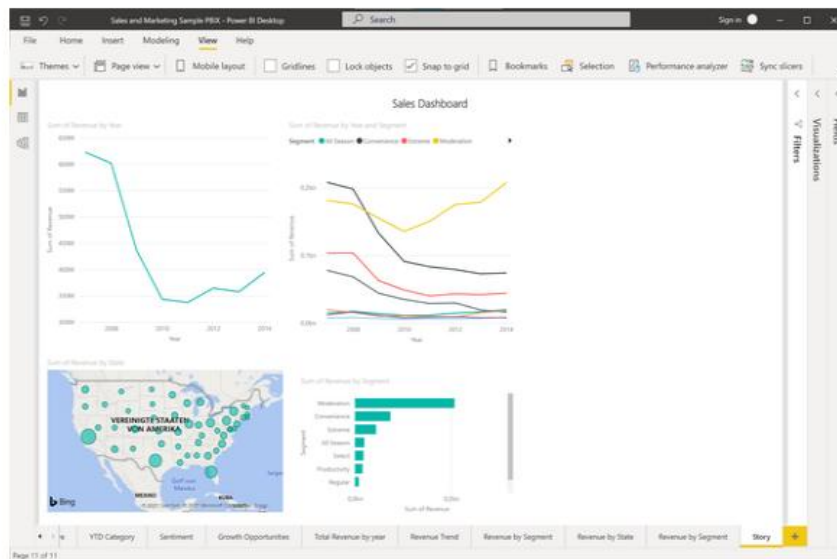
Kita ingin membuat empat visualisasi berikut:

- Pendapatan per tahun
- Pendapatan per negara bagian
- Pendapatan per segmen
- Pendapatan per tahun dan segmen

Anda dapat menggunakan alat Tanya Jawab untuk membuat grafik ini dengan cepat. Cukup konversikan ke visual standar setelah memasukkan kueri dan selesai. Apakah Anda melihat betapa cepatnya pembuatan visual ini dengan kemampuan AI NLP di Power BI dibandingkan dengan pendekatan tradisional yang memilih visual secara manual, dikombinasikan dengan pemetaan data, dan pemfilteran manual?

Susun visual pada halaman sedemikian rupa sehingga terdapat ruang di sebelah kanan untuk narasi. Halaman Anda akan terlihat seperti Gambar 5.13.



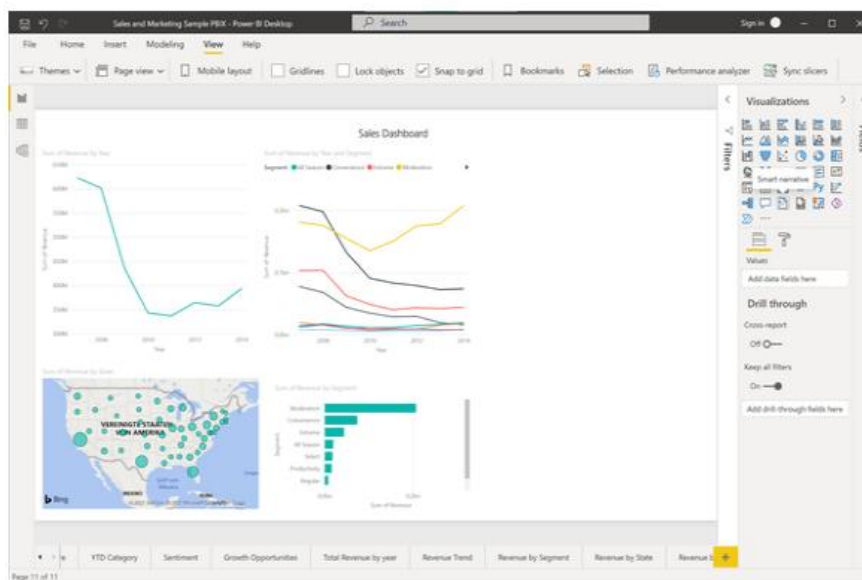


Gambar 5.13. Dasbor Penjualan

Buka panel Visualisasi. Anda akan melihat visual "Narasi Cerdas" seperti pada Gambar 5.14.

CATATAN

Jika Anda tidak melihat "Narasi Cerdas" di panel visualisasi, Anda mungkin menggunakan Power BI versi lama dan perlu mengaktifkan opsi narasi cerdas. Buka File > Opsi dan Pengaturan > Opsi > Fitur Pratinjau dan pastikan visual Narasi Cerdas diaktifkan.

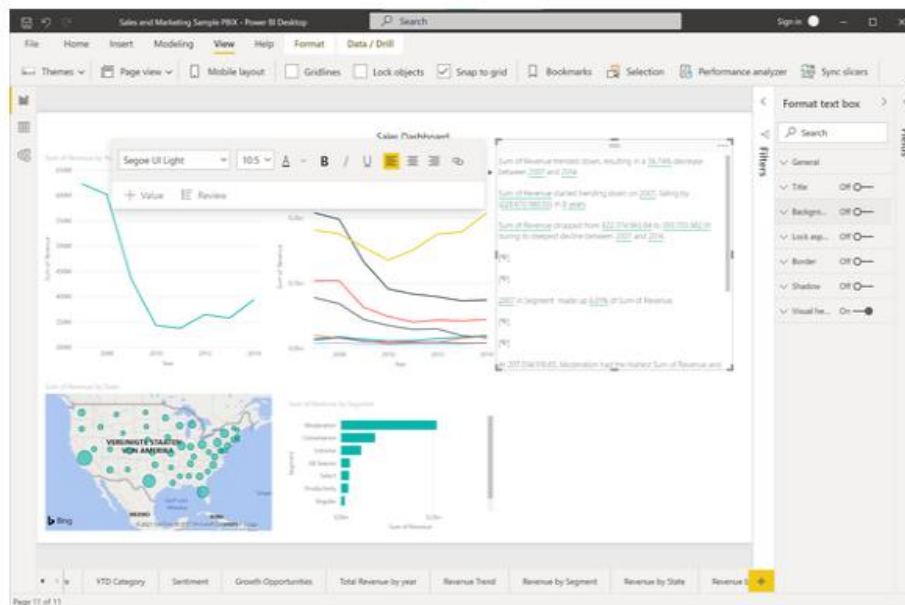


Gambar 5.14. Visualisasi Narasi Cerdas

Klik dua kali pada visual Narasi Cerdas akan menambahkan kotak teks di sisi kanan halaman Anda. Atau, Anda dapat menyeret dan melepas visual ke lokasi yang diinginkan di halaman. Kotak teks ini akan berisi deskripsi yang disarankan Power BI untuk semua visualisasi di

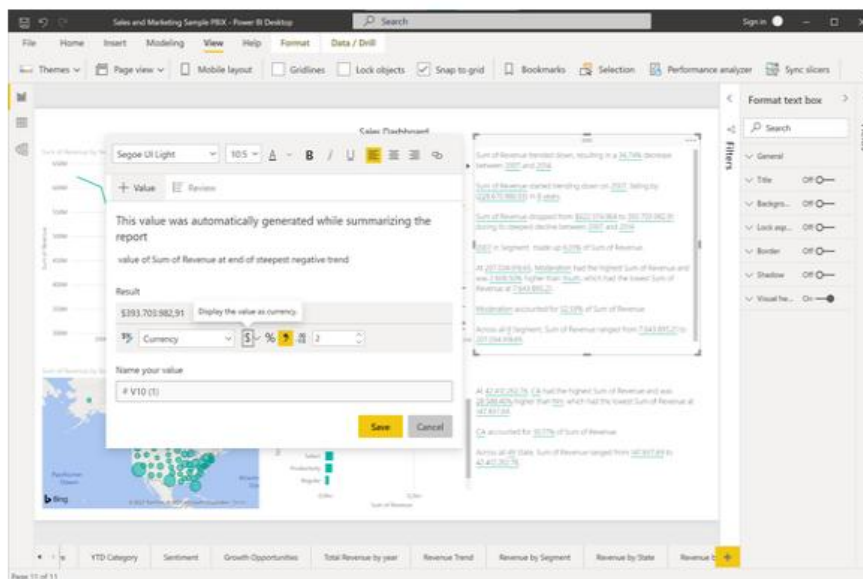


halaman. Dalam kasus kami, deskripsinya akan terlihat seperti output pada Gambar 5.15. Jika Anda mengklik kotak teks, Anda akan melihat nilai-nilai yang disorot yang dihitung Power BI di sini.



Gambar 5.15. Keluaran narasi cerdas

Ada beberapa hal menarik yang perlu diperhatikan di sini. Pada baris pertama, lihat bagaimana Power BI mengenali tren penurunan pendapatan dan menghitung persentase awal dan akhir deret waktu pendapatan (lihat Gambar 5.16). Hal ini juga menunjukkan bahwa segmen produk "Moderasi" menyumbang 52,59% dari total pendapatan pada tahun 2014. Jika Anda merasa suatu pernyataan kurang relevan atau bahkan berlebihan, Anda cukup menghapus baris ini dari ringkasan. Anda juga dapat menjelajahi nilai yang dihitung secara lebih detail dan menerapkan beberapa pemformatan. Misalnya, klik nilai yang menghitung angka pendapatan. Di menu konteks, kita dapat memilih agar nilai ini ditampilkan sebagai mata uang tanpa desimal setelah koma.



Gambar 5.16. Pemformatan Narasi Cerdas

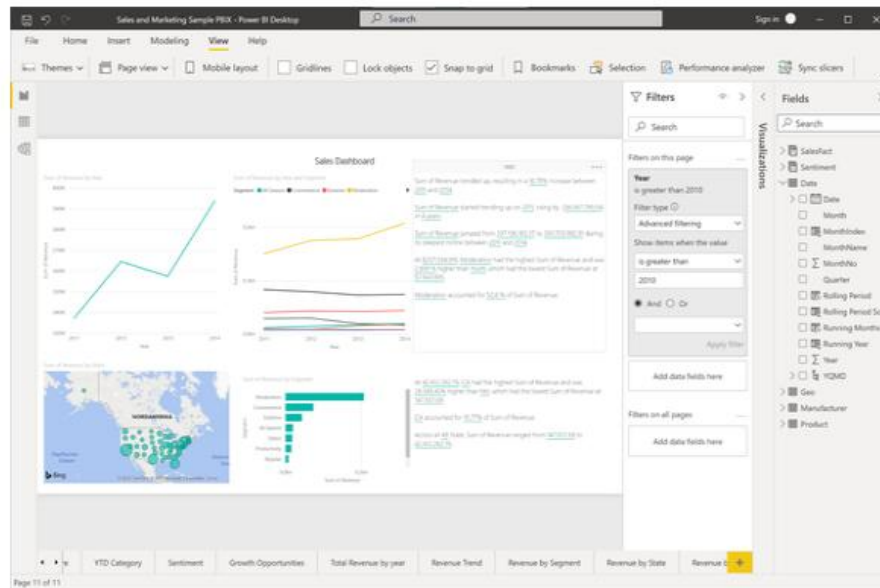
Memformat keluaran teks otomatis akan membuatnya lebih mudah diakses oleh pembaca dan secara keseluruhan terlihat jauh lebih baik.

Namun, ketika kita melihat ringkasannya, kita menyadari bahwa tidak ada informasi yang diberikan tentang rincian penjualan regional. Mungkin informasi tersebut tidak cukup penting bagi Power BI? Untuk menambahkannya secara manual, klik kanan pada plot peta yang menampilkan angka penjualan berdasarkan negara bagian dan klik "Ringkas" di menu konteks. Power BI akan menambahkan kotak teks lain yang menampilkan deskripsi hanya untuk plot yang Anda pilih. Sekali lagi, Anda dapat menghapus atau mengubah deskripsi sesuai keinginan.

Satu hal penting yang perlu diperhatikan tentang keterangan otomatis adalah sifatnya yang sepenuhnya dinamis dan merespons perubahan pada grafik yang mendasarinya. Untuk mendemonstrasikan hal ini, mari kita tambahkan filter halaman ke dasbor kita untuk hanya menampilkan data setelah tahun 2010 seperti yang ditunjukkan pada gambar 5.17. Untuk melakukannya, buka panel Bidang, lalu seret dan letakkan bidang "tahun" ke panel opsi "Filter di halaman ini". Setelah Anda menerapkan filter, lihat bagaimana konten kotak teks berubah.

Power BI mengenali tren pendapatan yang kini meningkat dan memperbarui persentase yang sesuai.





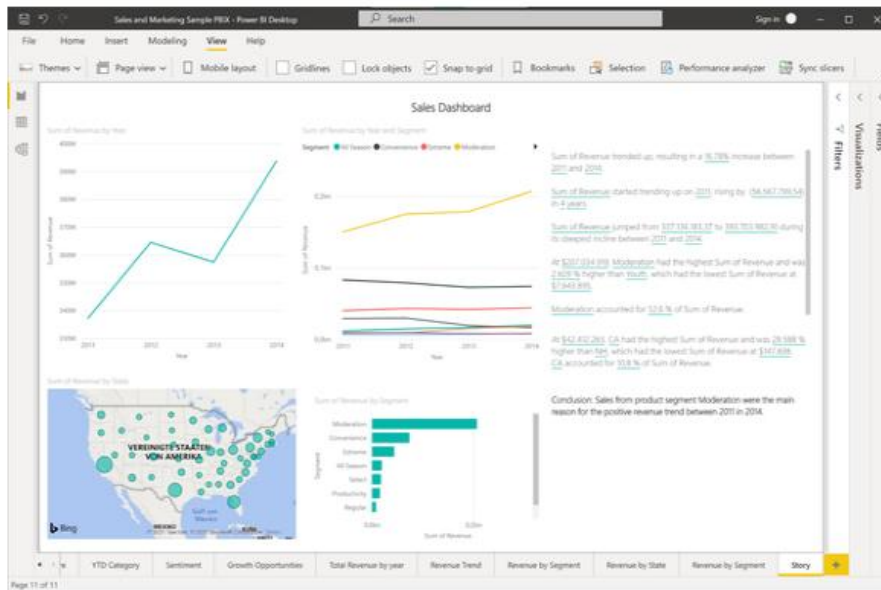
Gambar 5.17. Narasi pintar merespons pemfilteran global dan perubahan data

Yang masih kurang adalah penjelasan untuk salah satu bagan terpenting di dasbor ini: Rincian pendapatan berdasarkan segmen. Anda dapat melihat dengan jelas bahwa segmen "Moderasi" adalah satu-satunya segmen dengan tren positif yang relevan ini seharusnya mudah dikenali oleh alat Narasi pintar, bukan? Klik kanan pada plot dan klik "Ringkas" dari menu konteks. Tapi apa yang terjadi? Power BI hanya merespons dengan satu baris teks di kotak Narasi pintar yang sangat sulit diinterpretasikan.

Kita telah mencapai batas fitur Narasi pintar di sini. Pada saat penulisan ini, tidak mungkin untuk meringkas bagan yang sangat kompleks dengan banyak kategori atau tren yang mendasarinya. Keterbatasan lainnya adalah alat Narasi pintar tidak dapat menghasilkan ringkasan visual yang kolomnya dikelompokkan berdasarkan kolom lain dan untuk visual yang dibangun di atas bidang grup data. Selain itu, penggantian nama nilai dinamis atau pengeditan nilai dinamis yang dihasilkan secara otomatis saat ini tidak didukung dan ringkasan tidak dapat dibuat untuk visual yang berisi perhitungan instan seperti pengukuran kompleks atau persentase.

Karena fitur Narasi Cerdas awalnya baru dirilis pada tahun 2021, akan menarik untuk melihat batasan mana yang akan hilang. Anda dapat menemukan dokumentasi lengkap tentang alat Narasi Cerdas menggunakan dokumen sumber daya Microsoft ini.

Untuk saat ini, mari kita tambahkan kesimpulan akhir untuk dasbor kita secara manual. Tambahkan baris baru ke kotak teks terakhir dan ketik: "Kesimpulan: Penjualan dari segmen produk Moderasi merupakan alasan utama tren pendapatan positif antara tahun 2011 dan 2014." Dasbor penjualan akhir Anda sekarang akan terlihat seperti Gambar 5.18.



Gambar 5.18. Dasbor penjualan akhir dengan anotasi

Anda dapat dengan mudah mengekspor halaman ini ke dokumen PDF atau menerbitkannya ke server Power BI.

Meskipun terdapat lebih banyak kustomisasi dan konfigurasi untuk alat narasi Cerdas daripada yang dapat dicakup buku ini (misalnya, menentukan nilai kustom menggunakan kueri bahasa bergaya Tanya Jawab), kita akan membiarkannya seperti ini. Jika Anda ingin mengetahui lebih lanjut tentang alat narasi Cerdas, lihat tutorial video dari Microsoft ini.

5.3 RINGKASAN

Dalam bab ini, Anda telah mempelajari cara menggunakan kemampuan Bahasa Alami berbasis AI dalam bentuk alat Tanya Jawab di Power BI untuk membuat visualisasi lebih cepat dan memberikan pengalaman BI yang lebih baik kepada pengguna Anda. Anda juga telah mempelajari cara mengotomatiskan proses anotasi visualisasi Anda dan menambahkan deskripsi secara langsung menggunakan alat narasi Cerdas di Power BI. Di bab berikutnya, kita akan mengeksplorasi data kita lebih lanjut dan mencari tahu faktor-faktor apa saja yang memengaruhi tren pendapatan dalam data kita. Kita akan menggunakan alat berbasis AI untuk mengurangi waktu kita dalam mendapatkan wawasan dan menyisir data secara otomatis untuk menemukan pola-pola yang menarik.

BAB 6

ANALISIS DIAGNOSTIK BERTENAGA AI

6.1 KASUS PENGGUNAAN 3: WAWASAN OTOMATIS

Kita melanjutkan studi kasus dari bab sebelumnya: Di sebuah perusahaan manufaktur fiktif, kita membantu manajemen penjualan dalam proses pengambilan keputusan mereka. Di bab sebelumnya, kita menemukan bahwa penjualan perlahan pulih setelah periode penurunan yang tajam. Sekarang, kita akan menggali lebih dalam dan berkonsentrasi untuk memahami mengapa tren tertentu berevolusi.

Pernyataan Masalah

Sebagai analis bisnis dalam tim, kami ingin membantu manajemen penjualan menemukan penjelasan untuk dua tren pendapatan yang diamati: Mengapa angka penjualan menurun drastis antara tahun 2006 dan 2010, dan faktor apa yang menjelaskan pemulihan yang lambat antara tahun 2010 dan 2014?

Proses ini biasanya melibatkan analis yang memeriksa data secara manual, yang membutuhkan waktu dan bergantung pada ketersediaan analis. Karena kami tidak memiliki kapasitas untuk melakukan sesi tatap muka dengan setiap pengguna bisnis, kami mencari cara untuk memberikan wawasan kepada mereka secara lebih otomatis dan interaktif.

Ikhtisar Solusi

Tujuan kami adalah menemukan pola dalam kumpulan data kami secara otomatis dengan memindai semua informasi yang tersedia sesedikit mungkin dengan bantuan manusia. Untuk tujuan ini, kami memanfaatkan teknik berbasis AI untuk memberikan wawasan ini secara cepat dan interaktif. Hal ini akan memungkinkan pengguna bisnis mendapatkan wawasan dengan lebih sedikit atau bahkan tanpa bantuan analis data profesional. Untuk menghadirkan pengalaman ini, kami kembali menggunakan dua alat Power BI: Power BI Key Influencers dan Power BI Decomposition Tree.

Visual Key Influencers membantu memahami aspek-aspek yang memengaruhi metrik yang diminati dengan menampilkan kontributor teratas untuk metrik yang dipilih. Alat ini memeriksa informasi yang Anda berikan dan memunculkan aspek terpenting dengan melabelinya sebagai key influencer.

Visual decomposition tree Power BI memungkinkan Anda memvisualisasikan data dalam berbagai dimensi. Alat ini mengagregasi data secara otomatis dan memungkinkan Anda menelusuri dimensi apa pun dalam urutan apa pun. AI akan menyarankan dimensi paling relevan berikutnya untuk ditelusuri, tergantung pada area minat Anda. Hasilnya, ini merupakan alat yang berguna untuk investigasi ad hoc dan analisis akar penyebab.

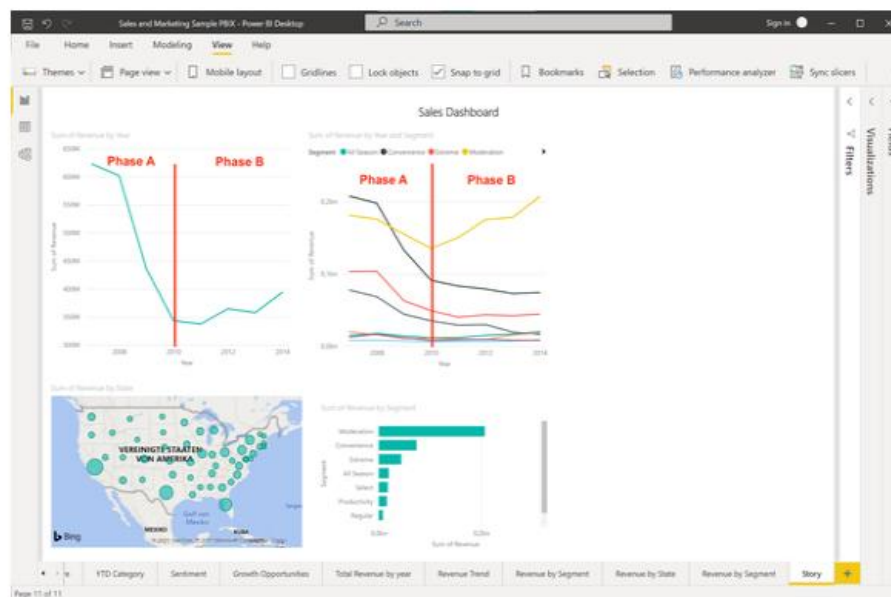
Kami akan menerapkan teknik-teknik ini pada contoh laporan Power BI Penjualan dan Pemasaran untuk mengetahui: Faktor apa saja yang mendorong tren pendapatan dan pola apa yang dapat kita amati dari data kita? Dalam skenario perusahaan, kedua alat ini dapat



diberikan kepada pengguna bisnis dalam bentuk halaman laporan Power BI sehingga mereka dapat berinteraksi dengan data mereka sendiri.

Panduan Power BI

Buka file contoh PBIX Penjualan & Pemasaran yang telah kita gunakan di bab sebelumnya. Singkatnya, kumpulan data ini berisi berbagai data penjualan dan pemasaran dari sebuah perusahaan manufaktur dan telah diisi sebelumnya dengan beberapa laporan untuk pangsa pasar, volume produk, angka penjualan, dan skor sentimen. Di Bab 2, kita menemukan bahwa pendapatan keseluruhan dalam empat tahun terakhir perlahan pulih setelah mengalami peningkatan tajam di tahun sebelumnya. Berikut cuplikan dasbor penjualan yang telah kita buat sebelumnya:



Gambar 6.1. Dua fase tren pendapatan

Kami secara khusus tertarik pada apa yang terjadi selama fase penurunan pendapatan antara tahun 2006 dan 2010 ("Fase A") dan fase pemulihan pendapatan secara perlahan antara tahun 2010 dan 2014 ("Fase B"). Untuk kedua fase ini, kami ingin menjawab pertanyaan sederhana namun mendalam: Apa yang terjadi?

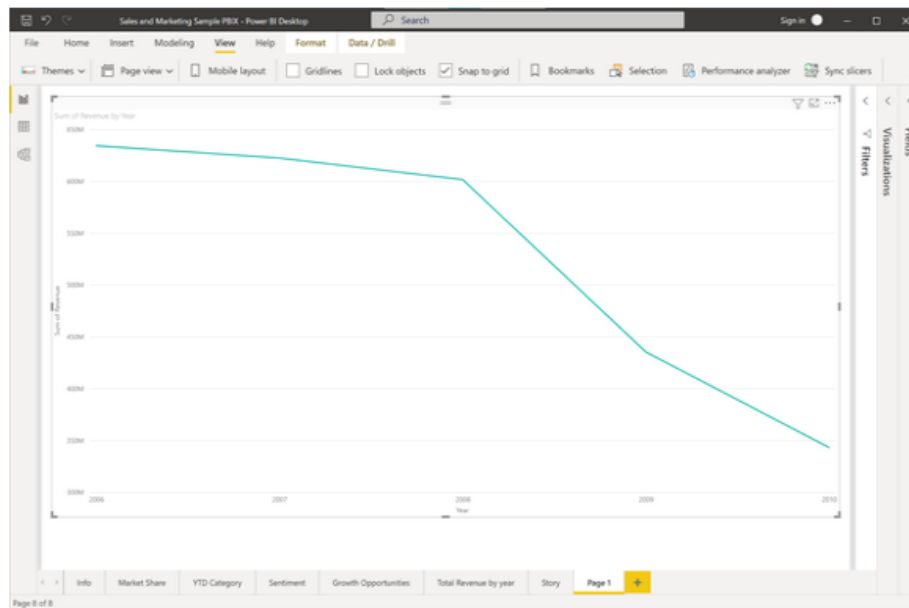
Kita akan mulai dengan memeriksa Fase A. Untuk membuat ulang visual pendapatan, buat halaman laporan Power BI baru, seret dan lepas visual Tanya Jawab, lalu ketik:

"Pendapatan per tahun antara 2006 dan 2010"

CATATAN

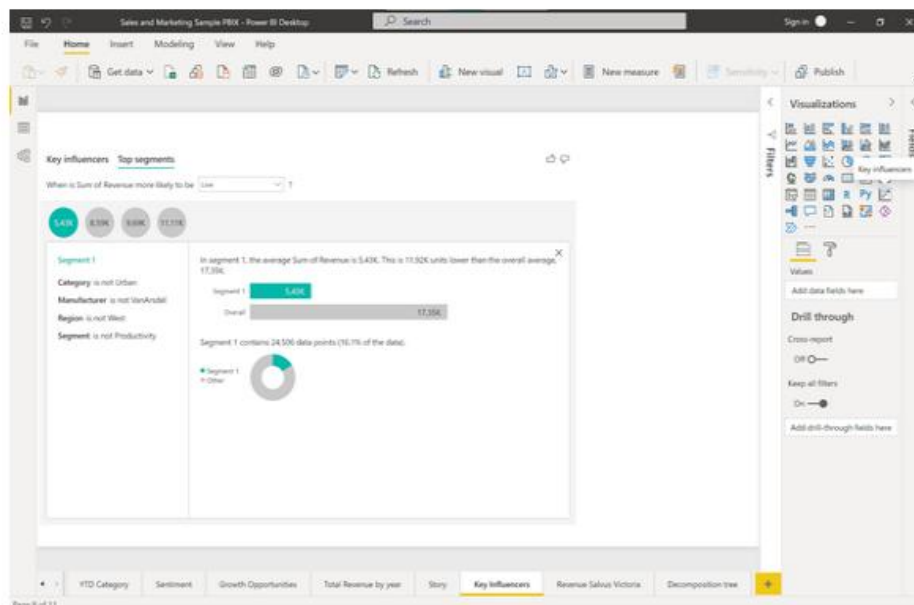
Jika Anda tidak melakukan kasus penggunaan dari bab sebelumnya, Anda mungkin perlu mengganti Pendapatan dengan "jumlah pendapatan".

Klik ikon konversi untuk mengubah visual Tanya Jawab menjadi visual Power BI standar. Bagan akan terlihat seperti ini:



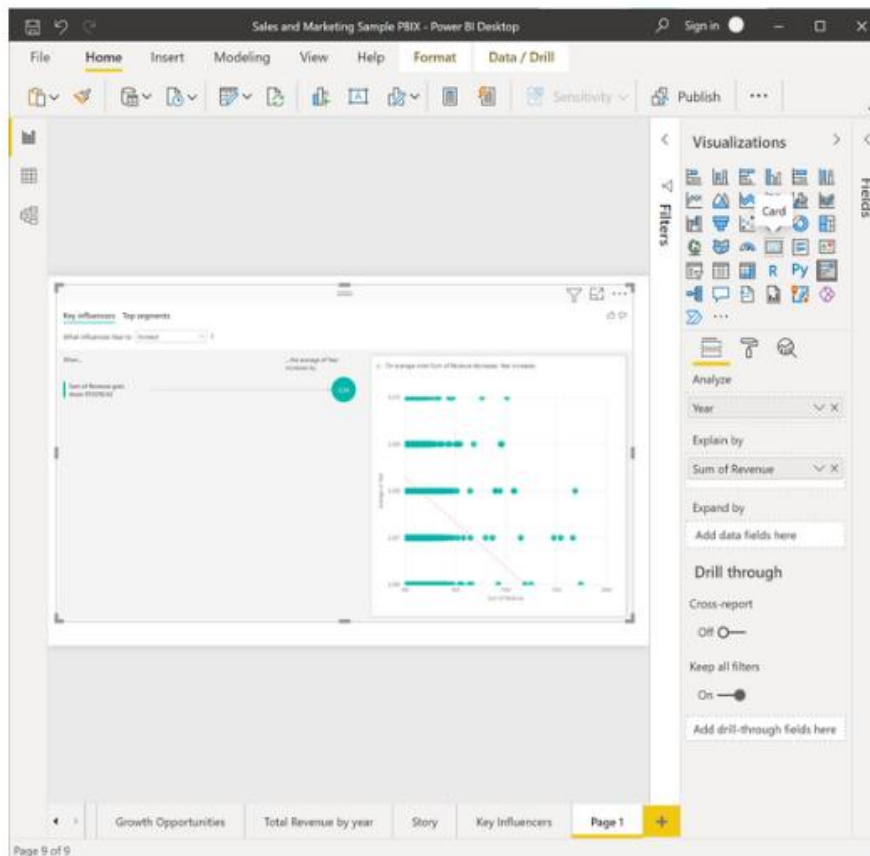
Gambar 6.2. Pendapatan per tahun antara 2006 dan 2010

Pilih diagram ini dan buka panel visualisasi di sebelah kanan. Klik ikon "Influencer Utama" untuk mengubah diagram garis menjadi visual "Influencer Utama".



Gambar 6.3. Ikon visual influencer utama

Setelah Anda memilih jenis visual baru, bagan akan otomatis diperbarui menjadi tampilan yang mirip dengan Gambar 6.4. Mari kita telusuri tampilan visual ini lebih detail.

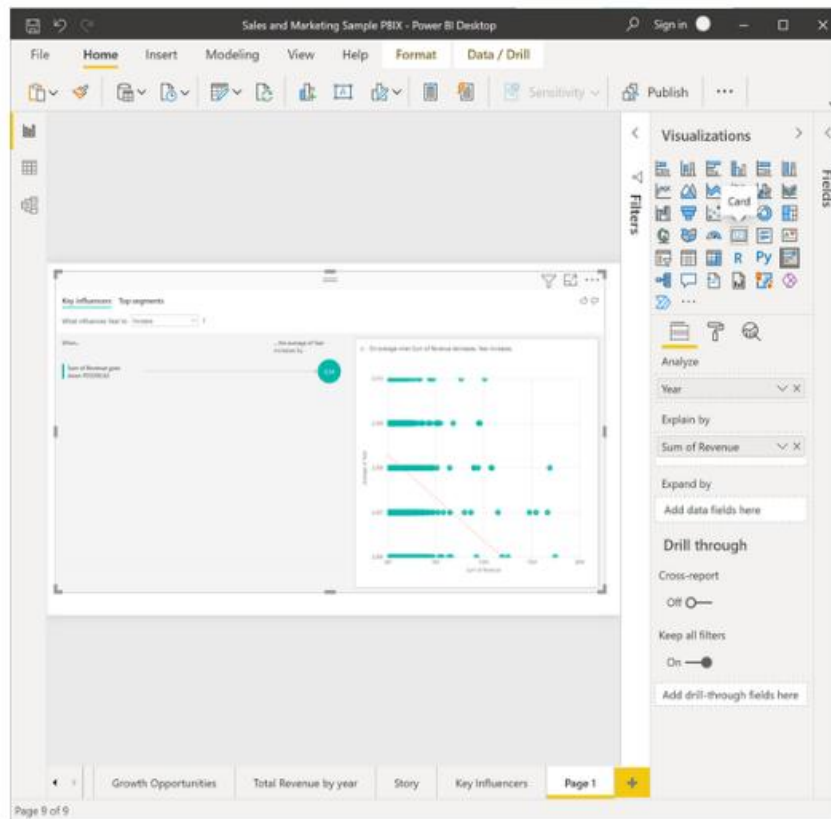


Gambar 6.4. Output standar Key Influencer

Nah, output ini mungkin sama sekali tidak sesuai harapan Anda dan terlihat agak aneh. Apa yang salah? Buka panel visualisasi dan periksa properti visual Key Influencer. Seharusnya tampilannya masih seperti yang ditunjukkan pada Gambar 6.5.

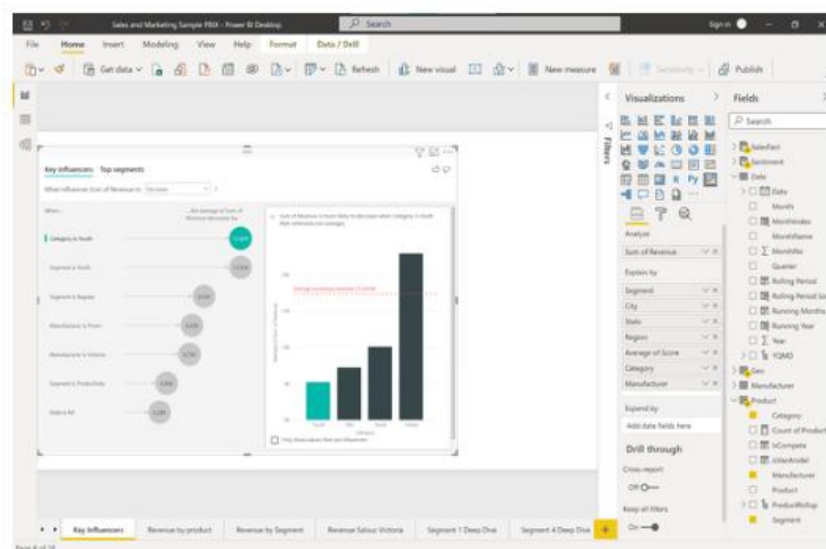
Di properti visual, Anda akan menemukan dua kolom penting: "Analisis" dan "Jelaskan berdasarkan". Sesuai namanya, "Analisis" mengacu pada metrik yang ingin Anda jelajahi dan "Jelaskan berdasarkan" mengacu pada berbagai dimensi yang mungkin Anda anggap memengaruhinya. Secara default, berdasarkan diagram garis kita sebelumnya, Power BI menyarankan untuk menjelaskan dimensi Tahun dengan kolom Jumlah Pendapatan, yang sama sekali tidak masuk akal.

Mari kita perbaiki hal ini dengan menyeret "Jumlah Pendapatan" dari kolom "Jelaskan berdasarkan" ke kolom "Analisis", alih-alih "Tahun". Oleh karena itu, tambahkan dimensi tambahan ke kolom "Jelaskan oleh" dengan mengambilnya dari repositori kolom data di sebelah kanan. Kolom kandidat yang masuk akal dari sudut pandang bisnis mungkin adalah: Segmen, Kota, Negara Bagian, Wilayah, Rata-rata Skor, Kategori, dan Produsen. Kita akan mengabaikan dimensi Produk karena terlalu banyak titik data yang hilang. Misalnya, beberapa produk digantikan oleh produk lain, beberapa produk baru saja diluncurkan, beberapa produk sudah tidak digunakan lagi.



Gambar 6.5. Properti utama yang memengaruhi setelah konversi dari diagram garis

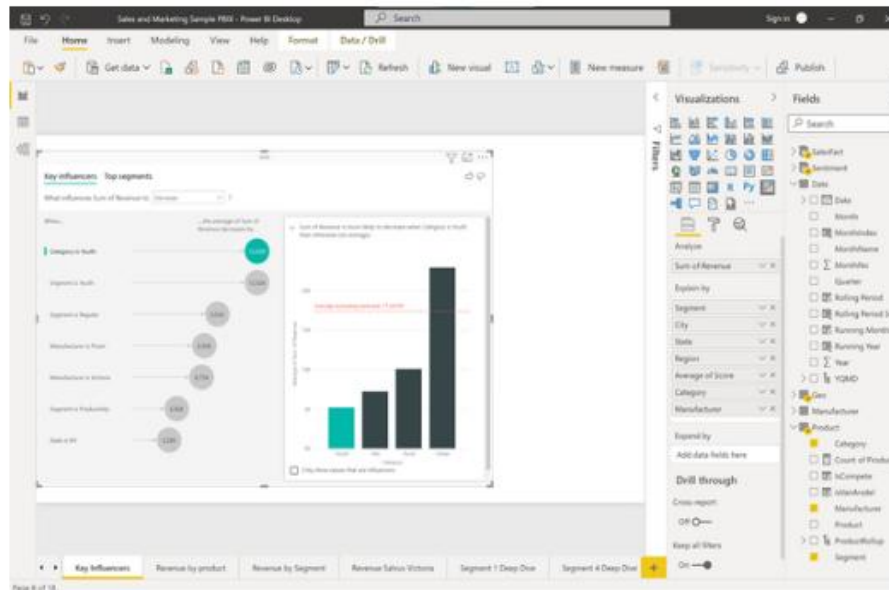
Jika kita ingin menganalisis Key Influencers hingga ke tingkat produk, sebaiknya kita melihat periode satu tahun, bukan periode empat tahun. Properti terbaru dari alat Key Influencers sekarang akan terlihat seperti pada Gambar 6.6:



Gambar 6.6. Properti Key Influencers (diedit)

Di dalam visual Key Influencers, alihkan menu tarik-turun di bagian atas dari Naik ke Turun, karena kita ingin mengetahui penyebab penurunan pendapatan. Visual yang diperbarui seharusnya sekarang terlihat mirip dengan yang ada di Gambar 6.7.

Visual ini secara intuitif lebih masuk akal daripada versi sebelumnya. Perhatikan bahwa ini akan menjadi halaman laporan yang dapat Anda bagikan atau publikasikan untuk pengguna bisnis agar mereka dapat berinteraksi dengan data mereka sendiri. Mari kita bahas visual ini untuk melihat cara kerjanya lebih detail.



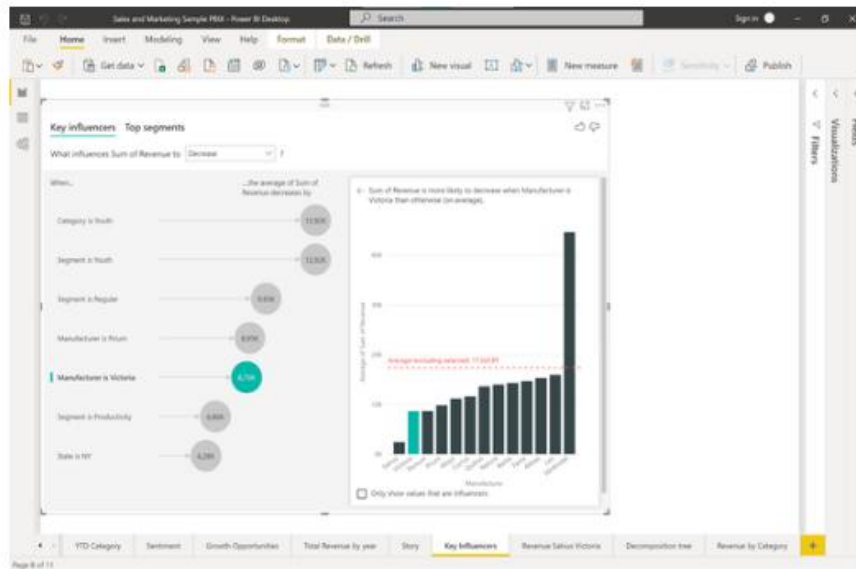
Gambar 6.7. Alat Influencer Utama yang diedit

Di sisi kiri visual, kita dapat melihat variabel mana yang diidentifikasi Power BI sebagai influencer utama untuk metrik yang diberikan, dalam hal ini pendapatan. Dapat dilihat bahwa Power BI mengidentifikasi atribut "Remaja" di seluruh dimensi Kategori dan Segmen sebagai pendorong utama yang menyebabkan metrik Pendapatan menurun.

Di sisi kanan, kita disajikan dengan distribusi data dasar untuk influencer yang dipilih, yang menunjukkan rata-rata metrik pendapatan di semua kategori dari dimensi yang dipilih, dalam contoh di Gambar 6.7 dimensi Kategori. Anda dapat membaca bagan di sebelah kanan sebagai berikut: Pendapatan rata-rata untuk Kategori Remaja hanya Rp 51.819.800 sementara pendapatan rata-rata untuk semua Kategori lain kecuali Remaja adalah Rp 173.498.900. Artinya, ketika Kategorinya adalah Remaja, pendapatan rata-rata Rp 120.920.000.000 lebih rendah dibandingkan dengan semua nilai Kategori lainnya.

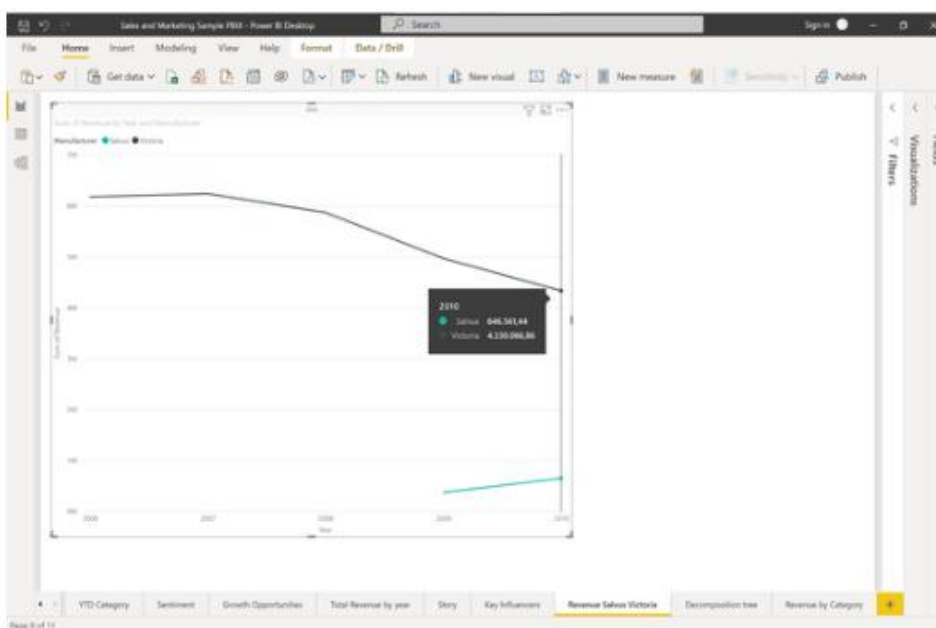
Penting untuk dicatat bahwa dalam kasus ini kita tidak menilai berdasarkan nilai absolut suatu variabel, tetapi berdasarkan rata-rata dan jumlah observasinya. Bergantung pada ukuran yang Anda pilih untuk dianalisis, baik kategorikal maupun berkelanjutan, alat Key Influencer akan beradaptasi. Alih-alih rata-rata, alat ini akan menunjukkan probabilitas seperti kemungkinan hasil x-kali lebih besar jika nilai tertentu terpenuhi.

Alat Key Influencer tidak selalu mempertimbangkan nilai terendah dalam suatu kategori sebagai influencer. Misalnya, influencer "Produsen adalah Victoria" dalam analisis kami. Jika Anda mengklik influencer ini, Anda akan melihat grafik pada Gambar 6.8. Kita dapat melihat bahwa Victoria bukanlah produsen dengan pendapatan rata-rata terkecil, melainkan Salvus. Lalu, mengapa Salvus tidak diidentifikasi sebagai Key Influencer?



Gambar 6.8. Visual Key Influencer untuk Fase A (Tren pendapatan negatif)

Alasannya menjadi jelas jika kita melihat perbandingan langsung antara Salvus dan Victoria pada Gambar 6.9, hanya untuk memperkuat pemahaman kita tentang cara kerja alat Key Influencer:

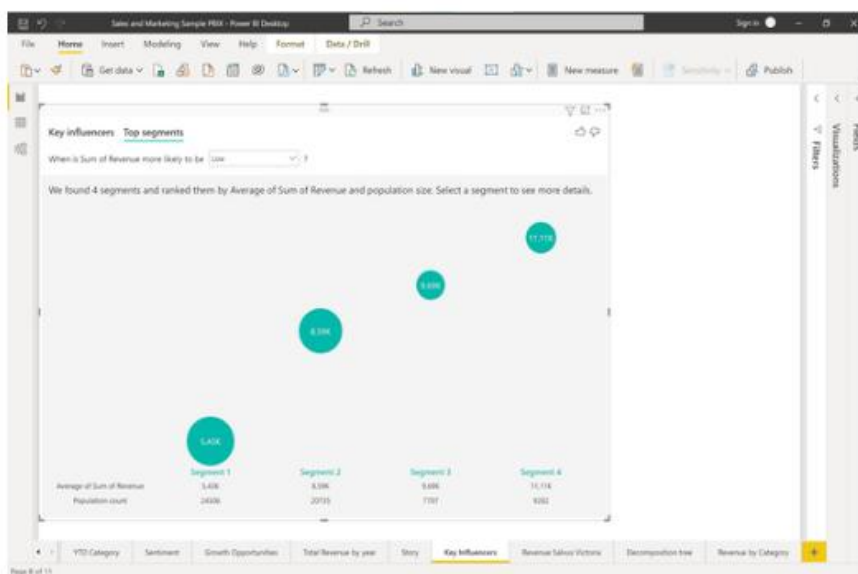


Gambar 6.9. Membandingkan penjualan produsen Salvus vs. Victoria

Ada tiga faktor yang mendiskualifikasi Salvus sebagai key influencer: Pertama, produsen ini baru diperkenalkan pada tahun 2009, sehingga total poin datanya lebih sedikit dibandingkan dengan produsen lain. Kedua, kontribusi pendapatan absolut Salvus sangat kecil sehingga kemungkinan besar tidak memberikan dampak keseluruhan.

Dan yang terakhir, tren pendapatan aktual produsen ini tidak negatif, melainkan positif. Meskipun pendapatan Victoria turun dari 6,2 juta pada tahun 2006 menjadi 4,3 juta pada tahun 2010, Salvus justru meningkatkan pendapatan mereka antara tahun 2009 dan 2010. Dengan informasi latar belakang ini, wajar jika Salvus tidak dianggap sebagai key influencer untuk penurunan pendapatan secara keseluruhan. Alat key influencer memudahkan kita menemukan pola-pola menarik tersebut tanpa harus menelusuri semua data secara manual.

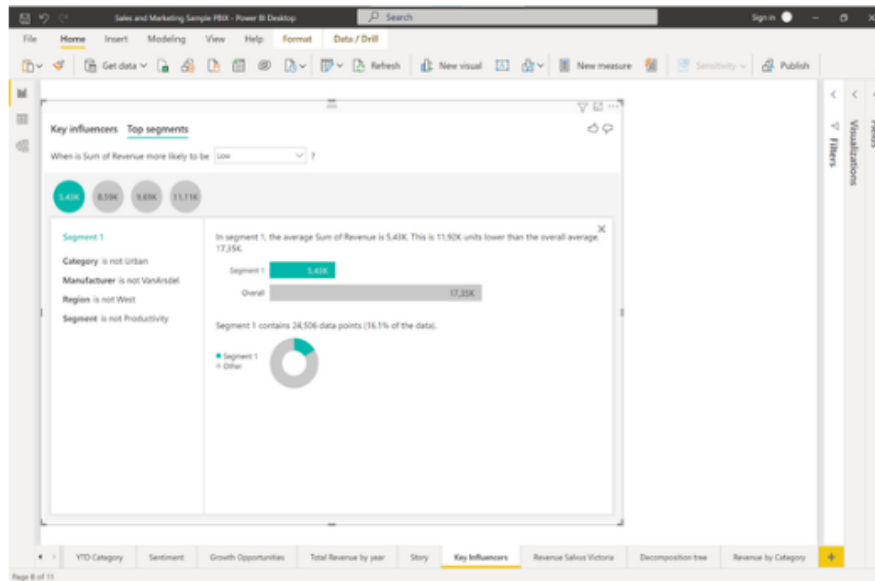
Mari kita jelajahi fitur lain yang bermanfaat dari alat influencer: Segmen. Segmen dapat dianggap sebagai grup dalam data Anda yang menunjukkan dampak tinggi pada metrik yang diminati. Beralihlah ke Segmen Teratas dan pilih "Kapan Jumlah Pendapatan lebih mungkin Rendah" dari menu tarik-turun di bagian atas. Anda akan melihat layar berikut pada Gambar 6.10 sebagai hasilnya:



Gambar 6.10. Menganalisis segmen yang berkontribusi terhadap tren pendapatan negatif

Kita dapat melihat bahwa ada empat segmen yang diidentifikasi oleh Power BI. Segmen terbesar, ditunjukkan oleh ukuran gelembung, berisi 24.506 observasi (yaitu produk yang terjual) dan menghasilkan pendapatan rata-rata hanya 5,4 ribu per penjualan. Segmen berkinerja rendah dengan pendapatan rata-rata tertinggi (namun di bawah total) sebesar 11,11 ribu berisi 9.282 observasi. Untuk mengetahui lebih lanjut tentang apa yang terjadi di sini, klik gelembung 5,43 ribu.

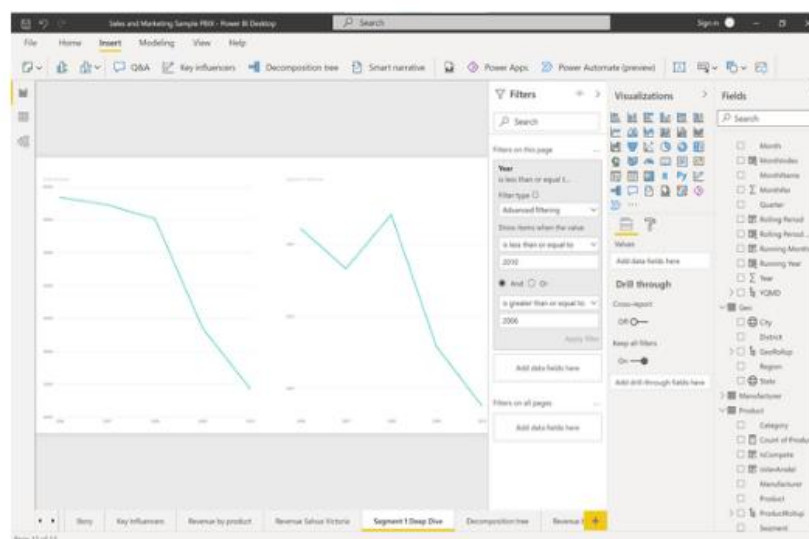
Anda akan melihat tampilan detail segmen ini yang dalam hal ini terlihat seperti ini pada Gambar 6.11:



Gambar 6.11. Detail Segmen 1

Dari diagram ini, kita dapat melihat bahwa segmen ini mencakup hampir 16% dari seluruh penjualan produk dan oleh karena itu memiliki relevansi yang sangat tinggi bagi kita. Segmen ini mencakup semua penjualan di mana kategorinya bukan Urban, produsennya bukan VanArsdel, wilayahnya bukan Barat, dan Segmennya bukan Produktivitas. Segmen ini menyumbang kelompok titik data terbesar dan memiliki pendapatan rata-rata terendah, yang menjadikannya segmen penting untuk dicermati dalam menjelaskan tren penurunan pendapatan di Fase A (2006-2010).

Untuk memahami maknanya secara detail, mari kita periksa tren pendapatan di segmen 1 ini dibandingkan dengan tren pendapatan total antara tahun 2006 dan 2010 seperti yang ditunjukkan pada Gambar 6.12.



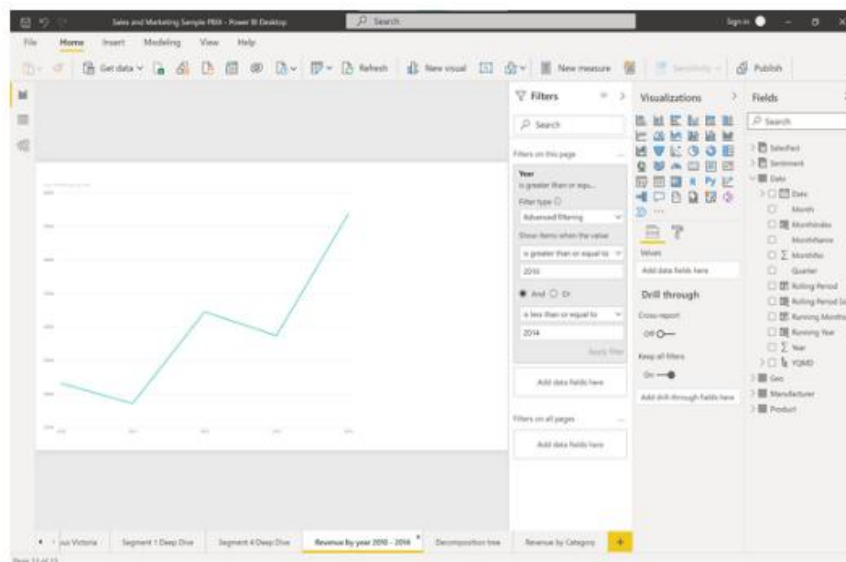
Gambar 6.12. Tren pendapatan keseluruhan antara total pendapatan dan pendapatan segmen 1

Meskipun pendapatan keseluruhan turun 46,6% dari 643,28 juta menjadi 343,13 juta dalam kumpulan data keseluruhan, segmen 1 sendiri menyumbang kerugian sebesar 12,3 juta dari 31,10 juta pada tahun 2006 menjadi 18,8 juta pada tahun 2010. Segmen yang lebih terfokus ini seharusnya lebih mudah ditangani oleh pengguna bisnis dan mengidentifikasi area masalah potensial, dibandingkan dengan tren yang lebih besar seperti Kategori "Pemuda".

Kita dapat mengeksplorasi segmen lain dengan cara yang sama, tetapi mari kita selesaikan analisis kita untuk penurunan pendapatan sebelumnya untuk saat ini dan beralih ke mencari tahu apa yang terjadi selama periode pemulihan antara tahun 2010 dan 2014, yang disebut Fase B. Jika Anda tertarik dengan fitur dan deskripsi lebih lanjut tentang cara kerja alat Key Influencers, Anda dapat melihat sumber daya komprehensif ini dari Microsoft.

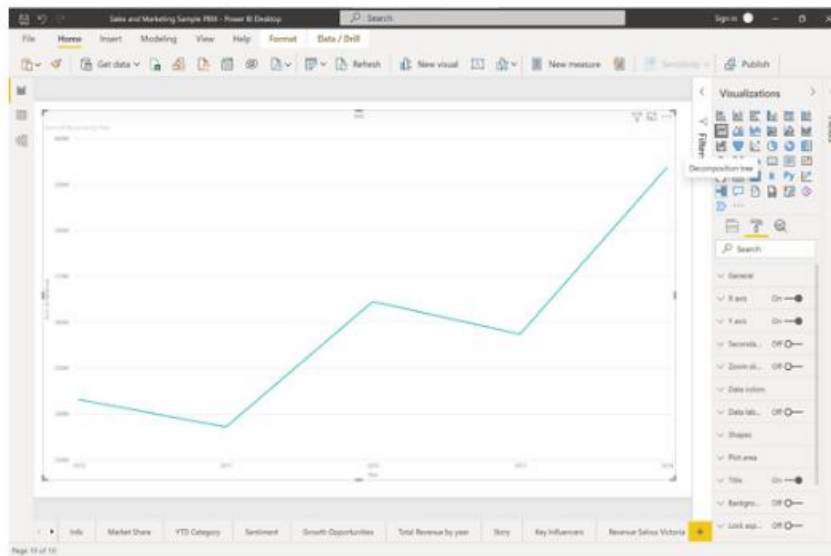
Untuk analisis Fase B, kita akan beralih ke fitur pohon dekomposisi di Power BI. Pohon dekomposisi adalah alat yang hebat untuk melakukan analisis akar penyebab. Anda dapat menggunakannya sebagai drilldown cerdas di mana Power BI menyarankan tingkat drilldown berikutnya secara otomatis berdasarkan metrik tertentu yang ingin Anda jelajahi.

Untuk tujuan ini, mulailah dengan halaman laporan kosong di file Power BI Anda. Buat ulang grafik pendapatan per tahun, kali ini untuk periode antara 2010 dan 2014. Ubah grafik ini menjadi visual statis jika Anda menggunakan alat Tanya Jawab untuk ini. Grafik pendapatan sekarang akan terlihat seperti Gambar 6.13:



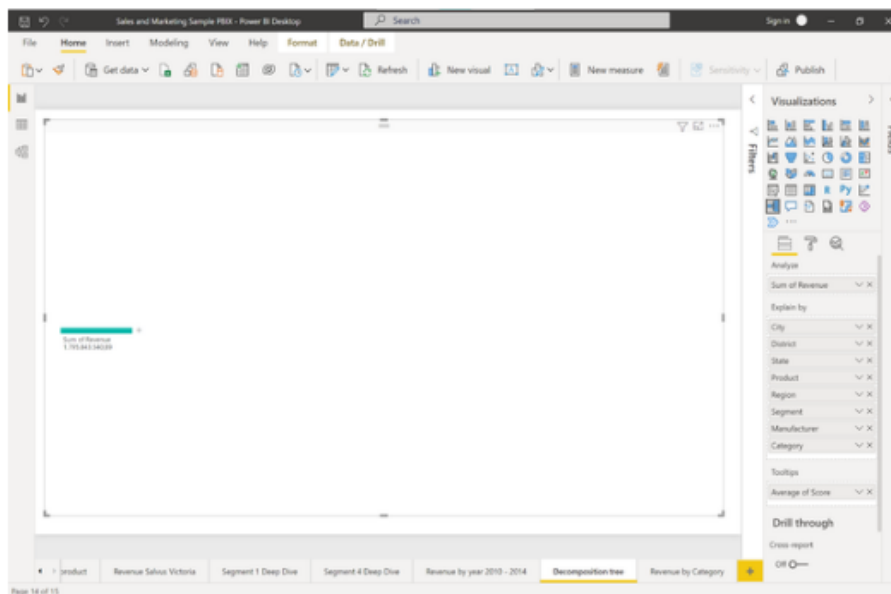
Gambar 6.13. Total pendapatan antara tahun 2010 dan 2014

Pilih diagram garis dan ubah jenis visualisasi menjadi "Pohon Dekomposisi" seperti yang ditunjukkan pada Gambar 6.14:



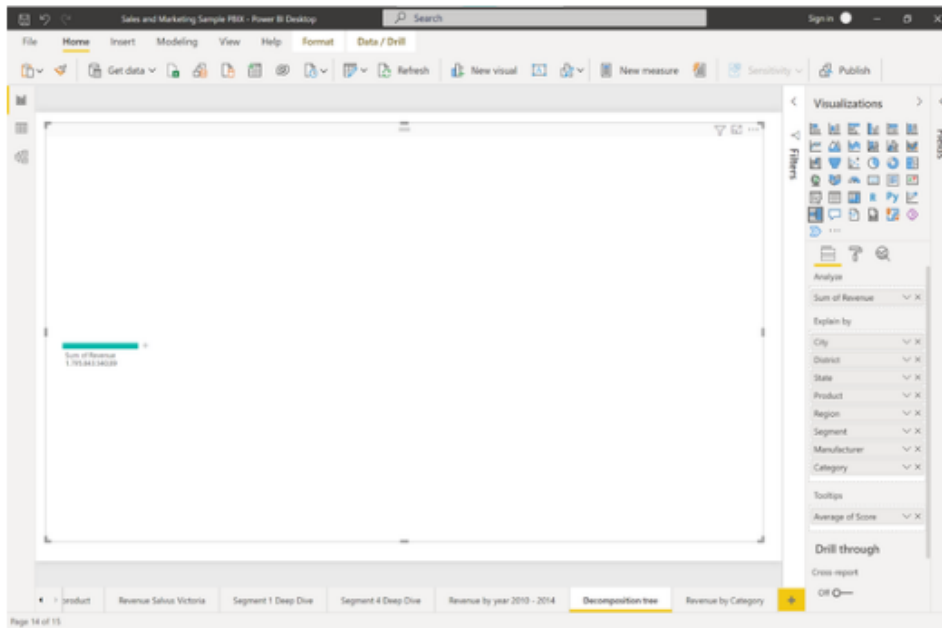
Gambar 6.14. Visual pohon dekomposisi

Serupa dengan alat influencer, kita harus memberi tahu Power BI variabel mana yang ingin kita analisis dan dimensi mana yang kita anggap sebagai faktor penjelas. Ubah properti visual agar terlihat seperti pada Gambar 6.15.



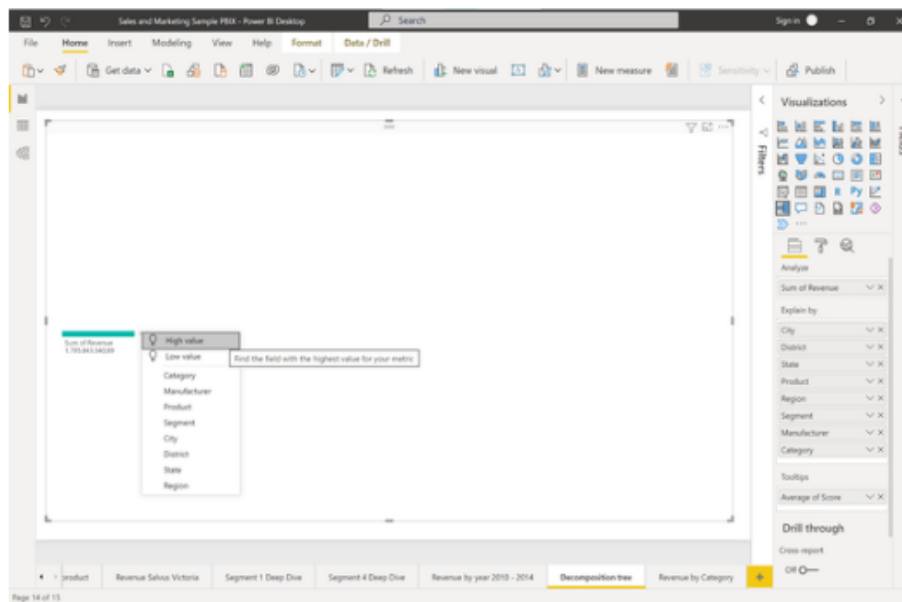
Gambar 6.15. Properti Pohon Dekomposisi

Keluaran awal visual pohon dekomposisi hanyalah ringkasan metrik yang kita minati di kanvas kosong. Anda dapat melihat keluaran ini pada Gambar 6.16.

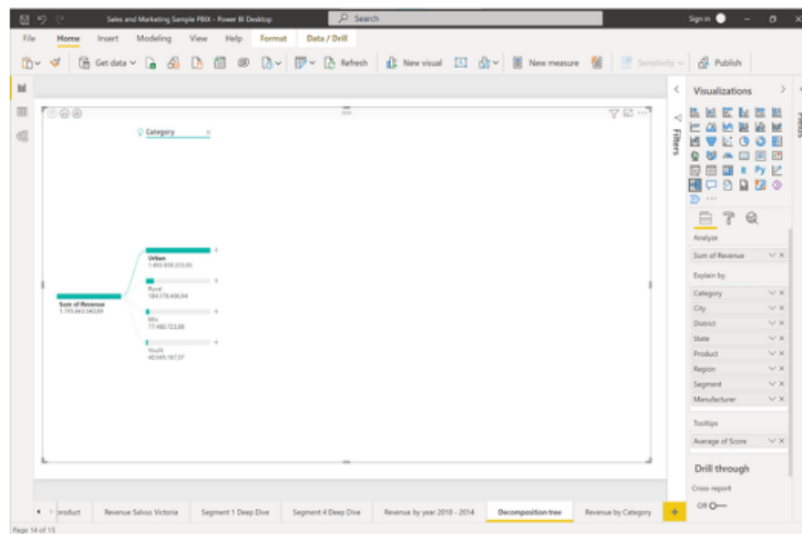


Gambar 6.16. Awal Pohon Dekomposisi

Tugas yang ada sekarang adalah menguraikan metrik agregat ini menjadi potongan-potongan yang lebih kecil, idealnya membaginya sedemikian rupa sehingga kita menangkap metrik terpenting di bagian atas pohon.



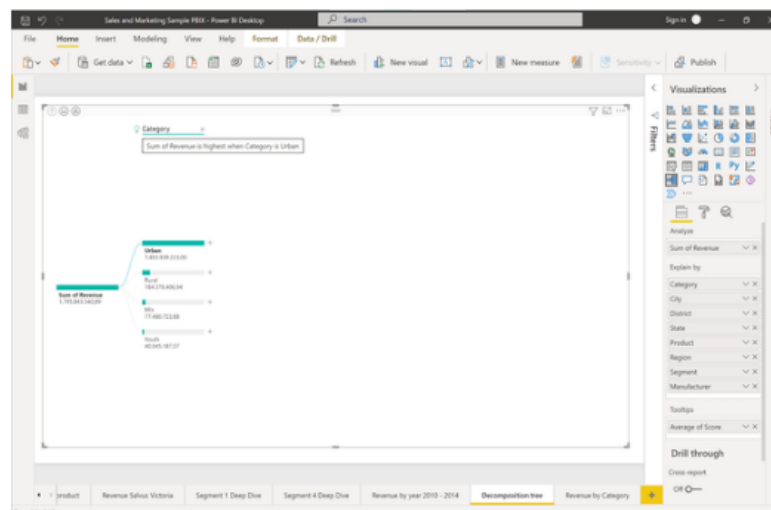
Gambar 6.17. Menggunakan pemisahan AI dalam pohon dekomposisi



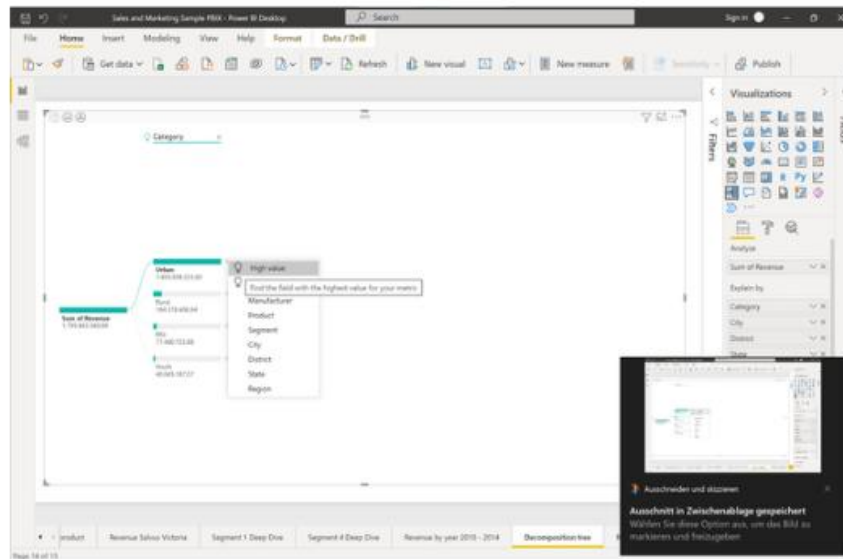
Gambar 6.18. Pembagian Tingkat Pertama pada pohon dekomposisi

Ketika Anda mengklik ikon "plus" kecil di sebelah kanan pendapatan agregat, Anda akan melihat menu pop-up yang memungkinkan Anda memilih pemisahan berikutnya yang ingin Anda buat. Anda dapat memilih pemisahan secara manual, misalnya berdasarkan logika bisnis atau preferensi Anda sendiri, atau Anda dapat menggunakan fitur yang disebut "pemisahan AI". Kita akan kembali membahas konsep ini nanti untuk mengetahui dalam kasus mana mungkin masuk akal untuk memilih tingkat drill down secara manual. Pemisahan AI ditunjukkan oleh bola lampu kecil dan secara otomatis akan memilih tingkat drill down terbaik berikutnya untuk Anda, tergantung apakah Anda ingin memengaruhi metrik tingkat atas Anda menjadi lebih tinggi atau lebih kecil, seperti yang dapat Anda lihat pada Gambar 6.17.

Karena kita ingin menjelaskan pertumbuhan pendapatan di Fase B, mari kita pilih pemisahan AI untuk nilai tinggi. Anda akan melihat bahwa Power BI membuat pemisahan pertama untuk Anda berdasarkan kriteria "Kategori" seperti yang ditunjukkan pada Gambar 6.18.



Gambar 6.19. Penjelasan AI Split

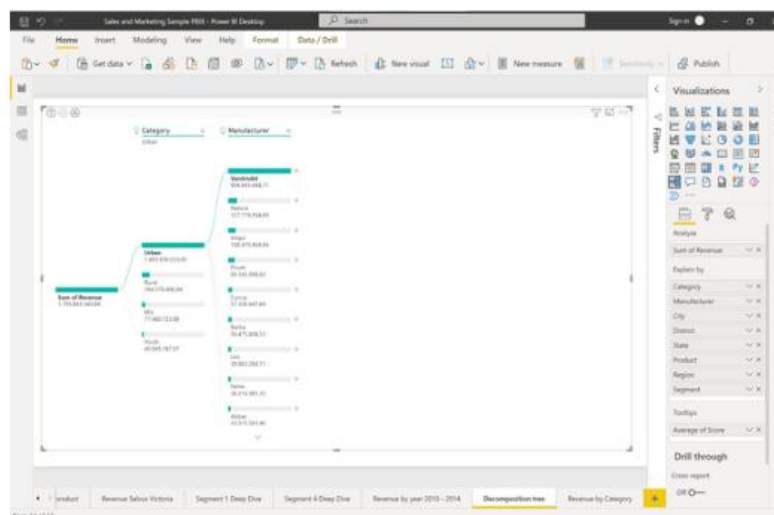


Gambar 6.20. Pemisahan AI Tingkat Kedua

Jika kita mengarahkan kursor ke bola lampu di sebelah pembagian Kategori, Anda akan melihat keterangan alat Power BI yang menjelaskan simpul pohon saat ini. Dalam hal ini, pembagian pertama ini memberi tahu kita bahwa pendapatan antara tahun 2010 dan 2014 paling tinggi ketika Kategorinya adalah Perkotaan (lihat Gambar 6.19).

Kita sekarang dapat menelusuri lebih lanjut sesuai keinginan, misalnya dengan mencari tahu kriteria mana yang menghasilkan pertumbuhan pendapatan dalam kategori Perkotaan. Sekali lagi, kita dapat memilih tingkat penelusuran berikutnya secara manual atau membiarkan Power BI yang menentukannya, tergantung apakah kita tertarik pada Nilai Tinggi atau Rendah (Lihat Gambar 6.20).

Jika kita memilih "Nilai Tinggi" lagi, kita dapat melihat bahwa pemisahan berikutnya akan dilakukan pada kolom "Produsen" seperti yang ditunjukkan pada Gambar 6.21.

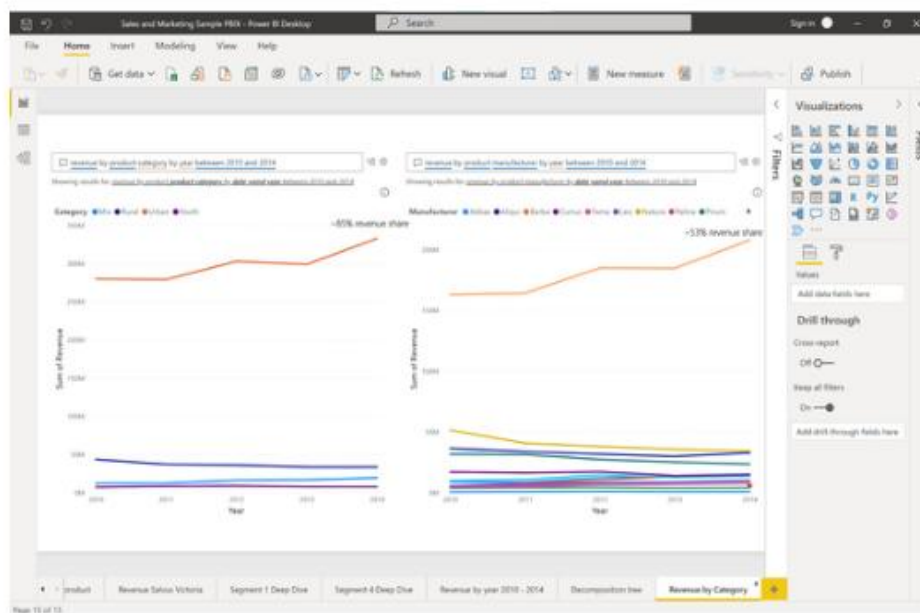


Gambar 6.21. Pembagian Tingkat Kedua dalam pohon dekomposisi



Pada titik ini, mari kita berhenti sejenak dan merangkum secara singkat mengapa kita melihat pembagian ini dalam urutan ini. Mengapa data tidak dibagi terlebih dahulu untuk Produsen dan baru kemudian berdasarkan Kategori? Setidaknya kita telah melihat bahwa produsen "VanArsdel" mengklaim sebagian besar pendapatan.

Hal ini terjadi karena algoritma dasar di Power BI yang melakukan pembagian tersebut bersifat greedy. Algoritma tersebut akan memilih kategori drill-down berikutnya di mana pembagian tersebut akan memberikan keuntungan terbesar. Untuk mendemonstrasikan fenomena ini, mari kita lihat perbandingan berdampingan antara pendapatan berdasarkan kategori dan pendapatan berdasarkan produsen seperti yang ditunjukkan pada Gambar 6.22.



Gambar 6.22. Pendapatan vs. Produsen vs. Kategori

Kita dapat melihat dengan jelas bahwa Urban dan VanArsdel menonjol di kelasnya sebagai penggerak pendapatan utama. Namun, jika ditelusuri lebih lanjut, kita akan melihat bahwa Urban mengklaim sekitar 85% pangsa pendapatan, sementara VanArsdel hanya mendapatkan 53%. Hal ini disebabkan oleh jumlah produsen yang jauh lebih banyak daripada kategori, sehingga mengurangi pangsa pendapatan dari penggerak utama di bidang ini. Itulah sebabnya Power BI membagi data terlebih dahulu berdasarkan Kategori, baru kemudian berdasarkan Produsen.

Meskipun pendekatan greedy ini umumnya cukup efektif, Anda harus berhati-hati dalam beberapa kasus. Ketika kita memeriksa data setelah pemisahan tertentu, kita hanya dapat menganalisis data yang benar-benar ada dalam pemisahan ini. Misalnya, dengan memisahkan berdasarkan Kategori terlebih dahulu, lalu menelusuri kategori Urban, kita hanya akan melihat produsen di cabang pohon data tersebut yang benar-benar memproduksi produk dalam kategori Urban. Jika ada produsen dalam kumpulan data yang tidak memproduksi

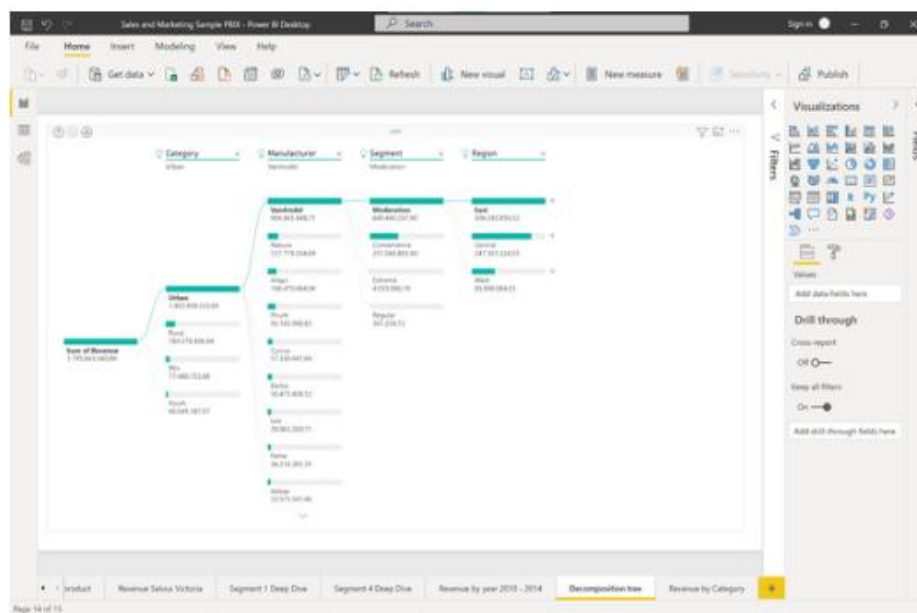


barang apa pun untuk kategori Urban, produsen ini tidak akan muncul dalam pemisahan hilir cabang pohon data tersebut.

Sebaliknya, ini berarti Power BI kemungkinan besar tidak akan pernah (atau hanya di bagian paling akhir pohon) menyarankan pemisahan berdasarkan bidang di mana semua observasi terdistribusi secara merata. Kita dapat mengamati hal ini misalnya dalam dimensi "Negara Bagian" dari kumpulan data kita, di mana kontributor pendapatan tunggal terbesar (California) hanya memperoleh sekitar 10% dari total pendapatan dan kontributor terkecil sekitar 4,8%.

Negara bagian terdistribusi secara merata dalam hal pendapatan, sehingga bukan kandidat yang baik bagi Power BI untuk melakukan pemisahan berdasarkan negara bagian tersebut. Jika negara bagian memang berperan untuk bisnis Anda (atau pengguna bisnis yang berbeda), Anda dapat memilih pemisahan tersebut secara manual di tingkat pertama atau menambahkan filter halaman sesuai dengan negara bagian yang Anda minati.

Kembali ke contoh Pohon Dekomposisi kita, kita ingin menambahkan dua pemisahan AI lagi untuk nilai tinggi. Jika Anda menambahkan pemisahan ini, Anda akan melihat pohon yang terlihat seperti pada Gambar 6.23:



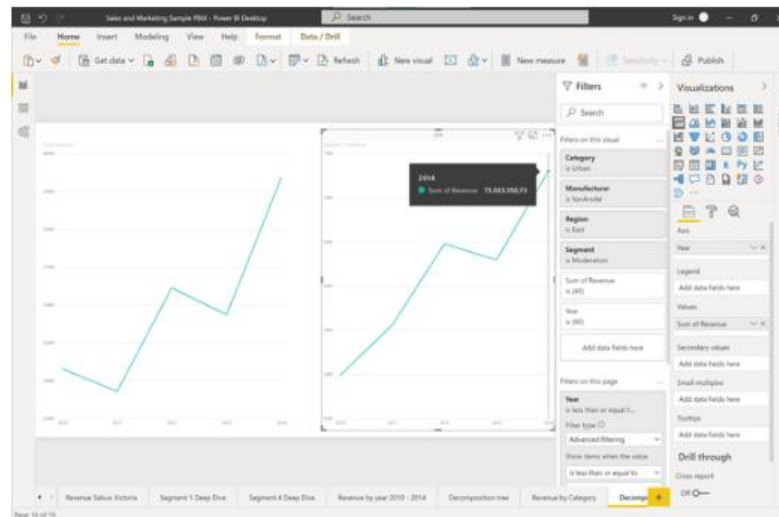
Gambar 6.23. Pohon Dekomposisi Akhir

Ketika kita melihat pohon ini, kita dapat langsung melihat bahwa penjualan dalam kategori Urban, dari Produsen VanArsdel dengan Moderasi Segmen, dan di wilayah Timur merupakan kontributor utama pertumbuhan pendapatan pada periode 2010-2014. Jika kita membuat diagram garis dengan kriteria filter ini dan membandingkannya dengan total pendapatan keseluruhan, kita dapat melihat dengan jelas bagaimana hal ini terjadi pada Gambar 6.24.

Penjualan dari segmen ini saja berkontribusi lebih dari 20 juta pendapatan terhadap total perolehan pendapatan kami dan tumbuh lebih kuat secara relatif (+46%) dibandingkan tren pendapatan absolut (+15%). Dengan Decomposition Tree, pengguna bisnis dapat dengan



mudah berinteraksi dengan data dan menemukan kontributor yang relevan pada tingkat granularitas yang mereka butuhkan. Setelah menyisir data secara otomatis menggunakan fitur AI Power BI, jauh lebih mudah untuk melibatkan analisis data, jika perlu, untuk memverifikasi kesimpulan mereka atau mengajukan pertanyaan yang muncul selama proses analisis.



Gambar 6.24. Perbandingan antara total pendapatan dan cabang Decomposition Tree

Laporan Power BI dengan Decomposition Tree dapat dibagikan dengan pengguna bisnis sehingga mereka dapat menelusuri data sendiri. Untuk berbagi laporan dengan pengguna lain di organisasi Anda, Anda memerlukan server Power BI atau lisensi Power BI pro. Untuk mengetahui lebih lanjut tentang Decomposition Tool, Anda dapat memeriksa sumber daya Microsoft berikut.

6.2 RINGKASAN

Dalam bab ini, kami telah menemukan dua fitur canggih yang didukung AI di Power BI yang membantu pengguna bisnis dan analisis mengurangi waktu mereka untuk mendapatkan wawasan dengan mengotomatiskan bagian-bagian dari proses analisis data. Alat Key Influencer dapat digunakan untuk memunculkan faktor pendorong utama dari metrik tertentu yang diminati dan mengidentifikasi segmen yang relevan. Pohon Dekomposisi yang dikombinasikan dengan pemisahan AI sangat berguna untuk penelusuran terarah dalam dimensi data apa pun sambil menjaga metrik bisnis secara keseluruhan tetap. Hal ini menjadikannya alat yang hebat untuk eksplorasi data ad hoc dan analisis akar penyebab mandiri. Di bab berikutnya, kita akan meninggalkan ranah data masa lalu dan masa kini dan melihat bagaimana AI dapat membantu kita mengantisipasi peristiwa di masa mendatang atau membuat prediksi yang lebih baik tentang hasil di masa mendatang. Oleh karena itu, kita akan memulai dengan tugas klasifikasi otomatis yang membantu kita mengkategorikan peristiwa bisnis di masa mendatang berdasarkan pengalaman masa lalu kita.

BAB 7

ANALISIS PREDIKTIF BERTENAGA AI

7.1 PRASYARAT

Untuk mengikuti Bab ini, Anda memerlukan akun Microsoft Azure untuk melatih dan menghosting model AI seperti yang ditunjukkan di Bab 4, pengetahuan dasar tentang Python atau R untuk mendapatkan prediksi dari model, dan alat BI yang mampu mengeksekusi skrip Python atau R untuk mengintegrasikan prediksi model ke dalam dasbor Anda.

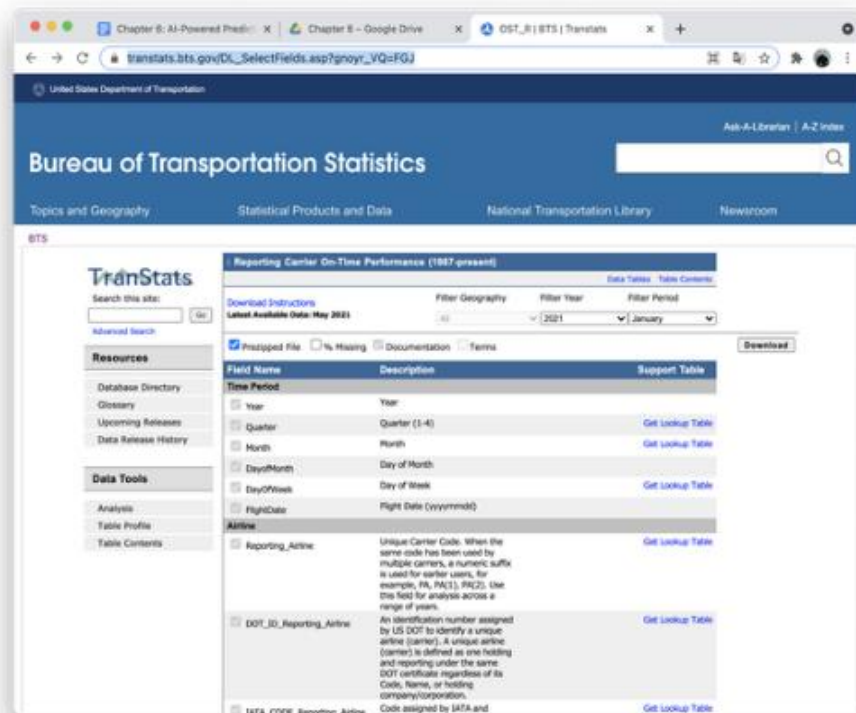
Meskipun kami menggunakan Microsoft Azure untuk melatih model, fungsionalitas yang sama kurang lebih juga tersedia di penyedia platform cloud lain seperti Google Cloud atau Amazon Web Services. Setelah Anda memahami cara kerja Azure, seharusnya relatif mudah untuk memulai dan menjalankannya di platform lain dengan cepat.

Untuk membangun dasbor, kami akan menggunakan Power BI nanti agar konsisten dengan kasus penggunaan sebelumnya (dan juga karena Azure bekerja dengan baik bersama Power BI). Namun, buku ini tidak terbatas pada pengguna Power BI. Itulah sebabnya saya akan menunjukkan cara mendapatkan inferensi dari model Anda menggunakan bahasa pemrograman populer seperti Python atau R. Anda dapat mengintegrasikan kode Python atau R ke dalam Power BI atau alat BI lainnya. Alternatifnya, Anda dapat menjalankan prediksi langsung di basis data (prediksi batch) dan menampilkan hasilnya di perangkat BI apa pun. Hal ini akan dibahas lebih lanjut di bab 10.

Dapatkan Set Data

Perhatian: Kita akan menggunakan set data dunia nyata. Set data ini belum dimodifikasi atau disesuaikan untuk latihan pembelajaran mesin. Jadi, jangan berharap wawasan yang mencengangkan atau grafik yang menarik. Sebaliknya, Anda akan mendapatkan gambaran tentang bagaimana skenario dunia nyata akan terlihat, yang merupakan keuntungan yang dapat Anda harapkan dari pembelajaran mesin dan bagaimana Anda dapat menggunakan teknologi ini untuk melampaui baseline Anda sendiri dan mengotomatiskan prediksi. Prosesnya mungkin akan sedikit rumit, tetapi lebih dekat dengan kehidupan nyata Anda daripada latihan akademis apa pun yang akan Anda temukan.





Untuk mengunduh set data, kunjungi situs web layanan TranStats dari Biro Statistik Transportasi AS. Sumber data ini menyimpan informasi terbaru tentang detail penerbangan di AS. Kami akan menggunakan data penerbangan dari Januari 2021 dan Februari 2021. Anda dapat mengunduh dataset langsung dari situs web dengan memilih opsi "Prezipped file" dan menekan tombol Unduh seperti yang ditunjukkan pada Gambar di bawah.

Untuk mengikuti tutorial buku, saya sarankan untuk menggunakan berkas Excel yang disediakan dalam sumber daya buku. Langkah-langkah pemrosesan berikut telah diterapkan agar berkas tersebut sesuai dengan studi kasus dalam bab ini:

- Dataset telah difilter hanya untuk penerbangan American Airlines.
- Penerbangan yang dibatalkan telah dihapus dari dataset.
- Penerbangan yang tidak memiliki informasi tentang status kedatangan telah dihapus (<1%>).

Silakan kembali ke sumber asli jika Anda ingin mencoba latihan dengan dataset yang diperbarui atau melatih model pada data yang lebih banyak dari satu bulan.

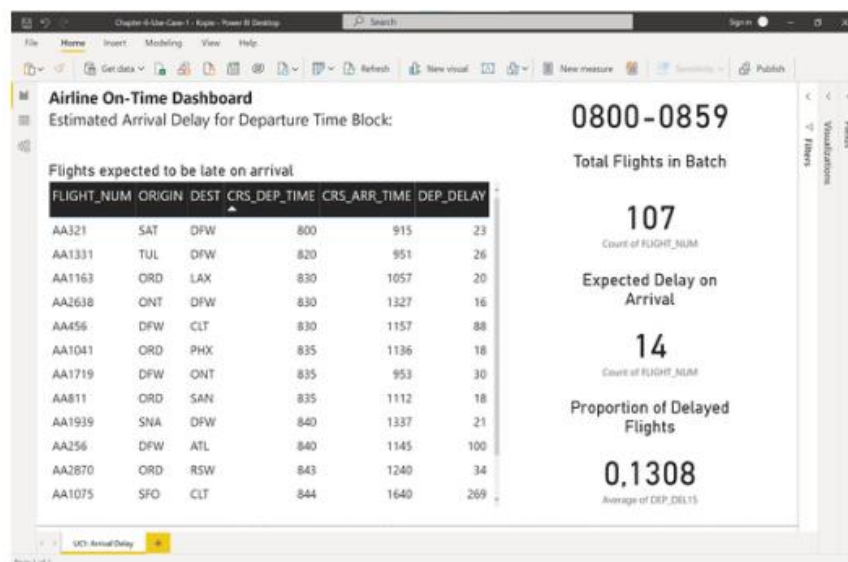
7.2 KASUS PENGGUNAAN: MENGOTOMATISKAN TUGAS KLASIFIKASI

Untuk kasus penggunaan pertama kita, kita menangani masalah klasifikasi di mana kita memprediksi apakah suatu penerbangan akan mengalami penundaan kedatangan atau tidak. Meskipun Anda bukan ahli dalam industri penerbangan, Anda seharusnya sudah familiar dengan tantangan keseluruhannya. Untuk menjelaskannya lebih detail, mari kita perkenalkan skenario konkret dan rumusan masalah sebagai berikut.

Pernyataan Masalah

Sebagai anggota tim BI American Airlines, kami dibekali dengan tugas untuk membangun dasbor pelaporan berdasarkan metrik penting bisnis. Metrik ini adalah persentase penerbangan American Airlines yang saat ini sedang mengudara dan diperkirakan akan tertunda lebih dari 15 menit. Metrik ini penting untuk tujuan pengendalian kualitas dan menjaga kepuasan pelanggan tetap tinggi, serta mempertahankan jadwal penerbangan secara keseluruhan.

Data penerbangan terbaru masuk dalam batch per jam karena alasan yang tidak dapat kami pengaruhi atau ubah. Jadi, misalnya, ketika kami melaporkan metrik kapan saja antara pukul 09.00 dan 09.59, data terbaru yang dapat kami lihat adalah dari batch keberangkatan antara pukul 08.00 dan 08.59. Metrik ini saat ini dilaporkan melalui dasbor BI yang terlihat seperti Gambar di bawah:



Tujuan kami adalah menghasilkan pengklasifikasi baru yang mampu melampaui nilai dasar F1-Score 0,72 ini, sehingga kami dapat membuat prediksi yang lebih akurat tentang penundaan penerbangan.

Dasbor menampilkan jumlah total penerbangan saat ini (107) yang dijadwalkan untuk periode keberangkatan pukul 08.00-08.59 pada tanggal 7 Februari 2021. Sebagian besar penerbangan ini akan tetap mengudara sebelum data batch berikutnya masuk. Untuk memperkirakan apakah akan ada penundaan kedatangan atau tidak, saat ini kami menggunakan informasi penundaan keberangkatan sebagai proksi. Sejauh ini, tim BI telah menggunakan variabel DepDel15, yang menunjukkan apakah penerbangan tertunda setidaknya 15 menit setelah keberangkatan, untuk memperkirakan apakah penerbangan juga akan terlambat lebih dari 15 menit setelah kedatangan.

Estimasi ini merupakan langkah awal yang baik, tetapi masih jauh dari sempurna. Mari kita buka buku kerja Excel AA_Flights_2021_1.xlsx seperti yang terlihat pada Gambar di bawah

untuk mengetahui kinerja estimasi ini secara historis. Spreadsheet Excel berisi semua penerbangan American Airlines dari Januari 2021 hampir 38.000 baris data. Setiap baris adalah penerbangan American Airlines yang terjadi selama Januari 2021.

Karena ini adalah kumpulan data historis, kita dapat melihat keterlambatan kedatangan aktual yang terjadi. Ambil baris pertama sebagai contoh. Penerbangan Nomor AA1 dari Bandara John F. Kennedy ke Los Angeles lepas landas pada 1 Januari 2021. Penerbangan tersebut tidak mengalami keterlambatan keberangkatan sehingga kolom DepDelay15 diatur ke 0. Karena mendarat bahkan lebih cepat dari jadwal tanpa keterlambatan, kolom ArrDelay15 juga diatur ke 0.

Penerbangan pertama yang mengalami keterlambatan keberangkatan lebih dari 15 menit terjadi di baris 21: Ini lagi-lagi penerbangan AA1 dari JFK ke LAX, tetapi kali ini pada 20 Januari 2021. Atribut DepDelay15 adalah 1 karena keterlambatan keberangkatan adalah 75 menit. Penerbangan mendarat dengan penundaan 61 menit dan oleh karena itu atribut ArrDel15 adalah 1.

Year	Quarter	Month	DayOfMonth	DayOfWeek	FlightDate	Reporting_Airline	DOT ID	Reporting_Airline	IATA_CODE_Reporting_Airline	Tail Numbr
2021	1	1	1	5	2021-01-01	AA	19805	AA	AA	N767AJ
2021	1	1	2	6	2021-01-02	AA	19805	AA	AA	N795AN
2021	1	1	3	7	2021-01-03	AA	19805	AA	AA	N772AN
2021	1	1	4	1	2021-01-04	AA	19805	AA	AA	N788AN
2021	1	1	5	2	2021-01-05	AA	19805	AA	AA	N781AN
2021	1	1	6	3	2021-01-06	AA	19805	AA	AA	N771AN
2021	1	1	7	4	2021-01-07	AA	19805	AA	AA	N781AN
2021	1	1	8	5	2021-01-08	AA	19805	AA	AA	N781AN
2021	1	1	9	6	2021-01-09	AA	19805	AA	AA	N780AN
2021	1	1	10	7	2021-01-10	AA	19805	AA	AA	N794AN
2021	1	1	11	1	2021-01-11	AA	19805	AA	AA	N757AN
2021	1	1	12	2	2021-01-12	AA	19805	AA	AA	N781AN
2021	1	1	13	3	2021-01-13	AA	19805	AA	AA	N772AN
2021	1	1	14	4	2021-01-14	AA	19805	AA	AA	N772AN
2021	1	1	15	5	2021-01-15	AA	19805	AA	AA	N787AL
2021	1	1	16	6	2021-01-16	AA	19805	AA	AA	N793AN
2021	1	1	17	7	2021-01-17	AA	19805	AA	AA	N791AN
2021	1	1	18	1	2021-01-18	AA	19805	AA	AA	N783AN
2021	1	1	19	2	2021-01-19	AA	19805	AA	AA	N787AL
2021	1	1	20	3	2021-01-20	AA	19805	AA	AA	N761AJ
2021	1	1	21	4	2021-01-21	AA	19805	AA	AA	N787AL
2021	1	1	22	5	2021-01-22	AA	19805	AA	AA	N793AN
2021	1	1	23	6	2021-01-23	AA	19805	AA	AA	N783AN
2021	1	1	24	7	2021-01-24	AA	19805	AA	AA	N788AN
2021	1	1	25	1	2021-01-25	AA	19805	AA	AA	N760AN

Jadi, dalam kasus ini, atribut DepDelay15 merupakan indikator yang baik untuk mengetahui apakah penerbangan tersebut benar-benar mengalami penundaan kedatangan. Nah, seberapa akurat prediksi kita jika kita hanya mengandalkan informasi DepDelay15? Buka lembar kerja

kedua pada berkas Excel untuk melihat "Tabel Kebingungan". Anda akan melihat tabel seperti pada Gambar di bawah.

	A	B	C	D
1	Classifier Evaluation			
2				
3			COUNT	PERC
4	BASELINE PREDICTION	0	34.169	90,1%
5	ACTUAL	0	32.991	96,6%
6	ACTUAL	1	1.178	3,4%
7	BASELINE PREDICTION	1	3.737	9,9%
8	ACTUAL	0	988	26,4%
9	ACTUAL	1	2.749	73,6%
10		Total	37.906	100,0%
11				
12		ArrDelay15	True	10%
13			False	90%

Sebelum kita membahas detailnya, perhatikan dua baris terakhir yang menunjukkan ArrDelay15 Benar = 10% dan Salah = 90%. Ringkasan ini menunjukkan bahwa kita sedang menghadapi apa yang disebut masalah "klasifikasi tidak seimbang". Hanya 10% dari semua penerbangan yang ditandai dengan atribut ArrDelay15 dan dengan demikian terlambat lebih dari 15 menit saat kedatangan.

Mengapa ketidakseimbangan ini penting bagi kita? Karena hal ini memengaruhi metrik evaluasi kita. Bayangkan jika kita memprediksi bahwa semua penerbangan akan tepat waktu dengan keterlambatan kurang dari 15 menit, ini akan memberi kita skor akurasi yang mencengangkan, yaitu 90%. Apakah prediksi ini berguna? Tidak. Itulah sebabnya kita perlu membuat metrik evaluasi yang lebih cerdas.

Lihat tabel pivot di atas. Ini menunjukkan kinerja prediksi dasar kita (label DepDelay15). Jika kita menggunakan heuristik Keterlambatan Keberangkatan sebelumnya, kita akan salah mengklasifikasikan 2.166 observasi: 1.178 penerbangan sebenarnya tertunda meskipun kita memperkirakan akan tepat waktu. Dan 988 penerbangan sebenarnya tepat waktu, meskipun kita memperkirakan akan ada penundaan.

Bagaimana kita mengompresi informasi ini menjadi metrik yang mudah dilacak? Kita menggunakan metrik yang disebut "F1-Score" yang merupakan kombinasi dari dua metrik lainnya: Presisi dan perolehan kembali.

Presisi adalah ukuran yang menangkap ketepatan pengklasifikasi, di mana presisi yang lebih tinggi menunjukkan lebih sedikit positif palsu. Jika kita berhasil meningkatkan metrik ini, 988 penerbangan yang diprediksi tertunda, tetapi sebenarnya tepat waktu, akan berkurang. Presisi dihitung dengan membagi nilai positif benar (2,749) dengan jumlah positif benar (2,749) dan positif palsu (988). Recall mengukur kelengkapan pengklasifikasi, di mana recall yang lebih tinggi menunjukkan lebih sedikit negatif palsu (1,178 penerbangan tertunda meskipun diprediksi tepat waktu). Recall dihitung dengan membagi positif benar (2,749) dengan positif benar ditambah negatif palsu (1,178).



Skor F1 adalah rata-rata harmonik presisi dan recall, dan dihitung sebagai berikut:

$$F1 = 2 * (\text{presisi} * \text{recall}) / (\text{presisi} + \text{recall}) = 2TP / (2TP + FP + FN)$$

Seperti yang Anda lihat pada gambar di bawah, Skor F1 untuk pengklasifikasi dasar kami adalah 0,72.

DASAR		
Benar Positif	2.749	
Positif Palsu	988	
Benar Negatif	32.991	
Negatif Palsu	1.178	
Presisi = TP / (TP + FP)		0,74
Ingat = TP/(TP + FN)		0,7
Skor F1 = 2TP / (2TP + FP + FN)		0,72

Tujuan kami adalah menciptakan pengklasifikasi baru yang mampu melampaui nilai dasar Skor F1 0,72 ini, sehingga kami dapat membuat prediksi yang lebih akurat tentang keterlambatan penerbangan.

Ikhtisar Solusi

Untuk meningkatkan prediksi keterlambatan, kami akan menggunakan pengklasifikasi AI yang dilatih berdasarkan data historis. AI akan mempelajari pola dari masa lalu yang memengaruhi keterlambatan dan membantu kami mengklasifikasikan titik data baru (yang belum pernah terlihat) berdasarkan contoh dari masa lalu. Masalah ini disebut masalah klasifikasi.

Ada banyak pendekatan untuk menyelesaikan masalah klasifikasi. Karena kami tidak ingin melibatkan pakar AI atau ML, kami akan menggunakan AutoML untuk melatih model. Dalam kasus khusus ini, kami menggunakan Microsoft Azure sebagai platform AutoML, tetapi platformnya bisa apa saja semuanya bekerja kurang lebih sama.

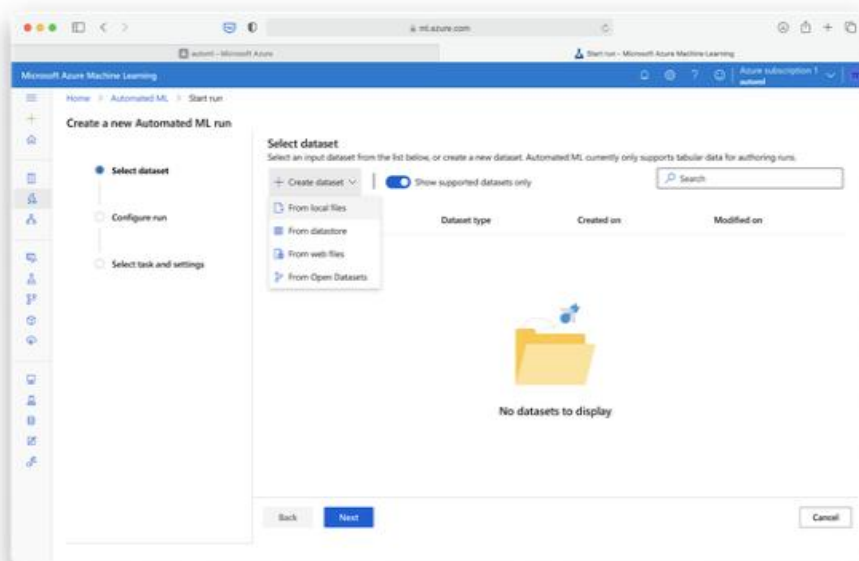
Pertama-tama, kami akan membangun model di Azure, lalu mengujinya pada data baru yang belum pernah terlihat. Setelah evaluasi akhir ini, kami akan mempelajari cara mengirimkan permintaan ke model menggunakan API daring dan akhirnya mengintegrasikan wawasan ini ke dalam dasbor BI. Kami akan menggunakan Power BI untuk tujuan ini guna menyajikan hasil dalam format dasbor.

Panduan Pelatihan Model dengan Microsoft Azure

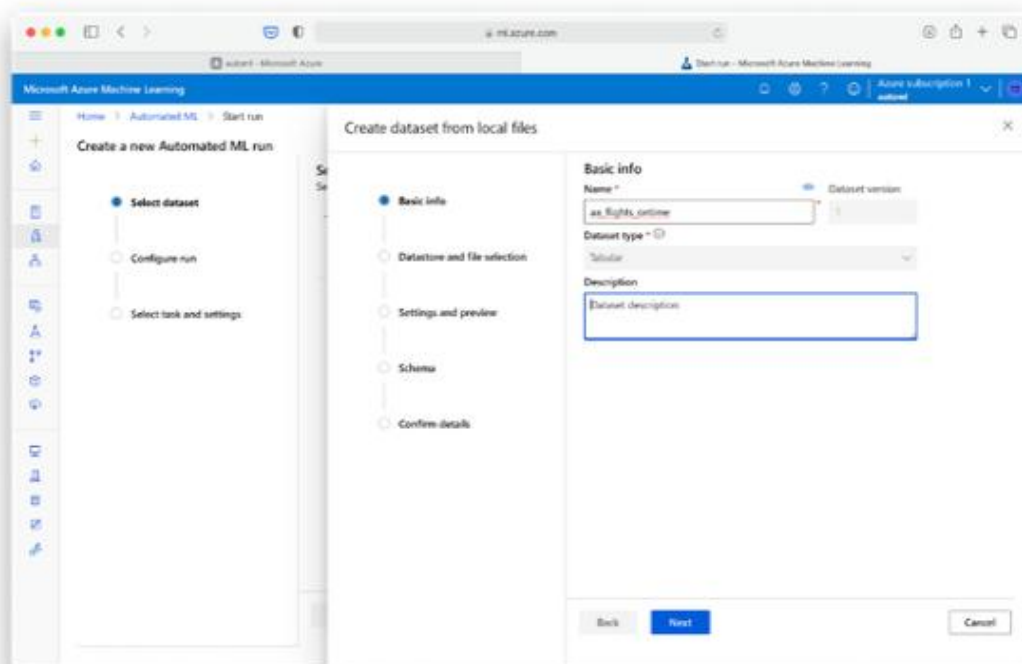
Kunjungi Azure Machine Learning Studio dan pilih sumber daya yang Anda buat di bab 3. Karena kita ingin melatih model AI menggunakan AutoML, pilih "Mulai Sekarang" pada panel Automated ML dan pilih "New Automated ML run" untuk mengisi baris kosong.

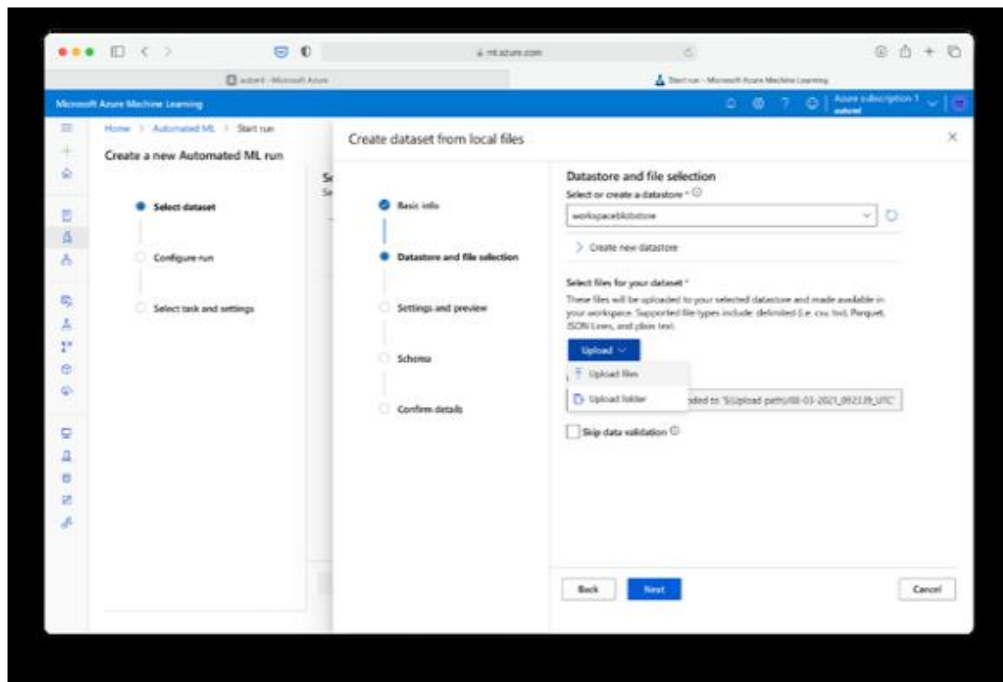
Sebelum kita dapat menyiapkan AutoML run, kita memerlukan beberapa data. Di Azure, data disusun dalam set data yang terhubung ke ruang kerja Anda. Set data akan memastikan bahwa data Anda diformat dengan benar dan dapat digunakan oleh machine learning run. Seperti yang ditunjukkan pada Gambar di bawah, pilih "buat set data".





Anda dapat mengimpor data dari berbagai sumber. Dalam kasus kami, kami akan mengunggah berkas lokal, jadi silakan pilih "Dari berkas lokal" seperti yang ditunjukkan pada Gambar di bawah. Pada formulir Info Dasar yang ditunjukkan pada Gambar di bawah, berikan nama yang jelas untuk dataset Anda dan tambahkan deskripsi jika diinginkan. Dalam kasus kami, saya menamai dataset ini aa_flights_ontime. Versi default-nya adalah 1 dan tipe dataset akan berupa Tabular karena AutoML saat ini hanya mendukung data tabular.



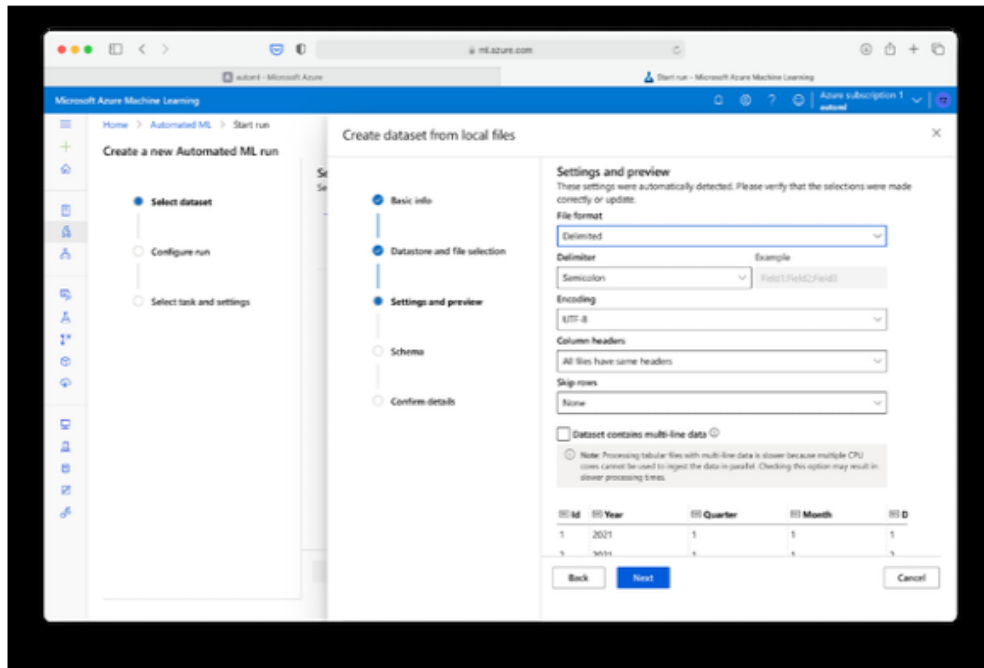


Klik Berikutnya dan lanjutkan ke bagian berikutnya. Anda perlu memilih lokasi penyimpanan data dan dari mana data tersebut akan diunggah. Pilih penyimpanan data default yang telah diatur secara otomatis saat Anda membuat ruang kerja (dalam contoh saya, ini disebut "workspaceblobstore").

Di sinilah Anda akan mengunggah berkas data secara fisik agar tersedia di ruang kerja Anda saat ini. Klik unggah dan pilih berkas "AA-Flights-2021-01-Training.csv". Berkas ini berisi data yang sama dengan lembar kerja "AA_Flights_2021_1.xlsx" yang telah kita bahas sebelumnya. Saya baru saja mengonversinya ke CSV untuk kenyamanan Anda. Pilih Berikutnya di bagian bawah layar untuk mengunggah berkas. Tergantung koneksi jaringan Anda, proses ini mungkin memerlukan waktu beberapa saat.

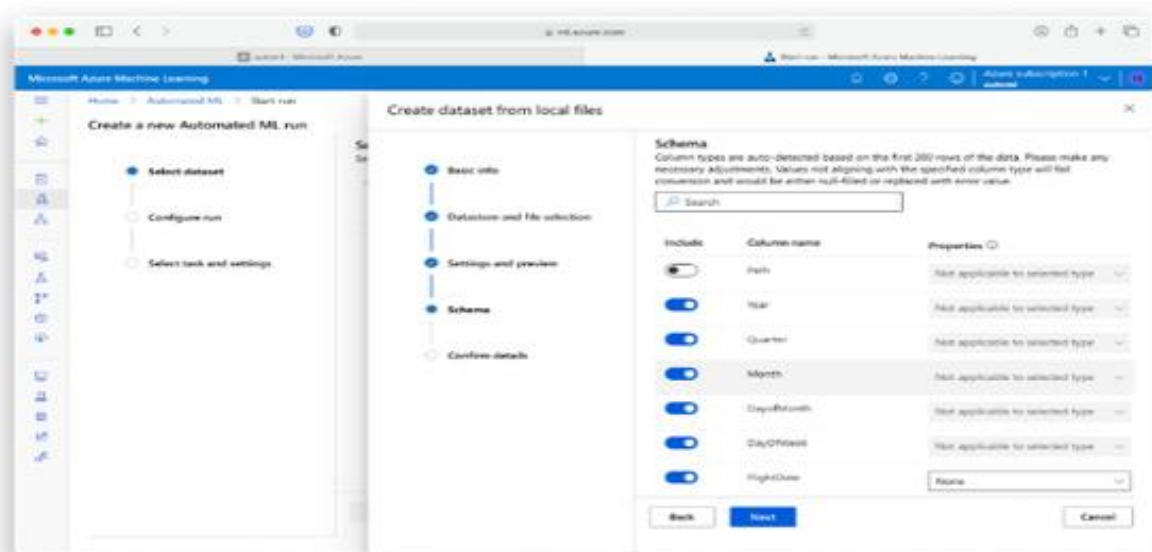
Setelah berkas berhasil diunggah, Anda akan melihat formulir "Pengaturan dan pratinjau" dengan beberapa informasi yang telah diisi sebelumnya berdasarkan berkas yang diunggah.





Untuk tabel contoh kita, periksa kembali apakah pengaturannya sudah seperti yang ditunjukkan pada Gambar di bawah dan pratinjau data pada tombol menampilkan beberapa baris pertama dengan benar. Jika semuanya sudah benar, klik Berikutnya.

Formulir berikut akan menentukan Skema dataset Anda, dan hal ini sangat penting karena dua alasan: Pertama, skema akan memungkinkan Anda menentukan kolom mana dalam dataset Anda yang harus dipertimbangkan oleh algoritma AutoML. Dan kedua, skema memungkinkan Anda menentukan tipe data masing-masing kolom.



Langkah ini penting karena kolom dan tipe data yang digunakan hampir tidak dapat disimpulkan oleh data itu sendiri. Anda perlu mengetahui hal ini terlebih dahulu. Dalam contoh



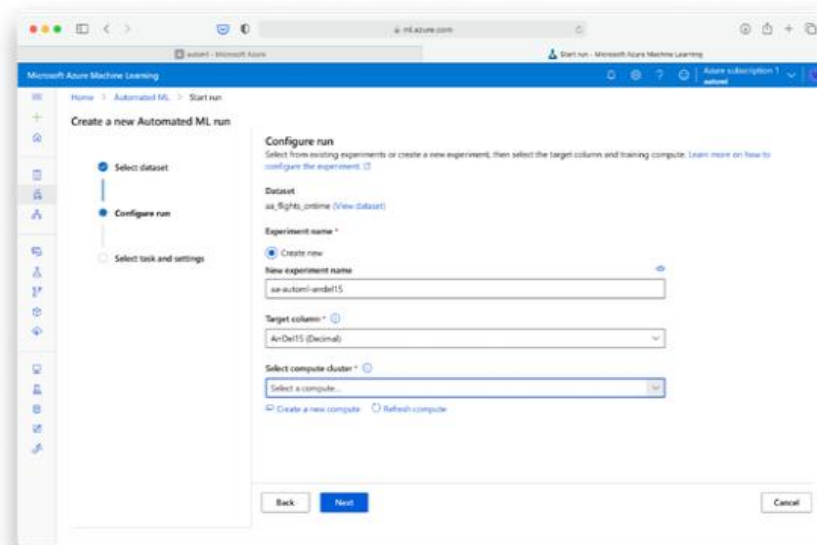
ini, kami tidak memerlukan semua data yang tersedia dalam kumpulan data. Beberapa nilai tidak memberikan wawasan yang berarti sama sekali dan hanya akan meningkatkan kompleksitas proyek (seperti Tahun yang semuanya sama). Dan beberapa nilai lain tidak ingin kami sertakan dalam prediksi kami karena mengandung sesuatu yang disebut "kebocoran target" dalam konteks ML. Misalnya, penundaan kedatangan memberi tahu kami dengan sempurna apakah ArrDel15 benar atau salah. Dan kedua, kami ingin mengatur tipe data dengan benar. Misalnya, masing-masing grup data harus berupa string dan bukan numerik karena Anda tidak dapat menghitung dengan benar dengannya (hari kerja 7 + 1 bukan hari kerja 8, tetapi hari kerja 1). Jadi, pilih hanya data berikut dengan tipe data yang sesuai:

- DayofWeek - String
- Asal - String
- Tujuan - String
- DepDelay - Des
- DepDelayMinutes - Des
- DepDel15 - String
- DepartureDelayGroups - String
- DepTimeBlk - String
- TaxiOut - Des
- ArrDel15 - String
- ArrTimeBlk - String
- Jarak - Des
- DistancingGroup - Str

Pilih Berikutnya. Pada halaman Konfirmasi detail, periksa kembali informasi yang Anda berikan dan klik Buat untuk menyelesaikan pembuatan set data Anda. Anda dapat memilih set data dari daftar yang muncul. Proses dengan memilih Berikutnya. Anda akan melihat layar Konfigurasi Jalankan seperti yang ditunjukkan pada Gambar di bawah.

Apa itu AutoML Run? Setelah kita mendefinisikan set data kita, kita perlu menentukan sumber daya komputasi mana yang akan digunakan untuk pelatihan AutoML. Di Azure, satu pelatihan AutoML pada set data tertentu disebut AutoML run. Run diatur dalam eksperimen. Eksperimen menyimpan informasi seperti ukuran lingkungan komputasi Anda dan menentukan kolom mana yang ingin Anda prediksi.

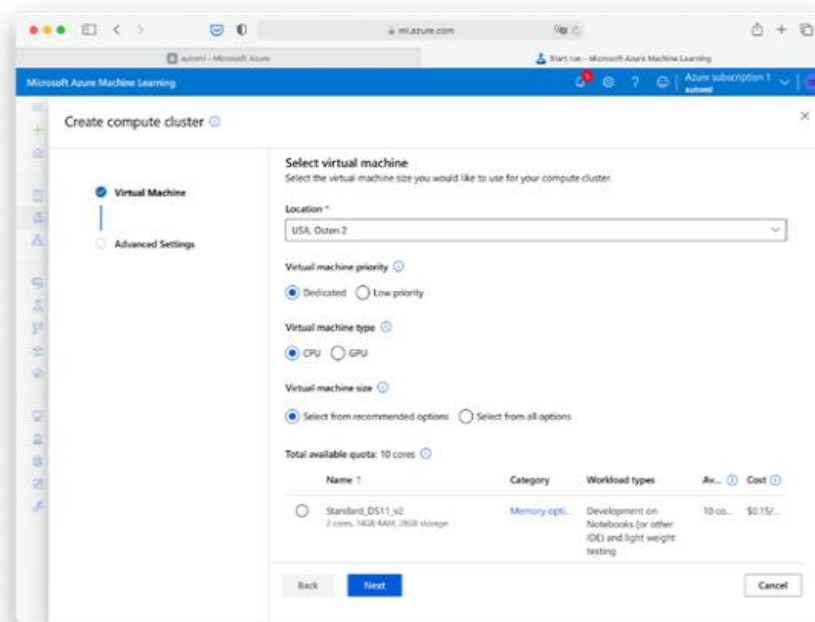




Karena kita belum menyiapkan eksperimen, pilih tombol radio "Buat baru". Masukkan nama eksperimen kustom. Dalam kasus ini, saya memilih aa-automl-arrdel15. Tentukan ArrDel15 (String) sebagai kolom target. Terakhir, kita harus menentukan sumber daya komputasi mana yang akan digunakan untuk menjalankan pelatihan ini. Klik "Buat komputasi baru" dan Anda akan melihat layar seperti yang ditunjukkan pada Gambar di bawah.

Di sini, Anda dapat mengonfigurasi sumber daya komputasi yang digunakan untuk eksperimen. Semakin tinggi kinerja yang Anda alokasikan, semakin cepat pelatihan akan berlangsung. Perlu diketahui bahwa Anda akan dikenakan biaya untuk sumber daya komputasi dan pelatihan AutoML. Meskipun sumber daya komputasi yang lebih besar mungkin akan meningkatkan kecepatan AutoML (dan dengan demikian mengurangi biaya layanan AutoML), biasanya lebih hemat biaya untuk memilih sumber daya komputasi yang lebih murah dan menerima pelatihan AutoML yang sedikit lebih lama.





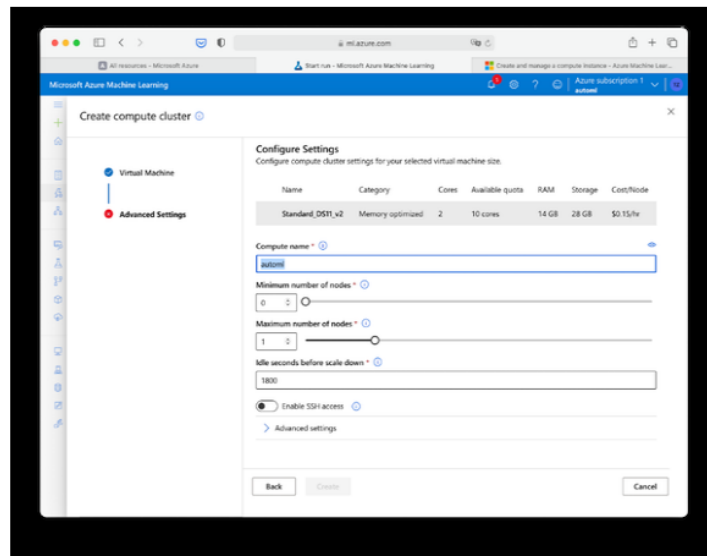
Anda dapat melihat biaya yang tercantum di bawah ini pada sumber daya yang disarankan: mengurangi waktu yang dibutuhkan untuk AutoML berlangsung. Mesin "Standard_DS11_v2" dengan 2 inti, RAM 14GB, dan penyimpanan 28GB berharga sekitar Rp 1.500 per jam di wilayah AS Timur. Azure akan menyarankan daftar ukuran yang direkomendasikan berdasarkan data dan jenis eksperimen Anda. Untuk eksperimen kita, pilih pengaturan berikut:

- Lokasi: AS Timur 2
- Prioritas mesin virtual: Khusus
- Jenis mesin virtual: CPU
- Ukuran mesin virtual: Standard_DS12_V2

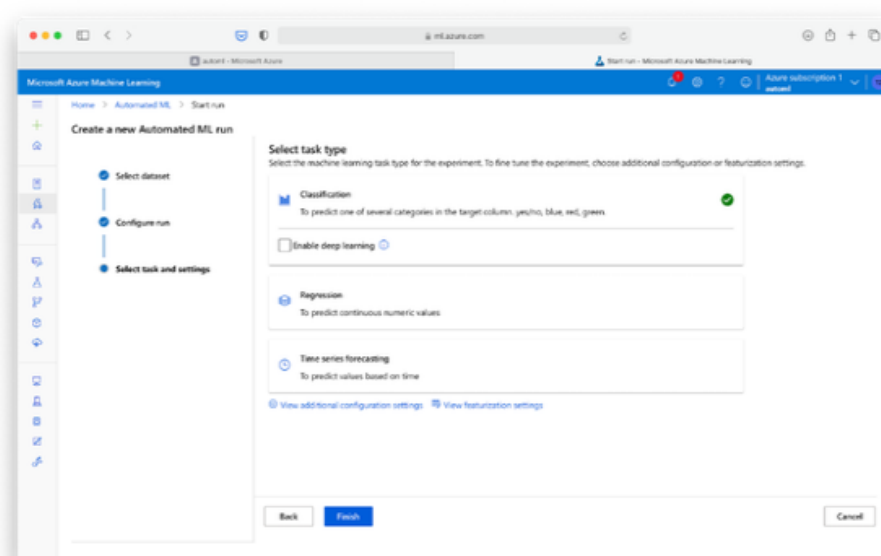
Pilih Berikutnya. Anda akan melihat formulir yang mirip dengan yang ada pada Gambar di bawah. Masukkan nama untuk instans komputasi Anda. Saya memilih "automl" pada contoh di bawah ini. Berikan juga jumlah minimum dan maksimum node komputasi. Seperti yang Anda lihat, sumber daya komputasi cloud sebenarnya dirancang untuk menjadi kluster mesin virtual yang dapat menangani set data yang sangat besar sekalipun. Karena set data kita tidak terlalu besar, kita tidak memerlukan kluster mesin, cukup satu mesin saja. Pilih Jumlah minimum node 0 dan jumlah maksimum node 1.

Atur "Detik diam sebelum skala turun" ke 1800. Ini akan memastikan mesin kita akan siaga setelah 30 menit tidak aktif dan mencegah kita dikenakan biaya. Nonaktifkan akses SSH dan lewati pengaturan lanjutan. Klik Buat. Proses ini mungkin memerlukan waktu beberapa menit. Setelah selesai, Anda akan diarahkan ke layar "Konfigurasi proses" pada Gambar di bawah dengan sumber daya komputasi baru yang telah terisi sebelumnya di menu tarik-turun sumber daya komputasi. Lanjutkan dengan Berikutnya.





Sekarang saatnya menentukan tugas AutoML yang ingin kita lakukan.

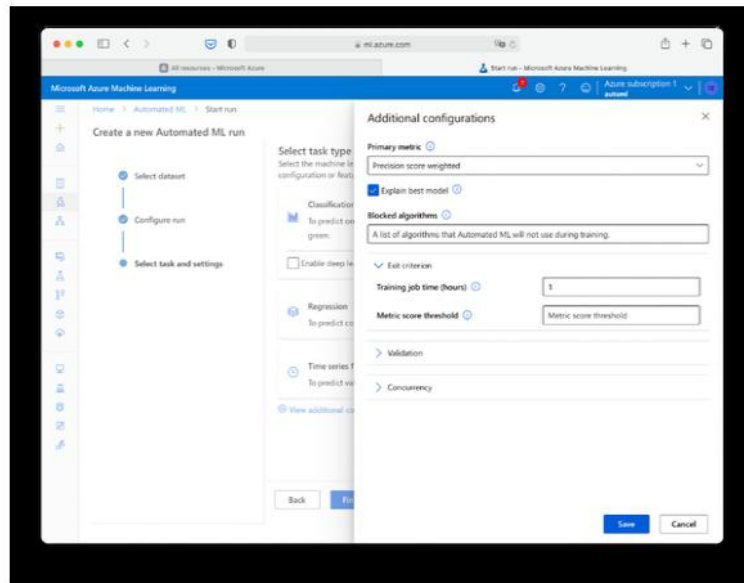


Kita dapat memilih di antara tiga jenis berbeda di sini:

- Klasifikasi: Memprediksi variabel kategoris
- Regresi: Memprediksi variabel numerik kontinu
- Peramalan deret waktu: Regresi untuk data deret waktu

Apakah Anda memahami mengapa penting untuk mengatur tipe data dengan benar sebelum kita membuat set data kita? Jika kita membiarkan tipe data variabel ArrDel15 menjadi Desimal, maka kita tidak akan dapat memilih Klasifikasi di sini. Demikian pula, jika kita ingin memprediksi nilai numerik, kita harus memastikannya diatur dengan benar dalam skema untuk mengaktifkan regresi. Auto ML tidak dapat menebak tipe data itu sendiri dengan andal. Itu adalah sesuatu yang harus kita definisikan secara manual, terutama untuk kolom target.

Sebelum Anda memilih Selesai, klik "Lihat pengaturan konfigurasi tambahan". Anda akan melihat panel konfigurasi tambahan seperti yang ditunjukkan pada Gambar di bawah.



Ada beberapa hal yang ingin kami sesuaikan di sini. Pertama, kami ingin mengubah metrik Primer. Secara default, metrik ini diatur ke "Akurasi". Seperti yang telah kita pelajari sebelumnya, Akurasi bukanlah ukuran yang baik untuk masalah klasifikasi yang tidak seimbang. Kita tidak dapat memilih Skor F1 di sini. Sebagai gantinya, pilih "Skor Presisi Tertimbang". Ini juga akan mempertimbangkan Recall dan oleh karena itu akan menghasilkan Skor F1 yang lebih baik. Pastikan juga bahwa "Jelaskan model terbaik" dicentang.

CATATAN

Memilih metrik evaluasi yang tepat merupakan langkah penting dalam proses Auto ML Anda dan dapat memengaruhi hasil secara signifikan. Di saat yang sama, sulit untuk menebak metrik mana yang Anda butuhkan untuk layanan Auto ML, karena hal ini bergantung pada masalah yang dihadapi. Misalnya, apakah Anda lebih memprioritaskan hasil Positif Palsu daripada Negatif Palsu seperti dalam uji klinis? Kenali berbagai metrik dan bereksperimenlah dengan metrik tersebut dalam berbagai proses AutoML. Kunjungi "Eksperimen" di Azure ML Studio untuk membandingkan proses Auto ML Anda berdasarkan berbagai metrik.

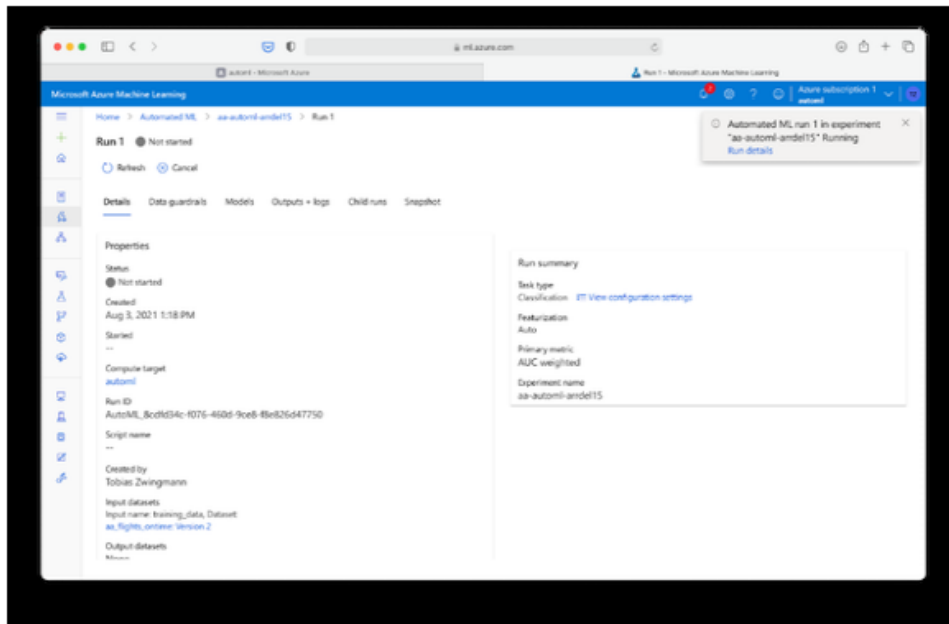
Sumber Daya

Pengaturan lain yang ingin kami sesuaikan adalah "Kriteria Keluar". Ini akan memberi tahu algoritme AutoML kami kapan modelnya "cukup baik". Jika kriteria terpenuhi, pekerjaan pelatihan dihentikan. Biasanya ada dua pendekatan yang dapat Anda lakukan. Anda dapat mengakhiri pelatihan berdasarkan batas waktu atau berdasarkan kriteria penerimaan



minimum untuk metrik tersebut. Dalam kasus kami, pilih 1 jam sebagai batas waktu maksimum. Bahkan, kemungkinan besar pelatihan dapat berakhir lebih awal, jika Algoritma tidak dapat memperoleh peningkatan yang lebih signifikan setelah melatih model yang lebih baik.

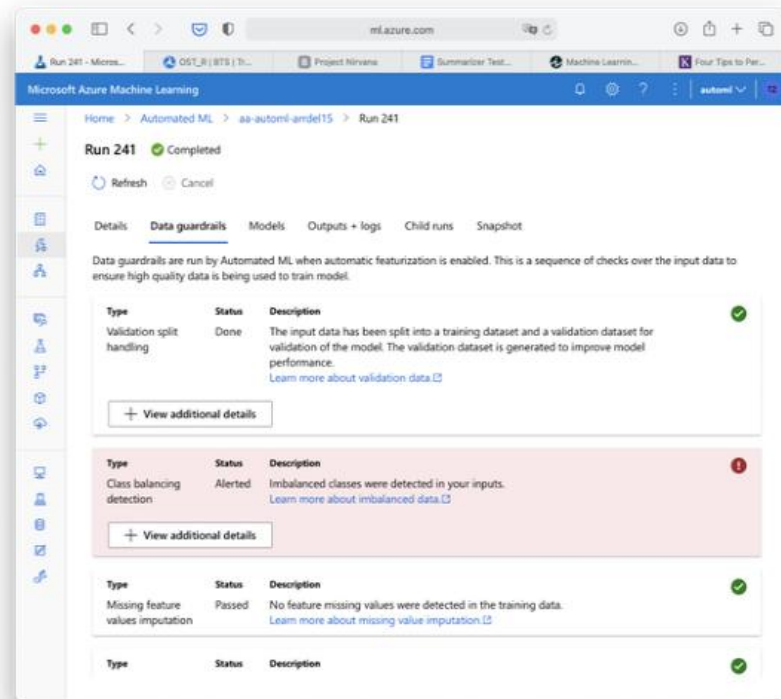
Konfirmasikan pengaturan tambahan dengan "Simpan". Dan klik selesai untuk memulai proses AutoML Anda. Ini akan menginisialisasi proses AutoML Anda dan membawa Anda ke layar "Detail Proses" seperti yang ditunjukkan pada Gambar di bawah.



Status ini diperbarui seiring berjalannya eksperimen. Setelah beberapa menit, Status akan berubah dari "Belum dimulai" menjadi "Berjalan".

Kemungkinan akan memakan waktu sekitar 10 menit sebelum model pertama benar-benar dilatih. Ada beberapa prosedur persiapan dan pra-pekerjaan yang sedang dilakukan sebelumnya. Salah satu persiapan ini adalah fitur yang disebut "Data guardrails". Klik tab "Data guardrails" untuk mempelajari lebih lanjut. Setelah beberapa menit, tampilannya akan seperti pada gambar di bawah. Jika area ini masih kosong, periksa kembali setelah beberapa menit dan segarkan halaman.





Data guardrails pada Azure Auto ML adalah serangkaian pemeriksaan otomatis atas data pelatihan untuk meningkatkan peluang hasil pelatihan berkualitas tinggi.

Pemeriksaan ini mencakup pemisahan data menjadi set pelatihan dan validasi secara otomatis, atau pemeriksaan nilai yang hilang di kolom pelatihan. Fitur ini juga memeriksa beberapa asumsi yang terkait dengan tugas pelatihan yang sedang dilakukan, misalnya mengenali kelas yang tidak seimbang dalam label dan menandainya sebagai area masalah potensial.

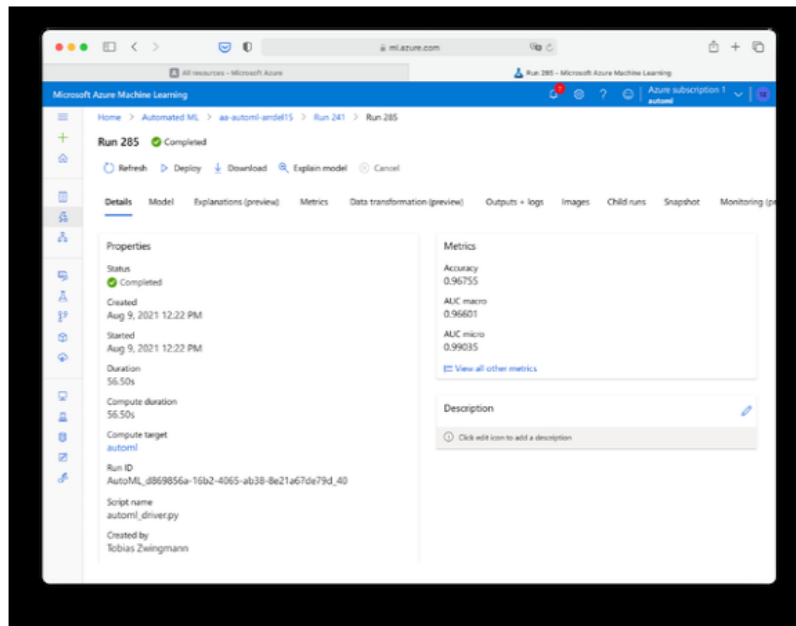
Jika Anda mengklik tombol "Lihat detail tambahan", Anda akan melihat distribusi label kelas yang mendasarinya. Untuk mempelajari lebih lanjut tentang cara mengatasi masalah ini, Azure menyediakan tautan ke "Pelajari lebih lanjut tentang data yang tidak seimbang". Selain beberapa fitur bawaan yang membantu mengatasi ketidakseimbangan kelas seperti menambahkan kolom bobot, satu poin utama yang mereka sarankan adalah memilih metrik evaluasi yang tangguh terhadap kelas yang tidak seimbang.

Keputusan yang tepat karena kami memilih Presisi daripada Akurasi sebagai metrik utama kami sebelumnya. Selagi proses pelatihan secara keseluruhan masih berjalan, buka tab "Model". Di sini Anda akan melihat daftar semua model dan metrik evaluasi awal yang telah dihasilkan oleh algoritma AutoML. Setelah 15 hingga 20 menit, Anda seharusnya sudah melihat beberapa model awal dengan metrik evaluasinya masing-masing.

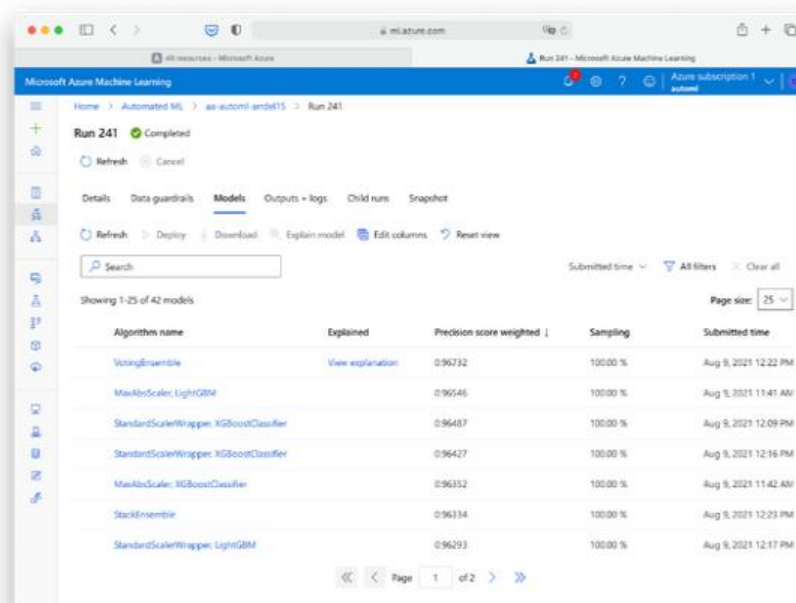
Pertimbangkan model-model ini karena hasilnya akan terus meningkat. Jika algoritma AutoML menemukan model yang lebih baik, model tersebut akan muncul di bagian atas daftar ini. Anda dapat menjelajahi setiap model di sini dan melihat lebih lanjut detailnya. Namun, ide AutoML adalah kita tidak terlalu khawatir tentang apa yang terjadi selama pelatihan. Biarkan algoritma melakukan tugasnya, ambil kopi, dan periksa kembali dalam 30 menit atau lebih.



Apakah pelatihan sudah selesai? Jika sudah, Anda akan melihat layar seperti pada gambar di bawah dengan status "Selesai".



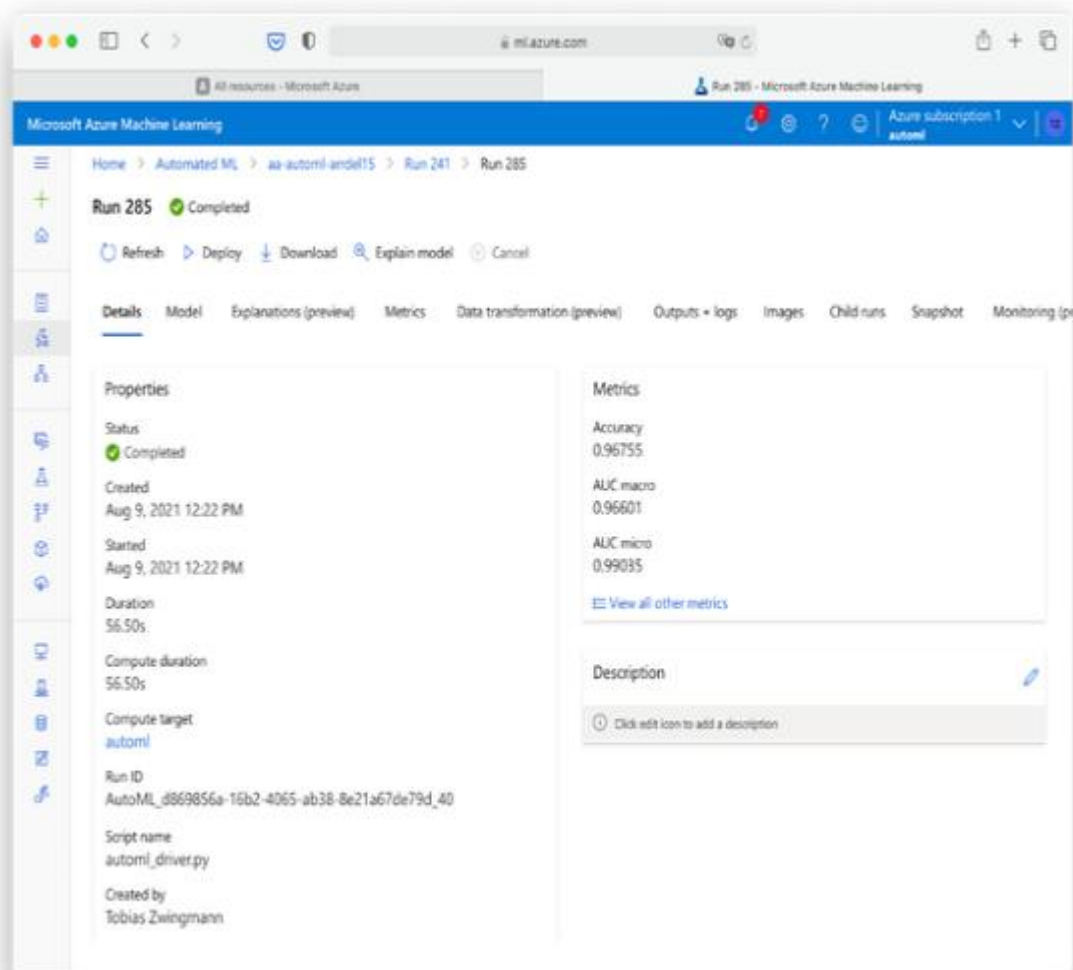
Sekarang saatnya untuk melihat model terbaik yang ditemukan algoritma dan mengeksplorasinya lebih detail. Di sisi kanan layar "Detail", Anda akan menemukan "Ringkasan Model Terbaik". Bagian ini menampilkan beberapa metrik penting seperti nama algoritma, dalam hal ini adalah "Voting Ensemble" dan skor Presisi akhir yang dicapai, yaitu 0,96732. Hasil aktual bergantung pada sumber daya komputasi dan waktu pelatihan yang dihabiskan. Hasil mungkin terlihat sedikit berbeda jika Anda memilih pengaturan yang berbeda untuk konfigurasi proses.



Buka model dengan mengklik tab "Model". Anda akan melihat daftar yang mirip dengan Gambar di bawah. Dari ikhtisar ini, kita dapat melihat bahwa 5 model teratas sebenarnya sangat berdekatan, dengan skor Presisi berkisar antara 0,96352 dan 0,96732 untuk model terbaik. Seperti yang telah kita pilih sebelumnya, model terbaik juga menyediakan penjelasan. Kita akan membahasnya nanti. Namun sebelumnya, klik nama algoritma "VotingEnsemble" untuk melihat halaman detail model seperti yang ditunjukkan pada Gambar di bawah.

CATATAN

Hasil aktual untuk proses AutoML Anda mungkin sedikit berbeda dari hasil di buku ini. Meskipun proses AutoML bersifat deterministik, Anda mungkin melihat hasil yang berbeda karena prosedur stokastik pada proses persiapan data (misalnya, pengambilan sampel data otomatis untuk pelatihan dan pengujian). Namun, gambaran besarnya seharusnya sangat mirip.

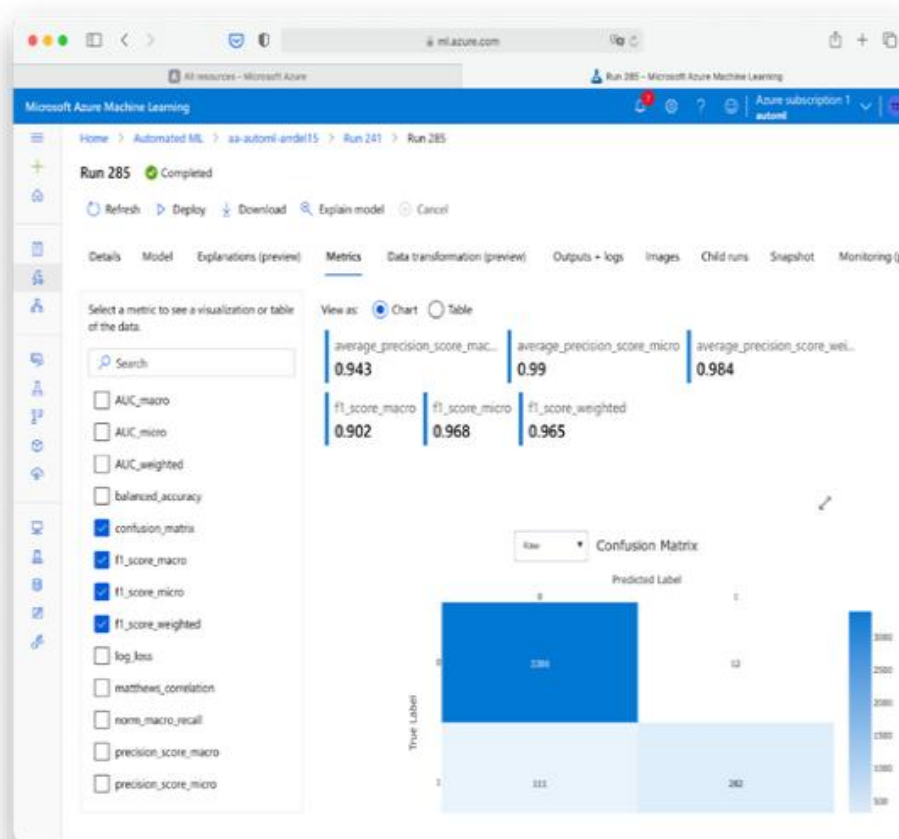


Yang terpenting, kita ingin mengetahui bagaimana kinerja model ini pada data kita. Untuk mengetahui lebih lanjut, klik tab "Metrik". Di bagian ini, aktifkan metrik evaluasi berikut dan nonaktifkan semua metrik lainnya untuk saat ini:

- average_precision_score_macro
- average_precision_score_micro
- average_precision_score_weighted confusion_matrix
- f1_score_macro f1_score_micro
- f1_score_weighted

Layar Anda akan terlihat seperti pada gambar di bawah.

Sebelum kita terlalu memikirkan arti metrik mikro, makro, dan rata-rata, mari kita lihat tabel kebingungan yang disediakan oleh evaluasi model. Tabel kebingungan ini berisi informasi yang sama persis dengan tabel Excel kita dalam pernyataan masalah, hanya tata letaknya yang sedikit berbeda. Tabel ini membandingkan label data sebenarnya dengan label yang diprediksi oleh model.

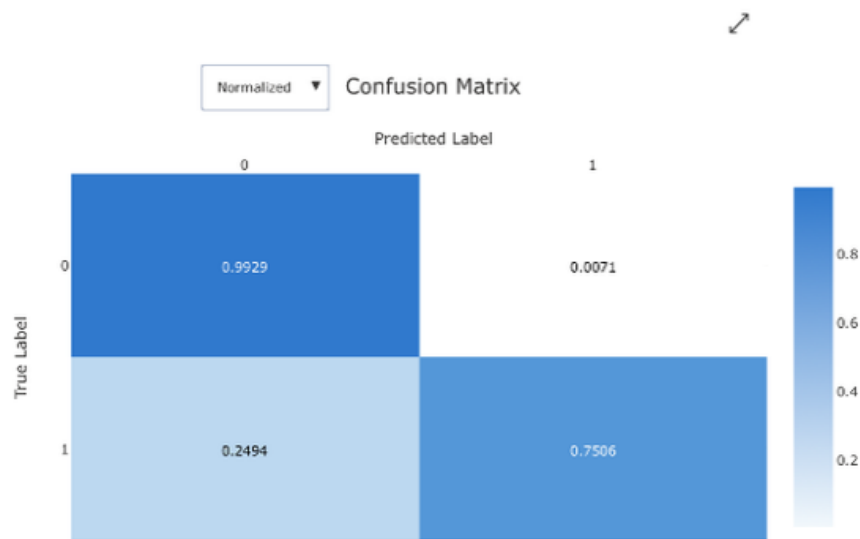


Catatan: Model dan hasil Anda mungkin terlihat sedikit berbeda. Cobalah format yang berbeda. Tampilkan tabel eksperimen.

Misalnya, terdapat 3374 observasi dengan label aktual "0" (tidak ada keterlambatan kedatangan > 15 menit) dan nilai yang diprediksi oleh model juga "0". Sebaliknya, 98 observasi diprediksi tepat waktu (Label Prediksi = "0"), tetapi ternyata tertunda lebih dari 15 menit (Label

Aktual = "1"). Perhatikan bahwa tabel ini tidak berisi semua observasi dari kumpulan data, tetapi hanya titik data sampel dari set validasi. Untuk membuatnya lebih sebanding dengan lembar kerja Excel yang telah kita lihat dalam rangkaian soal, alihkan menu tarik-turun dari "Mentah" ke "Dinormalkan" yang akan menunjukkan persentase masing-masing seperti yang terlihat pada gambar di bawah.

Mari kita bandingkan model ini dengan model dasar kita, prediksi ini semata-mata didasarkan pada fakta apakah pesawat mengalami penundaan lebih dari 15 menit setelah keberangkatan. Untuk meninjau, lihat Gambar di bawah dengan tabel kebingungan asli.



	A	B	C	D
1	Classifier Evaluation			
2				
3			COUNT	PERCENTAGE
4	BASELINE PREDICTION	0	34.169	90,
5	ACTUAL	0	32.991	96,
6	ACTUAL	1	1.178	3,4
7	BASELINE PREDICTION	1	3.737	9,5
8	ACTUAL	0	988	26,
9	ACTUAL	1	2.749	73,
10		Total	37.906	100
11				
12		ArrDelay15	True	10
13			False	90

Seperti yang Anda lihat, model kami menghasilkan peningkatan yang signifikan dibandingkan dengan nilai dasar untuk penerbangan yang diprediksi akan tertunda ("1") tetapi sebenarnya tepat waktu ("0"). Model AI hanya salah mengklasifikasikan 0,71% observasi di sini, sementara nilai dasar kami memiliki kesalahan sebesar 3,4% dalam kategori ini. Demikian pula, untuk penerbangan yang diprediksi akan tepat waktu ("0") tetapi sebenarnya mendarat

dengan penundaan lebih dari 15 menit ("1"), kami dapat sedikit mengurangi kesalahan dari nilai dasar dari 26,4% menjadi 24,94%.

Mari kita lihat apa artinya ini bagi kriteria penerimaan kami, yaitu presisi, perolehan kembali, dan Skor F1. Metrik untuk model dasar adalah sebagai berikut:

- Presisi: 0,74
- Recall: 0,70
- Skor F1: 0,72

Kita dapat menghitung metrik yang sama untuk model AI kita secara manual:

- Presisi = $TP / (TP + FP) = 295 / (295 + 24) = 0,92$
- Recall = $TP / (TP + FN) = 295 / (295 + 98) = 0,75$
- Skor F1: $2TP / (2TP + FP + FN) = 2 * 295 / (2 * 295 + 24 + 98) = 0,83$

Seperti yang Anda lihat, kita dapat meningkatkan prediksi dasar kita sebesar 11 poin persentase berkat pendekatan Pembelajaran Mesin Otomatis.

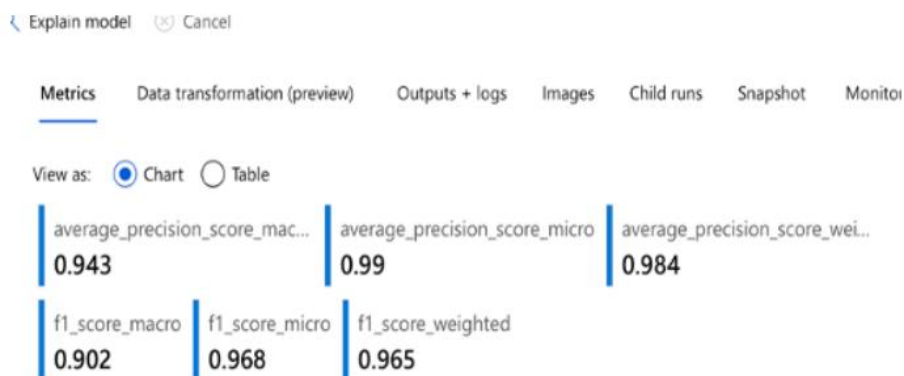
Mari kita kembali ke metrik Presisi dan Skor F1 di tab metrik model. Mengapa kita melihat nilai yang jauh lebih tinggi di sini? Itu karena kita diberikan skor mikro, makro, dan tertimbang untuk algoritma klasifikasi kita. Istilah-istilah ini memiliki arti sebagai berikut:

- Mikro: Hitung metrik secara global dengan menghitung total positif benar, negatif salah, dan positif salah.
- Makro: Hitung metrik untuk setiap label, dan temukan rata-rata tanpa bobotnya. Ini tidak memperhitungkan ketidakseimbangan label.
- Terbobot: Hitung metrik untuk setiap label, dan temukan rata-ratanya yang dibobot berdasarkan dukungan (jumlah kejadian benar untuk setiap label).

Untuk masalah klasifikasi yang tidak seimbang, akan lebih informatif untuk menggunakan rata-rata makro di mana kelas minoritas diberi bobot yang sama dengan kelas mayoritas.

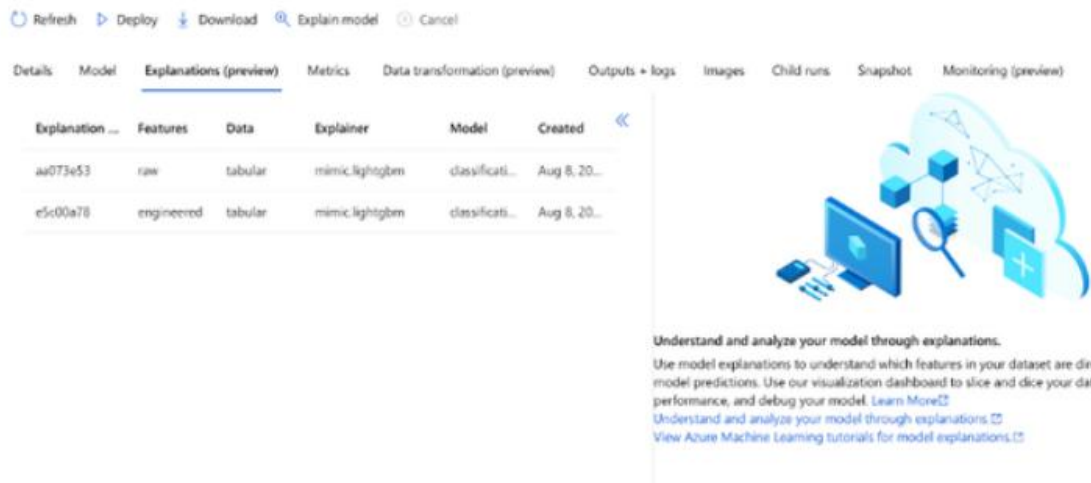
Dalam kasus masalah klasifikasi biner (dua kelas), Skor F1 Makro hanyalah rata-rata Skor F1 kelas positif dan kelas negatif:

Skor F1 Makro = $(\text{Skor F1 kelas positif} + \text{Skor F1 kelas negatif}) / 2 = (0,83 * 0,98) / 2 = 0,905$. Seperti yang Anda lihat, itulah yang ditunjukkan oleh metrik evaluasi untuk Skor F1 Makro di Azure pada Gambar di bawah:



Berhati-hatilah saat membandingkan metrik evaluasi agregat ini. Melihat hal-hal secara terperinci, misalnya tabel kebingungan, akan memberi Anda gambaran yang lebih baik tentang apa yang terjadi secara keseluruhan.

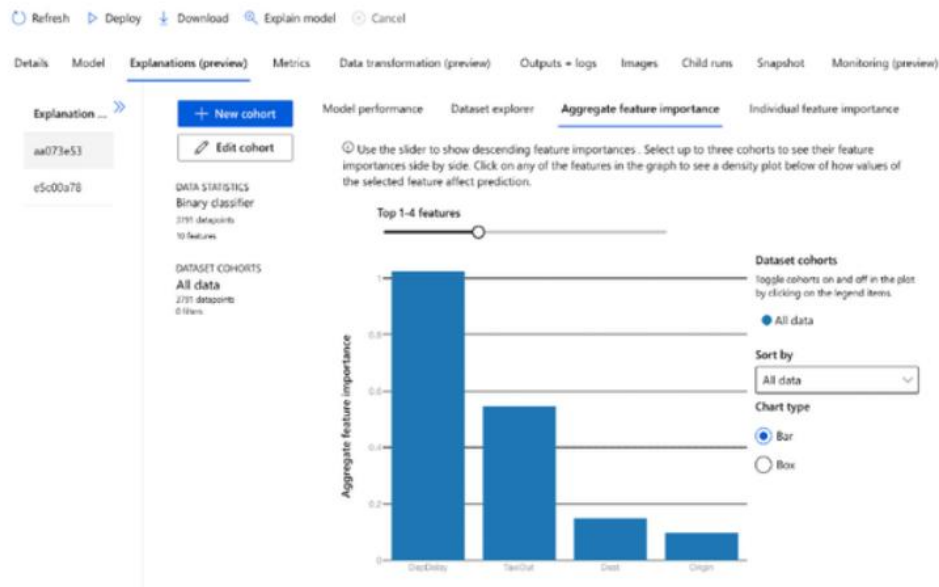
Setelah kita mendapatkan konfirmasi bahwa model kita sebenarnya berfungsi lebih baik sebagai prediksi dasar naif kita, mari kita pahami mengapa hal ini terjadi. Untuk tujuan ini, buka tab "Penjelasan" dari model terbaik kita. Pilih "Mentah" dari tabel pada Gambar di bawah dan perkecil panel daftar dengan mengklik panah ganda.



Selanjutnya, klik tab "Agregated feature importance". Ini akan menghasilkan tampilan seperti Gambar di bawah. Visual ini secara singkat menunjukkan variabel (fitur) terpenting secara keseluruhan yang memengaruhi hasil prediksi tertentu. Seperti yang dapat kita lihat dari grafik, atribut "Departure Delay" tidak mengherankan sejauh ini merupakan faktor pengaruh yang paling penting. Namun menariknya, ada tiga variabel lain yang memengaruhi prediksi dengan cukup signifikan. Yang pertama adalah atribut TaxiOut yang hampir setengahnya sama pentingnya dengan atribut Departure Delay. Hal ini seharusnya tidak mengejutkan karena TaxiOut sebenarnya terkait erat dengan penundaan keberangkatan. Namun yang mengejutkan, kita melihat bahwa variabel Destination dan Origin juga berperan dalam memprediksi penundaan kedatangan. Bergantung pada tempat pesawat lepas landas dan bandara tempat pesawat dijadwalkan tiba, probabilitas kedatangan tepat waktu meningkat atau menurun. Fitur-fitur ini tidak sepenting penundaan keberangkatan dan taxiout, tetapi keduanya berperan dalam membuat prediksi yang lebih baik.

Dengan lebih dari 400 kombinasi unik bandara keberangkatan dan kedatangan dalam kumpulan data kami, ini adalah area di mana Auto ML dapat menunjukkan kekuatannya, karena akan sangat merepotkan bagi kami untuk membuat aturan buatan tangan untuk kombinasi ini, hanya untuk meningkatkan kualitas keseluruhan prediksi kami hingga beberapa persen.

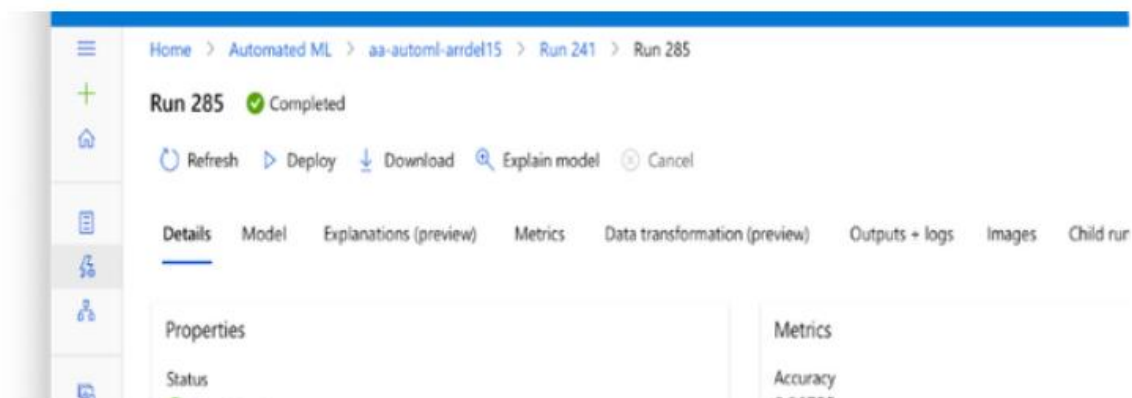




Setelah kita mendapatkan konfirmasi bahwa model kita bekerja lebih baik daripada baseline dan memiliki gambaran kasar mengapa hal ini terjadi, mari kita terapkan model kita untuk membuat prediksi pada set data baru yang belum pernah dilihat sebelumnya.

Panduan Penerapan Model dengan Microsoft Azure

Setelah kita melatih, memvalidasi, dan memahami model pembelajaran mesin kita, saatnya untuk menerapkannya agar dapat digunakan untuk membuat prediksi pada data baru. Di Azure Machine Learning Studio Anda, pilih model yang ingin Anda sediakan dan tekan "deploy". Panel "Deploy a model" akan muncul di sisi kanan seperti yang ditunjukkan pada Gambar di bawah. Beri nama untuk deployment model Anda. Saya sarankan menggunakan kombinasi tujuan model dan jenis model yang Anda gunakan. Dalam kasus kami, nama model yang baik adalah "arrdel-votingensemble".

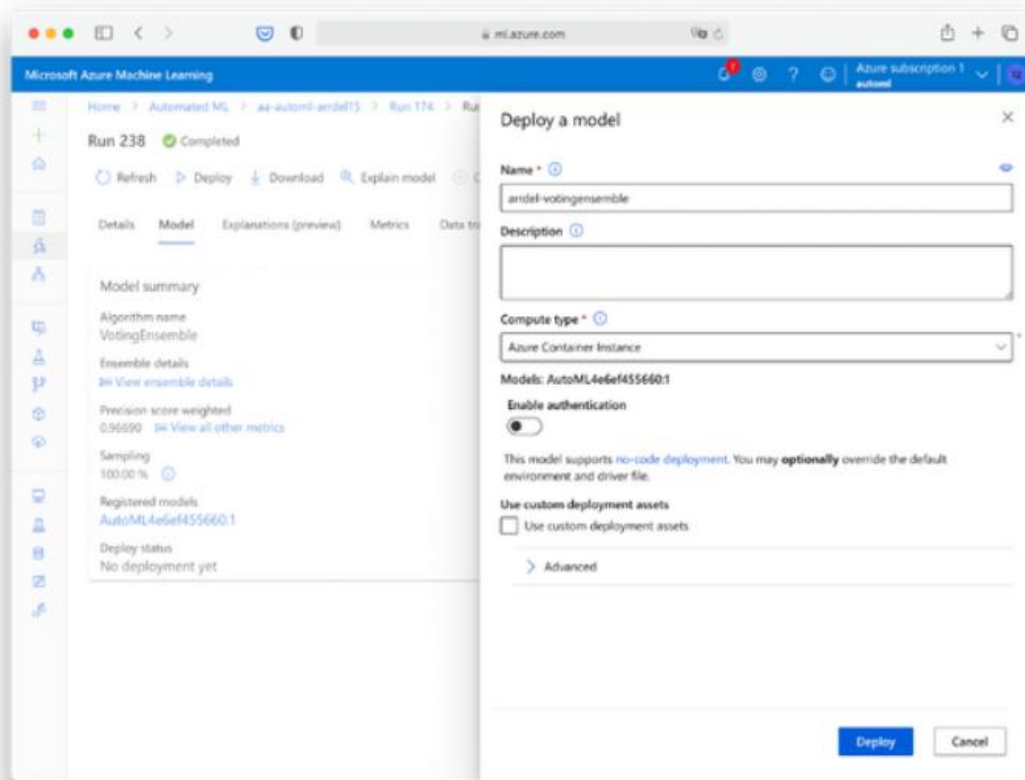


CATATAN

Terkadang model terbaik bukanlah model yang tepat untuk tujuan Anda. Terutama ketika metrik evaluasinya sangat berdekatan, Anda mungkin ingin memeriksa tiga model pertama

atau lebih dan mempelajari detailnya, misalnya tabel kebingungan seperti yang telah kita lakukan sebelumnya.

Tergantung preferensi Anda, masuk akal untuk memilih model terbaik kedua atau ketiga jika model tersebut lebih sesuai dengan permasalahan yang Anda hadapi. Mungkin Anda akan menemukan model yang menawarkan Recall yang lebih baik dengan mengorbankan presisi?



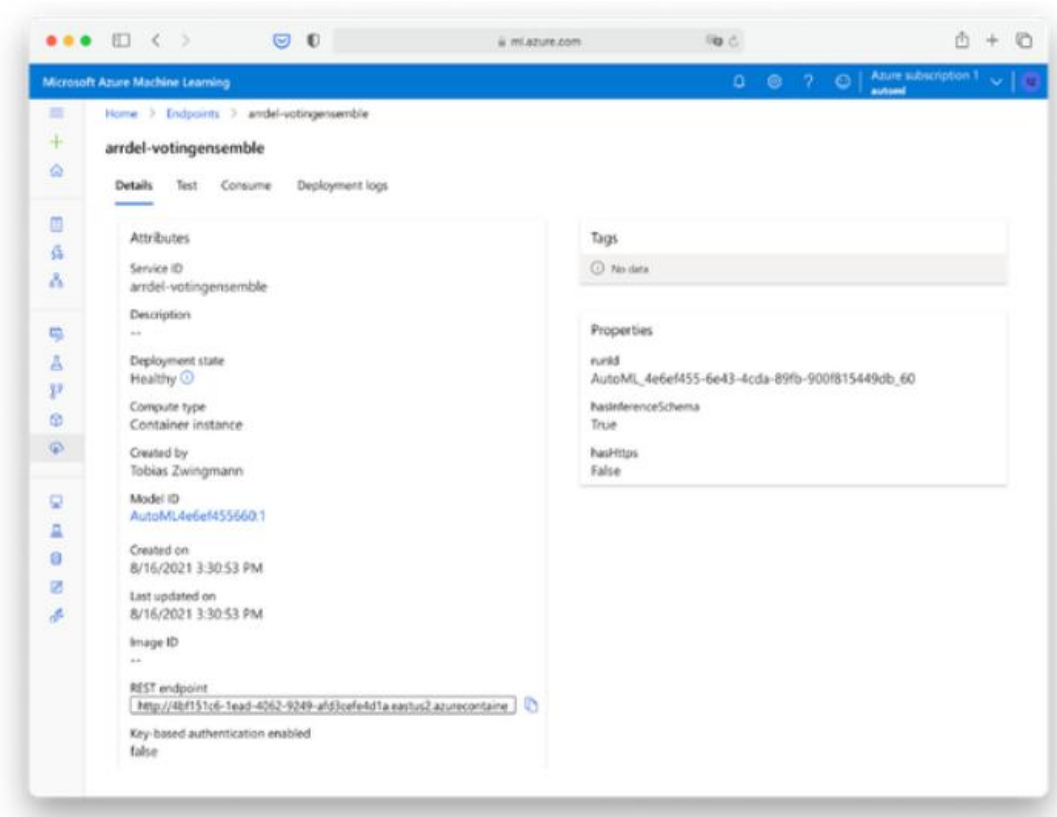
Selanjutnya, pilih jenis komputasi. Anda dapat memilih antara Azure Kubernetes Services dan Azure Container Instance. Kedua sumber daya ini biasanya digunakan untuk inferensi real-time. Layanan Kubernetes biasanya digunakan untuk deployment produksi skala tinggi yang membutuhkan waktu respons cepat dan penskalaan otomatis. Hal ini justru kebalikan dari persyaratan prototipe kami. Kami sudah cukup baik dengan Azure Container Service yang digunakan untuk beban kerja berbasis CPU skala rendah yang membutuhkan RAM kurang dari 48 GB. Pastikan autentikasi dinonaktifkan untuk saat ini dan tekan "Deploy".

Setelah itu, Anda akan melihat notifikasi yang memberi tahu Anda bahwa penerapan sedang berlangsung. Setelah beberapa menit, Anda akan melihat notifikasi lain yang menyatakan bahwa penerapan telah selesai. Klik "Titik Akhir" dan pilih model yang telah diterapkan di sini.

Periksa detail model yang telah diterapkan. Yang terpenting, pastikan status model menunjukkan "Sehat" seperti pada Gambar di bawah. Sekarang, model siap menerima permintaan inferensi (prediksi). Parameter terpenting yang Anda perlukan dari halaman ini

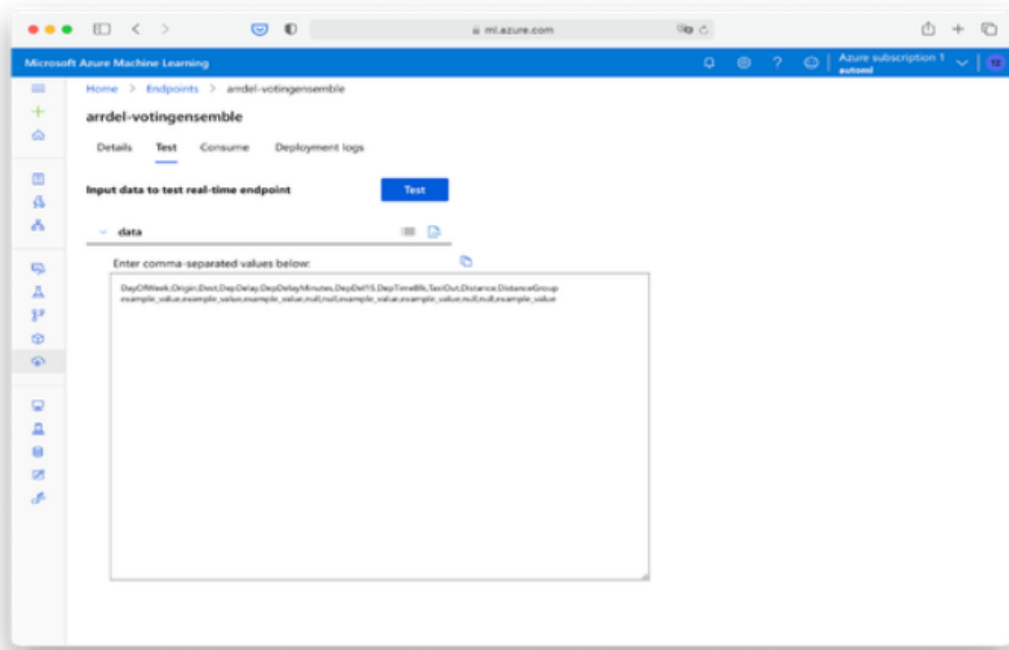


adalah titik akhir REST yang tercantum di bagian bawah layar. Kita akan menggunakan titik akhir REST ini nanti.

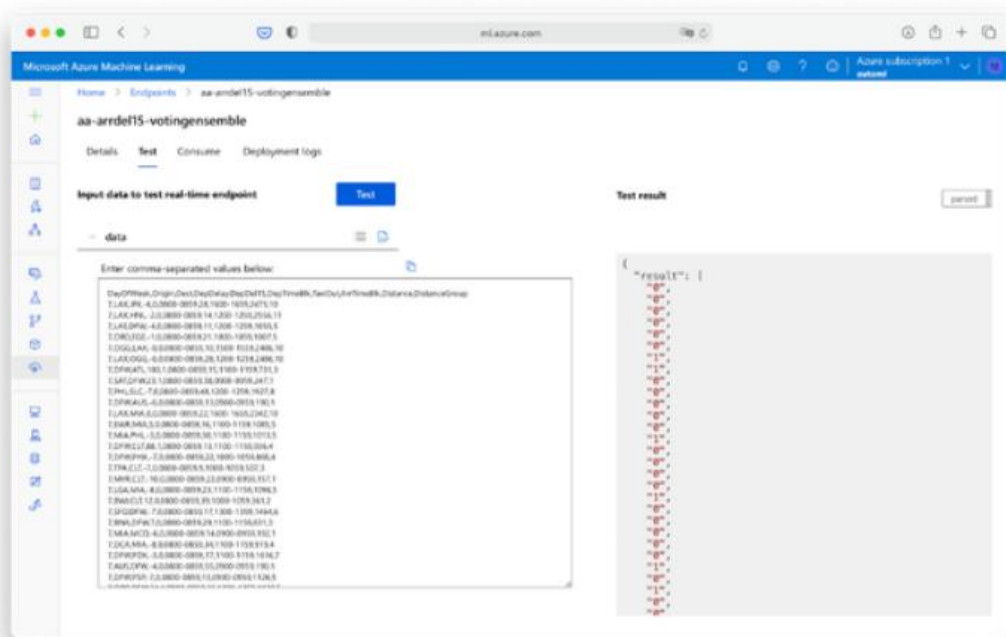


Sebelum kita melanjutkan mendapatkan prediksi dari BI kita, mari kita uji modelnya secara cepat menggunakan fitur pengujian bawaan di Azure ML Studio. Dari menu tab, pilih "Uji". Di sini, Anda dapat dengan cepat mendapatkan prediksi daring untuk beberapa titik data sampel. Anda dapat memasukkan nilai-nilai ini secara manual menggunakan formulir yang disediakan. Atau, Anda dapat mengaktifkan ikon CSV kecil untuk menempelkan data tabular seperti yang ditunjukkan pada Gambar di bawah.





Di sumber daya buku, temukan berkas "AA_Hourly_Batches_2021-02-07_0800-0859-inference-testing".



Ini adalah satu batch data penerbangan per jam yang ingin kita gunakan untuk membuat prediksi tentang perkiraan keterlambatan kedatangan. Buka berkas di editor teks, pilih semua, salin dan tempel isinya ke area teks pada halaman pengujian. Tekan tombol "Uji" dan dalam beberapa detik Anda akan melihat hasilnya di sebelah kanan seperti yang ditunjukkan pada



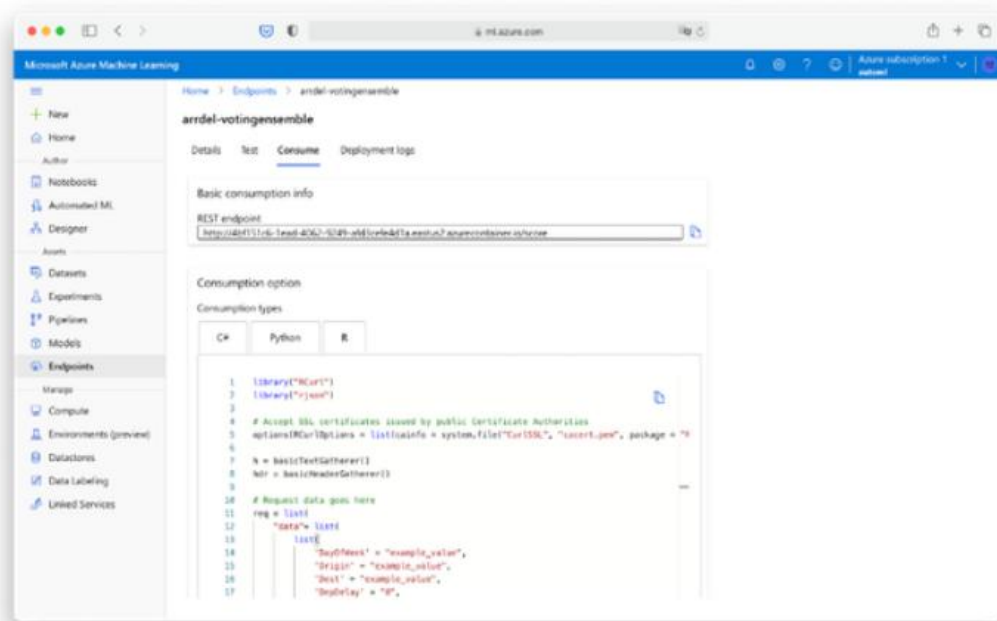
Gambar di bawah. Sekarang model kita sudah aktif dan berjalan, kita akan mulai mendapatkan kesimpulan dari BI kita menggunakan R atau Python.

CATATAN

Jelaskan perbedaan antara inferensi batch dan online, serta mengapa kami menggunakan inferensi online di sini untuk membuat semacam prediksi batch. Alasan utama: Portal Azure tidak mendukung inferensi batch, harus dibuat secara terprogram. Dan: Ukuran kumpulan data sangat kecil sehingga sebenarnya merupakan prediksi online.

Panduan Inferensi Model dengan R

Kunjungi Azure Machine Learning Studio. Pilih "Titik Akhir" dan pilih model yang ingin Anda gunakan untuk meminta inferensi. Pada layar berikutnya, pilih "Konsumsi" dan pilih R dari menu tab. Anda akan melihat dua hal penting di sini seperti yang ditunjukkan pada Gambar di bawah: URL titik akhir REST publik yang perlu Anda gunakan dan skrip R yang telah ditulis sebelumnya untuk mengirimkan permintaan.



Skrip yang telah ditulis sebelumnya sudah cukup baik jika Anda ingin memproses satu permintaan untuk satu observasi. Jika Anda ingin memproses seluruh tabel atau kerangka data R, Anda perlu menulis ulang skrip sedemikian rupa sehingga beberapa rekaman dikirim ke API, bukan hanya satu. Dalam sumber daya buku, Anda dapat menemukan berkas bernama "arrdel15-inference.Rrrdel15-inference.R".

Masukan skrip ini adalah kerangka data yang berisi kolom-kolom yang diperlukan untuk permintaan inferensi. Keluarannya adalah kerangka data yang dimodifikasi yang berisi semua informasi sebagai kolom asli satu ditambah satu yang disebut "ArrDel15_Prediction" yang berisi nilai prediksi aktual.



Perubahan utama di sini adalah fungsi `inference_request` yang mengambil daftar vektor (kolom kerangka data) dan mengirimkan satu permintaan API dalam format JSON. Keluarannya kemudian dikembalikan sebagai vektor. Ganti URL titik akhir dengan URL dari tab "Konsumsi" di portal Azure Anda. Selain itu, jika Anda membuat perubahan apa pun pada kumpulan data atau memilih kolom berbeda untuk prediksi, Anda perlu memperbaruinya juga dalam skrip, baik di dalam fungsi maupun untuk panggilan fungsi sebenarnya.

```
# API request function
inference_request <- function(DayOfWeek, Origin, Dest, DepDelay,
DepDel15, DepTimeBlk, TaxiOut, ArrTimeBlk, Distance, DistanceGroup)
{
  # Accept SSL certificates issued by public Certificate
  Authorities
  options(RCurlOptions = list(cainfo = system.file("CurlSSL",
"cacert.pem", package = "RCurl"), ssl.verifypeer = FALSE))

  h = basicTextGatherer()
  hdr = basicHeaderGatherer()

  # Bind columns to dataframe
  request_df <- data.frame(DayOfWeek, Origin, Dest, DepDelay,
DepDel15, DepTimeBlk, TaxiOut, ArrTimeBlk, Distance, DistanceGroup)

  # Dataframe to request list of lists
  req = list("data"=apply(df,1,as.list))

  body = enc2utf8(toJSON(req))
  api_key = "" # Replace this with the API key for the web service
  authz_hdr = paste('Bearer', api_key, sep=' ')

  h$reset()
  curlPerform(
    # Insert your custom endpoint URL here
    url = "http://xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx.azurecontainer.io/score",
    httpheader=c('Content-Type' = "application/json",
'Authorization' = authz_hdr),
    postfields=body,
    writefunction = h$update,
    headerfunction = hdr$update,
    verbose = FALSE
  )

  headers = hdr$value()
  httpStatus = headers["status"]
  if (httpStatus >= 400)
  {
    return(paste("The request failed with status code:",
httpStatus, sep=" "))
  }

  result = h$value()
```



```
result = fromJSON(fromJSON(result))$result
result = strtoi(result)
return(result)
}

# Submit request
result <- inference_request(df$DayOfWeek, df$Origin, df$Dest,
df$DepDelay, df$DepDel15, df$DepTimeBlk, df$TaxiOut, df$ArrTimeBlk,
df$Distance, df$DistanceGroup)

# Add column to dataset
df$ArrDel15_Prediction <- result

df
```

Panduan Inferensi Model dengan Python

- Menjelaskan inferensi model (Pemanggilan API) dengan Python

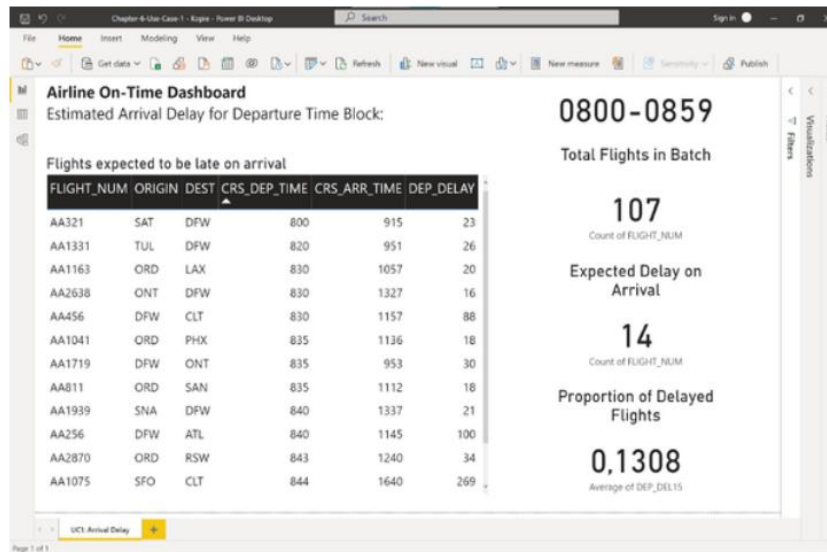
Panduan Inferensi Model dengan Power BI

Setelah kita mengetahui cara mendapatkan prediksi dari model kita secara terprogram, saatnya untuk menerapkan pengetahuan kita ke dalam Power BI. Perlu diketahui bahwa meskipun kita menggunakan Power BI di sini, Anda sebenarnya dapat menggunakan alat BI apa pun yang dapat memproses kode R atau Python.

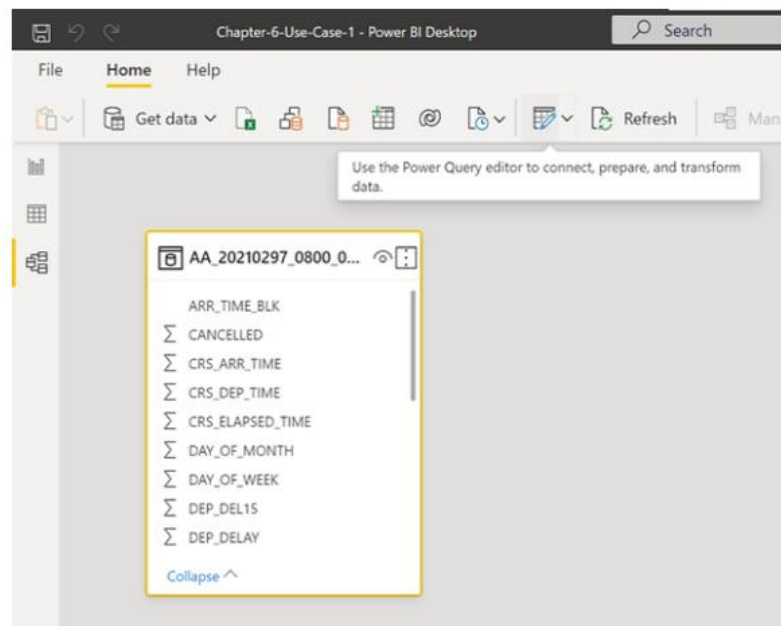
Tujuan kita adalah untuk meningkatkan prediksi keterlambatan kedatangan untuk dasbor yang telah kita temukan dalam pernyataan masalah (lihat Gambar di bawah). Singkatnya: Dasbor membaca data penerbangan per jam dan menunjukkan perkiraan proporsi penerbangan yang tertunda pada saat kedatangan. Kita sekarang akan mengambil data ini, mengirimkannya ke model yang dihosting, mendapatkan prediksi, dan menampilkan skor agregat untuk penggunaan lebih lanjut dan pengambilan keputusan yang lebih baik.

Untuk mengikuti langkah-langkah di Power BI, buka file "Chapter-6-Use-Case-1.pbix". Pastikan data dimuat dengan benar dan Anda dapat melihat dasbor seperti yang ditunjukkan pada Gambar di bawah.



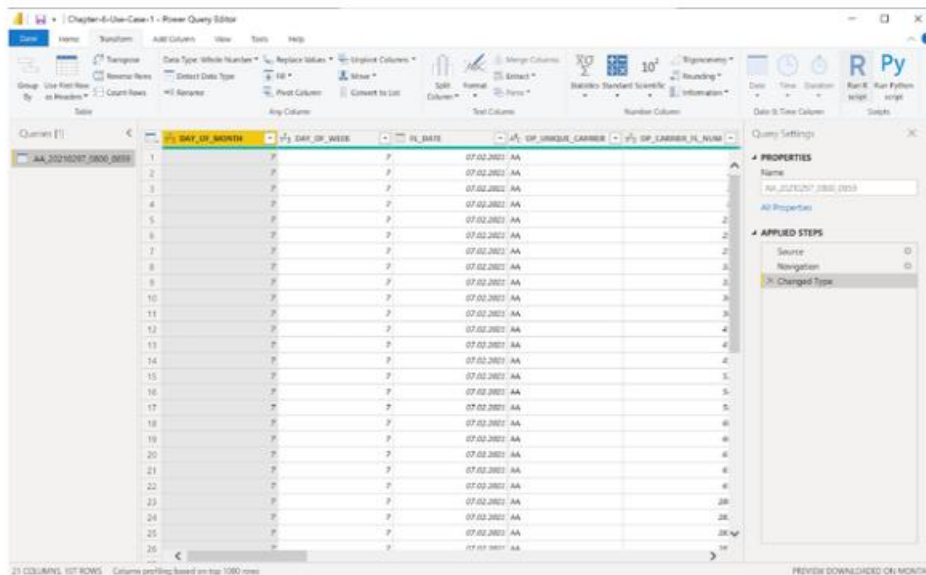


Untuk menambahkan prediksi AI, kita perlu menerapkan beberapa transformasi data. Buka "Model" di sebelah kiri dan pilih Power Query dari bilah alat seperti yang ditunjukkan pada Gambar di bawah:

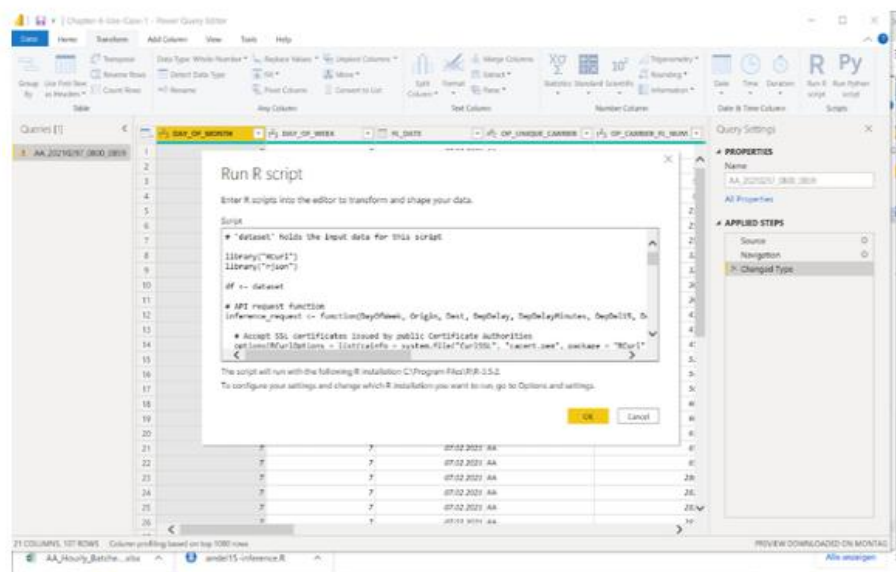


Ini akan membuka Editor Power Query. Alat ini memungkinkan Anda memanipulasi data, menerapkan kalkulasi, transformasi, dan menjalankan skrip Python dan R!

Dari bilah alat, pilih Transform, lalu Jalankan skrip R atau Jalankan skrip Python, tergantung pilihan Anda (lihat Gambar di bawah). Untuk tutorial ini, saya akan melanjutkan dengan contoh R.



Di editor kode, cukup tempel kode dari berkas `arrdel-inference.R` seperti yang ditunjukkan pada Gambar di bawah. Periksa kembali apakah Anda telah mengganti URL titik akhir dengan URL dari model Anda seperti yang ditunjukkan sebelumnya saat kita membahas skrip ini.



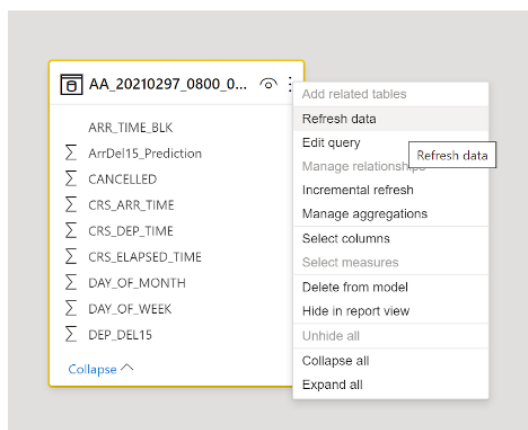
Klik OK dan Power BI akan mengevaluasi skrip. Proses ini akan memakan waktu beberapa saat, tergantung pada jumlah baris yang Anda kirim ke titik akhir API. Perlu diingat bahwa kita masih dalam tahap pembuatan prototipe. Dalam pengaturan produksi, Anda lebih suka menerapkan prediksi AI ini di tingkat basis data dan langsung menggunakannya dari Power BI atau sistem pelaporan Anda. Lihat bab 10 untuk detail selengkapnya. Setelah selesai, Anda akan melihat dua hal di editor: Pertama, ada langkah lain yang ditambahkan di panel "Applied Steps" di sebelah kanan.

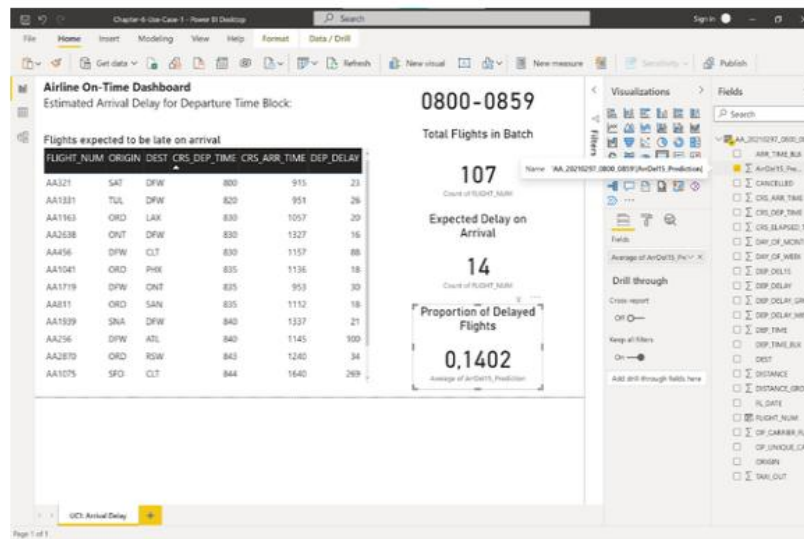
Dan kedua, Anda akan melihat kolom baru "ArrDel15_Prediction" jika Anda menggulir ke kanan seperti yang terlihat pada Gambar di bawah.

	DISTANCE_GROUP	ArrDel15_Prediction
2475	10	0
2556	11	0
1055	5	0
1007	5	0
2486	10	0
2486	10	0
731	3	1
247	1	1
1927	8	0
190	1	0
2342	10	0
1085	5	0

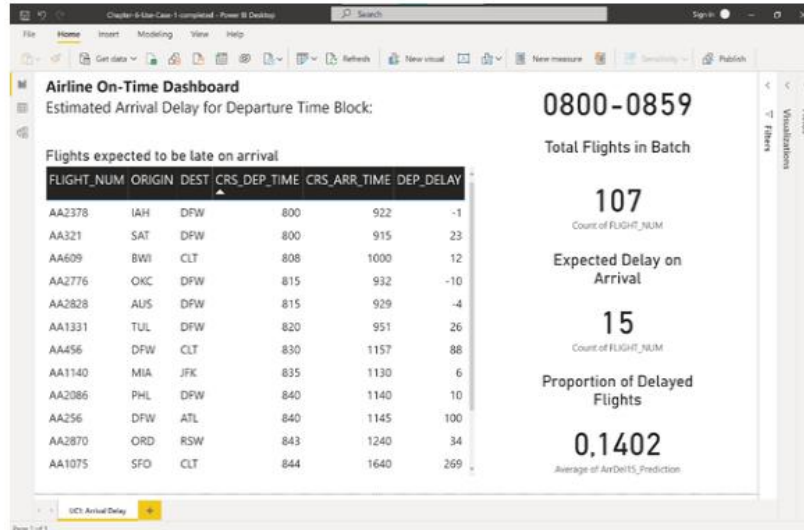
Di Editor Power Query, pilih File, lalu Tutup dan Terapkan. Model data Anda sedang diperbarui dan diisi dengan informasi tambahan baru. Anda dapat melihat kolom baru dalam model yang telah diperbarui. Setiap kali Anda ingin memperbarui data atau prediksi, Anda perlu menyegarkan model data seperti yang ditunjukkan pada Gambar di bawah. Ini akan memuat ulang data dari file atau sumber eksternal dan menjalankan kembali langkah-langkah di Editor Power Query.

Kita sekarang siap untuk mengisi dasbor kita dengan informasi baru. Kembali ke Laporan dan pilih visual Kartu dengan "Proporsi Penerbangan Tertunda". Seret dan letakkan dimensi baru "ArrDel15_Prediction" dari panel bidang ke kolom nilai visual kartu yang menggantikan metrik lama DEP_DEL15. Klik kanan pada kolom yang diperbarui dan pilih "Rata-rata". Panel pengaturan sekarang akan terlihat seperti pada Gambar di bawah dan metrik akan diperbarui menjadi 0,1402.





Silakan ganti kolom DEP_DELAY dengan dimensi ArrDel15_Predicted yang baru untuk visual lainnya di dasbor. Ini mencakup nilai filter untuk visual Kartu "Expected Delay on Arrival" serta filter pada tabel penerbangan. Atau, Anda dapat membuka berkas Chapter-6-Use-Case-1-completed.pbix dan melanjutkan dari sana. Dasbor akhir Anda akan terlihat seperti pada Gambar di bawah.



Dari jumlah penerbangan yang tertunda saja, kita mungkin secara intuitif berasumsi bahwa prediksi dari baseline kita dan model AI tidak jauh berbeda. Tapi jangan terjebak! Hanya karena AI memprediksi total satu penerbangan lagi yang akan tertunda, penerbangan aktual yang diperkirakan akan tertunda justru sangat berbeda. Lihatlah tabel penerbangan. Kita dapat melihat di dasbor yang didukung AI, penerbangan-penerbangan yang ditandai yang tidak mengalami penundaan keberangkatan selama 15 menit. Contohnya termasuk penerbangan AA2378 dari IAH ke DFW, penerbangan AA2776 dari OKC ke DFW, dan penerbangan AA2828 dari AUS ke DFW yang bahkan berangkat lebih awal.

Seperti yang kita ketahui dari dataset nyata seperti yang ditunjukkan pada Gambar XX, kita dapat mengonfirmasi bahwa penerbangan AA2378 dan penerbangan AA2828 memang tertunda lebih dari 15 menit. Penerbangan AA2776 tidak memiliki tanda ArrivalDelay15, tetapi pesawat mendarat terlambat 12 menit, meskipun memiliki keunggulan waktu 10 menit. Meskipun secara teknis prediksi AI di sini salah, itu jelas merupakan tebakan yang bagus. Anda dapat memeriksa hasil ini dan hasil lainnya dengan melihat tabel AA_Hourly_Batches_2021-02-07_0800-0859-ground_truth.xlsx dari sumber buku.

Saya harap Anda menikmati pembuatan dasbor ini dan menemukan bagaimana AI dapat membantu Anda membuat prediksi yang lebih baik untuk dataset Anda. Pantau data dasar dan lihat apakah dan seberapa baik AI dapat mengungguli data ini.

FL_DATE	OP_UNI	OP_CAR	OP_RIER_FL	ORIGIN	DEST	CRS_DEP_TIME	DEP_TIME	DEP_DELAY	DEP_DEL15	CRS_ARR_TIME	ARR_TIME	ARR_DELAY	ARR_DEL15
07.02.21	AA	2828	AUS	DFW	815	811	-4	0	929	954	25	1	
07.02.21	AA	2378	IAH	DFW	800	759	-1	0	922	1003	41	1	
07.02.21	AA	2776	OKC	DFW	815	805	-10	0	932	944	12	0	

CATATAN

Jika Anda mencermati tabel penerbangan, Anda akan menemukan sedikit kendala dalam pendekatan kami yang juga sangat mungkin terjadi di dunia nyata. Lihat misalnya baris terakhir, penerbangan dari SFO ke CLT. Penerbangan tersebut tertunda 269 menit. Inilah yang kami sebut sebagai training-serving-skew. Ketika ini menggunakan data dunia nyata, informasi ini sebenarnya baru tersedia hampir 4,5 jam kemudian. Jadi, tergantung pada titik waktu di mana kita melihat data tersebut, kita mungkin memiliki informasi yang tersedia atau tidak. Dalam pelatihan, kita memiliki semua informasi yang tersedia. Namun, kenyataannya tidak. Ada beberapa pendekatan yang dapat kita gunakan untuk mengatasi masalah ini, misalnya dengan memperkenalkan "penundaan minimum" yang menghitung penundaan "hingga saat ini" atau dengan tidak menyajikan prediksi untuk penerbangan ini. Atau, kita bisa menghilangkan variabel "Departure Delay" dari model AI kita dan tetap menggunakan kolom "Departure_Delay_15" sebagai proksi untuk penundaan penerbangan (di samping semua atribut lainnya) dan melihat kinerja model AI kita. Saat Anda memutuskan untuk memproduksi sebuah prototipe, pertanyaan-pertanyaan berikut perlu Anda ajukan: Seberapa sering kita mendapatkan data? Apakah frekuensinya cukup? Seberapa besar manfaat prediksi yang lebih baik bagi bisnis, dan apakah upaya peningkatan jalur data sepadan? Pertanyaan-pertanyaan inilah yang perlu Anda diskusikan dan fokuskan. Berkat prototipe Anda, Anda tidak perlu lagi membahas apakah AI dapat melampaui standar dasar saat ini. Anda sendiri yang membangun bukti konsepnya. Sekarang saatnya untuk fokus pada detail implementasi yang sebenarnya.



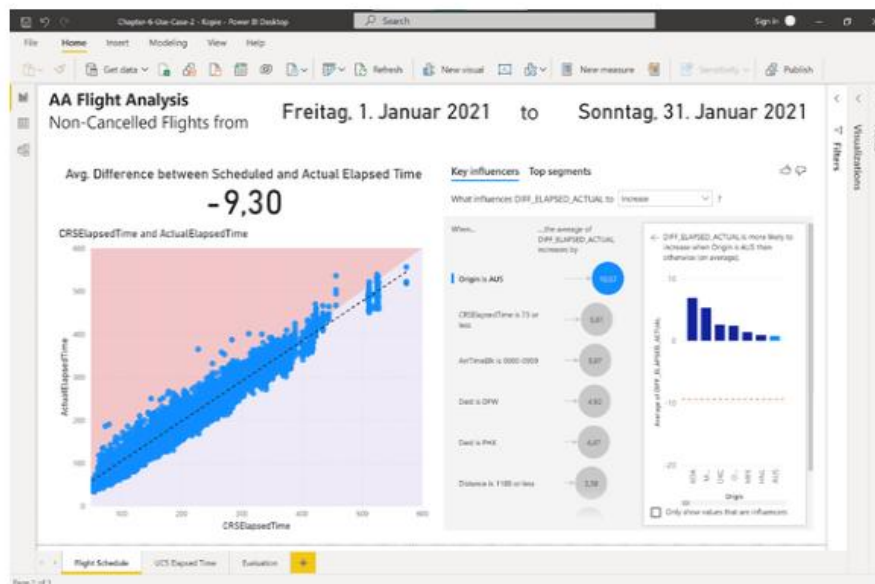
7.3 KASUS PENGGUNAAN: MENINGKATKAN PREDIKSI KPI

Untuk kasus penggunaan ini, kami akan melanjutkan skenario sebelumnya: Kami berada di tim BI American Airlines. Hanya saja kali ini kami tidak ingin meningkatkan pelaporan langsung, melainkan ingin meningkatkan perencanaan kami bahkan sebelum waktunya.

Untuk tujuan tersebut, kami perlu menangani metrik yang disebut "Waktu Berlalu" yang mengukur perbedaan antara waktu keberangkatan aktual dan waktu kedatangan aktual dari suatu penerbangan. Waktu berlalu ini direncanakan jauh sebelumnya dan disebut CRS Waktu Berlalu. Berbeda dengan kasus penggunaan sebelumnya, ini bukan variabel kategoris, melainkan variabel numerik kontinu. Selamat datang di dunia masalah regresi! Mari kita pelajari lebih lanjut tentang masalah ini dengan memperkenalkan pernyataan masalah sebagai berikut.

Pernyataan Masalah

Kami ingin mengidentifikasi hambatan dalam jadwal penerbangan untuk bulan berikutnya dan menandai penerbangan yang memiliki peluang tinggi untuk ditunda, yang berarti waktu berlalu aktual lebih tinggi daripada waktu berlalu yang direncanakan. Tim BI telah membangun dasbor yang menganalisis data historis dari masa lalu dan membandingkan Waktu Berlalu CRS (terjadwal) dengan Waktu Berlalu Aktual. Anda dapat melihat dasbor ini pada Gambar di bawah atau membukanya sendiri di Power BI dengan berkas "Chapter-6-Use-Case-2.pbix".



Dari dasbor, kami menemukan bahwa rata-rata penerbangan membutuhkan waktu 9,30 lebih sedikit dari yang direncanakan. Hal ini secara intuitif masuk akal karena kami tahu bahwa jadwal dioptimalkan untuk meminimalkan keterlambatan. Pada saat yang sama, kami dapat melihat dari diagram sebar di sebelah kiri bahwa terdapat penerbangan yang melebihi waktu tempuh yang direncanakan. Diagram sebar menunjukkan CRS Waktu Tempuh vs. Waktu Tempuh Aktual. Setiap titik adalah penerbangan. Titik-titik di area gelap atas pada diagram

membutuhkan waktu lebih lama dari yang dijadwalkan semula. Pertanyaan kuncinya adalah: Apakah kami dapat membuat model yang akan menandai penerbangan ini ketika jadwal penerbangan dibuat?

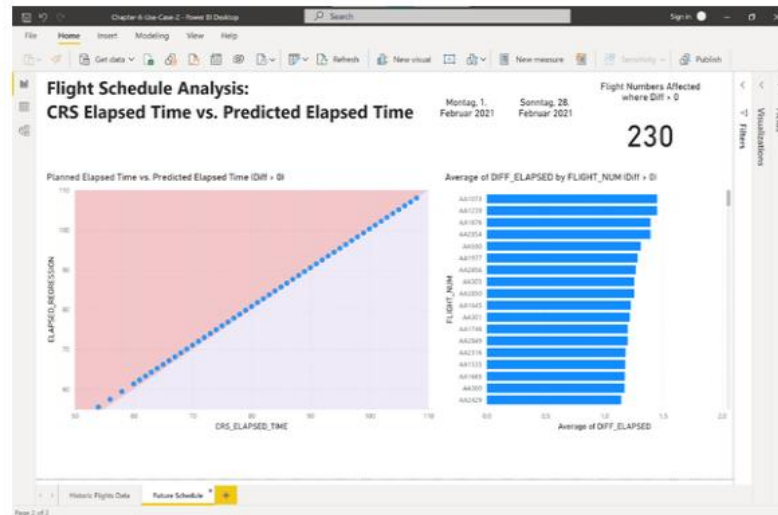
Berdasarkan data historis, para analis telah membangun model regresi yang sangat sederhana yang juga ditampilkan sebagai garis tren t pada dasbor di atas. Model tersebut mengambil CRS Waktu Tempuh yang dijadwalkan dan menghitung prediksi untuk waktu tempuh aktual berdasarkan data historis. Anda dapat melihat model regresi pada Gambar di bawah dan di lembar kerja "Model Regresi" pada berkas "AA_Flights_2021_1.xlsx". Model regresi memiliki nilai R2 sebesar 0,96 dan Standar Error (RMSE) sebesar 14,3. Artinya, [Jelaskan metrik error di sini]. Meskipun model ini tampak baik dalam makalah ini, ternyata model ini kurang efektif dalam memprediksi waktu aktual.

	A	B	C	D	E	F	G	H	I
SUMMARY OUTPUT									
<i>Regression Statistics</i>									
Multiple R		0,98047362							
R Square		0,96132851							
Adjusted R Square		0,96132749							
Standard Error		14,3004254							
Observations		37906							
<i>ANOVA</i>									
		<i>df</i>	<i>SS</i>	<i>MS</i>	<i>F</i>	<i>Significance F</i>			
Regression		1	192692108	192692108	942249,708	0			
Residual		37904	7751450,18	204,502168					
Total		37905	200443558						
		<i>Coefficients</i>	<i>Standard Error</i>	<i>t Stat</i>	<i>P-value</i>	<i>Lower 95%</i>	<i>Upper 95%</i>	<i>Lower 95,0%</i>	<i>Upper 95,0%</i>
Intercept		-4,4558099	0,19001978	-23,44919	9,781E-121	-4,8282538	-4,0833661	-4,8282538	-4,0833661
X Variable 1		0,97229165	0,00100164	970,695476	0	0,9703284	0,9742549	0,9703284	0,9742549
Squared Errors		7751450,18							
Root Mean Square Error (t		14,300							

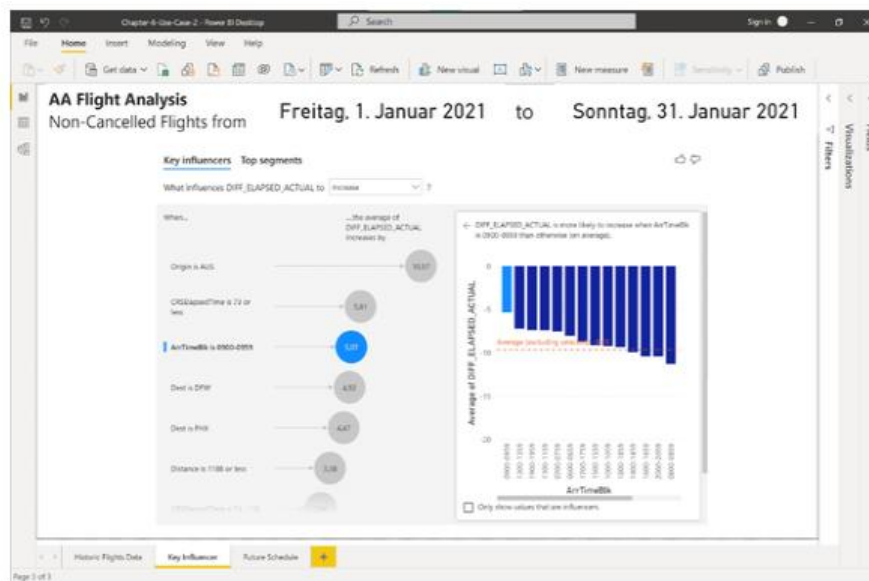
Jika kita menerapkan model ini pada jadwal penerbangan Februari 2021, model tersebut akan menandai 230 nomor penerbangan yang waktu tempuh aktualnya kemungkinan besar melebihi waktu tempuh yang dijadwalkan. Namun, dengan melihat diagram batang di sebelah kanan, kita dapat melihat bahwa perbedaan terbesar antara waktu tempuh yang dijadwalkan dan waktu tempuh aktual yang diprediksi hanya sangat tipis, yaitu 1,45 menit lebih lama dari yang direncanakan. Itu adalah perbedaan tertinggi yang dapat diprediksi oleh model regresi. Lihat diagram sebar di sebelah kiri.

Diagram ini menunjukkan semua penerbangan yang waktu tempuhnya diperkirakan melebihi waktu tempuh yang dijadwalkan. Anda dapat melihat bahwa semua penerbangan ini sebenarnya sangat dekat dengan ambang batas yang tepat sesuai dengan waktu tempuh yang dijadwalkan.





Model regresi yang telah dibangun tidak cukup fleksibel untuk menangkap variasi waktu berlalu yang disebabkan oleh variabel lain selain waktu CRS yang dijadwalkan.



Namun, apa sebenarnya variabel-variabel yang memengaruhi ini? Tim BI telah melakukan analisis lebih lanjut dan mengidentifikasi faktor-faktor pendorong utama yang menyebabkan peningkatan selisih antara waktu berlalu yang dijadwalkan dan waktu berlalu aktual. Anda dapat menemukan tangkapan layar analisis ini pada Gambar 4 atau di halaman laporan "Key Influencer" dalam berkas Power BI "Chapter-6-Use-Case-2.pbix".

Dari analisis kami menemukan bahwa berbagai faktor mendorong perbedaan antara waktu yang dijadwalkan dan waktu yang telah berlalu. Faktor-faktor tersebut antara lain bandara asal, jarak penerbangan, bandara tujuan, blok waktu kedatangan, dan hari kerja. Jadi, Anda mungkin tidak ingin menjadwalkan rapat segera setelah kedatangan jika Anda terbang ke Dallas yang diperkirakan akan mendarat antara pukul 09.00 - 09.59.

Namun, bagaimana kami sebagai Maskapai Penerbangan dapat mengubah wawasan ini menjadi informasi yang dapat ditindaklanjuti bagi pelanggan kami? Bagaimana kami dapat menandai hambatan kritis tersebut dalam jadwal penerbangan dan memunculkannya secara otomatis tanpa harus melakukan analisis faktor-faktor kunci yang memengaruhinya berulang kali?

Ikhtisar Solusi

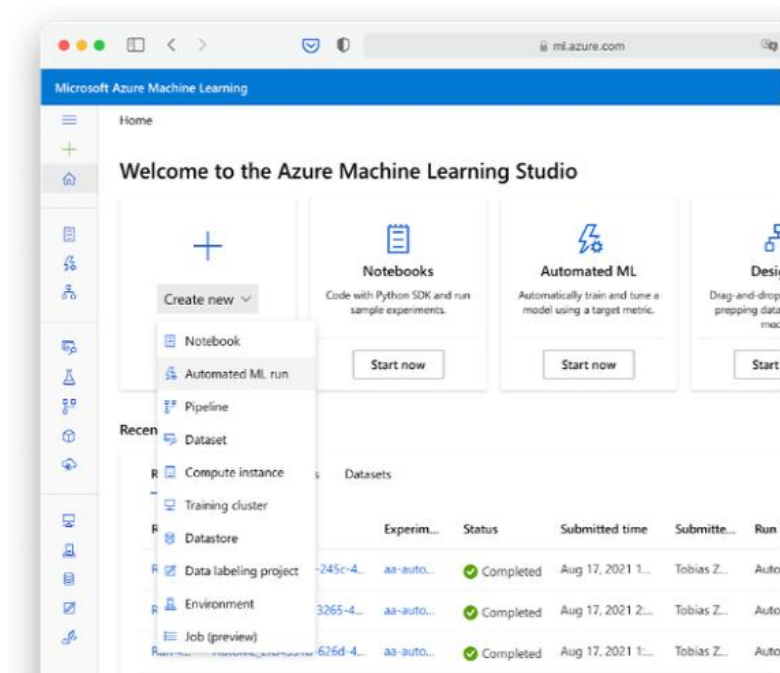
Untuk mengidentifikasi hambatan dalam jadwal penerbangan, kami akan mengambil semua informasi yang tersedia saat jadwal dibuat dan melatih model pembelajaran mesin berdasarkan data historis untuk memprediksi waktu yang telah berlalu sebenarnya menggunakan informasi ini. Informasi tersebut mencakup atribut seperti bandara asal, bandara tujuan, blok waktu kedatangan dan keberangkatan, dan lainnya. Semua atribut ini telah diketahui jauh sebelumnya sehingga kami tidak mengalami skew yang melayani pelatihan dan dapat bereaksi terhadap keluaran model jauh sebelumnya.

Kami kembali menggunakan Microsoft Azure Auto ML untuk membangun model pembelajaran mesin khusus dan menerapkannya di cloud Azure. Kemudian, kami akan mendapatkan inferensi dari model tersebut menggunakan skrip R atau Python yang kami panggil dari dalam alat BI kami. Dalam hal ini, kami juga menggunakan Power BI untuk membuat permintaan, tetapi umumnya hal ini juga dimungkinkan untuk alat BI lainnya.

Panduan Pelatihan Model dengan Microsoft Azure

Kunjungi Azure Machine Learning Studio dengan menavigasi ke ml.azure.com. Jika Anda melewati kasus penggunaan sebelumnya, silakan atur lingkungan Azure Anda sebagaimana disebutkan di awal bab ini.

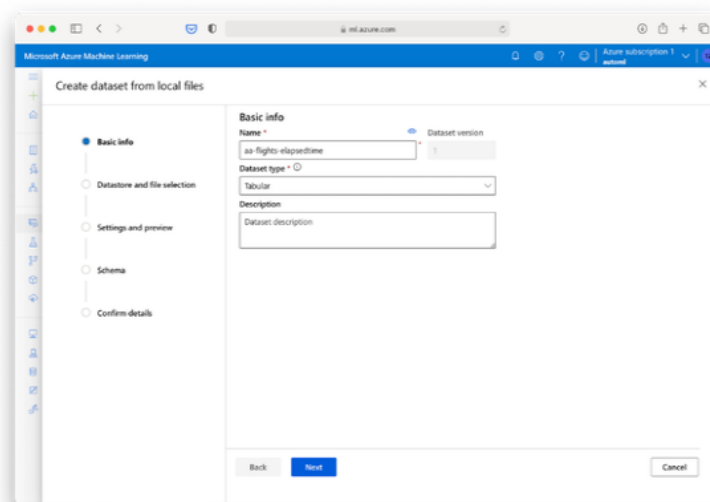
Dari layar beranda, pilih "Baru" -> "Jalankan Auto ML" seperti yang ditunjukkan pada Gambar di bawah



Pilih "Buat Dataset" -> Dari berkas lokal. Anda akan melihat formulir untuk memberikan informasi dasar tentang dataset Anda seperti yang ditunjukkan pada Gambar di bawah. Berikan nama pada dataset Anda yang sesuai dengan tugas pelatihan Auto ML. Dalam kasus saya, saya memilih "aa-flights-elapsedtime". Lanjutkan dengan "Berikutnya".

CATATAN

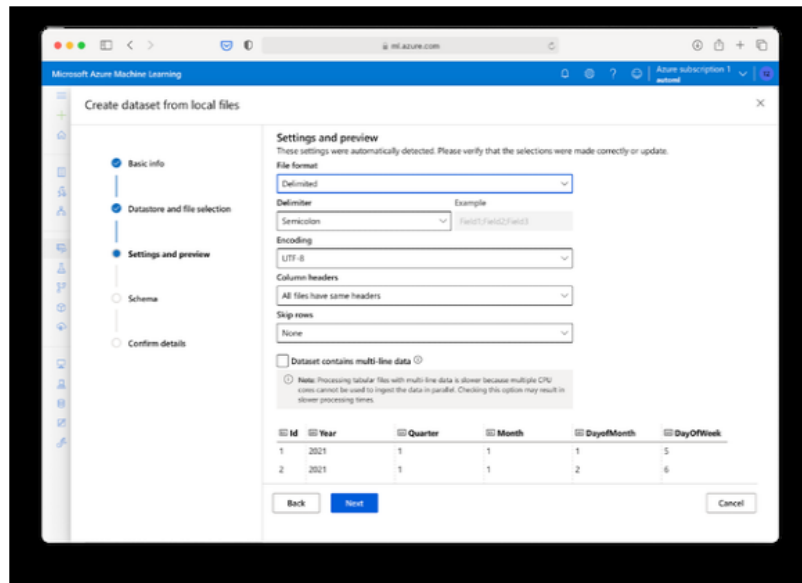
Meskipun secara teknis kita menggunakan sumber data yang sama seperti pada kasus penggunaan sebelumnya, sebaiknya unggah ulang berkas sebagai dataset terpisah di Microsoft Azure agar semuanya tetap teratur. Dataset pelatihan yang kita gunakan sekarang memiliki skema yang berbeda dari dataset pelatihan pada kasus penggunaan pertama. Misalnya, kali ini kita memprediksi atribut "Waktu Berlalu Aktual". Anda juga dapat memperbarui skema di Azure untuk suatu dataset dengan menambahkan versi baru, tetapi ini akan rumit untuk dilacak jika Anda ingin melatih ulang pengklasifikasi ArrDel15 dari kasus penggunaan sebelumnya. Aturan praktis yang baik: Simpan dataset terpisah untuk setiap tugas Auto ML, artinya kolom target harus tetap. Gunakan fungsi pembaruan skema dan kontrol versi untuk menambah atau menghapus fitur atau mengubah tipe data dari data input Anda, tetapi jangan sentuh kolom target.



Klik tombol "Unggah" dan pilih berkas "AA-Flights-2021-01-Training.csv". Setelah unggahan selesai, tombol "Berikutnya" akan muncul di bagian bawah layar.

Lanjutkan di sini untuk melanjutkan ke formulir "Pengaturan dan Pratinjau". Periksa kembali apakah format berkas diatur ke "Dibatasi", Pembatas ke "Titik Koma", dan pengodean ke "UTF-8". Anda akan melihat pratinjau data yang diformat dengan benar di bawah ini seperti yang ditunjukkan pada Gambar di bawah.

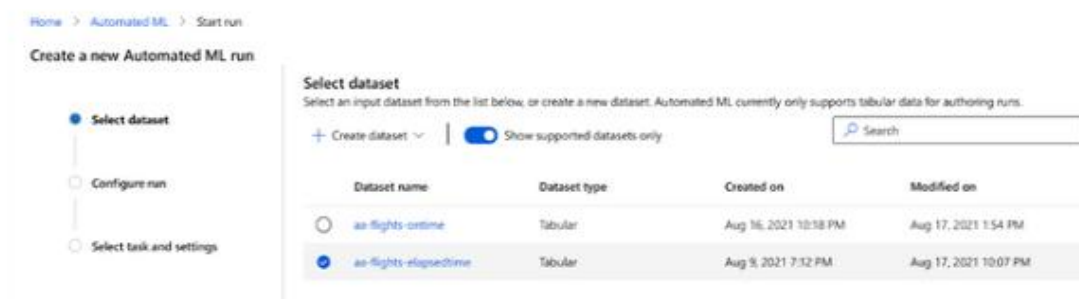




Klik Berikutnya untuk melanjutkan ke pengaturan skema. Aktifkan hanya atribut berikut dengan tipe data yang sesuai:

- DayOfWeek: String
- Asal: String
- Tujuan: String
- DepTimeBlk: String
- ArrTimeBlk: String
- CRSElapsedTime: Desimal (Koma)
- AktualElapsedTime: Desimal (Koma)
- Jarak: Desimal (Koma)
- DistanceGroup: String

Pilih Berikutnya, konfirmasi detailnya, dan Anda akan melihat dataset baru muncul dalam daftar dataset yang tersedia seperti yang ditunjukkan pada Gambar di bawah.



Pilih set data baru dan klik Berikutnya. Pada layar berikutnya, saya sarankan untuk membuat eksperimen baru agar model dan hasil dari tugas klasifikasi dan regresi tetap terpisah. Pilih nama eksperimen yang mudah dibedakan dari eksperimen sebelumnya, misalnya "aa-automl-

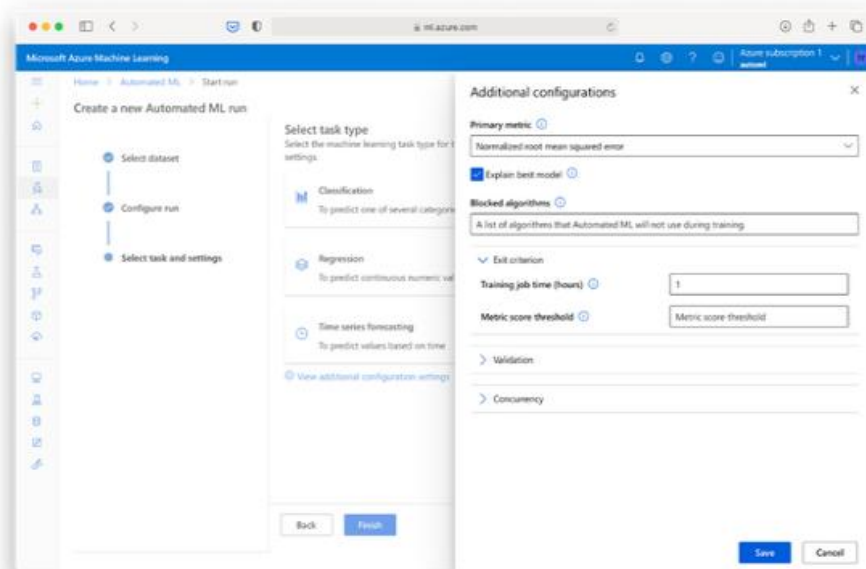


time". Sebagai kolom target, pilih "ActualElapsedTime (Desimal). Untuk sumber daya komputasi, Anda dapat menggunakan sumber daya yang sama yang telah kita buat pada kasus penggunaan sebelumnya "automl". Pengaturan Anda akan terlihat seperti pada Gambar di bawah. Lanjutkan dengan "Berikutnya".

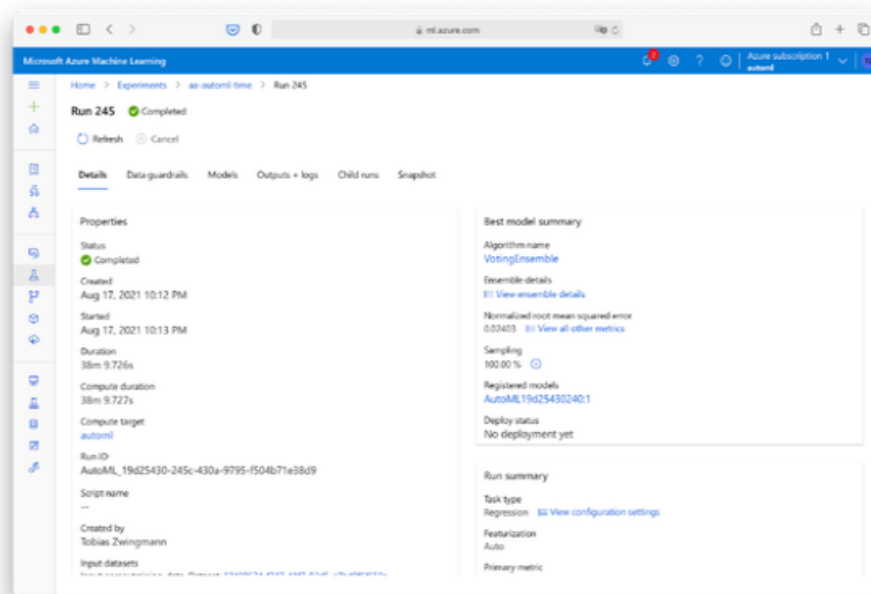
Pada layar berikutnya, Auto ML akan menebak bahwa Anda sedang mencoba memecahkan masalah regresi karena Anda menentukan variabel target numerik kontinu. Dalam kasus kita, ini sepenuhnya benar! Sebelum kita menyelesaikannya, buka pengaturan lanjutan dengan mengklik "Lihat pengaturan konfigurasi tambahan".

Pada konfigurasi tambahan, pastikan metrik Utama diatur ke Normalized root mean squared error. Dengan cara ini, kita dapat membandingkan model Auto ML dengan cukup baik dengan baseline yang ada dari model regresi yang sudah kita miliki. Periksa kembali apakah opsi "Explain best model" dicentang. Pada kriteria keluar, atur pekerjaan pelatihan ke maksimum 1 jam.

Sebenarnya, kita mungkin tidak membutuhkannya, tetapi algoritmanya akan konvergen setelah sekitar 30 menit. Konfirmasikan pengaturan dengan menyimpan dan kirimkan proses Auto ML dengan menekan "Finish".



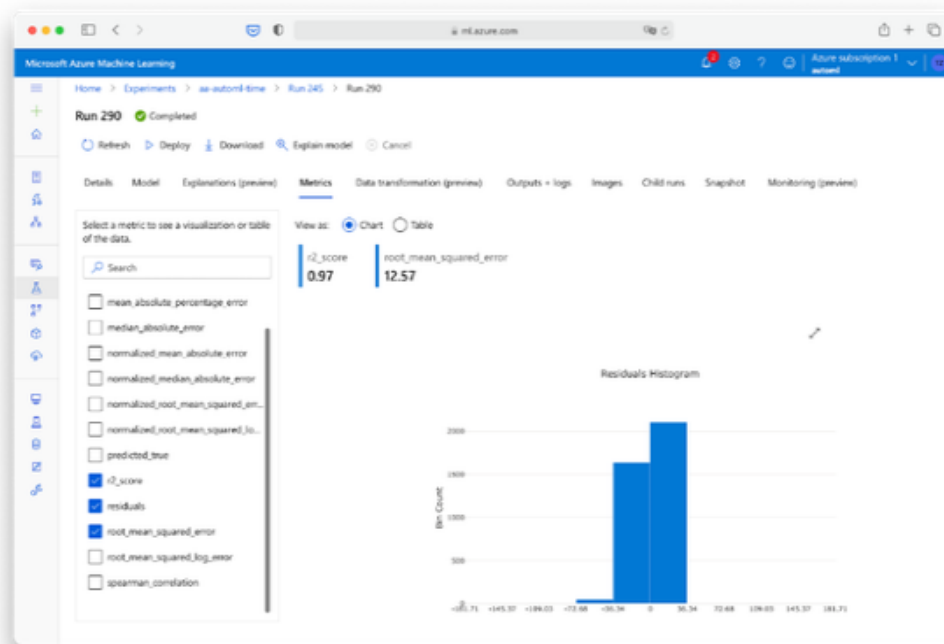
Beristirahatlah sejenak dan setelah beberapa saat, periksa status proses Anda dengan membuka Eksperimen -> aa-automl-time (atau nama apa pun yang Anda berikan) dan periksa status proses Anda saat ini. Jika Anda melihat kotak centang hijau di sini, klik Jalankan untuk melanjutkan ke detail Proses seperti yang ditunjukkan pada Gambar di bawah.



Seperti yang Anda lihat, dalam kasus saya, keseluruhan prosesnya memakan waktu sekitar 38 menit dan menghasilkan model yang menggunakan VotingEnsemble sebagai algoritmanya sekali lagi! Dan ini seharusnya tidak mengejutkan Anda. Faktanya, AutoML sering kali menghasilkan VotingEnsemble atau Stacked Ensemble sebagai model akhir. Keduanya merupakan teknik yang menggabungkan hasil dari sejumlah model berbeda yang "memilih" hasil akhir atau "ditumpuk" untuk menghasilkan keputusan akhir. Algoritma AutoML umumnya menghasilkan model ensemble.

Dalam kasus kami, model mencapai RMSE Ternormalisasi sebesar 0,2403. Bagaimana kita dapat membandingkannya dengan RMSE dasar sebesar 14,30 dari model regresi sederhana kami? Klik tautan "VotingEnsemble" dan pilih metrik. Hapus centang semua kotak kecuali untuk Skor R2, residual, dan Root Mean Squared Error seperti yang ditunjukkan pada Gambar di bawah.

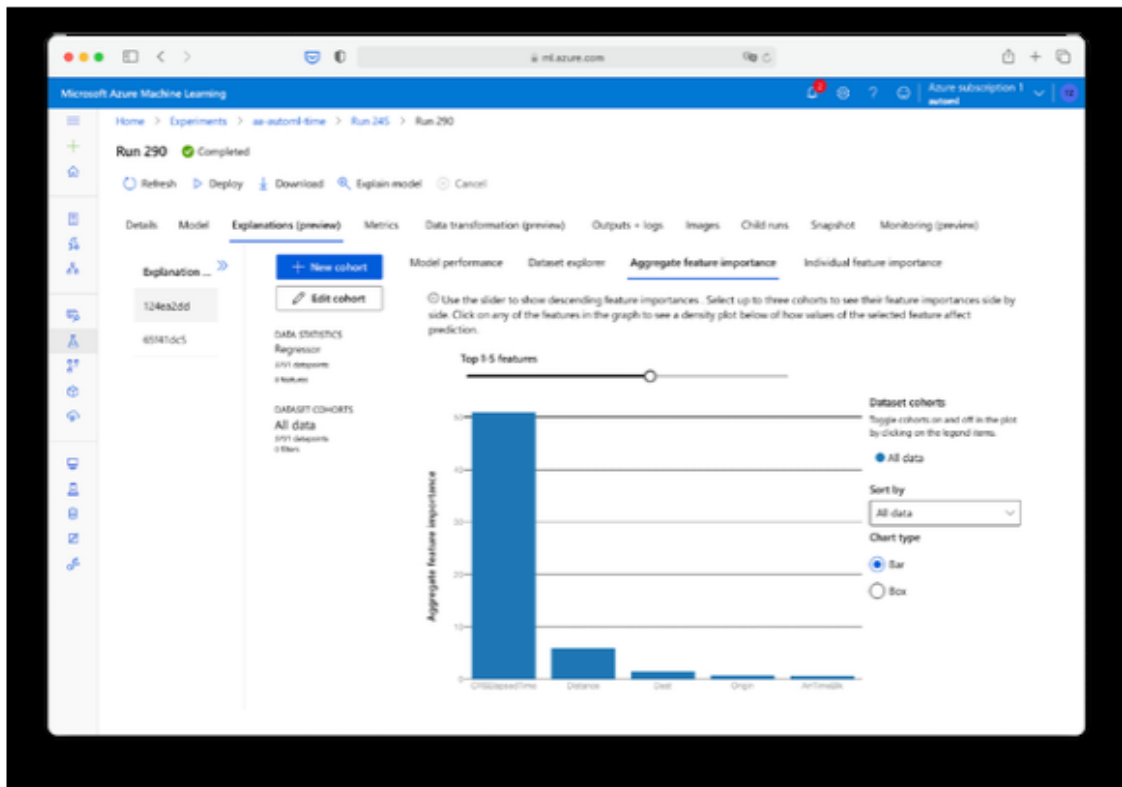




Dari metrik tersebut, Anda dapat melihat bahwa RMSE model regresi Auto ML kami adalah 12,57. Angka tersebut lebih rendah (dan karenanya lebih baik) daripada model regresi sederhana, tetapi sebenarnya tidak terlalu banyak mengingat besarnya daya komputasi yang kami gunakan untuk masalah ini. Namun, seperti yang telah kita pelajari sebelumnya dari kasus penggunaan pertama, kita sebaiknya bersikap skeptis terhadap metrik kualitas agregat dan lebih baik melihat hasilnya secara lebih detail.

Lihat histogram Residual. Residual adalah kesalahan yang dihasilkan model regresi, yaitu selisih antara hasil aktual dan hasil prediksi. Dalam kasus model regresi sederhana kami, regresi hampir selalu memprediksi nilai yang lebih rendah dari nilai sebenarnya. Dalam skenario Auto ML kami di sini, kami dapat melihat bahwa residual berpusat di sekitar nilai 0 dengan beberapa prediksi sedikit di atas metrik aktual dan beberapa prediksi di bawahnya. Varians jenis ini sebenarnya merupakan pertanda baik, karena variasi inilah yang kami butuhkan untuk mengidentifikasi penerbangan yang memiliki probabilitas tinggi melebihi waktu tempuh yang dijadwalkan. Jadi, dengan keyakinan yang kuat pada model Auto ML kita dan konfirmasi bahwa setidaknya di atas kertas model tersebut lebih baik daripada baseline regresi kita, buka tab Penjelasan untuk mengetahui faktor-faktor apa saja yang memengaruhi prediksi kita.

Di tab Penjelasan, pilih item daftar pertama dengan fitur mentah. Kemudian, pilih tab "Kepentingan fitur agregat" dari submenu. Pilih 5 fitur teratas dan Anda akan melihat layar seperti Gambar di bawah:



Dalam kasus saya, 5 fitur teratas adalah CRS Elapsed Time (Tidak mengherankan, sejauh ini) dan Distance, Dest, Origin, dan ArrivalTimeBlk. Pilihan ini masuk akal dari sudut pandang logis. Bandara yang lebih besar lebih ramai daripada yang lebih kecil, jadi seharusnya ada kemungkinan lebih tinggi terjadinya penundaan. ArrivalTimeBlk juga mengonfirmasi apa yang telah kita lihat dari alat-alat Key Influencer di atas. Ada waktu kedatangan tertentu, terutama di pagi hari, ketika bandara biasanya lebih ramai dan penundaan lebih mungkin terjadi.

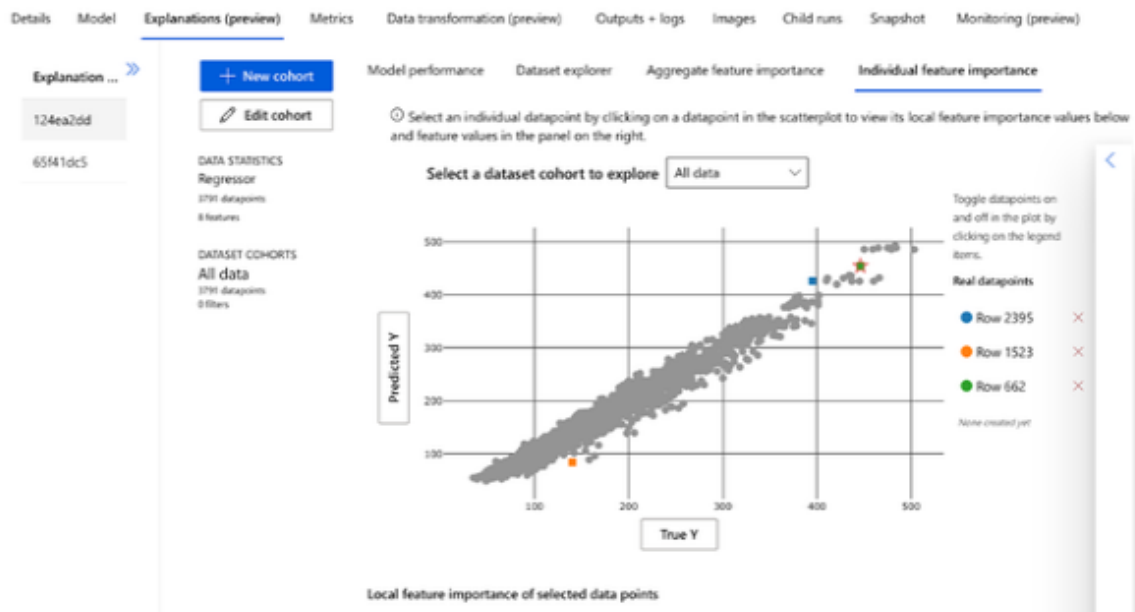
Perhatikan bahwa hasil aktual dalam kasus Anda mungkin terlihat sedikit berbeda karena ada beberapa elemen acak yang terlibat dalam proses Auto ML (misalnya cara data dibagi menjadi set pelatihan, pengujian, dan validasi). Sayangnya, kami tidak dapat menetapkan seed untuk prosedur acak ini di Azure sehingga kami memiliki hasil yang 100% dapat direproduksi. Namun, gambaran keseluruhannya seharusnya tetap sama.

Mari luangkan waktu sejenak untuk memahami model kita lebih baik dan melihat apa yang terjadi. Buka tab "Individual feature important". Pada diagram sebar berikut, klik judul sumbu Y dan pilih "Y yang Diprediksi". Lakukan hal yang sama untuk sumbu X dengan memilih "Y yang Benar". Ini akan memberi kita gambaran umum semua titik data (penerbangan) dan nilai benar serta nilai prediksinya untuk atribut Waktu Berlalu. Keunggulan visualisasi ini adalah kita dapat memilih satu titik data secara individual dan memeriksa tingkat kepentingan fitur konkret yang menghasilkan hasil prediksi individual. Pilih tiga titik data dari diagram ini:

- Satu titik dengan nilai prediksi lebih tinggi dari nilai aktual,
- satu titik dengan nilai prediksi hampir sama dengan nilai aktual, dan
- satu titik dengan nilai prediksi jauh lebih rendah dari nilai aktual.



Lihat Gambar di bawah untuk melihat tampilan pilihan saya:



Untuk memberikan konteks lebih lanjut, Anda juga dapat membuka panel menu di sebelah kanan untuk menampilkan detail lebih lanjut tentang titik data. Misalnya, titik data Baris 662 (Hijau) adalah penerbangan dari Bandara Honolulu ke Dallas Fort Worth yang berangkat pada Sabtu malam dan mendarat di pagi hari, seperti yang dapat Anda lihat pada Gambar di bawah. Prediksi untuk titik ini cukup akurat dengan prediksi Waktu Berlalu sebesar 454,2 menit dan waktu aktual sebesar 446 metrik.

Demikian pula, saya memilih titik di mana prediksi jauh lebih tinggi daripada aktual (426 menit vs. 395 menit untuk Baris 2395 (Biru) dan satu titik di mana prediksi meleset (84 menit vs. 140 menit) untuk Baris 1523 (Oranye).



dual feature importance

its local feature importance values below

> Data point info

What-If is currently not supported in studio. Run this widget in a jupyter notebook to enable What-If.

Data index
Row 662

True value: 446
Predicted value: 454,2

Feature values

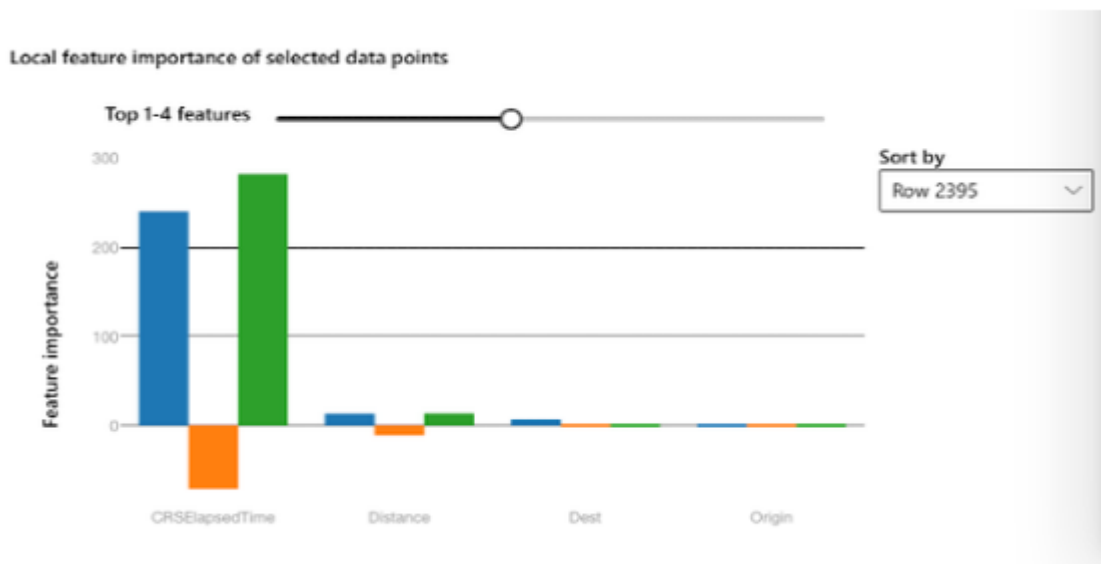
Search features

DayOfWeek
6

Origin
HNL

Dest
DFW

Sekarang, setelah tiga titik dipilih, gulir ke bawah dan Anda akan melihat grafik yang terlihat mirip dengan Gambar di bawah:



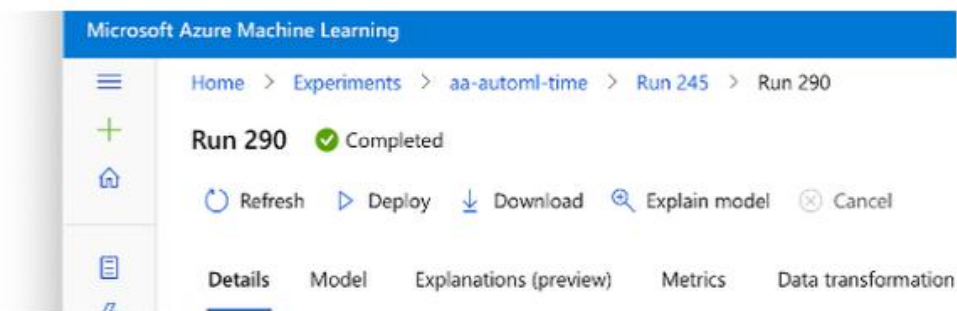
Grafik ini akan menunjukkan apa yang menyebabkan prediksi individual untuk titik data yang dipilih. Misalnya, kita dapat melihat bahwa dalam kasus Baris 1523 (batang oranye di tengah) titik di mana prediksi terlalu rendah fitur "CRSElapsedTime" justru dibebani oleh model. Entah mengapa, model memutuskan bahwa dalam kasus khusus ini, waktu terjadwal awal tidak terlalu dipertimbangkan.

Ternyata, dalam kasus ini, keputusan tersebut buruk. Sebaliknya, untuk Baris 662 (batang hijau di sebelah kanan) di mana prediksi hampir persis sama dengan hasil sebenarnya, waktu keberangkatan terjadwal memiliki faktor pengaruh yang tinggi. Model tidak mempertimbangkan faktor lain di sini sebagai faktor penting yang dapat menyebabkan penundaan.

Melihat prediksi tunggal dan memahami faktor mana yang memengaruhi prediksi akan membantu Anda merasa lebih nyaman dalam menerapkan dan juga menjelaskan model pembelajaran mesin Anda, alih-alih hanya menerimanya sebagai "kotak hitam". Ini juga merupakan alat yang baik untuk debugging. Dalam kasus kami, kami dapat melihat bahwa model tersebut cukup banyak berfungsi seperti yang diharapkan. Mari kita mulai dan terapkan model ini ke tahap produksi, agar kita bisa mendapatkan beberapa inferensi darinya.

Panduan Penerapan Model dengan Microsoft Azure

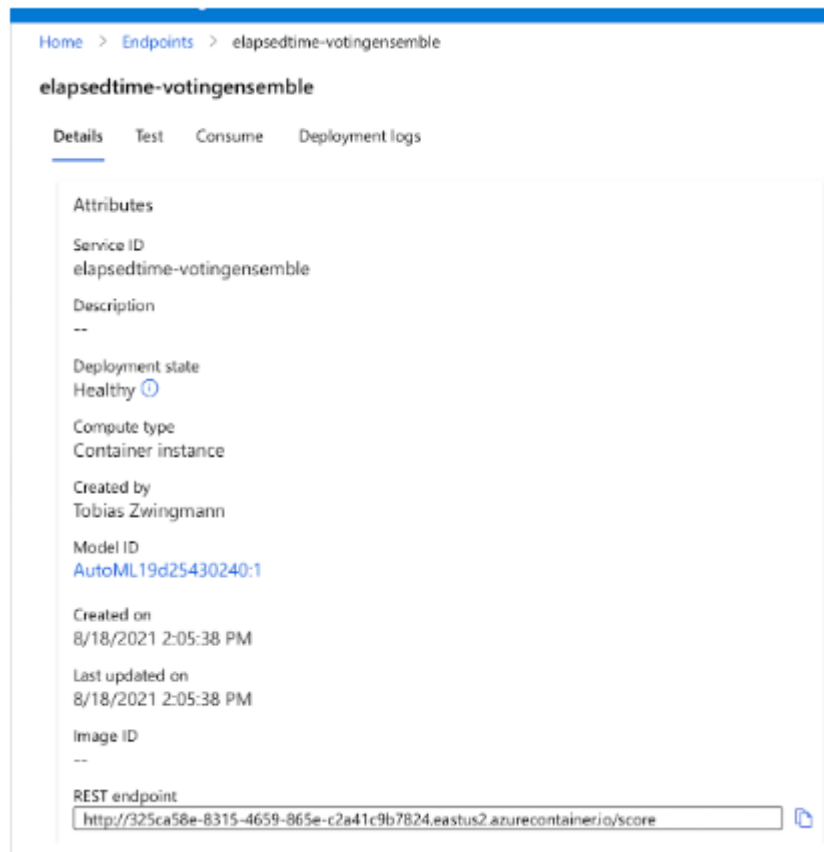
Langkah-langkah penerapan akan sama dengan kasus penggunaan sebelumnya, jadi saya akan menjelaskannya secara singkat di sini. Pilih model yang ingin Anda terapkan dan klik "Terapkan" dari bilah menu seperti yang ditunjukkan pada Gambar di bawah.



Di panel yang muncul, beri nama model Anda, misalnya "elapsedtime-votingensemble". Pilih Azure Container Instance sebagai Compute Type. Klik "Deploy".

Deploy selesai ketika model Anda tercantum di bawah "Endpoints" dan berstatus "Healthy" seperti yang terlihat pada Gambar di bawah. Pilih model untuk melihat URL titik akhir REST yang akan Anda perlukan di langkah selanjutnya.





Silakan uji model Anda secara daring menggunakan "Uji" dari menu navigasi atau pelajari cara membuat permintaan terhadap model Anda secara terprogram menggunakan tab "Konsumsi". Mari kita lanjutkan dan dapatkan beberapa prediksi dari titik akhir daring kita.

Inferensi Model dengan R

Untuk mengirim data ke titik akhir model yang dihosting dan mendapatkan prediksi kembali, kita akan menggunakan skrip yang sama seperti yang disediakan oleh Portal Azure, kali ini dengan beberapa modifikasi kecil. Anda dapat menemukan berkas skrip di buku resources elapsed-inference.R.

Modifikasi pertama adalah kita membungkus kembali permintaan di dalam fungsi yang menerima daftar vektor sehingga kita dapat meminta inferensi untuk lebih dari satu observasi secara bersamaan. Fungsi tersebut mengambil vektor-vektor ini dan membangun objek JSON yang dikirim ke API yang berisi semua observasi individual. Ini persis sama seperti pada kasus penggunaan sebelumnya.



```
# API request function:
# Expects a list of columns and gets predictions for each row
inference_request <- function(DayOfWeek, Origin, Dest, DepTimeBlk,
ArrTimeBlk, CRSElapsedTime, Distance, DistanceGroup) {

  # Accept SSL certificates issued by public Certificate
  Authorities
  options(RCurlOptions = list(cainfo = system.file("CurlSSL",
"cacert.pem", package = "RCurl"), ssl.verifypeer = FALSE))

  h = basicTextGatherer()
  hdr = basicHeaderGatherer()

  # Bind columns to dataframe
  request_df <- data.frame(DayOfWeek, Origin, Dest, DepTimeBlk,
ArrTimeBlk, CRSElapsedTime, Distance, DistanceGroup)

  # Dataframe to request list of lists
  req = list("data"=apply(request_df,1,as.list))

  body = enc2utf8(toJSON(req))
  api_key = "" # Replace this with the API key for the web service
  authz_hdr = paste('Bearer', api_key, sep=' ')

  h$reset()
  curlPerform(
    url = "http://325ca58e-8315-4659-865e-
c2a41c9b7824.eastus2.azurecontainer.io/score",
    httpheader=c('Content-Type' = "application/json",
'Authorization' = authz_hdr),
    postfields=body,
    writefunction = h$update,
    headerfunction = hdr$update,
    verbose = FALSE
  )

  headers = hdr$value()
  httpStatus = headers["status"]
  if (httpStatus >= 400)
  {
    return(paste("The request failed with status code:",
httpStatus, sep=" "))
  }

  # Get results
  result = h$value()
  result = fromJSON(fromJSON(result))$result

  return(result)
}
```



Modifikasi kedua membahas jumlah data yang akan dikirim ke API. Pada kasus penggunaan sebelumnya, kita hanya melihat kumpulan data berdurasi satu jam (sekitar 100 baris). Namun, dalam kasus ini, kita menangani data selama sebulan penuh totalnya lebih dari 32.000 baris! Meskipun ini mungkin sesuai dengan ukuran permintaan, kita telah mencapai batas inferensi daring waktu pemrosesan untuk permintaan ini hampir satu menit.

Selain itu, data ini mengandung banyak informasi redundan untuk API inferensi. Perlu diingat bahwa kita berlatih pada hari kerja. Beberapa penerbangan dalam jadwal dilakukan setiap hari, artinya setiap hari kerja akan berada dalam dataset empat kali atau lebih. Mengirim titik data dengan informasi yang sama sebanyak 4 kali ke API akan menjadi tidak efisien, baik secara komputasi maupun finansial. Kita tidak akan melakukan itu, dan berikut caranya menggunakan skrip R kita:

```
df <- dataset
# Create subset of dataframe that holds distinct flight information
for prediction
df_inference <- df %>% select(DAY_OF_WEEK, ORIGIN, DEST,
DEP_TIME_BLK, ARR_TIME_BLK, CRS_ELAPSED_TIME, DISTANCE,
DISTANCE_GROUP) %>%
  distinct()
```

Kerangka data `df` berisi dataset asli, artinya semua 32.000+ baris. Kita akan membuat subset dari dataset ini menggunakan kerangka data `df_inference`. Kerangka data `df_inference` hanya berisi kolom-kolom yang diperlukan untuk prediksi. Kita tidak menyertakan tanggal penerbangan di sini sehingga beberapa baris sekarang akan redundan. Menggunakan fungsi `distinct()` dari paket `dplyr`, kita menghapus semua baris duplikat.

Ini akan mengecilkan kerangka data kita dari 32.116 baris menjadi hanya 13.832 baris yang berisi informasi unik untuk inferensi. Ini merupakan pengurangan beban kerja data kita lebih dari 50%! Kita akan mengirimkan kerangka data `df_inference` ke API dan mendapatkan prediksi untuk itu. Namun, pada akhirnya, kita ingin melaporkan semua penerbangan, bukan hanya kombinasi unik yang kita kirimkan ke model kita. Dan begitulah cara kita akan menggabungkan hasil prediksi kembali ke kerangka data asli dengan fungsi berikut:

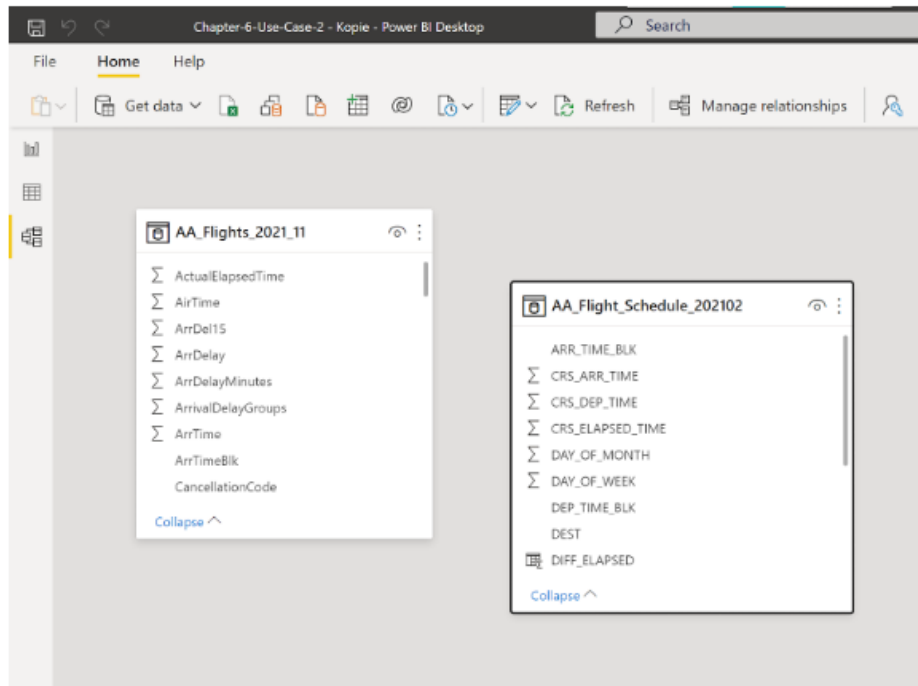
```
df <- df %>%
  left_join(df_inference, by = c("DAY_OF_WEEK", "ORIGIN", "DEST",
"DEP_TIME_BLK", "ARR_TIME_BLK", "CRS_ELAPSED_TIME", "DISTANCE",
"DISTANCE_GROUP"))
```

Hasilnya, kerangka data `df` berisi semua 32.116 observasi asli beserta hasil prediksinya masing-masing, tetapi kita hanya mengirimkan 13.832 baris darinya ke model kita. Indah, bukan? Saatnya menjalankan skrip kita dan melihat model kita beraksi di dalam perangkat BI kita!



Panduan Inferensi Model dengan Power BI

Buka file "Chapter-6-Use-Case-2.pbix" di Power BI desktop atau layanan Power BI. Pilih "model data" dari panel navigasi di sebelah kiri. Anda akan melihat dua sumber data di sini seperti yang terlihat pada Gambar di bawah.



Sumber data "AA_Flights_2021_1" berisi semua riwayat penerbangan dari Januari 2021. Ini adalah data yang kami gunakan untuk melatih model kami. Kami tidak perlu mengubah data ini. Sumber kedua "AA_Flight_Schedule_202102" berisi data jadwal penerbangan yang kami minati. Untuk model data ini, kami ingin menjalankan prediksi untuk atribut Waktu Berlalu. Kami akan melakukannya dengan menerapkan skrip R atau Python yang telah kami tulis sebelumnya untuk membuat kolom baru dalam model.

Pilih Editor Power Query dan pastikan Anda memilih tabel "AA_Flight_Schedule_202102" di sebelah kiri. Di panel menu, pilih "Transform", lalu pilih "Run Python Script" atau "Run R Script" sesuai preferensi Anda seperti yang ditunjukkan pada Gambar di bawah.

Salin dan tempel konten dari berkas skrip R atau Python, lalu tempelkan ke jendela kode di Power BI. Periksa kembali apakah URL yang diberikan dalam skrip tersebut merupakan URL titik akhir REST Anda yang valid. Pastikan juga Anda telah menginstal semua paket yang diperlukan (RCurl, rjson, dan dplyr) di lingkungan R yang digunakan oleh Power BI. Tekan "OK" dan Power BI akan menjalankan skrip Anda.



	DISTANCE_GROUP	CRS_ELAPSED_TIME	ELAPSED_REGRESSION	ELAPSED_TIME_PREDICTED
1	2475	38	382	374,4154091
2	2475	39	382	374,4154091
3	2475	39	382	374,4154091
4	2475	39	382	374,4154091
5	2475	39	382	374,4154091
6	2475	39	382	374,4154091
7	2475	39	382	374,4154091
8	2475	39	382	374,4154091
9	2475	39	382	374,4154091
10	2475	39	385	377,332284
11	2475	39	385	377,332284
12	2475	39	385	377,332284
13	2475	39	385	377,332284
14	2475	39	385	377,332284
15	2475	39	385	377,332284
16	2475	39	385	377,332284
17	2475	39	385	377,332284
18	2475	39	385	377,332284
19	2475	39	385	377,332284
20	2475	39	385	377,332284
21	2475	39	385	377,332284
22	2475	39	385	377,332284
23	2475	39	385	377,332284
24	2475	39	385	377,332284

Proses ini mungkin memerlukan waktu, tergantung pada mesin Anda. Perlu diingat bahwa kita melakukan kalkulasi di sini untuk lebih dari 32.000 baris data. Meskipun kita telah mengurangi jumlah data yang dikirim ke API, jumlahnya masih lebih dari 13.000 baris. Jika Anda ingin melihat kinerja komputer Anda, silakan buka Windows Task Manager, pilih Tab "Performance", dan lihat komputer Anda bekerja.

Setelah proses selesai, Anda akan melihat bahwa satu langkah telah ditambahkan di sebelah kanan di bawah "Applied Steps" dan dataset Anda sekarang berisi kolom tambahan bernama "ELAPSED_TIME_PREDICTED" seperti yang terlihat pada Gambar di bawah.

	CRS_ELAPSED_TIME	ELAPSED_REGRESSION	ELAPSED_TIME_PREDICTED
1	38	382	374,42
2	39	382	374,42
3	39	382	374,42
4	39	382	374,42
5	39	382	374,42
6	39	382	374,42
7	39	382	374,42
8	39	382	374,42
9	39	382	374,42
10	39	385	377,33
11	39	385	377,33
12	39	385	377,33

Pilih "File" -> "Tutup dan Terapkan" untuk keluar dari editor Power Query dan menerapkan transformasi Anda. Power BI akan memperbarui model data yang sesuai, yang mungkin memerlukan waktu beberapa menit. Waktu tunggu dalam kasus ini seharusnya tidak terlalu

mengganggu, karena proses ini hanya akan dilakukan sekali untuk setiap jadwal penerbangan bulanan.

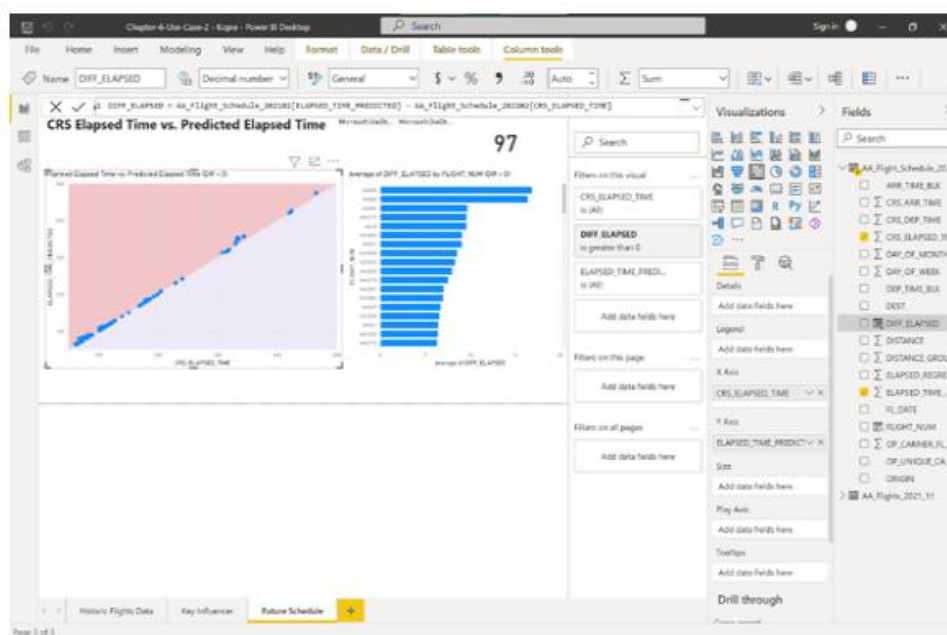
Setelah model data berhasil diperbarui, buka "Laporan" di Power BI. Kita ingin menerapkan prediksi baru kita. Pilih laporan "Jadwal Mendatang" di mana Anda dapat melihat prediksi regresi baseline. Saatnya untuk menyesuaikannya. Pilih diagram sebar di sebelah kiri. Ganti nilai untuk sumbu Y. Alih-alih "ELAPSED_REGRESSION", masukkan ukuran baru "ELAPSED_TIME_PREDICTED" di sini. Buka Filter untuk visual ini.

Anda dapat melihat bahwa data masih difilter untuk $DIFF_ELAPSED > 0$, di mana $DIFF_ELAPSED$ adalah selisih antara waktu yang diprediksi oleh model baseline (regresi) dan waktu berlalu yang dijadwalkan. Kami ingin kolom ini menunjukkan perbedaan antara model AI kami dan waktu tempuh yang dijadwalkan. Karena ini adalah kolom kalkulasi di Power BI, sangat mudah untuk mengubahnya. Pilih ukuran dan Anda akan melihat rumus muncul di bagian atas halaman laporan. Ganti $AA_Flight_Schedule_202102[ELAPSED_REGRESSION]$ dengan $AA_Flight_Schedule_202102[ELAPSED_TIME_PREDICTED]$ seperti yang ditunjukkan pada Gambar di bawah dan tekan Enter.



Dengan nilai yang diperbarui untuk pengukuran $DIFF_ELAPSED$, semua visual lain di halaman tersebut juga akan diperbarui sesuai dengan kriteria filter yang baru.

Gambar di bawah menunjukkan tampilan dasbor yang diperbarui:



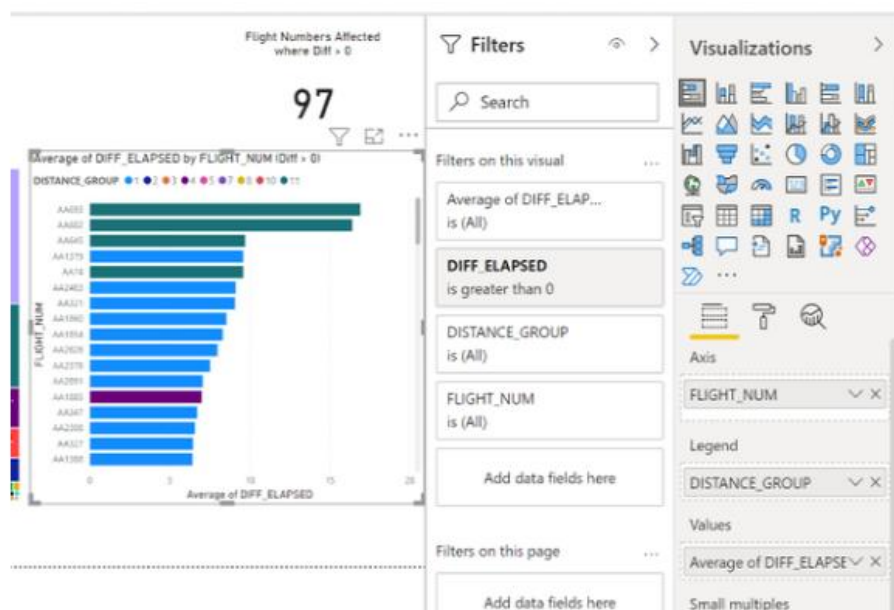
Apa yang telah kita capai sejauh ini?

Pertama, dengan menerapkan model AI, kita telah mengurangi jumlah penerbangan yang diprediksi membutuhkan waktu lebih lama dari yang diperkirakan menjadi 97 observasi.

Kedua, jika kita melihat diagram sebar di sebelah kiri, kita dapat melihat bahwa terdapat lebih banyak variasi dan titik-titiknya tidak membentuk satu garis lurus seperti pada model regresi sederhana.

Dan ketiga, jika kita melihat diagram batang di sebelah kanan, kita dapat melihat bahwa kita telah mengidentifikasi dua penerbangan yang diprediksi membutuhkan waktu 15 menit lebih lama dari yang direncanakan. Jika kita membandingkannya dengan model regresi sebelumnya, di mana perbedaan tertinggi hanya sekitar 1,5 menit, jumlah ini jauh lebih besar. Mari kita lihat apakah kita dapat sedikit memodifikasi laporan ini agar lebih interaktif dan dapat ditindaklanjuti.

Pertama, pilih diagram batang di sebelah kanan dan tambahkan kolom "DISTANCE_GROUP" ke legenda seperti yang ditunjukkan pada Gambar di bawah



Dari informasi tambahan ini, kita dapat melihat bahwa penerbangan dengan perbedaan tertinggi antara waktu tempuh yang diprediksi dan dijadwalkan sebenarnya adalah penerbangan jarak jauh (kelompok jarak 11). Namun, sebagian besar penerbangan pada grafik adalah penerbangan jarak pendek.

Selain itu, mari kita hilangkan diagram sebar. Diagram ini telah terbukti memberikan gambaran umum yang baik tentang cara kerja model, tetapi kurang interaktif. Pilih diagram sebar dan ganti dengan visual Peta Pohon. Sesuaikan pengaturan untuk peta pohon seperti yang terlihat pada Gambar di bawah:

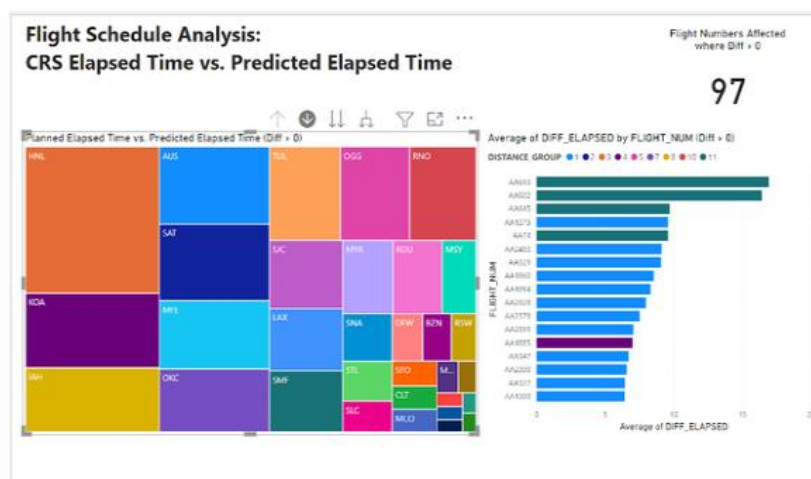
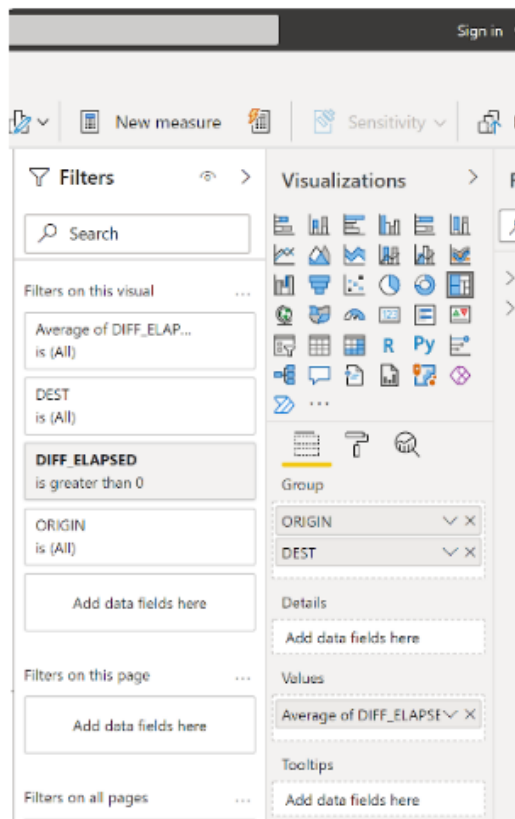
- Tambahkan kolom ORIGIN dan DEST ke Group.
- Masukkan kolom DIFF_ELAPSED ke dalam nilai. Agregat DIFF_ELAPSED berdasarkan Rata-rata.

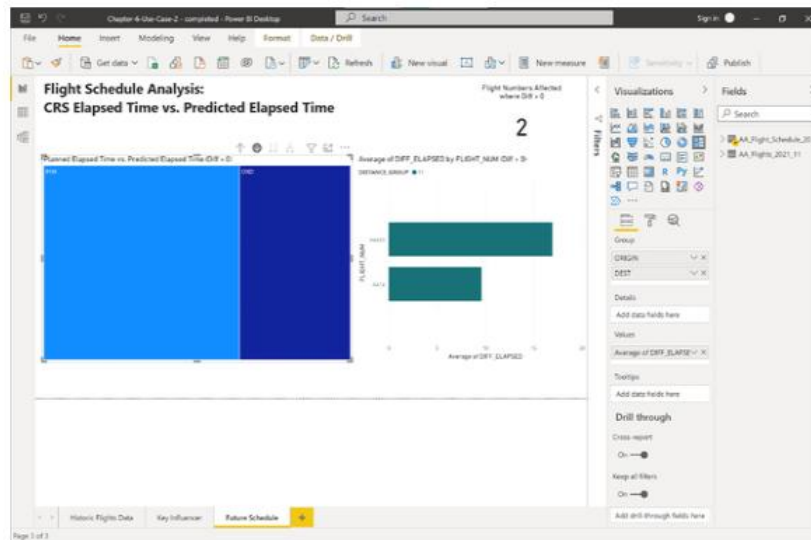


- Filter untuk visual ini diatur ke "DIFF_ELAPSED lebih besar dari 0".

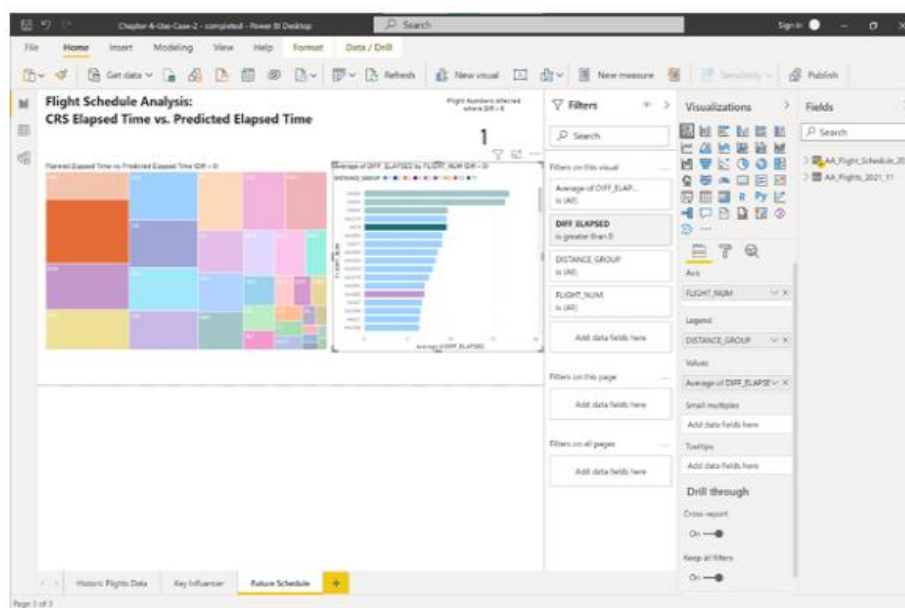
Hasilnya, Anda akan melihat visual yang diperbarui seperti yang ditunjukkan pada Gambar XX. Aktifkan fungsi drill-down di bagian atas.

Peta pohon menunjukkan bandara asal, dengan ukuran kolom rata-rata selisih antara waktu tempuh yang diprediksi dan dijadwalkan untuk penerbangan dari bandara tersebut. Setiap kali kita berinteraksi dengan pohon ini, diagram batang di sebelah kanan akan berubah dan kita dapat menelusurinya lebih jauh. Misalnya, klik bandara asal Honolulu. Anda akan melihat bahwa peta pohon diperbarui dan diagram batang juga ditampilkan seperti pada Gambar di bawah.





- 2 penerbangan, satu tertunda 10 menit dan yang lainnya tertunda 15 menit. Demikian pula, kita dapat memilih batang dari diagram batang di sebelah kanan dan Power BI akan menyortir bandara asal penerbangan ini seperti yang ditunjukkan pada Gambar di bawah.



Hal ini memberi kami interaktivitas dan sumber daya yang hebat untuk mengetahui di mana letak hambatan dalam perencanaan penerbangan kami. Kami dapat melihat dengan jelas bahwa penerbangan yang dijadwalkan berangkat dari Honolulu memiliki jadwal yang padat dan bahwa koneksi dari Honolulu ke Phoenix, Arizona, memiliki peluang tertinggi untuk melebihi waktu yang dijadwalkan, setidaknya berdasarkan pengetahuan terbaik yang kami miliki sejauh ini.

Meskipun modelnya belum sempurna, kami telah meningkatkan baseline kami dan siapa tahu mungkin kami dapat memikirkan lebih banyak kriteria yang dapat membuat model lebih akurat? Setidaknya sekarang kami memiliki contoh yang bagus untuk menunjukkan bagaimana model AI yang lebih baik dapat bermanfaat untuk meningkatkan jadwal kami sebelumnya.

Temukan file PBI final di repositori buku - Bab-6-Kasus-Penggunaan-2 - Completed.pbix

7.4 KASUS PENGGUNAAN: DETEKSI ANOMALI OTOMATIS

Dalam kasus penggunaan sebelumnya, kami mempelajari cara memanfaatkan AI untuk memprediksi variabel kategorikal dan numerik serta cara menggunakan keterampilan tersebut untuk menciptakan nilai bisnis. Dalam kasus penggunaan berikut, kami tidak terlalu tertarik apakah suatu nilai bersifat kategoris atau numerik, tetapi kami ingin mengetahui apakah suatu nilai tertentu abnormal. Untuk mengidentifikasi apakah suatu nilai merupakan anomali atau bukan, algoritma akan menganggap titik-titik tersebut sebagai serangkaian peristiwa dalam waktu, yang juga dikenal sebagai deret waktu. Mari kita bahas rumusan masalah dari kasus penggunaan kami.

Rumusan Masalah

Kami adalah bagian dari tim Operasional American Airlines dan tugas kami adalah menjaga tingkat layanan. Ini termasuk mengamati apa yang terjadi di bandara-bandara tempat American Airlines lepas landas, terutama yang paling sering. Bandara-bandara terpenting yang mencatat lebih dari 1.000 keberangkatan penerbangan pada Januari 2021 adalah *Dallas Fort Worth (DFW)*, *Bandara Charlotte Douglas (CLT)*, *Phoenix Sky Harbor (PHX)*, *Bandara Internasional Miami (MIA)*, *Chicago O'Hare (ORD)*, *Bandara Internasional Philadelphia (PHL)*, dan *Bandara Internasional Los Angeles (LAX)*.

Di antara metrik lainnya, kami harus memantau secara ketat waktu keberangkatan taksi yang terjadi di bandara-bandara ini. Waktu taksi keluar didefinisikan sebagai waktu antara pesawat didorong mundur dari gerbang dan roda lepas landas. Waktu taksi keluar yang tinggi merupakan proksi untuk bandara yang sibuk atau penuh sesak dan dapat menyebabkan penundaan penerbangan yang tidak terduga (seperti yang telah kita lihat pada kasus penggunaan sebelumnya).

Meskipun fluktuasi alami waktu taksi keluar adalah normal dan diperhitungkan dalam jadwal penerbangan, terlalu banyak puncak waktu taksi dapat menyebabkan penundaan berulang. Waktu taksi keluar yang stabil merupakan prasyarat penting untuk menjamin kelancaran proses keberangkatan di bandara. Selain itu, waktu taksi keluar yang lama meningkatkan kemacetan dan emisi gas rumah kaca yang berlebihan.

Memantau status taksi keluar dan melaporkan lonjakan yang tidak diinginkan kepada operator bandara merupakan pendekatan efektif untuk menghilangkan penundaan dan meningkatkan pemanfaatan sumber daya.

Proses ini saat ini ditangani melalui dasbor BI yang menampilkan ikhtisar rata-rata waktu taksi keluar harian untuk bandara-bandara dengan lalu lintas tinggi yang disebutkan di atas. Dasbor tersebut ditunjukkan pada Gambar di bawah.



<6-taxi-dashboard-baseline>

Tujuan utama analisis yang mengamati dasbor ini adalah untuk mengidentifikasi puncak-puncak waktu yang tidak normal untuk suatu bandara guna menyampaikan masalah ini kepada tim Operasi untuk diselidiki lebih lanjut. Permasalahan utamanya adalah: Apa yang dimaksud dengan titik abnormal? Garis solid pada grafik menunjukkan rata-rata waktu taksi keluar harian untuk bulan Januari di 7 bandara yang disebutkan di atas.

Garis horizontal putus-putus menampilkan rata-rata waktu taksi keluar untuk masing-masing bandara ini pada bulan Januari. Seperti yang dapat kita lihat dari dasbor, nilai rata-rata ini bervariasi tergantung pada bandara. Rentang rata-rata taksi keluar adalah antara 15,3 menit untuk bandara Internasional Los Angeles dan hampir 20 menit untuk Dallas Fort Worth. Itulah sebabnya sulit untuk menilai waktu taksi keluar yang abnormal secara umum, tetapi sebaliknya harus dipertimbangkan oleh bandara.

Namun, menandai semua waktu taksi keluar di atas rata-rata tidaklah praktis karena waktunya memiliki terlalu banyak fluktuasi. Sebagai gantinya, tim telah memutuskan untuk menetapkan ambang batas persentil 90 untuk setiap bandara. Ambang batas ini mencakup 90% dari semua rata-rata taksi keluar untuk bandara tertentu. Semua rata-rata taksi keluar harian yang melebihi ambang batas ini telah diputuskan untuk ditandai dan dianggap sebagai anomali. Misalnya, nilai taksi keluar rata-rata untuk DFW pada tanggal 4 dan 10 Januari telah ditandai. Namun, pendekatan ini memiliki keterbatasan:

- Nilai persentil ke-90 sangat statis dan tidak mempertimbangkan tren apa pun yang terjadi dalam data.
- Selain itu, pendekatan ini sangat sensitif terhadap nilai ekstrem, yang berarti satu hari dengan nilai taksi keluar rata-rata yang tinggi dapat meningkatkan standar ke tingkat yang tidak perlu.
- Dan ketiga, sangat membosankan dan memakan waktu untuk melihat grafik ini secara manual dan menandai peristiwa bermasalah berdasarkan kasus per kasus.

Tim ini sedang mencari pendekatan yang lebih baik secara keseluruhan untuk mengidentifikasi nilai puncak waktu taksi keluar untuk bandara-bandara ini.

Ikhtisar Solusi

Tujuan kami adalah membuat prediksi titik data abnormal lebih dinamis dan lebih responsif terhadap perkembangan linimasa secara keseluruhan. Pendekatan yang kami ambil adalah deteksi anomali bertenaga AI yang menganalisis serangkaian peristiwa (nilai) selama periode waktu tertentu (stempel waktu). Deteksi anomali akan menghitung margin atas dan bawah dinamis untuk nilai yang diharapkan dan menandai semua nilai yang melebihi batas ini sebagai anomali positif (di atas batas atas) atau negatif (di bawah batas bawah).

Kali ini, kami tidak akan melatih model AI kami sendiri. Sebaliknya, kami akan menggunakan layanan AI siap pakai untuk pertama kalinya dalam buku ini. Artinya, kami akan menggunakan model AI yang telah dilatih sebelumnya pada tugas yang ingin kami capai.

Dalam kasus kami, kami menggunakan Detektor Anomali dari Azure Cognitive Services. Untuk melakukan ini, kita akan mengaktifkan titik akhir melalui portal Azure, mempelajari cara menyiapkan data dan mendapatkan prediksi dari titik akhir ini menggunakan Python dan R,

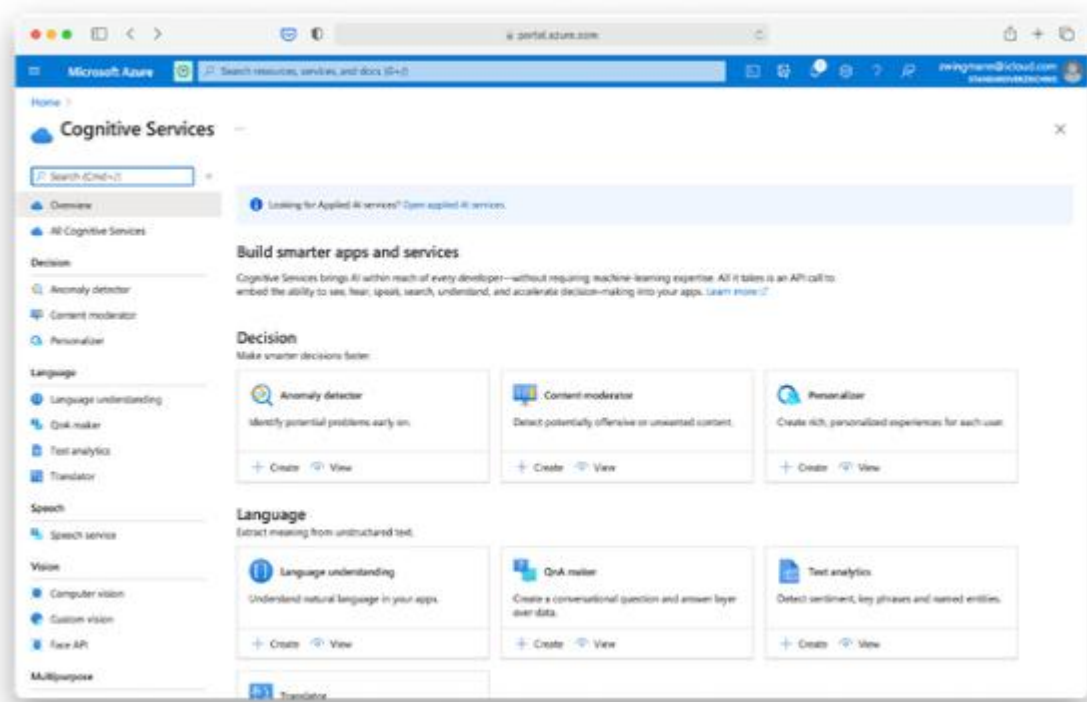


lalu menerapkan prediksi ini di BI kita untuk membangun versi yang disempurnakan dari laporan deteksi anomali taxi-out.

Perlu diketahui bahwa layanan deteksi anomali dapat digunakan untuk prediksi batch dan real-time. Prediksi real-time dalam hal ini berarti kita mengirimkan titik apa pun yang kita amati langsung ke API dan akan mendapatkan prediksi kembali apakah titik ini merupakan anomali berdasarkan titik yang kita kirimkan sebelumnya. Namun, untuk kasus penggunaan kita, kita menggunakan pendekatan batch, artinya kita mengunggah semua data yang tersedia sekaligus ke API. Tanpa basa-basi lagi, mari kita menuju ke portal Azure.

Panduan Mengaktifkan Layanan AI di Microsoft Azure

Tugas pertama kita adalah mengaktifkan API Cognitive Services Anomaly Detector di Azure dengan membuat sumber daya untuknya. Untuk melakukannya, Anda dapat mengunjungi tautan ini atau menavigasi ke sana secara manual: Kunjungi portal.azure.com, cari “Layanan Kognitif” dan di bagian “Keputusan” Anda akan menemukan kartu “Detektor Anomali” tempat Anda dapat menekan buat seperti yang ditunjukkan pada Gambar di bawah.

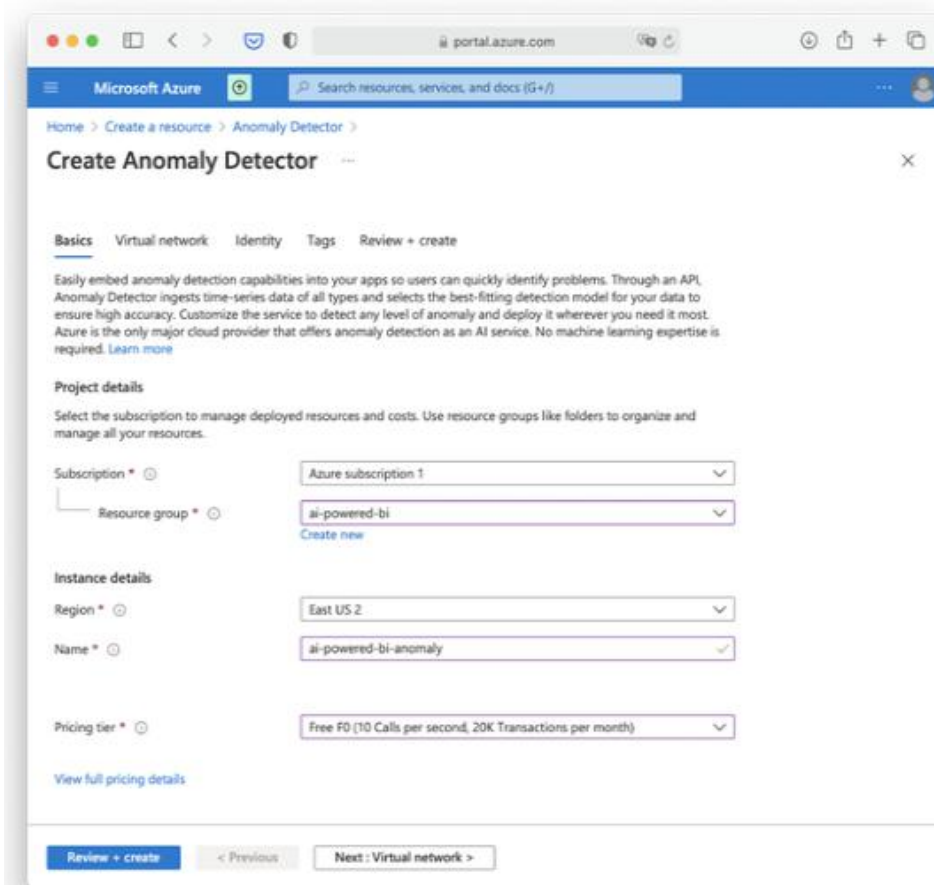


Formulir "Buat Detektor Anomali" mengharuskan Anda menentukan beberapa detail lebih lanjut. Pilih langganan Azure yang diinginkan (yang hanya akan berupa satu langganan jika Anda baru saja mendaftar) dan pilih grup sumber daya. Pilih grup sumber daya yang sama seperti yang Anda buat untuk kasus penggunaan sebelumnya. Selain itu, Anda perlu menentukan wilayah geografis tempat layanan akan dijalankan. Pilih wilayah yang diinginkan dan beri nama yang jelas untuk layanan Anda. Perhatikan bahwa nama ini unik secara global, anggap saja sebagai subdomain untuk titik akhir.



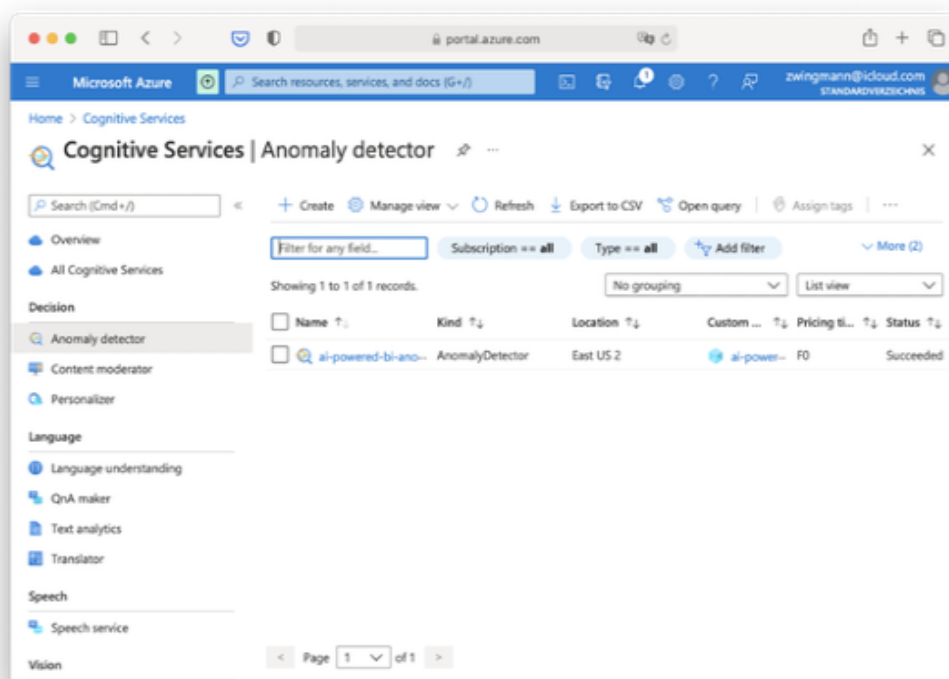
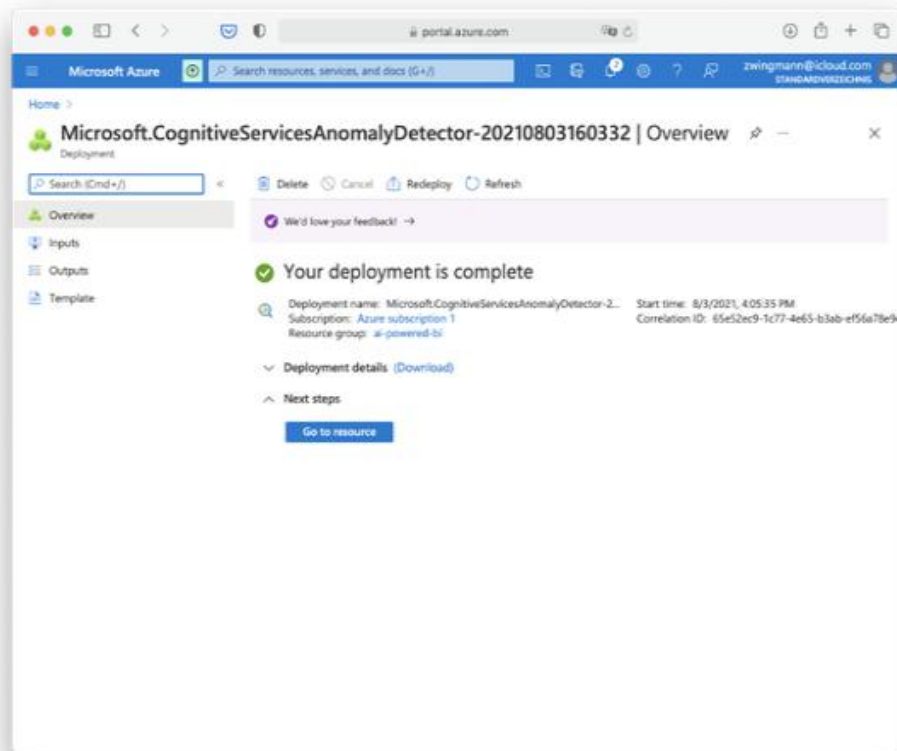
Terakhir, Anda perlu memilih tingkatan harga. Tingkatan harga yang berbeda memengaruhi performa dan model harga layanan AI. Untuk tujuan kami, tingkatan Gratis seharusnya cukup, yang memungkinkan 10 Panggilan per detik dan 20.000 transaksi per bulan. Lihat Gambar di bawah untuk contoh formulir yang telah diisi. Lanjutkan dengan memilih "Tinjau + Buat". Tinjau pengaturan Anda di halaman berikutnya dan konfirmasi dengan mengklik "Buat". Proses penerapan layanan mungkin memakan waktu beberapa menit. Setelah selesai, Anda akan melihat notifikasi di Azure dan pesan sukses seperti yang ditunjukkan pada Gambar di bawah.

Setelah penerapan berhasil diselesaikan, navigasikan ke sumber daya yang baru dibuat dengan mengklik "Buka sumber daya" seperti yang ditunjukkan pada Gambar di bawah. Atau, Anda dapat menavigasi dari halaman beranda Portal Azure ke Cognitive Services, pilih "Detektor anomali" dari menu di sebelah kiri, dan klik nama titik akhir yang akan Anda temukan dalam daftar seperti yang ditunjukkan pada Gambar di bawah.



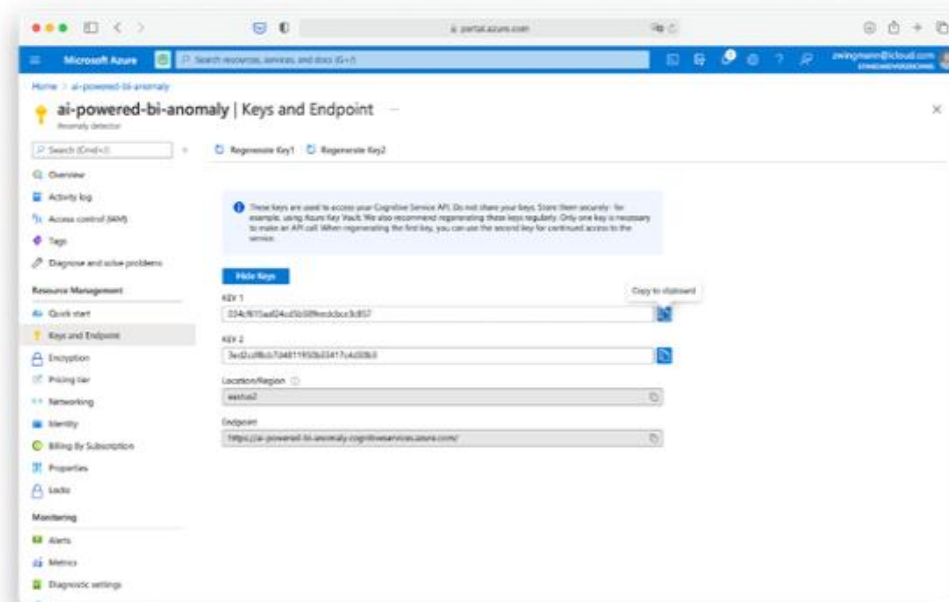
The image shows a screenshot of the Microsoft Azure portal's 'Create Anomaly Detector' form. The browser address bar shows 'portal.azure.com'. The page title is 'Create Anomaly Detector'. The form is divided into several sections: 'Basics', 'Virtual network', 'Identity', 'Tags', and 'Review + create'. The 'Basics' section is active and contains the following fields: 'Subscription' (Azure subscription 1), 'Resource group' (ai-powered-bi), 'Region' (East US 2), 'Name' (ai-powered-bi-anomaly), and 'Pricing tier' (Free F0 (10 Calls per second, 20K Transactions per month)). At the bottom of the form, there are three buttons: 'Review + create', '< Previous', and 'Next: Virtual network >'. The 'Review + create' button is highlighted in blue.





Setelah Anda memilih sumber daya Detektor Anomali, Anda akan diarahkan ke panduan mulai cepat. Ini adalah tempat yang tepat untuk kembali jika Anda ingin mengintegrasikan titik akhir ke aplikasi selanjutnya. Halaman terpenting yang kita perlukan untuk kebutuhan kita adalah

bagian "Kunci dan Titik Akhir", yang dapat Anda temukan di menu sebelah kiri seperti yang ditunjukkan pada Gambar di bawah.



Di halaman ini, Anda akan menemukan tiga komponen penting: Dua komponen pertama adalah kredensial rahasia yang Anda perlukan untuk mengakses layanan AI. Anggap saja seperti kata sandi yang memberi Anda otorisasi untuk mengajukan permintaan terhadap layanan tersebut. Seperti halnya kata sandi, Anda sebaiknya tidak menyimpan kunci ini di suatu tempat dalam bentuk teks biasa dan tidak membagikannya secara publik di mana pun. Jika Anda melakukannya, baik secara tidak sengaja maupun sengaja (seperti mencetaknya di buku), tekan tombol "Regenerate Key 1" atau "Regenerate Key 2" untuk membuat pasangan kunci baru. Ini akan membatalkan kunci sebelumnya dan membuat yang baru untuk Anda.

Mengapa kita membutuhkan dua kunci? Bayangkan seperti memiliki kunci tamu untuk rumah Anda. Ada satu kunci, yang dapat Anda bagikan dan distribusikan kepada rekan kerja untuk keperluan pembuatan prototipe cepat, jadi tidak ada salahnya jika kunci tersebut sering dibuat ulang. Kunci lainnya harus terintegrasi dengan aman ke dalam aplikasi Anda yang sedang berjalan di mana Anda tidak ingin terlalu sering mengganti kunci ini (kemungkinan besar tidak akan pernah).

Dan terakhir, komponen ketiga di halaman ini adalah titik akhir API. Namun, berhati-hatilah ini bukan titik akhir lengkap yang dapat Anda gunakan untuk membuat permintaan inferensi seperti yang kami lakukan pada contoh sebelumnya. Sebagai gantinya, Anda perlu menambahkan jalur ke layanan yang sebenarnya, versi layanan, dan pengaturan layanan ke dalamnya. Untuk mengetahui cara membuat titik akhir REST final, Anda dapat merujuk ke dokumentasi API atau panduan singkat yang Anda kunjungi sebelumnya.

Misalnya, titik akhir REST final untuk inferensi batch anomali terlihat seperti berikut (ganti "ai-powered-bi" dengan nama layanan Anda sendiri:

```
https://ai-powered-bi-  
anomaly.cognitiveservices.azure.com/anomalydetector/v1.0/timeseries  
/entire/detect
```

Selesai. Sekarang kita siap untuk mulai membuat permintaan prediksi dari layanan AI yang baru kita siapkan. Mari kita cari tahu cara melakukannya dengan Python atau R, tergantung preferensi Anda.

Panduan Inferensi Model dengan Python

- Menjelaskan inferensi model (Pemanggilan API) dengan Python

Panduan Inferensi Model dengan R

Di repositori buku, Anda akan menemukan berkas bernama "anomaly-inference.r" yang berisi skrip yang akan kita gunakan untuk mendapatkan inferensi untuk contoh kasus penggunaan kita. Mari saya jelaskan secara singkat apa yang dilakukan skrip tersebut:

Inti dari skrip ini adalah fungsi yang disebut `inference_request`. Fungsi ini sangat mirip dengan fungsi permintaan yang kita gunakan sebelumnya, tetapi mengandung beberapa variasi. Perubahan terbesarnya adalah badan permintaan telah disusun secara berbeda agar sesuai dengan persyaratan API Azure Anomaly Detector. API mengharapkan objek JSON yang berisi informasi tentang granularitas deret waktu (dalam kasus kita, yaitu harian) dan kemudian deret waktu aktual dengan stempel waktu dan nilai yang sesuai. Dua vektor yang Anda berikan sebagai input ke fungsi (vektor stempel waktu dan vektor nilai) akan ditempatkan di sini dalam format yang tepat.

Perubahan kedua adalah kami menambahkan `api_key` untuk tujuan otorisasi.

Saat Anda menggunakan skrip untuk latihan Anda sendiri, Anda perlu menggantinya dengan:

- `url`: Ganti "ai-powered-bi-anomaly" dengan nama layanan kustom Anda.
- `api_key`: Ganti dengan kunci 1 atau kunci 2 dari Portal Azure seperti yang dijelaskan sebelumnya.



```
inference_request <- function(timestamp, value) {

  # Accept SSL certificates issued by public Certificate
  Authorities
  options(RCurlOptions = list(cainfo = system.file("CurlSSL",
"cacert.pem", package = "RCurl"), ssl.verifypeer = FALSE))

  h = basicTextGatherer()
  hdr = basicHeaderGatherer()

  # Bind columns to dataframe
  request_df <- data.frame(timestamp, value)

  # Build JSON object for request body
  granularity = c("granularity" = "daily")
  series = list("series"=apply(request_df,1,as.list))
  req = c(granularity, series)
  body = enc2utf8(toJSON(req))
  api_key = "c3580b4786e44ba8ae95578ed83eb21f" # Replace this with
the API key for the web service

  # Perform HTTP request
  h$reset()
  curlPerform(
    url = "https://ai-powered-bi-
anomaly.cognitiveservices.azure.com/anomalydetector/v1.0/timeseries
/entire/detect",
    httpheader=c("Content-Type" = "application/json", "Ocp-Apim-
Subscription-Key" = api_key, "Ocp-Apim-Subscription-
Region"="eastus2"),
    postfields=body,
    writefunction = h$update,
    headerfunction = hdr$update,
    verbose = FALSE
  )

  headers = hdr$value()
  httpStatus = headers["status"]

  if (httpStatus >= 400)
  {
    return(paste("The request failed with status code:",
httpStatus, sep=" "))
  }

  # Get results
  result = h$value()
  result = fromJSON(result)
  return(result)
}
```



Setelah kita membahas fungsi inferensi, mari kita lihat beberapa hal lain yang dilakukan skrip ini untuk kita:

Pada baris 11 dan 11, kita menetapkan ambang batas untuk pemilihan bandara. Perlu diingat bahwa kita tidak ingin memeriksa anomali untuk semua bandara, tetapi hanya untuk bandara tersibuk, yang didefinisikan sebagai bandara dengan minimal 1000 penerbangan keberangkatan.

```
# Threshold for minimum flights per airport
MIN_FLIGHTS <- 1000
```

Variabel ini akan digunakan pada baris 60 - 64 di mana kita mendapatkan kerangka data yang berisi daftar bandara yang cocok dengan kriteria ini.

```
# Get airports where threshold is met
airports <- df %>%
  group_by(Origin) %>%
  count() %>%
  filter(n >= MIN_FLIGHTS) %>%
  arrange(desc(n))
```

Setelah kita memiliki fungsi inferensi dan daftar bandara yang akan kita gunakan untuk melakukan inferensi ini, kita memerlukan kode yang menerapkan fungsi inferensi tersebut ke rata-rata waktu taksi keluar harian untuk masing-masing bandara. Kita melakukannya dengan menjalankan skrip prapemrosesan di dalam for loop yang akan mengulang daftar bandara yang telah kita pilih berdasarkan kriteria ambang batas. Skrip berikut akan membuat kerangka data agregat untuk waktu taksi keluar setiap bandara dan meneruskan informasi ini ke fungsi `inference_request`. Semua hasil akan dikumpulkan dalam kerangka data yang disebut `df_inference_all`.



```
for (airport in airports$Origin) {
  # Create aggregated dataframe for average TaxiOut per Airport
  df_inference <- df %>%
    select(FlightDate, Origin, TaxiOut) %>%
    filter(Origin == airport) %>%
    group_by(FlightDate, Origin) %>%
    summarize(AvgTaxiOut = mean(TaxiOut)) %>%
    mutate(Timestamp = paste0(FlightDate, "T00:00:00Z"),
           AvgTaxiOut = round(AvgTaxiOut,2))

  # Submit request
  result <- inference_request(df_inference$Timestamp,
df_inference$AvgTaxiOut)

  # Convert result object to dataframe
  results_df <- as.data.frame(result)

  # Add inference results to inference_df
  df_inference <- bind_cols(df_inference, results_df) %>%
    select(-Timestamp)

  df_inference_all <- bind_rows(df_inference_all, df_inference)
}
```

Terakhir, kami menggabungkan kembali informasi tersebut ke kerangka data asli agar dapat menggunakan kolom-kolom baru di alat BI kami, seperti yang telah kami lakukan sebelumnya. Proses ini ditangani oleh kode berikut:

```
# Add inference information to original dataframe
df <- df %>%
  left_join(df_inference_all, by = c("FlightDate", "Origin")) %>%
  mutate(upperMargins = expectedValues + upperMargins,
         lowerMargins = expectedValues - lowerMargins)
```

Jangan khawatir jika kodenya mungkin terlihat terlalu rumit atau berlebihan pada pandangan pertama. Anda selalu dapat kembali dan mengulanginya langkah demi langkah jika perlu. Jika tidak, silakan salin dan tempel semuanya sambil mengganti variabel kustom Anda (url dan api_key).

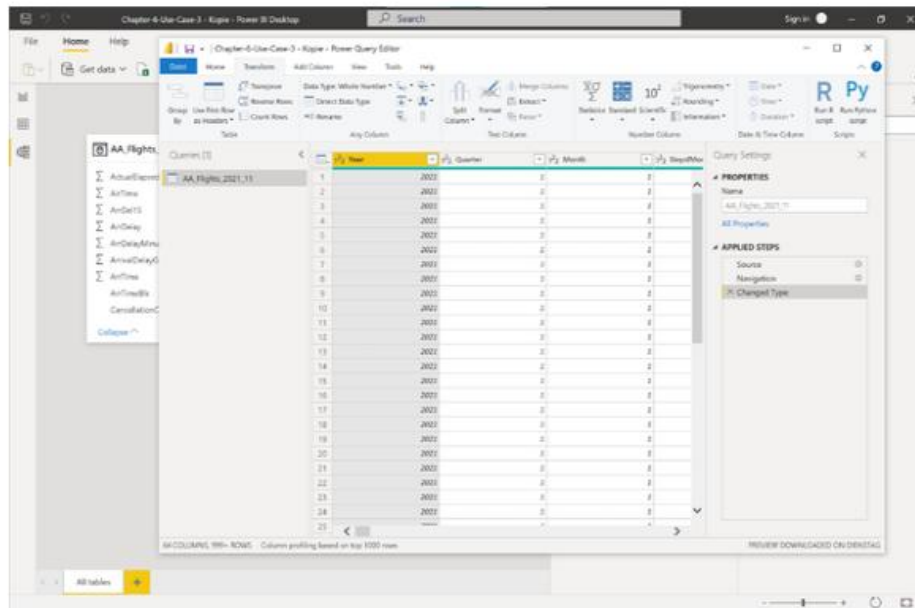
Panduan Inferensi Model dengan Power BI

Tujuan kami adalah memanfaatkan detektor anomali untuk membuat laporan yang lebih baik dan wawasan yang lebih mudah ditindaklanjuti. Kami ingin membuat laporan keseluruhan lebih akurat, interaktif, dan terbebas dari upaya manual.

Pertama, mari kita hubungkan AI ke dalam model data kita.

Buka file Chapter-6-use-case-3.pbix di PowerBI Desktop atau Power BI Service dan navigasikan ke model data. Buka Power Query Editor seperti yang ditunjukkan pada Gambar di bawah.





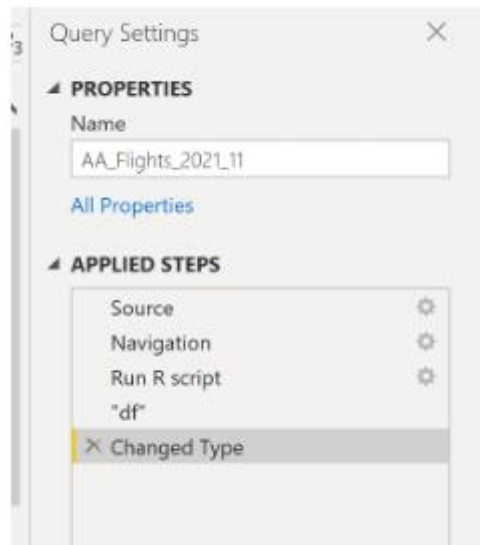
Pilih Transform di menu pita dan pilih "R script" atau "Python script" sesuai preferensi Anda. Saya akan melanjutkan contoh dengan R, tetapi langkah yang sama juga berlaku untuk Python. Saat editor skrip terbuka, jangan tempel kode dari berkas sumber kita dulu. Sebagai gantinya, masukkan baris berikut, baik Anda menggunakan R maupun Python:

```
df = dataset
```

Konfirmasikan dengan "OK" dan Anda akan melihat dua langkah ditambahkan di bawah "Langkah yang Diterapkan" di sebelah kanan.

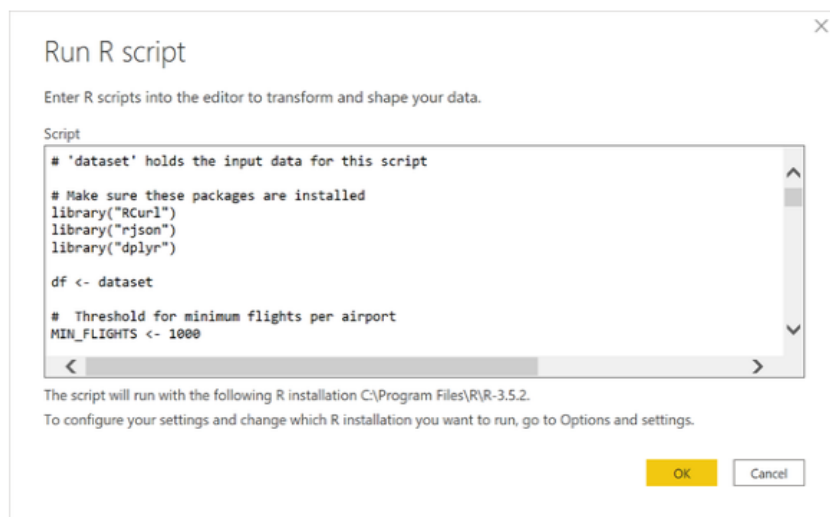
Alasan kami belum menyertakan skrip dari berkas sumber adalah karena pada tahap ini skrip tersebut tidak akan berfungsi. Power BI mengonversi kolom FlightDate dari String ke format Datetime yang dipatenkan, yang sulit kami proses dengan Python atau R. Namun, kami tidak ingin menghapus konversi tipe ini, karena akan memberikan pengalaman pelaporan yang lebih baik.

Yang perlu kami lakukan adalah menerapkan skrip R atau Python kami sebelum konversi tipe data dilakukan. Oleh karena itu, pilih langkah "Ubah Tipe" dan seret ke akhir daftar. "Langkah yang Diterapkan" sekarang akan terlihat seperti yang ditunjukkan pada Gambar di bawah.



Setelah langkah-langkah diterapkan dalam urutan yang benar, kita sekarang dapat memilih langkah "Jalankan skrip R" atau "Jalankan skrip Python" sekali lagi dan mengedit skrip dengan mengklik ikon roda gigi di sebelah kanan.

Sekarang, kita dapat menempelkan skrip dari berkas ke editor kode Power BI seperti yang ditunjukkan pada Gambar di bawah. Anda dapat mengganti konten yang sudah ada, juga baris kode yang baru saja kita tulis sebelumnya.



Klik OK dan tunggu skrip dieksekusi. Proses ini mungkin memakan waktu beberapa menit, tergantung kinerja komputer Anda.

Jika skrip sudah selesai, pilih langkah terakhir "Changed Type" dari daftar. Periksa kembali apakah kolom-kolom baru seperti expectedValues, isAnomaly, isPositiveAnomaly, dan upperMargins telah ditambahkan dengan benar ke tabel. Tidak masalah jika baris pertama menampilkan nilai null, karena kami tidak melakukan inferensi untuk seluruh dataset, tetapi



hanya untuk beberapa bandara. Jika Anda menggulir ke bawah, Anda akan melihat beberapa hasil muncul dari baris 32 dan seterusnya seperti yang terlihat pada Gambar di bawah.

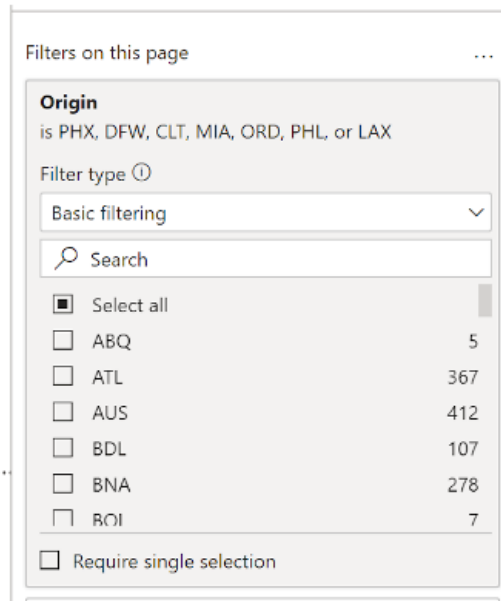
	IsIntegratesAnomaly	IsPositiveAnomaly	IsLowestOrigins	
29	null	null	null	null
30	null	null	null	null
31	null	null	null	null
32	FALSE	FALSE	FALSE	34.04499889
33	FALSE	FALSE	FALSE	34.00909904
34	TRUE	FALSE	TRUE	34.27461747
35	FALSE	FALSE	FALSE	33.79409105
36	FALSE	FALSE	FALSE	34.32661208
37	TRUE	FALSE	TRUE	34.3505298
38	FALSE	FALSE	FALSE	34.24612201
39	FALSE	FALSE	FALSE	34.36145381
40	FALSE	FALSE	FALSE	33.52106165
41	FALSE	FALSE	FALSE	33.37438712
42	FALSE	FALSE	FALSE	33.33124084
43	FALSE	FALSE	FALSE	33.62827324
44	FALSE	FALSE	FALSE	33.43388079
45	FALSE	FALSE	FALSE	33.748972164
46	FALSE	FALSE	FALSE	33.72290332
47	FALSE	FALSE	FALSE	34.61249075
48	FALSE	FALSE	FALSE	33.97399578
49	FALSE	FALSE	FALSE	34.36467209
50	TRUE	FALSE	TRUE	34.47652056
51	FALSE	FALSE	FALSE	34.35889268
52	FALSE	FALSE	FALSE	34.61404654
53	TRUE	FALSE	TRUE	34.64952231

Klik File -> Tutup dan Terapkan untuk keluar dari Editor Power Query. Kembali ke menu model data, Anda akan melihat daftar kolom baru dalam model Anda.

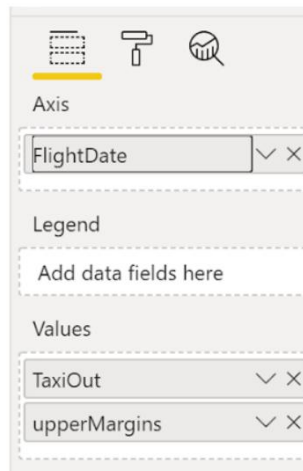
Proses yang berat sudah selesai. Sekarang, kita dapat melanjutkan ke dasbor dan membuat prediksi kita terlihat oleh pengguna laporan. Di bagian "Laporan", tambahkan halaman laporan lain dengan mengklik ikon plus di samping halaman laporan pertama. Kita tidak ingin memperbarui visual yang ada, tetapi mencari pendekatan yang lebih sesuai dengan informasi yang kita miliki.

Tujuan kita adalah membuat diagram garis yang tidak hanya menampilkan waktu taksi rata-rata aktual, tetapi juga batas atas yang diprediksi untuk deret waktu tersebut. Selain itu, kita ingin menyorot titik data yang melampaui batas atas dan oleh karena itu menandainya sebagai anomali. Dan terakhir, kita ingin menyediakan cara interaktif bagi pengguna untuk beralih antar bandara dengan mudah.

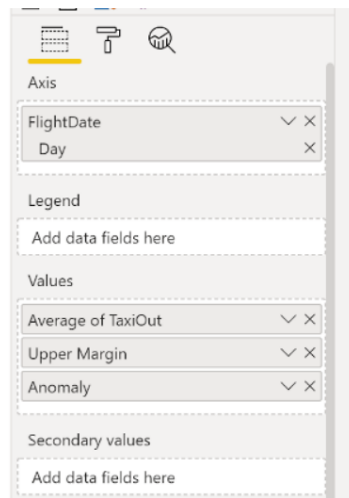
Sebelum menambahkan visual apa pun, buka panel Filter dan tambahkan filter halaman ke halaman laporan ini seperti yang ditunjukkan pada Gambar di bawah. Ingatlah bahwa kita hanya ingin menganalisis sekelompok bandara tertentu, sehingga akan lebih mudah untuk memfilternya untuk semua grafik. Tambahkan kolom data "Asal" ke area "Filter di halaman ini" dan pilih bandara berikut: CLT, DFW, LAX, MIA, ORD, PHL, dan PHX. Filter akhir akan terlihat seperti pada Gambar di bawah.



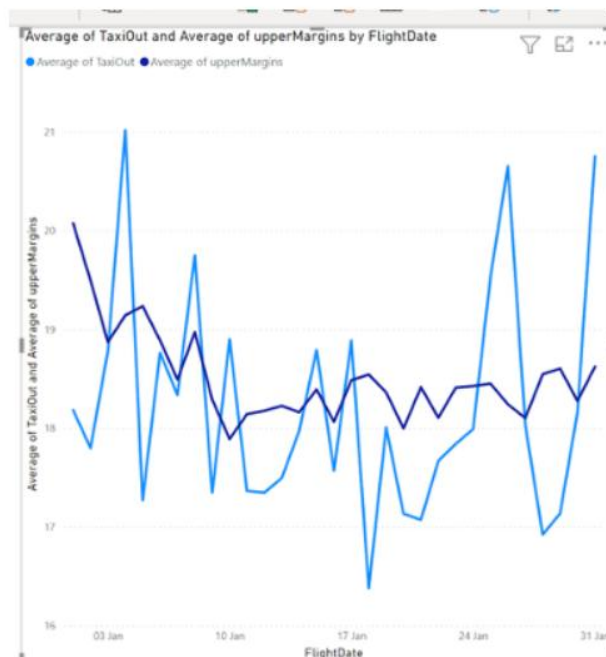
Mari kita lanjutkan dengan menambahkan visual diagram garis. Klik dua kali ikon diagram garis dari panel visualisasi. Seret diagram garis hingga menutupi sekitar setengah halaman laporan. Tambahkan kolom berikut ke area sumbu dan nilai seperti yang ditunjukkan pada Gambar di bawah: FlightDate (Sumbu), TaxiOut, upperMargins (Nilai).



Di bawah FlightDate, hapus hierarki Tahun, Kuartal, dan Bulan dengan mengeklik ikon "x" kecil sehingga hanya tersisa Hari seperti yang ditunjukkan pada Gambar di bawah. Selain itu, klik kanan pada TaxiOut dan upperMargins, lalu pilih "Average" sebagai fungsi agregasi, bukan sum. Label kolom sekarang seharusnya "Average of TaxiOut" dan "Average of upperMargin".



Sekarang Anda akan melihat dua garis seperti yang ditunjukkan pada Gambar di bawah

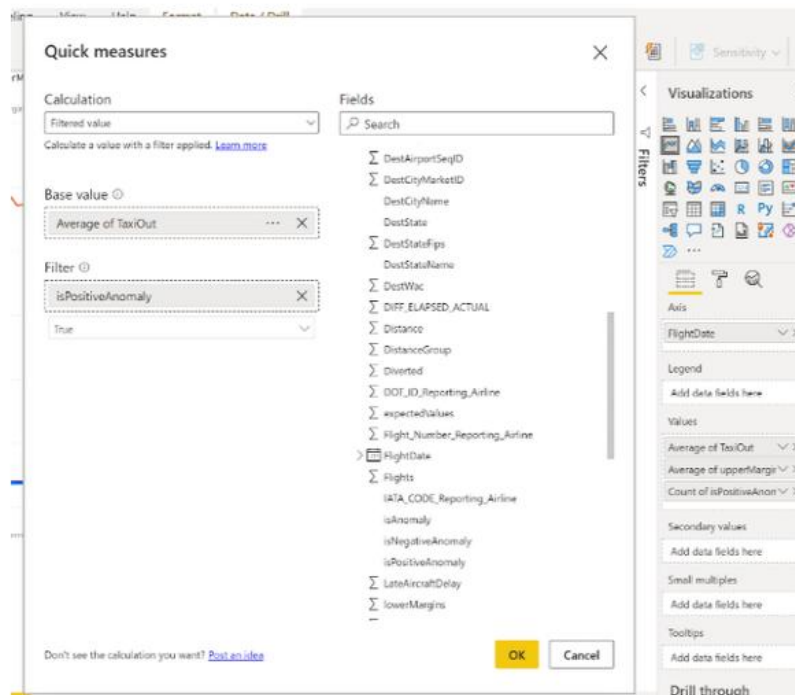


Grafik menunjukkan rata-rata taksi keluar dan rata-rata margin atas untuk bandara yang kami tentukan pada filter halaman. Namun, kami ingin menganalisis bandara satu per satu. Oleh karena itu, buka kembali panel filter dan atur filter khusus untuk visual ini, yaitu "Asal" dan "DFW" seperti yang ditunjukkan pada Gambar di bawah. Kami akan menambahkan interaktivitas nanti agar pengguna dapat memilih bandara yang ingin mereka analisis.

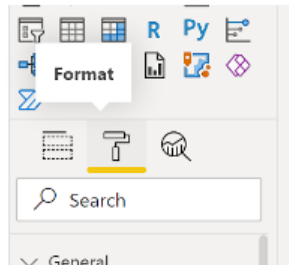


Anda seharusnya sudah menetapkan dua filter: Satu filter pada halaman untuk 7 bandara, dan satu filter pada visual untuk bandara DFW. Selain itu, kita sudah memiliki dua garis pada grafik: Satu garis menunjukkan rata-rata taksi keluar harian dan satu garis menunjukkan batas atas yang diprediksi. Yang masih kurang adalah sorotan outlier. Kita hampir sampai di sana!

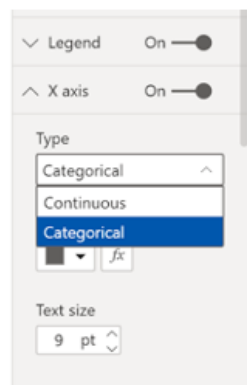
Tambahkan kolom `isPositiveAnomaly` ke nilai sebagai baris ketiga. Klik kanan dan pilih "New Quick Measure". Di jendela yang muncul, atur pengaturan seperti yang ditunjukkan pada Gambar di bawah: Pilih perhitungan menjadi "Nilai yang Difilter", atur nilai dasar menjadi "Rata-rata TaxiOut", dan Filter menjadi `isPositiveAnomaly` sama dengan `true`. Konfirmasikan dengan OK. Hapus "Jumlah `isPositiveAnomaly`" dari area nilai jika masih muncul di sini. Seharusnya sekarang sudah ada 3 nilai, "Rata-rata TaxiOut", "Rata-rata upperMargin", dan "Rata-rata TaxiOut untuk True". Klik kanan pada "Rata-rata TaxiOut untuk True" dan pilih "Ganti Nama untuk visual ini". Ganti nama kolom ini menjadi "Anomali". Demikian pula, ubah nama "Rata-rata Margin Atas" menjadi "Margin Atas" untuk visual ini.



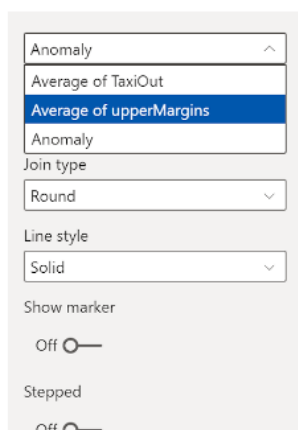
Dengan ketiga nilai ini, kita memiliki semua informasi yang kita butuhkan tentang plot. Sisanya "hanya" format. Jadi, buka pengaturan Format seperti yang ditunjukkan pada Gambar di bawah.



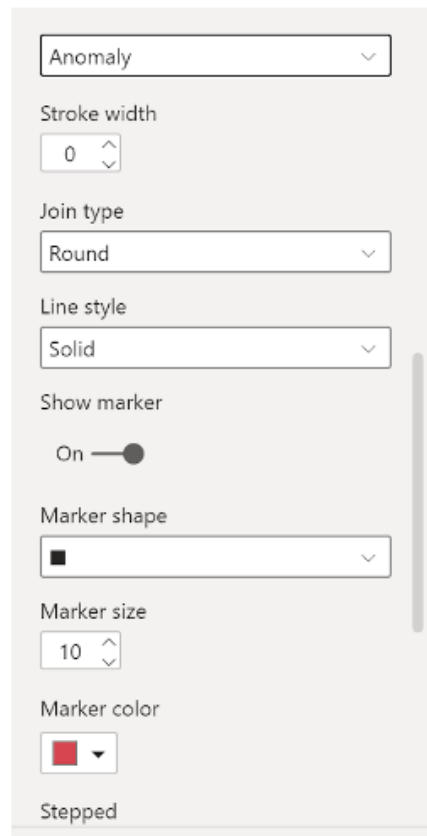
Pilih X-Axis dan ubah jenisnya dari kontinu menjadi kategoris.



Buka "Warna Data" dan ubah warna "Margin Atas" dan "Anomali" menjadi Merah. Selanjutnya, pilih "Bentuk" dan atur lebar guratan menjadi 6. Aktifkan "Sesuaikan Seri". Dari menu tarik-turun seperti yang ditunjukkan pada Gambar di bawah, pilih "Margin Atas".



Kita ingin memberikan gaya yang berbeda pada pita atas ini untuk memperjelas bahwa ini adalah batas, bukan nilai data yang sebenarnya. Jadi, atur gaya garis ke "Dotted" dan aktifkan "Stepped". Selanjutnya, pilih "Anomaly" dari menu tarik-turun di "Shapes" dan aktifkan "Show marker". Ubah bentuk marker dari lingkaran menjadi persegi dan tingkatkan ukuran marker menjadi 10 seperti yang ditunjukkan pada Gambar di bawah. Atur lebar goresan ke 0.



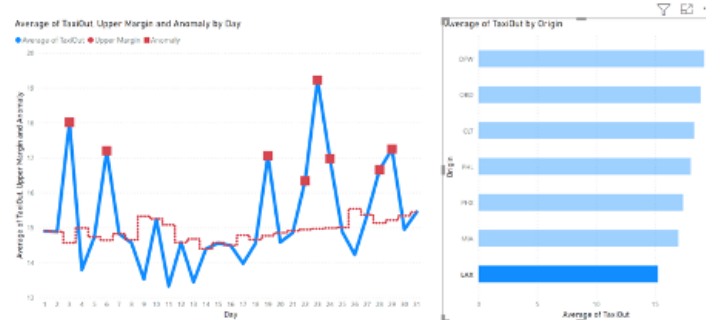
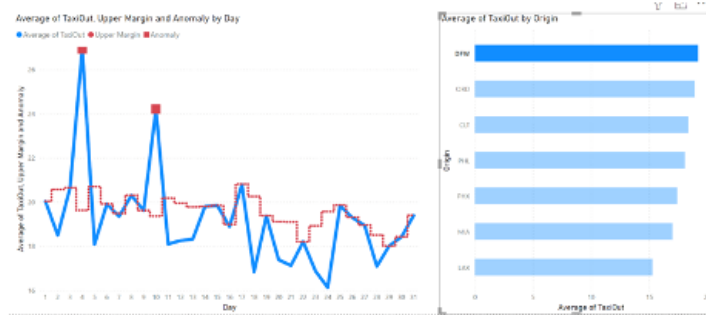
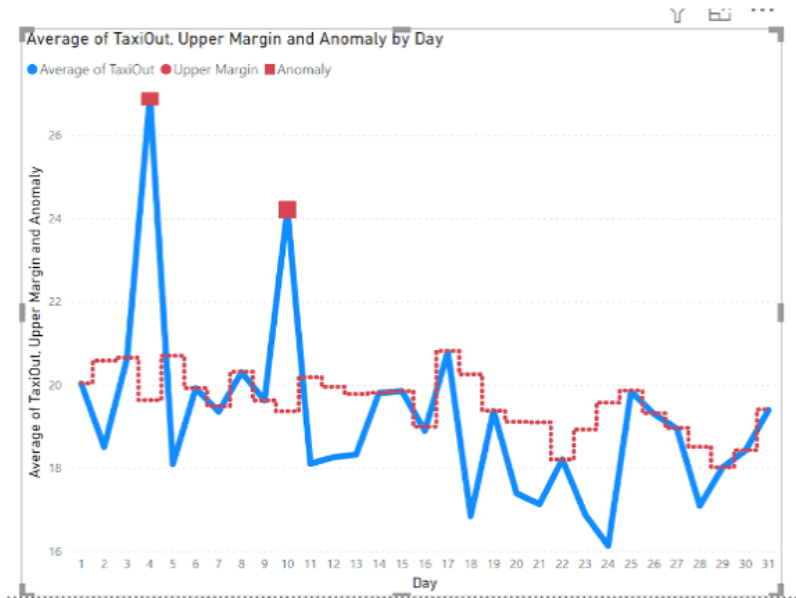
Sekarang, diagram garis Anda seharusnya terlihat seperti pada Gambar di bawah, dan kita hampir siap! Satu-satunya yang belum ada adalah pengguna dapat memilih bandara yang ingin mereka periksa. Kita akan menggunakan fitur crossfilter di Power BI agar proses ini lebih mudah bagi pengguna.

Tambahkan Bagan Batang Bertumpuk dari panel visualisasi ke halaman laporan dengan klik dua kali. Pastikan bagan garis Anda tidak dipilih, jika tidak, visual ini akan tergantikan. Bagan batang akan otomatis mengisi ruang kosong tepat di sebelah bagan garis. Tambahkan "Asal" ke sumbu dan "TaxiOut" ke nilai. Klik kanan TaxiOut dan pilih "Rata-rata" sebagai ukuran agregasi. Di bawah "Format", pilih "Sumbu Y" dan tingkatkan padding bagian dalam menjadi 50%.

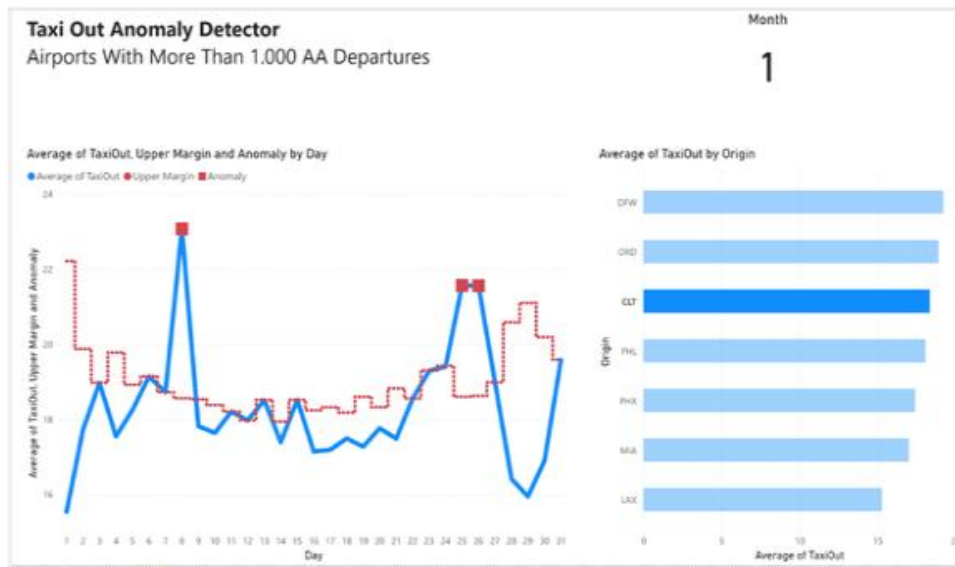
Fungsi filter silang di Power BI akan membuat setelah pengguna memilih batang dari bagan batang, visual lain di halaman yang sama akan difilter sesuai dengan pilihannya. Untuk mengaktifkan pemfilteran dinamis pada bagan garis kita, pilih bagan garis, buka panel filter, dan hapus filter visual "Asal adalah DFW" yang kita tetapkan untuk mendesain bagan. Biarkan filter halaman tetap utuh. Sekarang, setiap kali kita memilih batang di sisi kanan, sisi kiri akan



menampilkan waktu taksi keluar rata-rata harian dengan nilai anomali disorot seperti yang Anda lihat pada Gambar di bawah.



Untuk menyelesaikan laporan kita, mari tambahkan judul dan bulan berjalan di bagian atas laporan dengan menyalin elemen-elemen dari halaman laporan pertama. Gambar di bawah menunjukkan dasbor akhir kita yang sedang beraksi. Anda dapat menemukan status akhirnya di berkas sumber daya buku terlampir "Chapter-6-Use-Case-3-AI-Powered.pbi".



Karena sekarang kita memiliki atribut `isAnomaly` dalam dataset kita, kita dapat dengan mudah menjalankan analisis lebih lanjut atau laporan otomatis untuk metrik ini. Untuk mempelajari lebih lanjut tentang cara kerja deteksi anomali secara detail, cara menjalankan hasil inferensi secara real-time, dan praktik terbaik umum seputar API ini, saya merekomendasikan sumber daya Microsoft ini.

Dengan pendekatan berbasis AI, kami telah membantu tim operasi mengidentifikasi anomali lebih cepat dan tanpa harus mengikuti aturan baku. Berkat prediksi AI, label data `"isAnomaly"` kini juga tersedia dalam model data yang memungkinkan analisis lebih lanjut atau pelaporan otomatis metrik ini, sehingga mendukung tim data dengan mengurangi banyak beban kerja manual.

7.5 RINGKASAN

Selamat! Beri diri Anda pujian karena apa yang baru saja Anda lakukan tidak lain adalah menerapkan perangkat AI canggih ke data dunia nyata dan membangun Dasbor BI yang dapat ditindaklanjuti untuk analitik prediktif. Kita tidak hanya mempelajari cara menggunakan informasi historis dalam dataset untuk memprediksi variabel kategorikal atau numerik di masa mendatang. Kami juga telah membahas layanan AI yang memberikan estimasi otomatis untuk kasus ini, yaitu deteksi anomali. Tidak masalah apakah Anda pengguna Power BI atau Tableau, atau apakah Anda lebih suka AWS, Azure, atau Google Cloud Platform. Landasan fundamental yang telah Anda pelajari dan terapkan dalam bab ini akan membantu Anda menguasai platform tersebut dengan cepat. Cobalah dengan dataset Anda sendiri dan dapatkan pengalaman langsung. Di bab selanjutnya, kita akan membahas ranah analitik preskriptif, yang menghasilkan rekomendasi cerdas untuk keputusan di tingkat mikro.



BAB 8

ANALISIS PRESKRIPTIF BERTENAGA AI

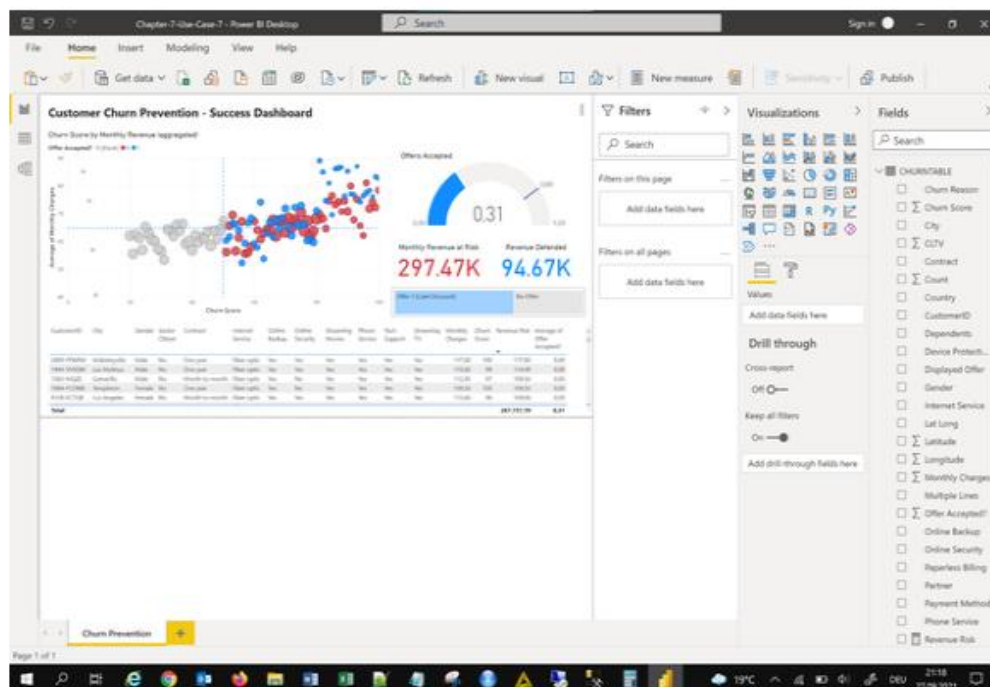
8.1 KASUS PENGGUNAAN: REKOMENDASI TINDAKAN TERBAIK BERIKUTNYA

Untuk kasus penggunaan ini, kami mengembangkan saran dari model prediktif dan memilih opsi terbaik untuk pelanggan tertentu.

Pernyataan Masalah

Kami bekerja di tim BI sebuah penyedia telekomunikasi besar. Perusahaan ini menjual berbagai produk seperti langganan kabel dan telepon seluler bulanan atau tahunan, dan memiliki jutaan pelanggan aktif setiap tahun. Perusahaan ini menghadapi masalah yang semakin meningkat, yaitu pelanggan yang berhenti berlangganan (churn). Para ilmuwan data dalam tim telah berhasil membangun model churn pelanggan yang memprediksi kemungkinan pelanggan tunggal untuk berhenti berlangganan (churn) pada akhir kuartal ini.

Prediksi churn dihitung sebagai skor churn, di mana 100 berarti probabilitas churn tertinggi dan 0 berarti probabilitas churn lebih rendah. Meskipun prediktor churn terbukti cukup akurat dan bermanfaat dalam mengidentifikasi pelanggan yang kemungkinan besar akan membatalkan langganan mereka, perusahaan masih kesulitan menemukan langkah yang tepat untuk mengatasi churn. Tim BI telah mengintegrasikan dasbor pencegahan churn yang ditunjukkan pada Gambar 8.1.



Gambar 8.1. Dasbor Pencegahan Churn Baseline

Dasbor menunjukkan diagram sebar di kiri atas yang membandingkan skor churn dengan pendapatan bulanan. Di kuadran kanan atas adalah pelanggan dengan paket bulanan yang mahal namun memiliki kemungkinan churn yang tinggi, sehingga mereka menjadi kelompok sasaran yang relevan untuk langkah-langkah pencegahan churn.

Saat ini, semua pelanggan yang berlabel "Churn" mendapatkan penawaran retensi yang sama. Rata-rata, penawaran ini diterima oleh 31% pelanggan, yang digambarkan oleh titik-titik biru pada diagram sebar. 31% penawaran yang diterima ini menghasilkan pendapatan yang dipertahankan sebesar 94,67 ribu dolar. Bagian Penjualan mengindikasikan bahwa mereka dapat memberikan penawaran lain tetapi tidak yakin penawaran mana yang akan ditampilkan kepada pelanggan. Pengujian A/B sederhana terbukti tidak efektif karena banyaknya variasi segmen pelanggan dan jenis penawaran yang berbeda.

Kepala departemen analitik menyarankan untuk menggunakan pendekatan berbasis data guna mengetahui penawaran retensi mana yang harus ditampilkan kepada berbagai segmen pelanggan. Tujuan kami adalah menemukan pendekatan yang setidaknya lebih baik dari baseline yang ada dan mendukung kemampuan untuk bereksperimen dengan jenis penawaran baru secara real-time.

Ikhtisar Solusi

Pertama, kami menggunakan hasil prediksi churn untuk memfilter semua pelanggan yang memiliki kemungkinan churn lebih tinggi dari 50%. Selanjutnya, kami akan memberikan penawaran terbaik yang disesuaikan untuk mengurangi kemungkinan pelanggan berhenti menggunakan layanan perusahaan. Untuk menemukan penawaran terbaik bagi pelanggan, kami akan menggunakan pendekatan yang disebut pembelajaran penguatan.

Dengan bereksperimen dengan berbagai penawaran untuk kelompok pelanggan yang berbeda, model penguatan akhirnya akan menentukan penawaran mana yang akan ditampilkan kepada kelompok pelanggan yang mana. Model akan belajar berdasarkan sistem penghargaan. Jika pelanggan menerima penawaran, model akan mendapatkan penghargaan, jika tidak, model akan diberi penalti. Kami akan menggunakan layanan AI Microsoft yang disebut Azure Personalizer untuk menjalankan dan memelihara model.

Untuk mendapatkan beberapa interaksi pengguna, kami juga menjalankan skrip kecil yang meniru proposal penawaran kepada pengguna dan menghasilkan penghargaan berdasarkan preferensi (tersembunyi) pengguna. Dalam dunia nyata, kita tentu tidak mengetahui preferensi pengguna ini dan harus menemukannya melalui eksperimen. Namun, demi simulasi kita, kita akan menyimpan data dasar dalam berkas JSON biasa. Tentu saja, model tidak akan memiliki akses ke berkas ini dan perlu menemukan polanya sendiri. Hasil simulasi kita akan dicatat menggunakan berkas CSV dan disimpan di penyimpanan blob Azure agar dapat diakses nanti melalui Power BI untuk analisis lebih lanjut.

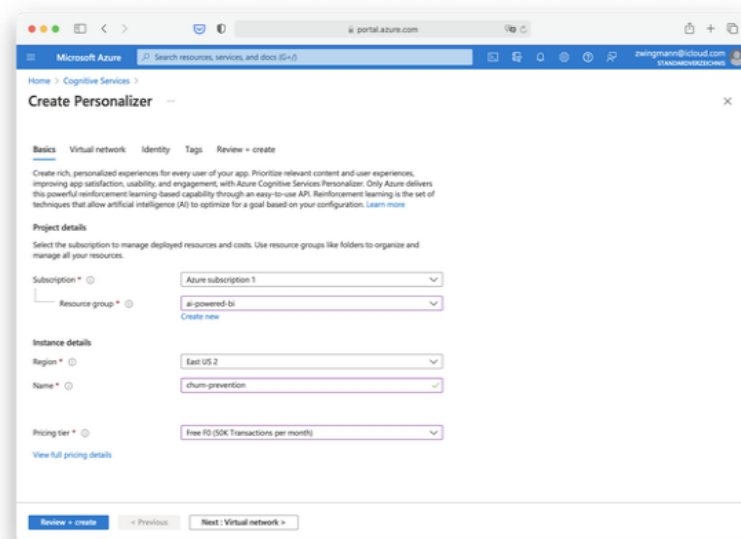
Terakhir, kita akan menggunakan Power BI untuk memantau kinerja model kita dan mendapatkan gambaran umum penawaran mana yang akan direkomendasikan untuk kelompok pengguna mana. Dengan pendekatan ini, kita seharusnya dapat melihat seberapa baik rekomendasi personalisasi kita dibandingkan dengan rekomendasi dasar.



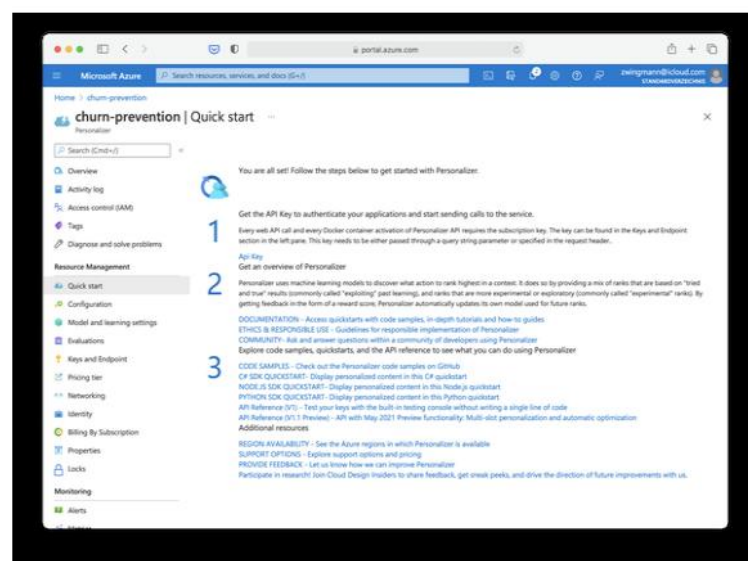
Menyiapkan Layanan AI

Buka Dasbor Azure Anda dengan mengunjungi *portal.azure.com*. Ketik "Personalizer" di bilah pencarian di bagian atas, lalu pilih "Personalizers" dari daftar layanan. Klik "Buat" yang akan membawa Anda ke formulir "Buat Personalizer" seperti yang ditunjukkan pada Gambar 8.2. Pilih grup sumber daya yang Anda buat di Bab 6 dan beri nama layanan rekomendasi Anda, misalnya "offer-engine". Pilih tingkatan harga "Free F0". Klik "Tinjau + Buat".

Setelah validasi akhir selesai, konfirmasi dengan "Buat". Proses penerapan mungkin memerlukan waktu beberapa menit untuk selesai. Setelah penerapan layanan Personalizer Anda selesai, navigasikan ke layanan tersebut dengan mengklik "Buka sumber daya". Mari kita lihat apa yang terjadi. Di halaman beranda layanan, Anda akan disambut dengan "Mulai cepat" yang terlihat seperti Gambar 8.3.



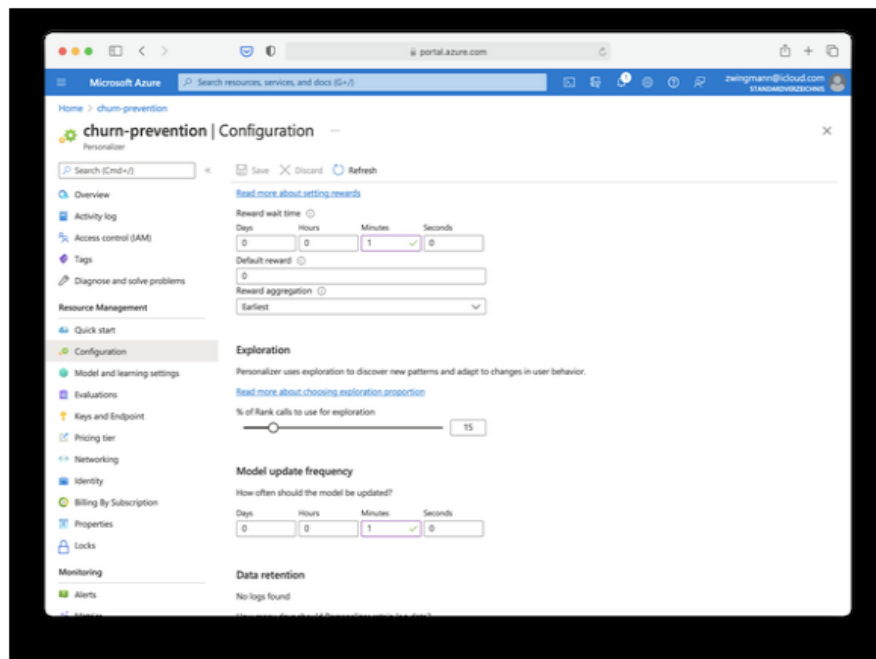
Gambar 8.2. Buat Formulir Personalizer



Gambar 8.3. Panduan Memulai Layanan Personalizer

Antarmukanya mungkin agak membingungkan pada awalnya, tetapi sebenarnya kita tidak perlu melakukan banyak hal di sini. Pertama, buka konfigurasi untuk meninjau pengaturan model Anda. Di sinilah Anda dapat menyesuaikan perilaku pembelajaran model Anda. Waktu yang dipilih bergantung pada data yang Anda gunakan. Biasanya tidak masalah untuk memulai dengan pengaturan standar dan melihat seberapa cepat model menangkap pola apa pun. Namun, untuk kasus penggunaan kita di sini, di mana kita akan mensimulasikan interaksi pengguna, kita ingin menjaga siklus tunggu dan pembaruan model tetap singkat.

Oleh karena itu, atur "Waktu tunggu reward" dan "Frekuensi pembaruan model" menjadi 1 menit seperti yang ditunjukkan pada Gambar 8.4. Kita juga dapat menurunkan persentase eksplorasi menjadi 15%, karena kita tahu bahwa preferensi pengguna tidak berubah selama eksperimen kita.

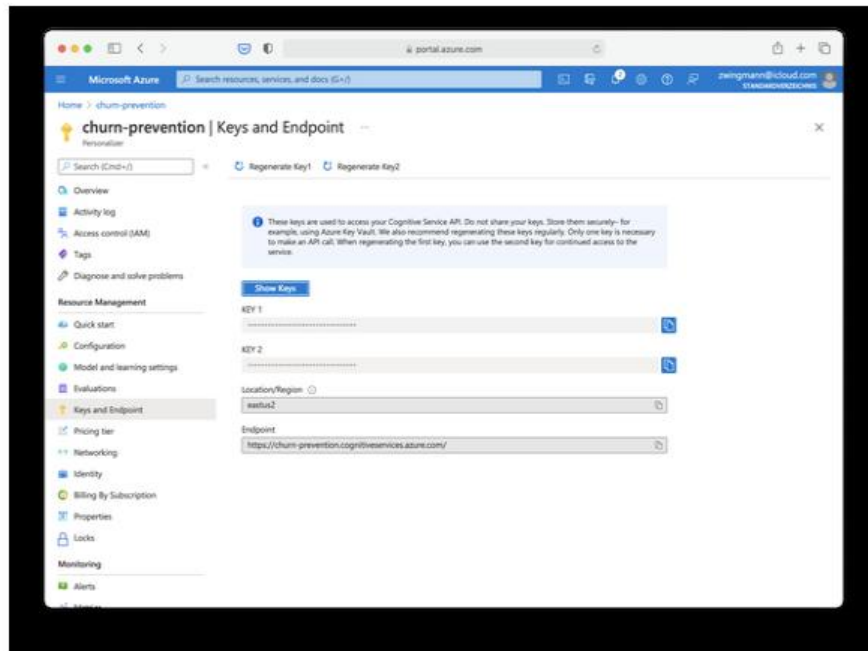


Gambar 8.4. Konfigurasi pembelajaran Personalizer.

Jangan lupa untuk menyimpan perubahan Anda menggunakan ikon cakram kecil di bagian atas dan verifikasi bahwa pengaturan telah diterapkan dengan menyegarkan halaman.

Selanjutnya, periksa perilaku pembelajaran pada tab. Ini adalah fitur yang sangat praktis untuk pengaturan di dunia nyata. Di sini Anda dapat menyesuaikan apakah layanan harus segera memberikan hasil yang diprediksi atau harus berjalan dalam apa yang disebut "mode magang" yang berarti menyajikan prediksi dasar dan mengumpulkan data hingga dasar ini dapat dilampaui. Anda dapat menemukan informasi lebih lanjut tentang perilaku pembelajaran layanan Personalizer dengan melihat dokumen dukungan Microsoft ini. Untuk pengaturan kita, kita akan membiarkan semuanya apa adanya dan tetap menggunakan "Kembalikan tindakan terbaik, pelajari online".

Untuk mengakses model dan mengirimkan permintaan ke sana, kita memerlukan URL model unik dan kunci akses yang sesuai. Anda dapat menemukan keduanya dengan membuka menu "Kunci dan titik akhir" di panel navigasi di sebelah kiri seperti yang ditunjukkan pada Gambar 8.5.

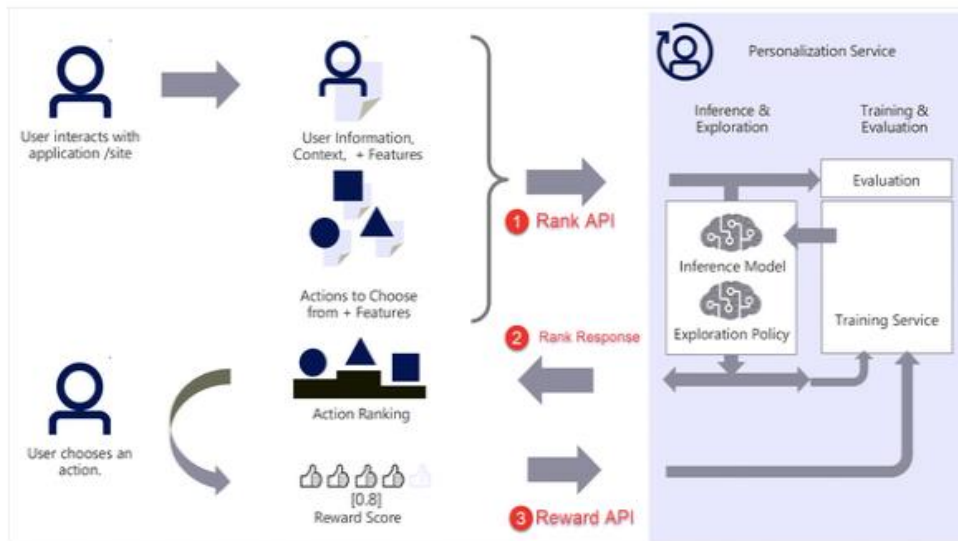


Gambar 8.5. Kunci akses dan URL ke layanan Personalizer

Layar ini menampilkan dua informasi penting:

Pertama, layar ini memberi Anda dua kunci akses yang perlu diautentikasi terhadap API, sehingga Anda dapat mengajukan permintaan peringkat dan reward. Kedua, Anda akan menemukan URL model yang diperlukan untuk mengaksesnya. Biarkan halaman ini terbuka untuk nanti, karena kita akan membutuhkan informasi ini saat berinteraksi dengan model.

Setelah kita menyiapkan model dan mendapatkan kredensial untuk menggunakannya, sekarang saatnya untuk meninjau bagaimana proses pembelajaran sebenarnya terjadi dan bagaimana model tersebut bekerja. Hal ini ditunjukkan secara konseptual pada Gambar 8.6:



Gambar 8.6. Alur konseptual layanan Personalizer. (Sumber)

Permintaan umum ke model menggabungkan panggilan peringkat dan hadiah. Keduanya disebut sebagai loop pembelajaran: Fungsi peringkat menyarankan sesuatu kepada pengguna dan memutuskan apakah akan mengeksploitasi, yaitu menampilkan tindakan terbaik berdasarkan data sebelumnya atau mengeksplorasi artinya memilih tindakan yang berbeda untuk melihat apakah hasil rekomendasi keseluruhan masih dapat ditingkatkan.

Setelah layanan menyarankan tindakan kepada pengguna, model mengharapkan hadiah. Ini bisa berupa apa saja, mulai dari tanda sederhana seperti 0 untuk "tidak berhasil" atau 1 untuk "berhasil", dalam kasus kami jika tawaran diterima atau tidak atau bisa juga berupa nilai kontinu yang pada dasarnya dapat mewakili apa saja, mulai dari kedalaman gulir artikel berita hingga keuntungan yang diperoleh dari perdagangan saham pembelajaran penguatan adalah konsep yang cukup universal yang dapat diterapkan dalam skenario apa pun di mana sistem mengikuti serangkaian aturan yang ditentukan.

Dalam kasus layanan Azure Personalizer, Anda dapat melihat dari Gambar 1 bahwa model mengharapkan dua hal untuk dikirim ke API peringkat: Informasi tentang pengguna (fitur konteks) dan informasi tentang tindakan yang dapat dipilih (1). Model akan melakukan tugasnya dan mengirimkan respons peringkat (2) yang menyarankan opsi yang menghasilkan peluang tertinggi untuk mendapatkan hadiah (jika dijalankan dalam mode eksploitasi). Pengguna akan memilih tindakan (atau tidak) dan hasilnya akan dikirim kembali ke model melalui API hadiah (3). Model akan menerima umpan balik ini dan memperbarui dirinya sendiri sesuai dengan kebijakan pembelajaran yang telah ditentukan sebelumnya.

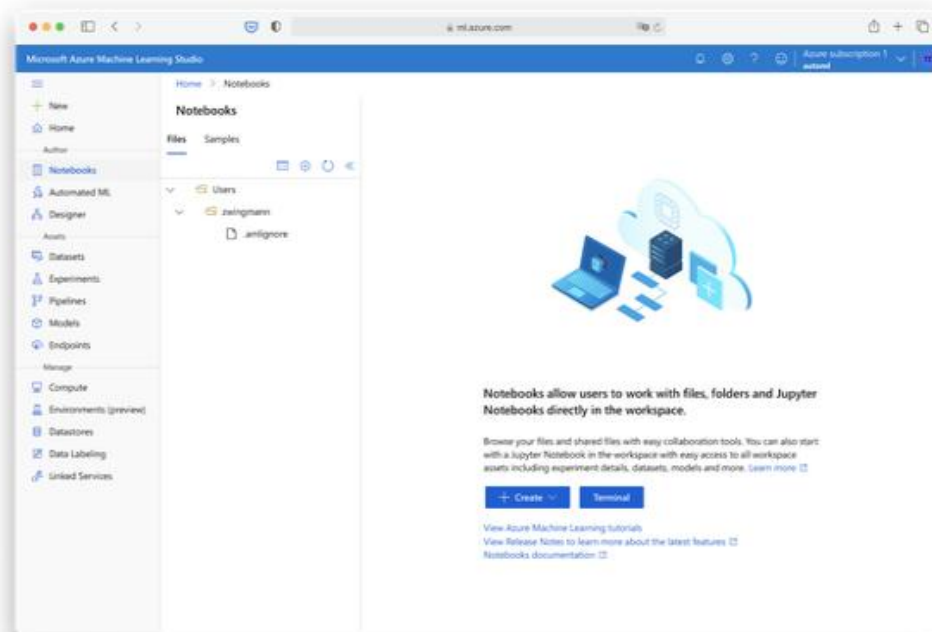
Berebel pengetahuan baru tentang Pembelajaran Penguatan ini, mari kita lanjutkan untuk mendapatkan pengalaman langsung tentang bagaimana loop pembelajaran terlihat dalam praktik dan bagaimana kita dapat memanfaatkannya dalam BI kita. Untuk mensimulasikan interaksi pengguna dengan penawaran kita, kita perlu menjalankan beberapa kode Python atau R di luar Power BI.



Menyiapkan Azure Notebook

Silakan lewati bagian ini jika Anda sudah menggunakan program di komputer untuk menjalankan kode Python atau R (seperti PyCharm, VSCode, Jupyter Notebook, RStudio, dll.) dan langsung lanjutkan ke "Simulasi interaksi pengguna".

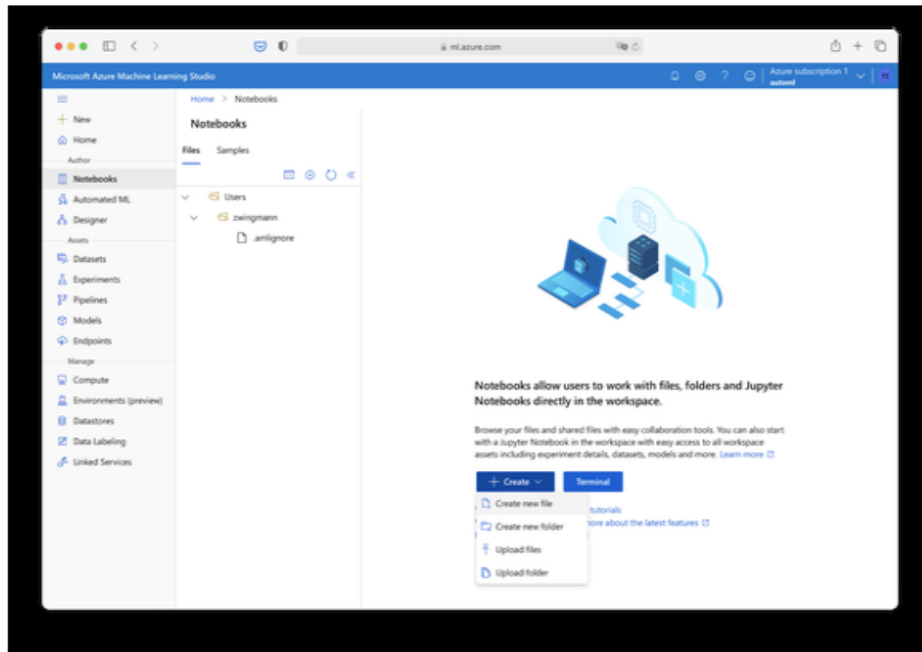
Jika belum, ikuti langkah-langkahnya dan saya akan menunjukkan cara cepat untuk mengeksekusi kode Python tanpa perlu menginstal perangkat lunak apa pun di komputer Anda. Buka ml.azure.com, pilih ruang kerja yang Anda inginkan, dan buka "Notebook" seperti yang ditunjukkan pada Gambar 8.7.



Gambar 8.7. Azure Notebook

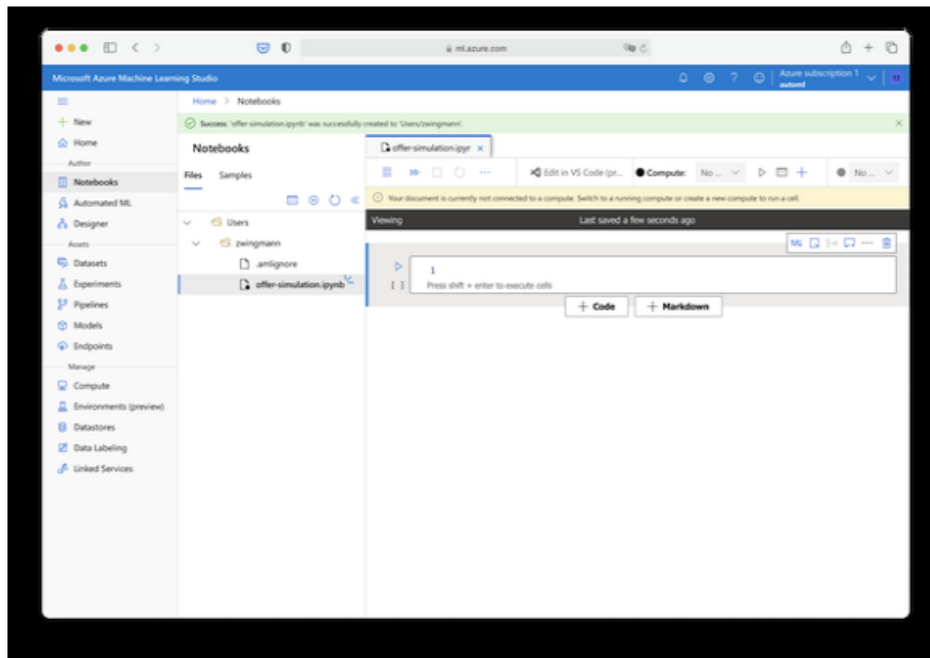
Pilih **Buat** → **Buat File Baru** seperti yang ditunjukkan pada Gambar 8.8. Beri nama notebook Anda, misalnya "offer-simulation.ipynb", dan pastikan jenis file diatur ke "Notebook (*.ipynb)". Lanjutkan dengan mengklik "Buat".





Gambar 8.8. Membuat Buku Catatan Azure Baru

Anda sekarang akan menemukan buku catatan baru di pohon berkas di sebelah kiri seperti yang ditunjukkan pada Gambar 8.9.

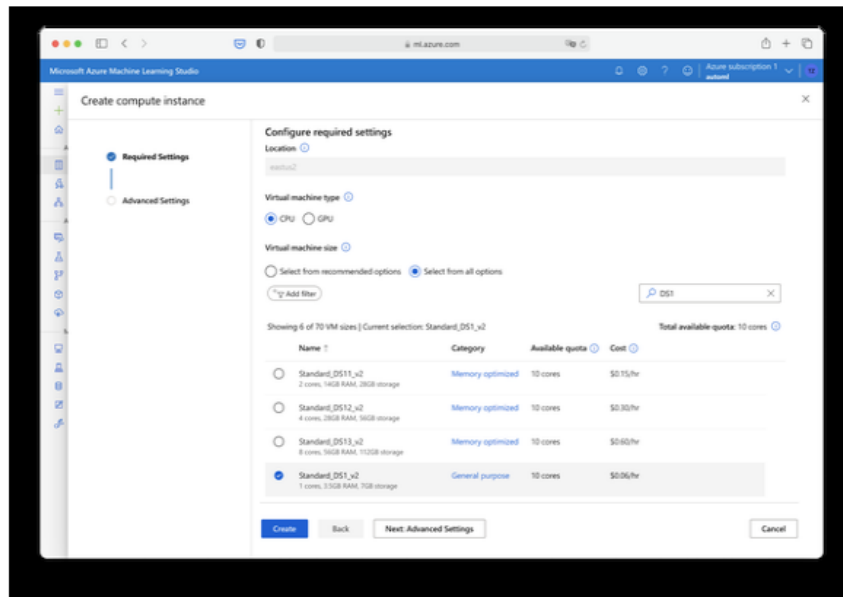


Gambar 8.9. Antarmuka Azure Notebook

Setelah kita menyiapkan editor kode yang memungkinkan kita berinteraksi dengan program, kita perlu menyediakan beberapa sumber daya komputasi tempat kode tersebut dieksekusi.

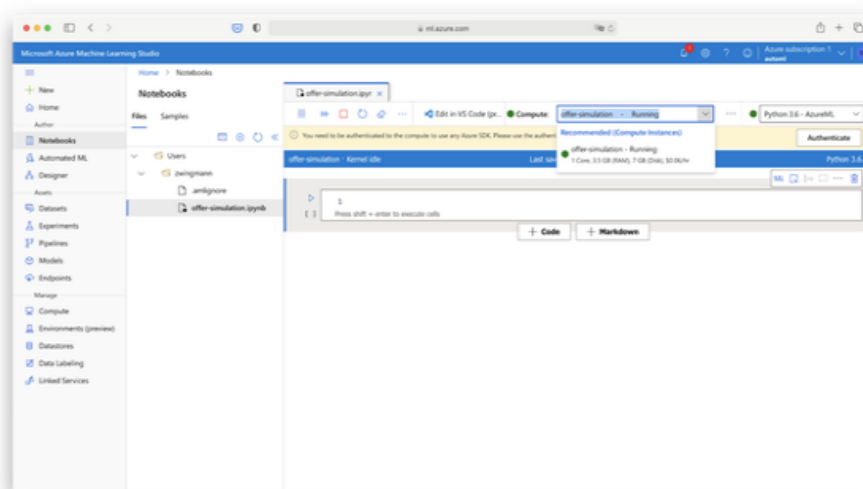


Tekan ikon plus seperti yang ditunjukkan pada Gambar 8.9 dan pilih "Buat Komputasi". Beri nama sumber daya komputasi Anda, misalnya "offer-simulation", pastikan jenis mesin diatur ke "CPU", dan pilih sumber daya termurah yang tersedia, yaitu jenis mesin "Standard_DS1_v2" seperti yang ditunjukkan pada Gambar 8.10. Anda sebenarnya dapat memilih jenis mesin apa pun di sini, pastikan saja harganya terjangkau karena kita tidak membutuhkan kinerja saat ini.



Gambar 8.10. Buat instans komputasi baru.

Klik "Buat" untuk menyediakan sumber daya komputasi baru Anda. Proses pembuatan sumber daya komputasi Anda mungkin memerlukan waktu beberapa menit. Setelah selesai, Anda akan melihat pesan sukses dan sumber daya baru akan ditampilkan seperti pada Gambar 8.11.



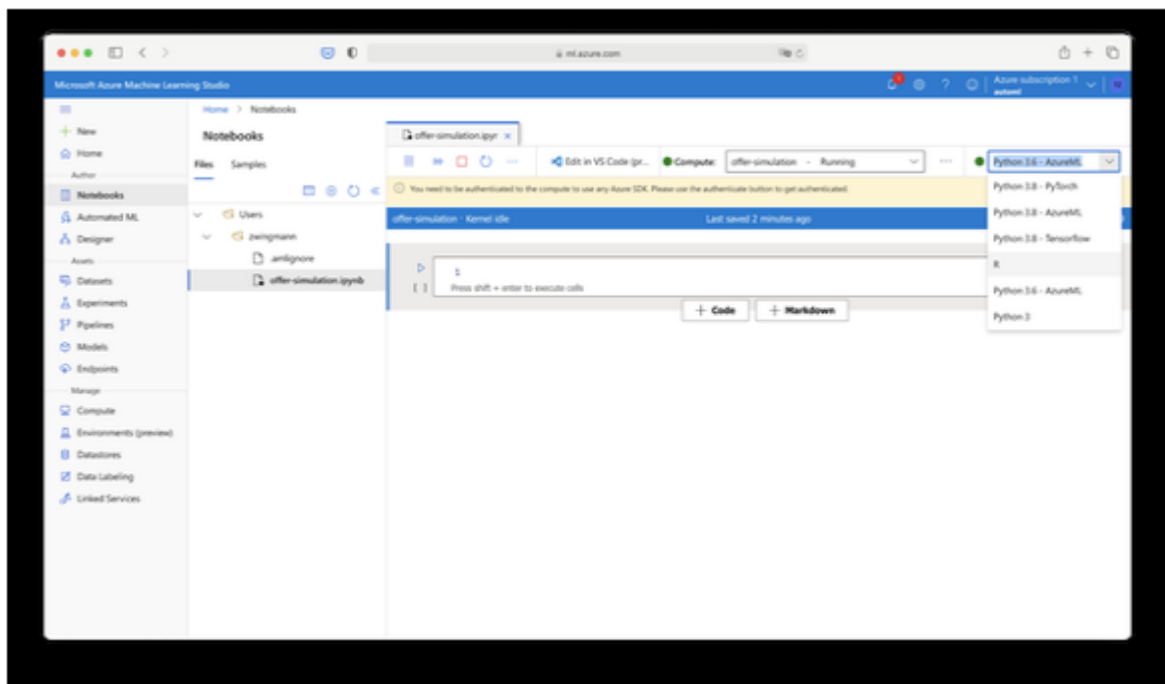
Gambar 8.11. Daftar instans komputasi



CATATAN

Setelah Anda membuat sumber daya komputasi dan sumber daya ini online, Anda akan dikenakan biaya sesuai harga yang tercantum dalam daftar sumber daya komputasi (misalnya, 0,6\$/jam). Jika Anda masih menggunakan kredit Azure gratis, biaya ini akan dipotong. Jika tidak, kartu kredit Anda akan dikenakan biaya sesuai ketentuan. Kami hanya membutuhkan instans komputasi ini selama kurang lebih satu jam untuk menjalankan simulasi. Setelah Anda menyelesaikan bab ini, pastikan untuk menghapus sumber daya Anda. Untuk melakukannya, buka Kelola → Komputasi dan pilih sumber daya komputasi Anda dari daftar. Pilih "hentikan" untuk menjeda layanan komputasi atau "hapus" untuk menghapusnya sepenuhnya.

Di pojok kanan atas, Anda dapat memilih kernel bahasa pemrograman yang ingin digunakan. Anda dapat memilih Python 3.6 atau R seperti yang ditunjukkan pada Gambar 8.12 untuk mengikuti contoh kode simulasi di langkah berikutnya.



Gambar 8.12. Beralih antar bahasa pemrograman di Azure Notebooks

Simulasi Interaksi Pengguna

Akhirnya, mari kita simulasikan beberapa interaksi pengguna dengan penawaran. Untuk ini, kita akan membuat dua asumsi: Pertama, kita mendefinisikan serangkaian kemungkinan penawaran yang terbatas dengan kompleksitas yang cukup sederhana. Dan kedua, kita berasumsi bahwa kita tahu segmen pengguna mana yang menerima penawaran yang mana. Dengan informasi ini, kita dapat membiarkan model kita mulai menyarankan



penawaran secara acak ke berbagai segmen pelanggan dan model tersebut akan mengambil pola yang mendasarinya melalui eksplorasi.

Kita juga menjaga kompleksitas segmen pengguna relatif rendah karena kita tidak ingin melebihi 50 ribu panggilan API untuk tingkat gratis kita. Ingat, kita membutuhkan satu panggilan untuk setiap fungsi peringkat dan hadiah, dan model penguatan yang umum membutuhkan beberapa ribu iterasi untuk mulai mempelajari pola yang bermakna. Oleh karena itu, kita akan menjaga kompleksitasnya tetap rendah. Namun, kita akan mematuhi satu prinsip: Model tidak mengetahui segmen pelanggan mana yang lebih menyukai jenis penawaran yang mana dan model tersebut harus mencari tahu sendiri.

Biasanya, model pembelajaran penguatan membutuhkan beberapa ribu iterasi untuk mempelajari pola yang bermakna. Itulah sebabnya kami memilih kompleksitas rendah di awal agar mendapatkan hasil pembelajaran yang baik dengan lebih sedikit putaran pembelajaran (dan Anda tetap berada dalam kredit gratis). Namun, jangan terbatas oleh contoh. Anda bebas menambahkan lebih banyak fitur konteks tentang segmen pelanggan, fitur tindakan, atau penawaran baru jika Anda ingin mencapai tujuan yang lebih tinggi.

Untuk simulasi ini, kami membutuhkan tiga komponen:

1. Informasi tentang jenis penawaran dan atribut penawaran yang tersedia
2. Informasi tentang pelanggan (fitur konteks) dan preferensi mereka yang sebenarnya (ground truth) untuk menghitung reward secara offline
3. Tempat penyimpanan data interaksi agar dapat dianalisis nanti melalui Power BI.

Anda dapat menemukan dua komponen pertama di repositori buku terlampir.

Tabel 8.1: Tabel Aturan dengan Preferensi Pelanggan

Warga lanjut usia	Kontrak	Penawaran Pilihan
Ya	Bulan ke Bulan	Diskon Tunai (Penawaran 1)
Ya	Setahun	Bulan Gratis (Penawaran 2)
Ya	Dua Tahun	Bulan Gratis (Penawaran 2)
Tidak	Bulan ke Bulan	Kinerja (Penawaran 3)
Tidak	Setahun	Kinerja (Penawaran 2)
Tidak	Dua Tahun	Diskon Tunai (Penawaran 1)

Komponen pertama adalah berkas bernama "offers.json" yang mencantumkan penawaran yang tersedia dan fitur-fitur terkait. Ada tiga penawaran yang tersedia untuk mencegah churn: Bulan Gratis, Upgrade Gratis, dan Diskon Tunai. Agar adil, kami berasumsi bahwa semua penawaran ini memiliki nilai yang sama, jadi tidak ada penawaran yang lebih baik dari yang lain. Ini murni masalah preferensi. Komponen kedua adalah berkas bernama "preferences.json". Berkas ini berisi informasi tentang segmen pelanggan mana yang lebih menyukai jenis penawaran tertentu.

Ini adalah pola-pola yang tidak diketahui oleh model kita dan kita ingin model tersebut mengambilnya secara otomatis. Seperti yang telah disebutkan sebelumnya, kita menjaga kompleksitas tetap rendah dengan hanya mempertimbangkan variabel "Warga Senior" dan

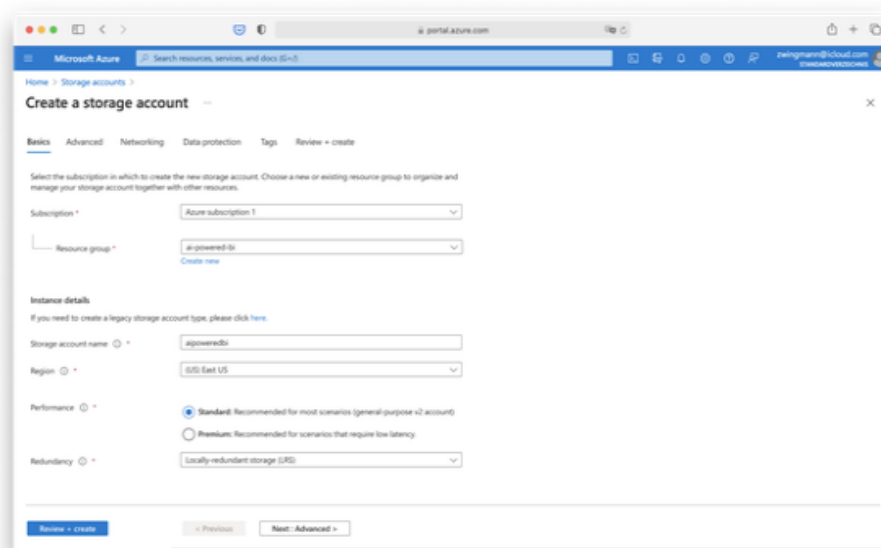


"Kontrak", tetapi tentu saja layanan Personalizer mampu menangani lebih banyak fitur. Bahkan, semakin banyak fitur yang kita sediakan terkait konteks penawaran, semakin baik - mengingat kita memiliki cukup waktu/kredit untuk belajar.

Dan terakhir, kita membutuhkan tempat untuk menyimpan hasil simulasi kita. Untuk menyederhanakannya (dan juga menjaga aksesibilitas melalui Power BI), kita akan menyimpan hasilnya sebagai file CSV biasa. Anda dapat menyimpan hasilnya secara offline di komputer Anda dan memuatnya nanti secara manual ke Power BI atau - jika Anda ingin pengalaman yang lebih nyata menulis output ke penyimpanan blob Azure. Percayalah, ini tidak serumit kedengarannya. Ikuti langkah-langkah singkat saya untuk menyiapkan penyimpanan blob Azure:

Menyiapkan Azure Blob Storage

Kunjungi portal.azure.com dan cari "storage accounts". Klik "Buat" dan buat akun penyimpanan baru di wilayah yang direkomendasikan, misalnya "AS Timur", lalu beri nama unik untuk akun ini.

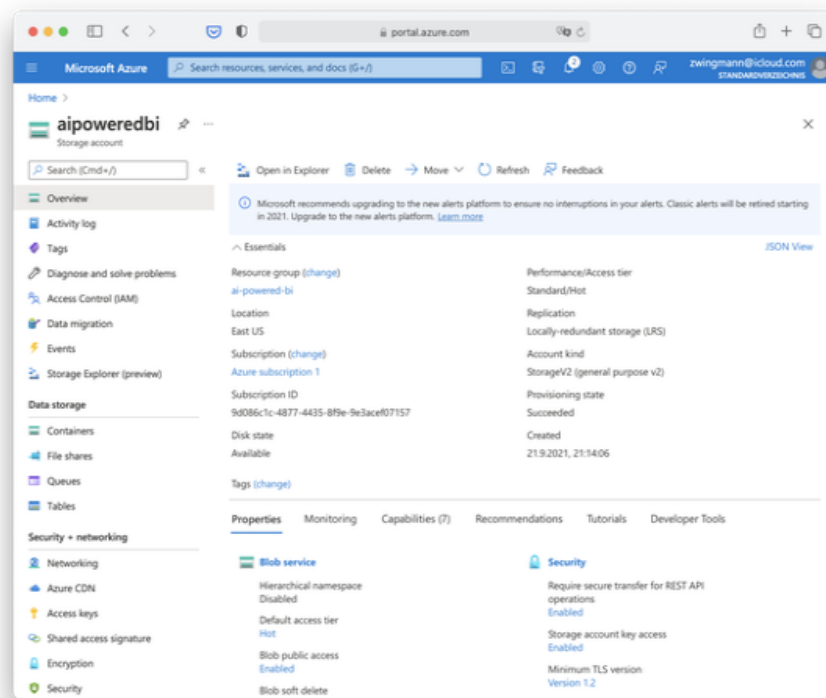


Gambar 8.13. Membuat akun penyimpanan

Pilih kinerja standar dan penyimpanan redundan lokal seperti yang ditunjukkan pada Gambar 8.13. Karena ini hanya data uji, kita tidak perlu membayar lebih untuk standar ketersediaan data yang lebih tinggi. Sekali lagi, kita hanya akan menyimpan data di sini untuk sementara, dan biayanya akan lebih dari cukup untuk menutupi anggaran uji coba gratis Anda.

Setelah penerapan selesai, klik "Buka sumber daya". Anda akan melihat antarmuka seperti yang ditunjukkan pada Gambar 8.14, yang sangat mirip dengan semua sumber daya Azure lain yang pernah kita temui sebelumnya.

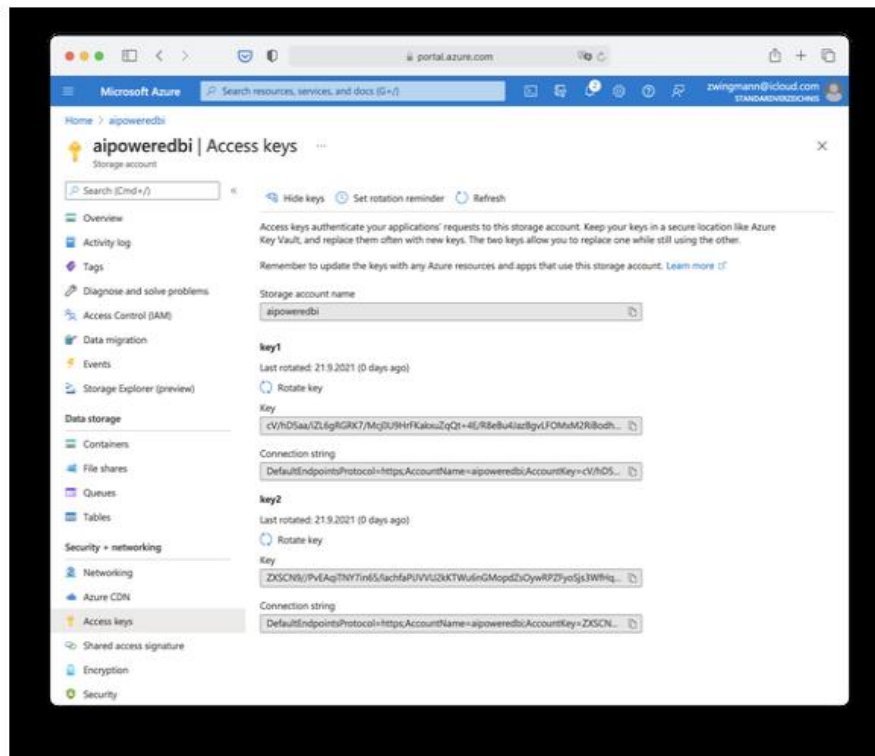




Gambar 8.14. Antarmuka akun penyimpanan Azure

Arahkan ke "Kunci Akses" seperti yang ditunjukkan pada Gambar 8.14 dan biarkan jendela ini terbuka karena kita akan membutuhkan kunci ini sebentar lagi.

Setelah kita memiliki akun penyimpanan dan kunci untuk memanipulasi data, kita perlu membuat kontainer, semacam folder file yang membantu kita mengelola file. Arahkan ke "Kontainer" di sisi kanan menu dan klik "+ Kontainer" di bagian atas. Atur tingkat akses ke "Kontainer" agar kita dapat mengakses file nanti melalui Power BI. Perlu diingat, jika Anda menangani data sensitif atau produksi, Anda tentu akan memilih "Pribadi" untuk memastikan otorisasi diperlukan sebelum data dapat diakses. Voilà, kontainer baru akan muncul dalam daftar dan siap untuk menyimpan beberapa file. Mari kita tinggalkan kontainer untuk saat ini dan lanjutkan menjalankan kode yang melakukan simulasi pengguna untuk kita.



Gambar 8.15. Kunci akses untuk akun penyimpanan Azure

Menjalankan Simulasi dengan Python atau R

Tergantung pada bahasa pemrograman pilihan Anda, lanjutkan dengan bagian meninjau kode Python atau R untuk tugas ini. Buka repositori buku dan temukan berkas "user-simulation.py" dan "user-simulation.R". Buka salah satu berkas ini di IDE favorit Anda. Jika Anda menggunakan Azure Notebooks, buka berkas tersebut di editor teks, pilih semua, lalu salin dan tempel isinya ke dalam notebook yang telah Anda buat sebelumnya.

Hasilnya akan terlihat seperti pada gambar tersebut. Pastikan kernel bahasa diatur ke R atau Python, tergantung pilihan Anda.

Namun, jangan jalankan kode ini dulu!

Ada beberapa pengeditan khusus yang harus Anda lakukan terlebih dahulu, tetapi pertamanya, mari kita bahas secara singkat apa yang dilakukan kode tersebut. Struktur kode sama untuk skrip versi Python dan R.

Kode ini terdiri dari lima bagian:

Bagian kode pertama disebut "Input Pengguna". Di area inilah Anda perlu menyesuaikan beberapa hal:

- OFFERS_FILE_PATH: Jalur berkas tempat berkas "offers.json" berada.
- PREFERENCES_FILE_PATH: Jalur berkas tempat berkas "preference.json" berada.
- USER_TABLE_FILE_PATH: Jalur berkas ke tabel pengguna yang churn
- PERSONALIZATION_BASE_URL: Titik akhir yang ditampilkan di "Kunci dan titik akhir" di Portal Azure



- KEY: Salah satu kunci akses dari halaman yang sama di portal Azure (kunci yang mana pun tidak masalah). Pastikan Anda mengganti kunci ini jika Anda memutuskan untuk membuatnya kembali.
- AZURE_CONNECTION_STRING: String koneksi Azure yang akan diotorisasi untuk penyimpanan blob, akan diabaikan jika kosong.
- AZURE_CONTAINER_NAME: Nama kontainer blob Azure, akan diabaikan jika kosong.

Bagian kode kedua disebut "Dependensi" dan memuat semua paket yang diperlukan untuk skrip ini. Pastikan Anda telah menginstal paket-paket ini di komputer Anda, jika tidak, eksekusi kode akan gagal. Paket `azure-storage-blob` khususnya hanya diperlukan jika Anda berencana mengunggah hasil ke penyimpanan blob Azure. Bagian ketiga disebut "Fungsi" dan pada dasarnya berisi empat fungsi utama:

- `add_event_id`: Fungsi ini menghasilkan ID unik untuk setiap panggilan peringkat. ID ini digunakan untuk mengidentifikasi informasi panggilan peringkat dan reward. Nilai ini dapat berasal dari proses bisnis seperti ID tampilan web atau ID transaksi.
- `add_action_features`: Fungsi ini menambahkan seluruh daftar penawaran ke objek JSON untuk dikirim ke permintaan Peringkat.
- `get_reward_from_preferences`: Fungsi ini dipanggil setelah API Peringkat dipanggil, untuk setiap iterasi. Fungsi ini membandingkan preferensi pengguna terhadap suatu penawaran, berdasarkan senioritas dan jenis kontrak mereka, dengan saran Personalizer untuk pengguna untuk filter tersebut. Jika rekomendasi benar, hadiah 1 akan dikembalikan, jika tidak, hadiahnya 0.
- `loop_through_user_table`: Fungsi ini berisi pekerjaan utama skrip. Fungsi ini akan mengulangi setiap baris tabel pengguna yang disediakan, meminta penawaran yang dipersonalisasi dari API, membandingkannya dengan preferensi pengguna yang sebenarnya, dan menghitung skor hadiah yang kemudian dikirim kembali ke layanan Personalizer. Semua hasil akan dikumpulkan dalam daftar yang disebut "hasil" dan ini akan dikembalikan oleh fungsi setelah selesai.

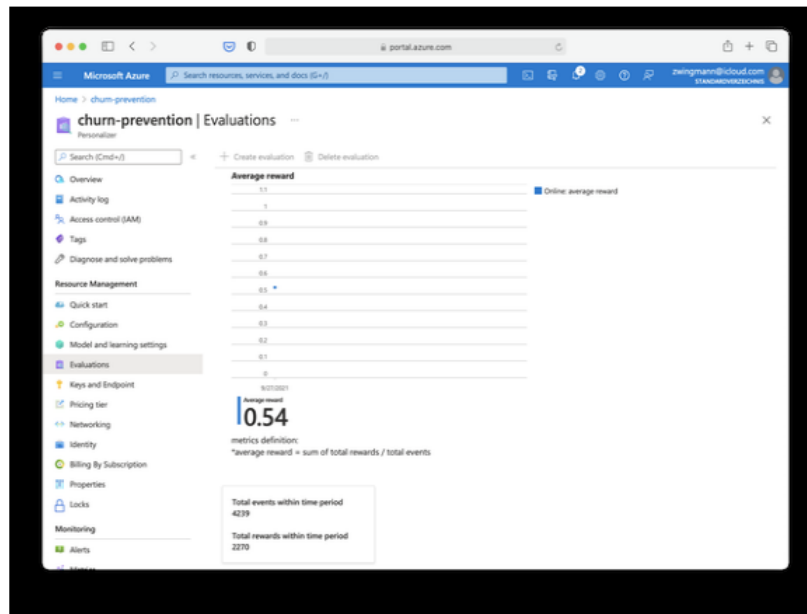
Bagian keempat yang disebut "Eksekusi" akan memanggil fungsi yang disebutkan di atas dan menjalankan alur kerja yang sebenarnya. Bagian kelima "Ekspor" akan menyimpan hasil dengan menyimpannya secara lokal ke file datar bernama "results.csv". Jika kredensial penyimpanan Azure diberikan, file hasil juga akan diunggah ke penyimpanan blob Azure.

Verifikasi bahwa Anda telah menyesuaikan semua input di bagian kode pertama. Sekarang, jalankan kodenya! Iterasi di seluruh tabel pelanggan akan memakan waktu sekitar 15 menit. Setelah proses selesai, Anda akan melihat file baru "results.csv" yang sedang dibuat di direktori kerja Anda. Ini pada dasarnya adalah dokumentasi eksperimen kami (dan akan menjadi file log dari skenario dunia nyata). Jika Anda menentukan string koneksi Azure dan nama kontainer yang valid di input, file ini akan otomatis diunggah ke kontainer Azure Anda. Jika terjadi kegagalan saat pengunggahan, pastikan Anda tidak menjalankan seluruh skrip lagi, tetapi hanya bagian terakhir di bagian 5 yang menangani pengunggahan file.



Evaluasi Performa Model di Portal Azure

Untuk melihat sekilas performa model, kita dapat memeriksa performa model di Portal Azure. Arahkan ke sumber daya Personalizer Anda dan pilih "Evaluasi" dari menu di sebelah kiri. Gulir ke bawah dan Anda akan melihat skor reward rata-rata seperti yang ditunjukkan pada Gambar 8.15.

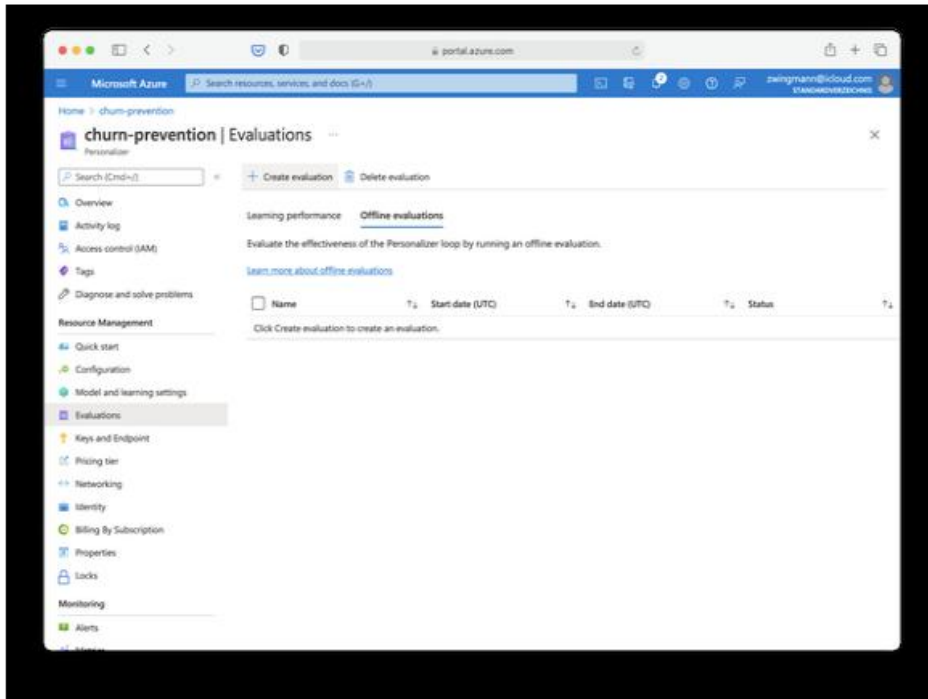


Gambar 8.16. Evaluasi Personalizer

Dalam kasus kami, Anda dapat menginterpretasikan angka ini sebagai persentase seberapa sering model memilih penawaran yang tepat untuk pelanggan. Pada saat ini, skor reward seharusnya masih relatif rendah.

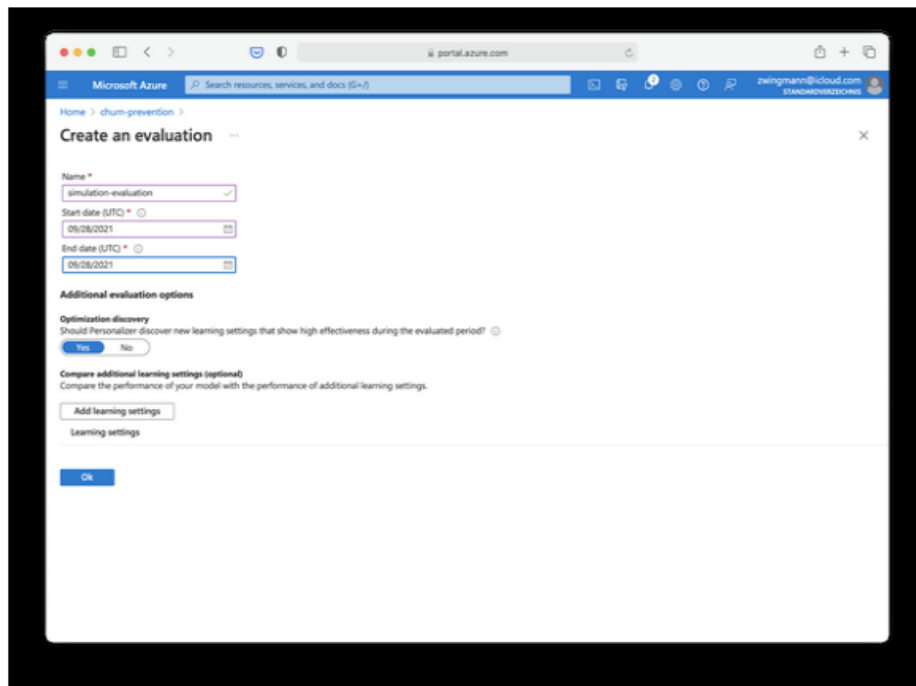
Ingatlah bahwa kita memulai dengan model naif yang membuat tebakan acak dan baru menemukan pola potensial setelah ribuan percobaan. Layanan personalisasi biasanya membutuhkan puluhan ribu contoh agar benar-benar berkinerja baik dalam skenario dunia nyata. Jika Anda menjalankan kode simulasi di atas 2-3 kali lebih banyak, Anda akan melihat angka ini meningkat. Jika tidak demikian, maka itu merupakan tanda yang jelas bahwa ada yang salah dengan perilaku pembelajaran model. Anda dapat memeriksa FAQ Microsoft ini untuk memecahkan masalah kesalahan dalam pembelajaran model.

Namun, ada satu hal yang masih dapat kita lakukan untuk meningkatkan kinerja tanpa harus menjalankan seluruh dataset lagi. Fitur ini disebut evaluasi offline. Ini adalah teknik yang memungkinkan Anda menggunakan data yang ada dan melatih ulang model Anda dengan berbagai kebijakan pembelajaran untuk mengetahui mana yang paling sesuai untuk data Anda. Mari kita coba ini untuk melihat apakah kita bisa mendapatkan model yang lebih baik daripada yang telah kita latih sejauh ini. Dari menu atas, pilih "Evaluasi offline" seperti yang ditunjukkan pada Gambar 8.16.



Gambar 8.17. Membuat evaluasi luring

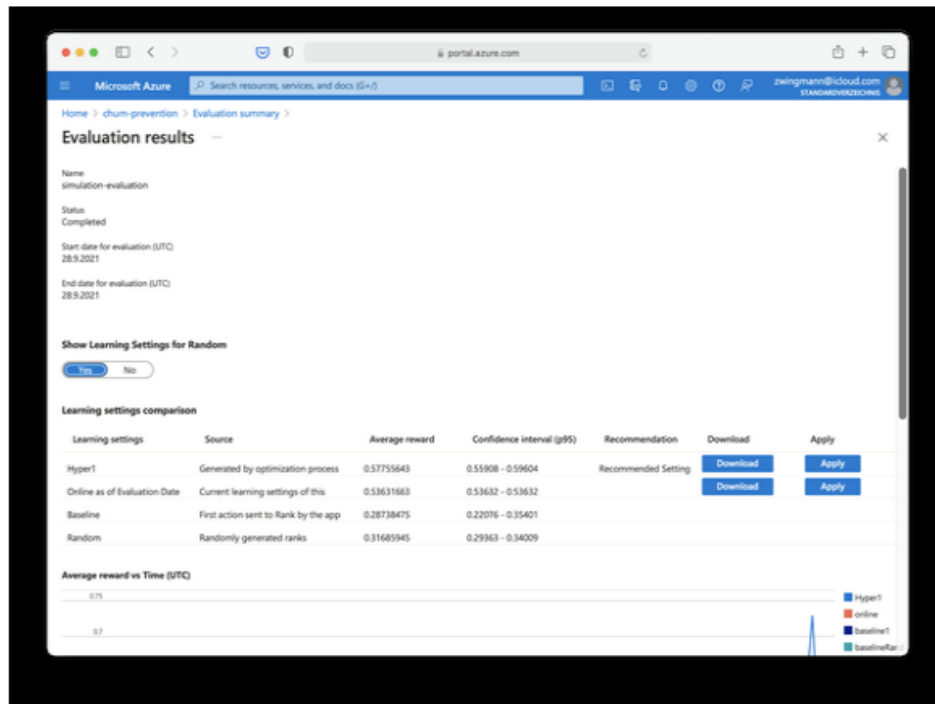
Periksa kembali apakah rentang tanggal sesuai dengan hari Anda menjalankan eksperimen simulasi dan biarkan semua pengaturan lainnya sebagaimana adanya. Klik "OK" untuk memulai evaluasi luring seperti yang ditunjukkan pada Gambar 8.17.



Gambar 8.18. Formulir Evaluasi Luring



Proses evaluasi memakan waktu beberapa menit. Setelah selesai, pilih evaluasi dari daftar untuk melihat detail lebih lanjut. Klik tautan "Bandingkan skor aplikasi Anda dengan pengaturan pembelajaran potensial lainnya" untuk melihat metrik evaluasi seperti yang ditunjukkan pada Gambar 8.18.



Gambar 8.19. Hasil evaluasi luring

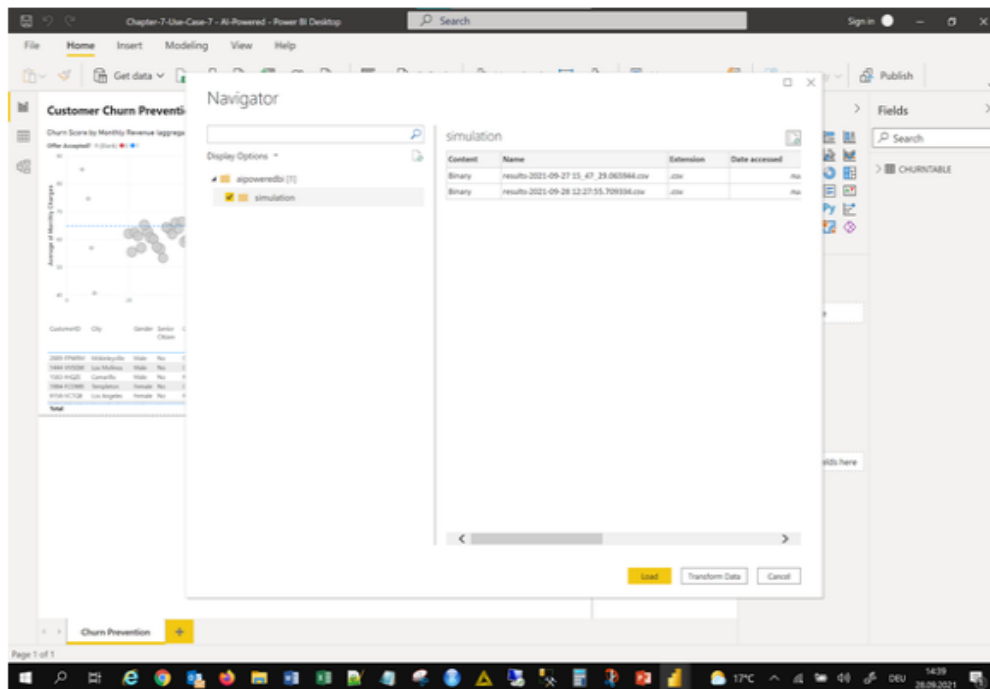
Seperti yang Anda lihat, dari evaluasi luring, kami mengetahui bahwa kami dapat meningkatkan model kami dari rata-rata reward 0,5363 ("Daring per Evaluasi") menjadi rata-rata reward 0,5775 jika kami menerapkan pengaturan pembelajaran "Hyper1". Ini mendekati skenario ideal 0,8 mengingat kami menyisakan 20% untuk eksplorasi. Pilih pengaturan pembelajaran baru dengan mengklik tombol "Terapkan" di sebelahnya. Ini akan melatih ulang model Anda dan menerapkannya dengan pengaturan pembelajaran baru.

Anda sekarang dapat kembali ke kode Anda dan menjalankan simulasi lagi untuk melihat bagaimana model tersebut meningkat. Namun sebelum melakukannya, kami ingin memasukkan rekomendasi AI kami ke dalam dasbor BI kami. Mari kita bahas ini di langkah terakhir dari kasus penggunaan ini.

Mengakses Hasil dari Power BI

Seperti yang mungkin Anda ingat, dalam skenario kasus penggunaan kami, kami telah diberikan dasbor BI yang menunjukkan prediksi churn pelanggan saat ini serta tingkat penerimaan keseluruhan dari penawaran standar, yang menunjukkan keberhasilan strategi retensi pelanggan kami. Yang ingin kami lakukan sekarang adalah menggabungkan informasi ini dengan wawasan tentang kinerja layanan Personalisasi kami dan melihat apakah inisiatif kami menghasilkan lebih banyak pelanggan yang menerima penawaran.





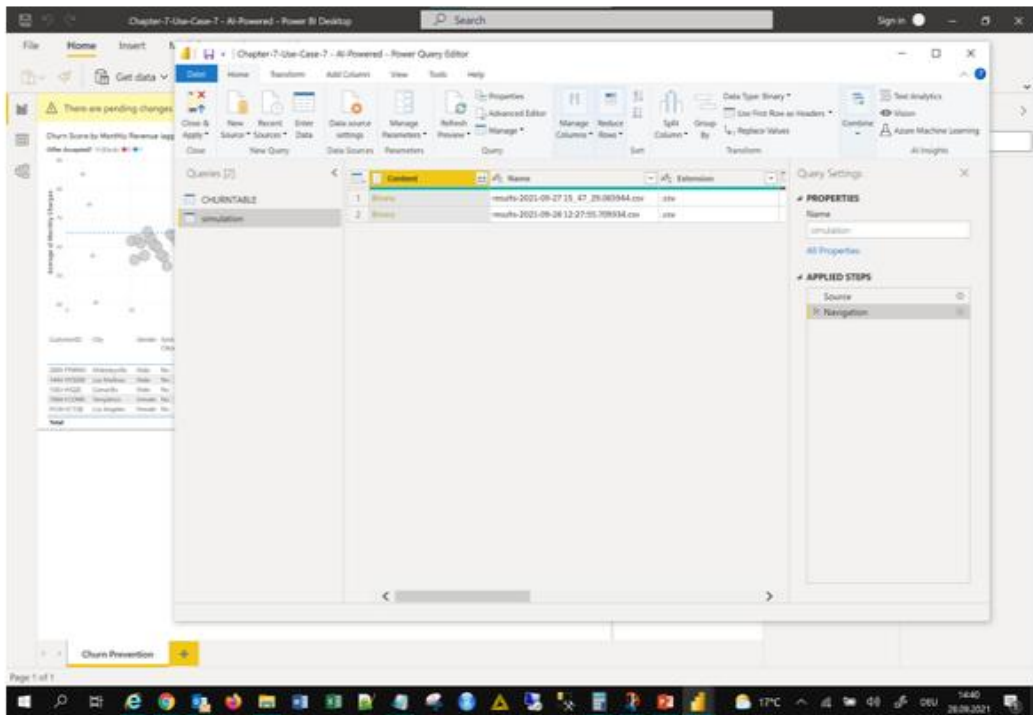
Gambar 8.20. Mengakses Azure Blob Storage dari Power BI

Untuk memulai, buka buku kerja Power BI "Use-Case-7.pbix" dari repositori buku dan buka model data. Anda akan melihat struktur tabel yang ada di sini untuk prediksi churn. Mari tambahkan hasil dari model kita dengan mengimpor file CSV hasil yang kita buat selama simulasi. Jika sebelumnya Anda memutuskan untuk menyimpan CSV secara lokal, klik "Dapatkan data" → impor Teks/CSV dan cari file CSV di komputer lokal Anda.

Jika Anda memutuskan untuk menyimpan file di penyimpanan Azure Blob, klik "Dapatkan data", pilih "Azure", lalu pilih "Penyimpanan Azure Blob". Berikan nama akun penyimpanan yang Anda berikan sebelumnya, dan pada langkah berikutnya, tempel kunci akses yang Anda dapatkan dari Portal Azure. Saat Anda mengonfirmasi, Anda akan melihat daftar berisi semua kontainer dan file di penyimpanan Blob Anda seperti yang ditunjukkan pada Gambar 8.19.

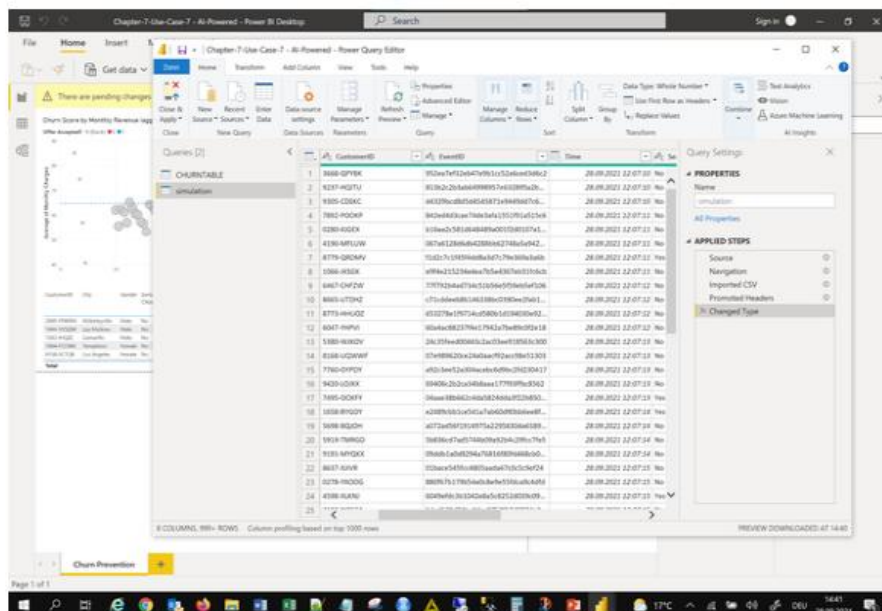
Pilih kontainer berisi data simulasi Anda. Namun, jangan muat dulu berkasnya! Sebagai gantinya, klik tombol "Transformasi data" seperti yang ditunjukkan pada Gambar 8.19. Ini akan memungkinkan kita menggunakan Power Query untuk mengekstrak konten sebenarnya dari berkas CSV. Jika tidak, kita hanya akan melihat metadata CSV seperti nama berkas, tanggal pembuatan, ukuran, dll. Di editor Power Query, klik tautan data biner seperti yang ditunjukkan pada Gambar 8.20.





Gambar 8.21. Mengakses file blob dengan Power Query Editor

Ini akan membawa Anda ke konten file CSV tempat konversi dari data biner ke tabel ditampilkan sebagai langkah pemrosesan di sisi kanan panel "Langkah yang Diterapkan" seperti yang terlihat pada Gambar 8.21.

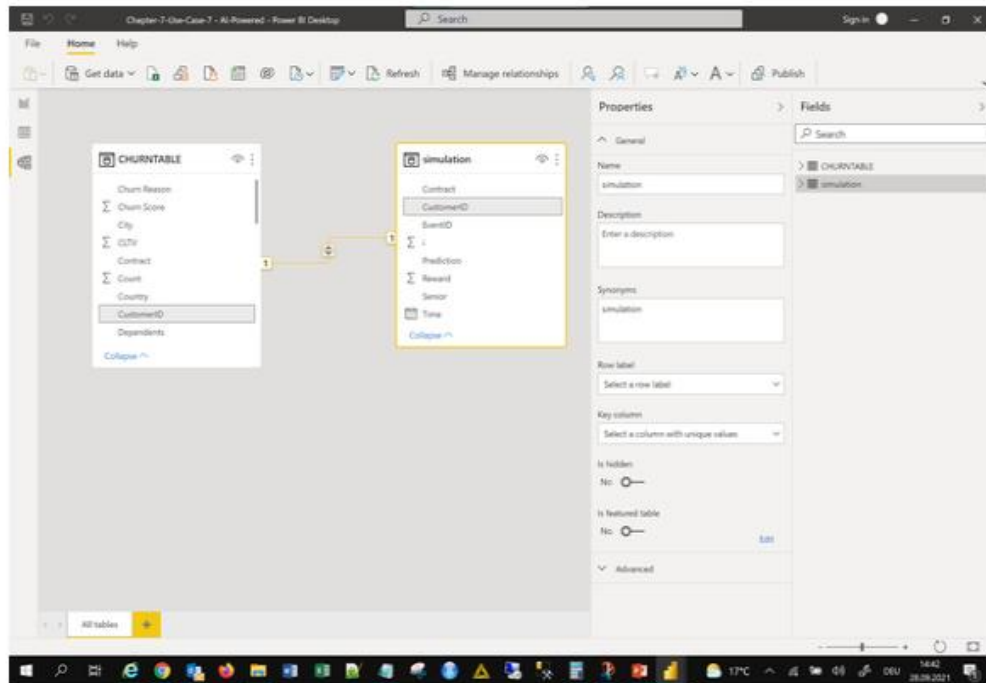


Gambar 8.22. Mengimpor data tabel dari penyimpanan blob

Konfirmasikan transformasi dengan memilih "Tutup & Terapkan" dari menu atas. Anda akan melihat bahwa isi berkas CSV baru telah ditambahkan ke model data Anda dan Power BI



seharusnya telah secara otomatis menghasilkan relasi antara model data "Churntable" dan "simulasi" berdasarkan kolom CustomerID seperti yang ditunjukkan pada Gambar 8.22.



Gambar 8.23. Relasi data di Power BI.

Buka halaman laporan BI dan mari kita lakukan beberapa modifikasi di sini untuk menyertakan rekomendasi berbasis AI baru kita. Lakukan perubahan berikut pada laporan:

Perbarui visual scatterplot:

- Kolom legenda: Ganti "Penawaran diterima?" dengan "Hadiah" dari model data simulasi
- Ubah skema warna pada format dan warna data: Atur "Kosong" menjadi abu-abu, "0" menjadi merah, dan "1" menjadi biru

Perbarui visual pengukur:

- Ganti "Rata-rata Penawaran diterima" dengan "Hadiah" dari tabel Simulasi. Terapkan perhitungan "Rata-rata" ke "Hadiah".

Perbarui visual metrik untuk pendapatan yang dipertahankan:

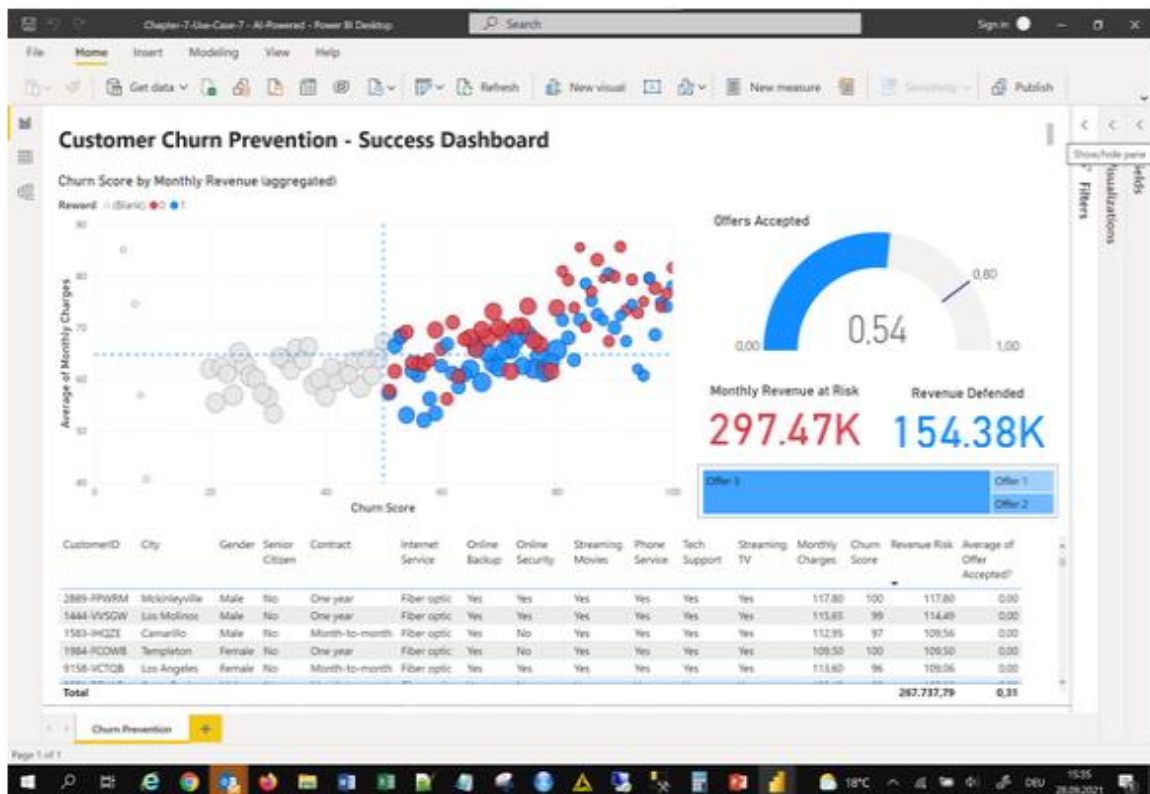
- Tambahkan Hadiah = 1 ke filter visual.
- Hapus "Penawaran Diterima" dari filter visual.

Perbarui visual peta pohon:

- Ganti "Penawaran yang ditampilkan" dengan "Prediksi" di kolom grup. Sesuaikan warna data di panel pemformatan agar "Kosong" berwarna abu-abu, dan semua penawaran berwarna biru.

Tampilan dasbor akhir kini mirip dengan yang ada pada Gambar 8.23:





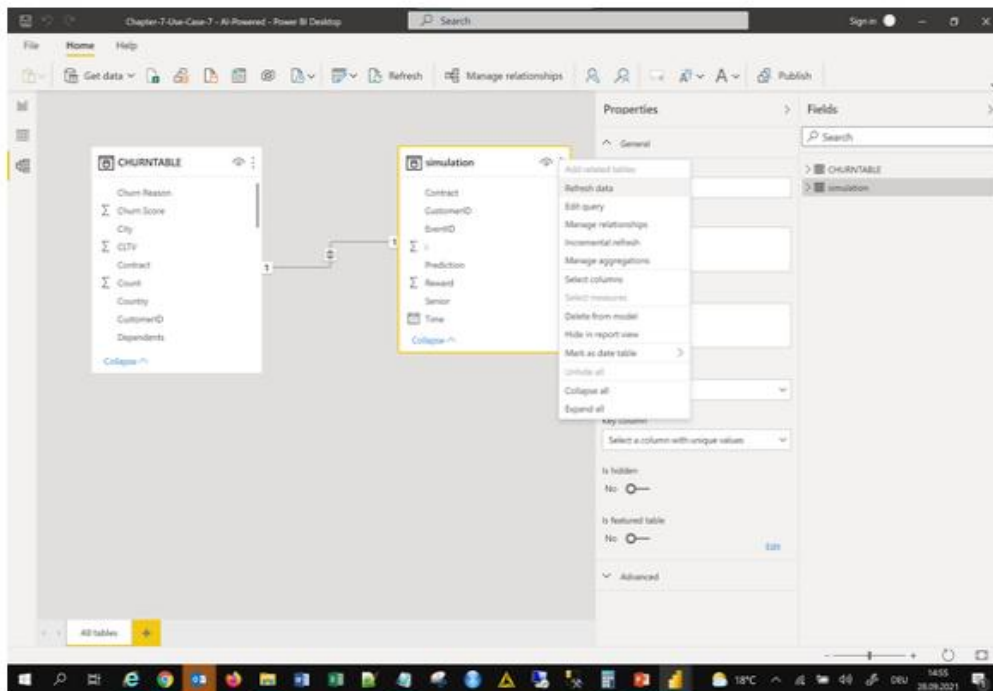
Gambar 8.24. Dasbor Rekomendasi Berbasis AI

Mari kita bandingkan dasbor ini secara singkat dengan dasbor asli yang kita lihat di awal. Kita akan melihat tiga perubahan besar di sini:

Pertama, indikatornya meningkat dari 0,31 menjadi 0,54, yang merupakan peningkatan akurasi sebesar 23 poin persentase dibandingkan dengan baseline kita saat ini. Ini berarti bahwa dengan menggunakan pendekatan berbasis AI, kita dapat meyakinkan lebih banyak pelanggan untuk menerima penawaran retensi kita. Pendapatan bulanan yang dipertahankan pun meningkat menjadi 154,38 ribu, yang merupakan nilai tambah hampir Rp 600 juta.

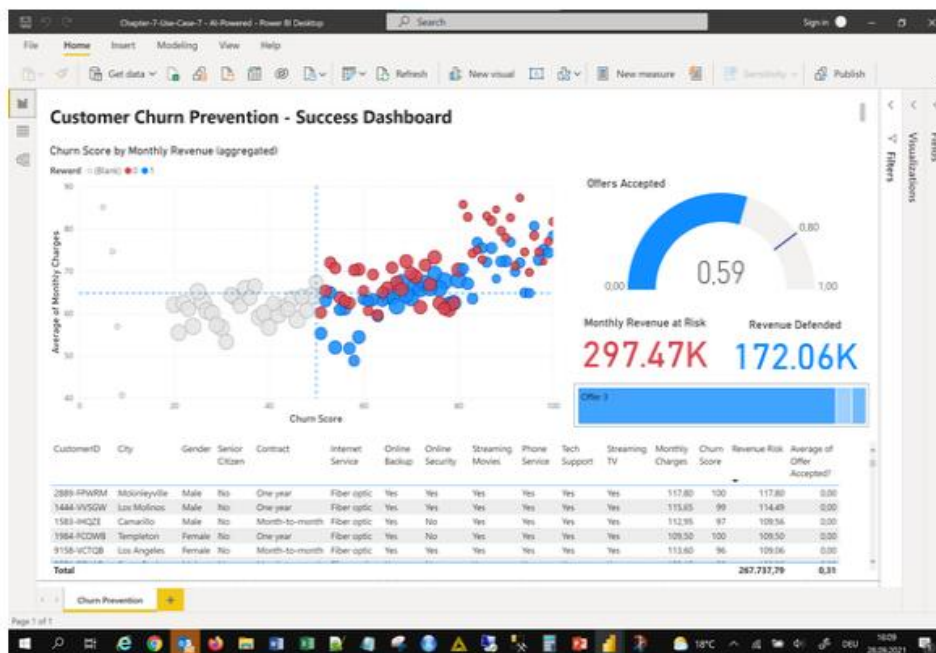
Dan kedua, kita dapat melihat bahwa proporsi penawaran meningkat. Penawaran 1 (Penawaran Tunai) yang sebelumnya paling populer kini sangat jarang terjadi. Sebagian besar pelanggan menerima Penawaran 3 yang berkinerja lebih tinggi. Bisakah kita meningkatkan KPI ini lebih jauh? Ingat bahwa kita telah menyempurnakan model kita melalui evaluasi offline sebelumnya. Namun sejauh ini, kita belum menerapkan model baru.

Kembali ke kode simulasi dan jalankan kembali tanpa membuat perubahan apa pun. Setelah 15 menit, kode akan melewati set data sekali sehingga layanan Personalizer dapat menggunakan pengaturan pembelajaran yang telah diperbarui. File results.csv di penyimpanan blob Azure Anda seharusnya telah diperbarui secara otomatis. Oleh karena itu, yang perlu kita lakukan di Power BI hanyalah membuka model data dan menekan "Refresh data" untuk tabel Simulasi seperti yang ditunjukkan pada Gambar 8.24.



Gambar 8.25. Penyegaran data

Setelah beberapa detik, Power BI telah mengambil data yang diperbarui dan jika kita kembali ke laporan, kita dapat melihat rekomendasi yang diperbarui dalam tindakan seperti yang Anda lihat pada Gambar 8.25.



Gambar 8.26. Dasbor bertnaga AI setelah pembaruan data

Setelah iterasi ulang dataset dengan pengaturan pembelajaran yang diperbarui, rekomendasi kami meningkat 5 poin persentase, mencapai akurasi 59%. Pendapatan yang dipertahankan meningkat sebesar Rp 176.800.000.

Untuk skenario dunia nyata, bayangkan Anda tidak melakukan iterasi pada dataset yang sama berulang kali, melainkan pada data pelanggan baru untuk setiap kuartal, bulan, atau hari, tergantung pada bisnis Anda. Dengan cara ini, Anda dapat memantau kinerja model rekomendasi Anda dan memutuskan apakah model tersebut cukup baik untuk dirilis ke tahap produksi atau tidak. Setidaknya, dasbor ini akan memberi Anda panduan yang solid untuk menilai nilai bisnis layanan rekomendasi berbasis AI dibandingkan dengan nilai dasar Anda saat ini.

Membersihkan Sumber Daya

Jika Anda menggunakan Azure Notebooks dengan mesin komputasi Azure untuk bab ini, pastikan untuk menghapus sumber daya komputasi setelah latihan agar tidak menagih tagihan. Pertimbangkan juga untuk menghapus file CSV dari penyimpanan blob jika Anda tidak membutuhkannya lagi.

8.2 RINGKASAN

Saya harap Anda dapat memahami dari kasus penggunaan ini bagaimana kita dapat memanfaatkan AI untuk mengubah prediksi menjadi tindakan dan bagaimana menemukan tindakan mana yang sesuai dalam konteks tertentu. Meskipun Azure Personalizer hanyalah salah satu dari banyak layanan, ide dan prinsip di balik mesin rekomendasi berbasis pembelajaran penguatan ini sangat mirip dan Anda akan mudah menggunakan layanan lain jika diperlukan. Selain itu, dengan penyimpanan Azure Blob, Anda mempelajari cara memantau pembaruan model secara real-time dan memasukkannya ke dalam BI Anda tanpa terlalu banyak hambatan. Di bab berikutnya, kita akan menangani data tidak terstruktur dan melihat bagaimana kita dapat memanfaatkan AI pada gambar, dokumen, dan file audio untuk menganalisis isinya dalam alur kerja BI reguler kita dan memadukannya dengan informasi yang ada dari bisnis.



BAB 9

MEMANFAATKAN DATA TAK TERSTRUKTUR DENGAN AI

9.1 KASUS PENGGUNAAN: MENDAPATKAN WAWASAN DARI DATA TEKS

Internet tidak hanya penuh dengan teks. Bahasa tulis adalah salah satu sumber data terbesar dan paling beragam yang telah dikumpulkan manusia. Dan bisnis pun tidak terkecuali. Pencipta data terbesar di sini adalah manusia, baik di dalam maupun di luar organisasi. Pelanggan misalnya berkat internet menjadi produsen konten dan berbagi pendapat mereka tentang produk atau layanan di seluruh web dan di berbagai saluran.

Dalam kasus penggunaan ini, kita akan menerapkan layanan AI yang akan membantu kita memahami data ini. Dalam masalah konkret yang dihadapi, kita akan menganalisis ulasan pengguna dalam skala besar dan mengomunikasikan wawasan utama melalui dasbor BI. Ayo!

Pernyataan Masalah

Kamar kecil, staf yang tidak ramah, dan sarapan yang buruk atau tidak? Sebagai pengelola hotel besar, manajemen kewalahan dengan beragamnya umpan balik pelanggan. Apakah memang ada masalah yang perlu ditangani manajemen atau hanya keluhan biasa yang harus diterima sebagai pengelola bisnis akomodasi? Kepala operasional telah mempekerjakan kami sebagai analis eksternal untuk mencari tahu bagaimana pelanggan menilai hotel dan bagaimana tren ini berkembang seiring waktu. Sebagai sumber data, mereka menyediakan contoh berkas teks berisi umpan balik pelanggan yang telah mereka kumpulkan dari waktu ke waktu melalui situs web mereka dan dari portal pemesanan. Karena musim baru akan segera dimulai, mereka menginginkan hasil yang lebih baik hari ini daripada besok, sehingga kecepatan jelas lebih diutamakan daripada akurasi. Manajemen ingin tahu apakah ada sesuatu yang salah secara fundamental dan memiliki kemungkinan untuk menyelidiki lebih dalam jika perlu.

Ikhtisar Solusi

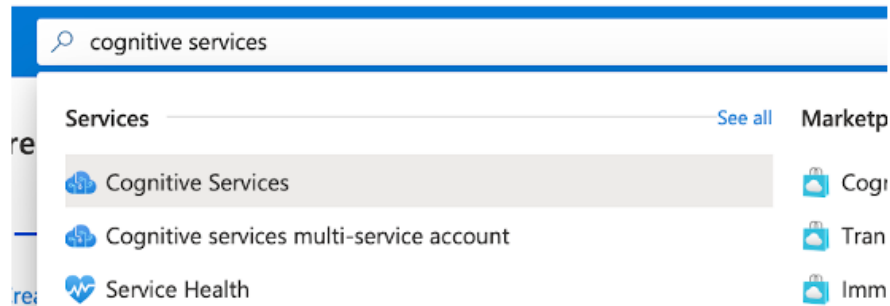
Untuk menganalisis banyak berkas teks secara otomatis, kami akan menggunakan layanan AI Pemrosesan Bahasa Alami (NLP). Tujuan kami adalah mengekstrak informasi tentang apakah opini dalam teks-teks ini negatif atau positif, yang juga disebut analisis sentimen. Selain itu, kami ingin mengekstrak kata kunci agar kami dapat menghubungkannya kembali dengan frasa kata yang mengandung emosi positif atau negatif. Presentasi harus dalam bentuk dasbor BI, dalam kasus kita Power BI.

Untuk menangani tugas ini, pertama-tama kita akan menyiapkan layanan AI di Azure yang disebut Cognitive Services Text Analytics. Kemudian, kita akan menyiapkan alur pemrosesan data kecil yang mengambil berkas teks mentah kita sebagai input dan membuat berkas csv datar sebagai output yang berisi semua hasil yang diperlukan untuk melakukan analisis di Power BI (atau alat BI pilihan Anda).

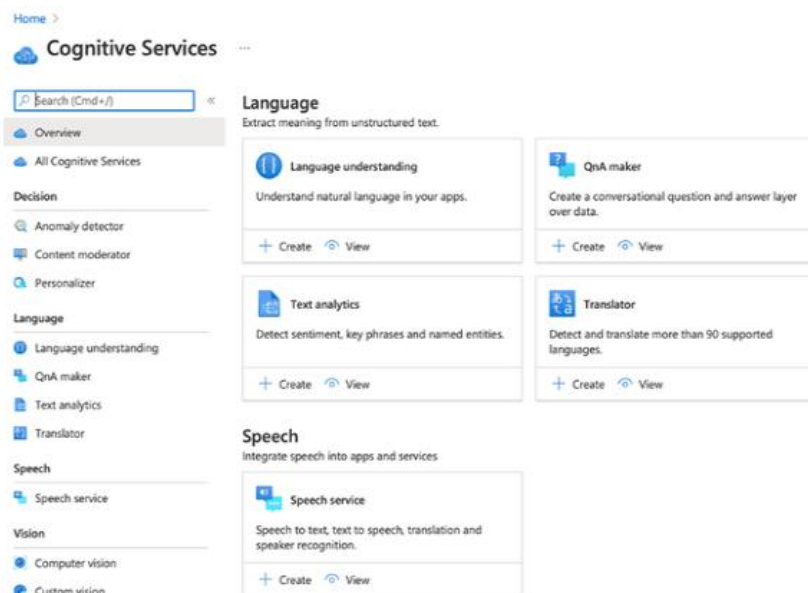


Menyiapkan Layanan AI

Pertama-tama, kita perlu mengaktifkan Cognitive Services Text Analytics di langganan Azure kita. Langkah ini sangat mudah dan sangat mirip dengan yang kita lakukan sebelumnya untuk layanan AI lainnya. Untuk mengaktifkan Cognitive Services Text Analytics, buka Portal Azure Anda dan cari Cognitive Services di bilah pencarian seperti yang ditunjukkan pada Gambar di bawah ini.



Pilih Layanan Kognitif dari daftar saran dan buka halaman sumber daya yang sesuai. Di halaman ini, Anda dapat mengaktifkan Layanan Kognitif untuk semua jenis data dan kasus penggunaan. Gulir ke bawah hingga Anda melihat bagian "Bahasa" seperti yang ditunjukkan pada Gambar di bawah ini.



Dari sini, Anda dapat menerapkan sumber daya analitik Teks baru dengan mengeklik tombol "+ Buat". Pada formulir berikut, yang seharusnya sudah cukup Anda kenal, pilih langganan Azure Anda, grup sumber daya, dan beri nama layanan analitik Teks Anda. Pastikan juga Anda memilih Tingkat Gratis (F0) yang memungkinkan 5.000 transaksi per 30 hari. Satu transaksi



adalah rekaman teks yang setara dengan jumlah unit 1.000 karakter dalam dokumen yang diberikan sebagai input ke API Analitik Teks jumlah tersebut lebih dari cukup untuk kasus penggunaan kita di sini. Anda dapat melihat contoh formulir yang telah diisi pada Gambar di bawah ini.

Home > Cognitive Services > Select additional features >

Create

Text Analytics

Basics Tags Review + create

Unlock insights from unstructured text using advanced natural language processing. Use sentiment analysis to find out what customers think of your brand. Find topic-relevant phrases using key phrase extraction and identify the language of the text with language detection. Detect and categorize entities in your text with named entity recognition. [Learn more](#)

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *

Resource group * [Create new](#)

Instance details

Region *

Name *

Pricing tier ([Learn More](#)) *

Responsible AI Notice

Microsoft provides technical documentation regarding the appropriate operation applicable to this Cognitive Service that is made available by Microsoft. Customer acknowledges and agrees that they have reviewed this documentation and will use this service in accordance with it.

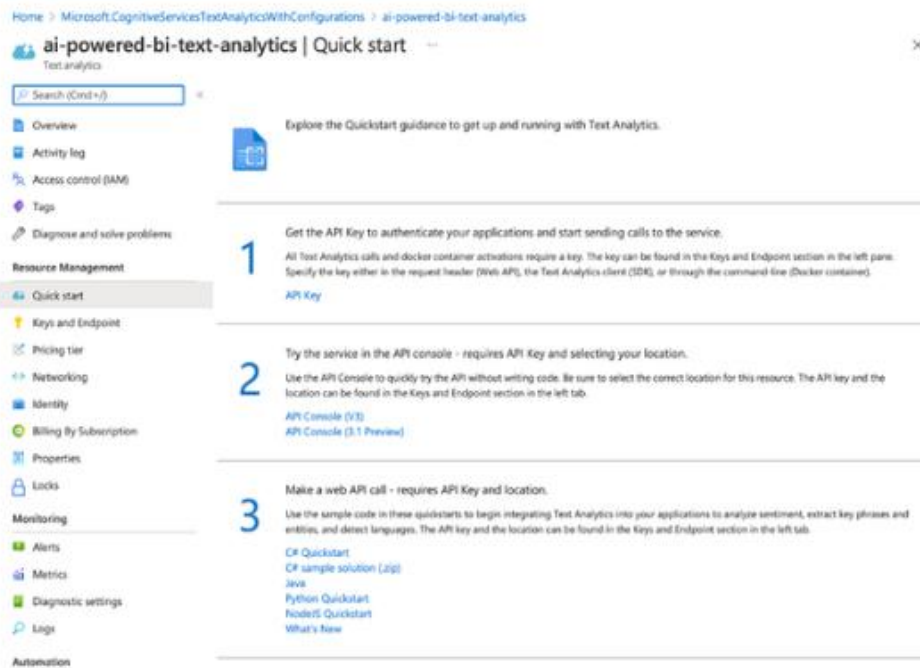
[Responsible Use of AI documentation for Text Analytics for Health](#)
[Responsible Use of AI documentation for Text Analytics PII](#)

I certify that I have reviewed and acknowledge the terms in the Responsible AI Notice. *

[Review + create](#) [Next : Tags >](#)

Pilih "Tinjau + buat" lalu "Buat" setelah proses peninjauan otomatis selesai. Proses penerapan akan memakan waktu beberapa menit, tetapi setelah itu Anda akan melihat notifikasi yang menyatakan bahwa sumber daya baru sudah siap.

Arahkan ke sumber daya baru dengan mengklik notifikasi tersebut, atau jika Anda meninggalkan halaman cukup cari nama sumber daya yang Anda berikan di formulir di bilah pencarian Azure. Dalam kedua kasus tersebut, Anda akan disambut dengan panduan "Mulai Cepat" sumber daya seperti yang ditunjukkan pada Gambar di bawah ini.



Ada banyak hal yang bisa kita lakukan di sini, tetapi untuk saat ini kita hanya ingin mengetahui dua hal:

Pertama, bagaimana kita mengautentikasi API? Dan kedua, di mana kita bisa mengetahui cara kerja API, yaitu kode apa yang perlu kita tulis (atau lebih tepatnya: salin dan tempel)?

Untuk menjawab bagian pertama pertanyaan ini, Anda dapat mengklik menu "Kunci dan Titik Akhir" di panel menu sebelah kiri. Di sini Anda akan menemukan dua kunci, sama seperti yang Anda lakukan pada contoh Azure Personalize di bab sebelumnya. Anda hanya perlu satu kunci ini untuk memanggil API. Biarkan jendela ini terbuka di tab baru karena kita akan segera membutuhkan kunci sumber daya ini.

Untuk mengetahui cara kerja API, Anda dapat menjelajahi sumber daya di "Langkah 3" dari panduan mulai cepat. Jika Anda mengklik "Python Quickstart", Anda akan menemukan dokumen yang menjelaskan berbagai skenario di mana Anda dapat menggunakan Text Analytics API dan kode apa yang akan Anda gunakan untuk ini. Ini mungkin tampak banyak untuk saat ini, tetapi jangan khawatir. Anda tidak akan membutuhkan semua informasi ini. Untuk saat ini, cukuplah untuk mengakui bahwa sebagian besar kode yang kita tulis nanti pada langkah persiapan data sebenarnya dapat diambil dari halaman dokumentasi ini dan diadaptasi sesuai kebutuhan Anda.

Dengan menelusuri dokumentasi, kita sebenarnya dapat menemukan beberapa informasi bermanfaat lainnya mengenai layanan AI baru kita. Buka "Konsep" dan "Batas Data", dan Anda akan menemukan tabel seperti yang ditunjukkan pada Gambar di bawah ini.



The maximum number of documents you can send in a single request will depend on the API version and feature you're using, which is described in the table below.

Version 3 | Version 2

The following limits are for the current v3 API. Exceeding the limits below will generate an HTTP 400 error code.

Feature	Max Documents Per Request
Language Detection	1000
Sentiment Analysis	10
Opinion Mining	10
Key Phrase Extraction	10
Named Entity Recognition	5
Entity Linking	5
Text Analytics for health	10 for the web-based API, 1000 for the container.
Analyze endpoint	25 for all operations.

Jika kita melihat tabel ini dan menemukan baris "Analisis Sentimen", kita dapat melihat bahwa API menerima 10 dokumen per permintaan. Artinya, kita dapat mengirim hingga 10 berkas teks sekaligus ke API, yang akan sangat mempercepat proses inferensi kita, dibandingkan dengan mengirim berkas teks satu per satu. Kita akan membahas hal ini lagi saat pemrosesan data. Sekarang, semuanya telah disiapkan untuk layanan AI. Mari kita lanjutkan untuk membangun alur data mini kita dan menjalankan beberapa inferensi pada data umpan balik pelanggan.

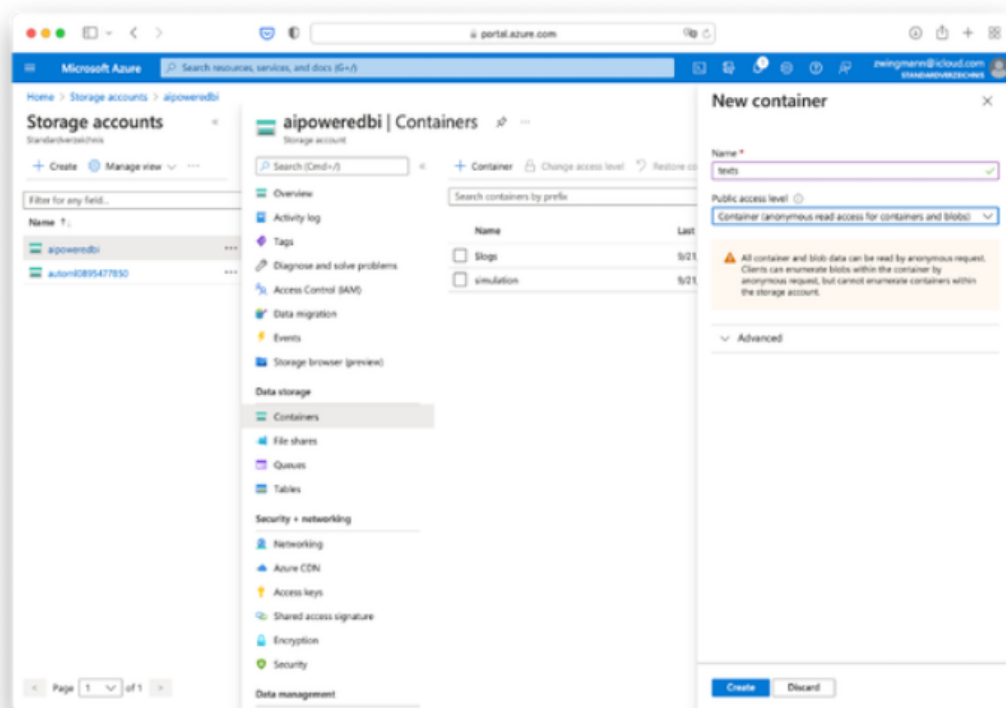
Menyiapkan Alur Data

Untuk mendapatkan dasbor BI yang dirancang dengan indah dari berkas teks mentah, kita harus menyelesaikan beberapa langkah perantara. Dalam lingkungan yang produktif, ini disebut proses ETL atau Ekstrak-Transformasi-Muat. Untuk prototipe kita, kita tidak akan menyebutnya proses ETL yang lengkap, tetapi pada dasarnya kita memang melakukan hal itu. Mari kita sebut saja ini pekerjaan ETL mini kita. Kita akan menulis skrip singkat yang akan menangani bagian-bagian penting dari proses ETL, yaitu:

- Langkah 1: Membaca berkas teks biasa dari berkas.
- Langkah 2: Mengirim isi berkas ke API layanan AI.
- Langkah 3: Mengumpulkan hasil, mentransformasi, dan menyimpannya dalam objek data terstruktur.
- Langkah 4: Mengekspor berkas sebagai tabel datar agar mudah digunakan oleh perangkat lunak BI kita.



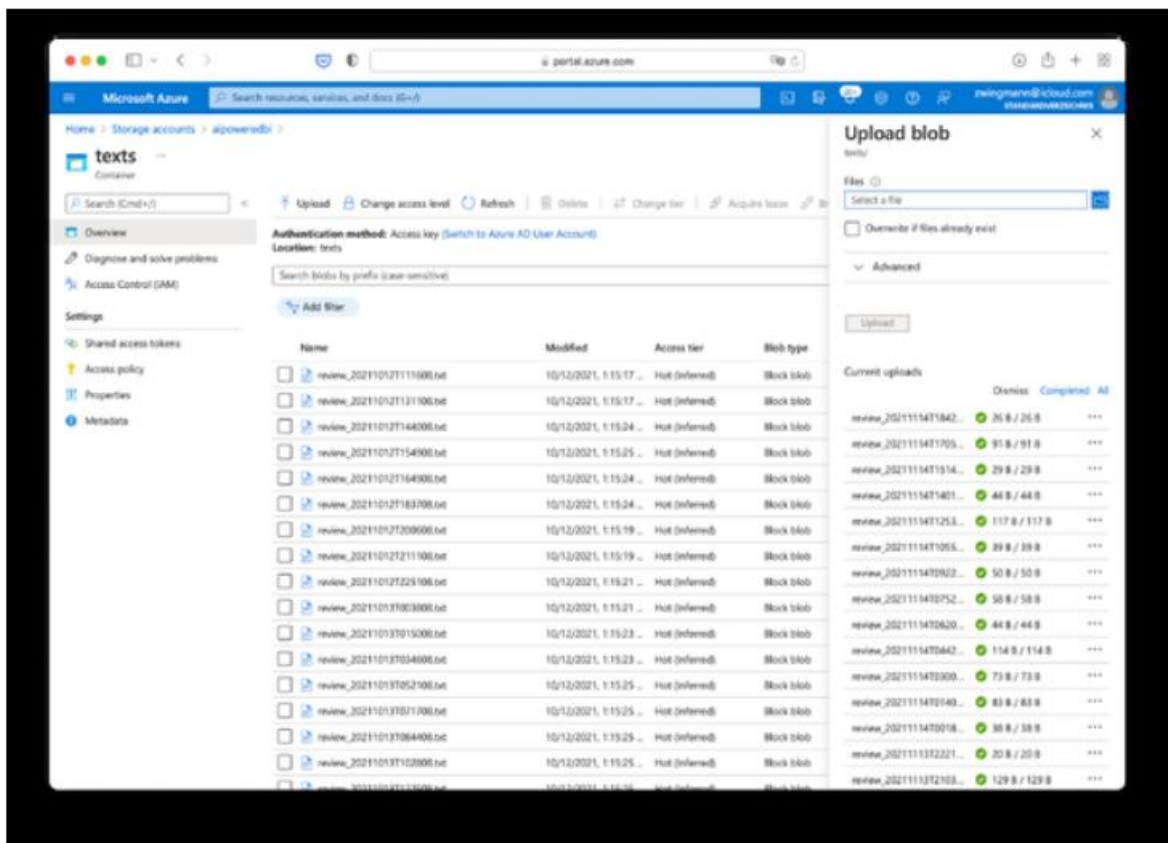
Untuk langkah 1, kita perlu mengambil berkas teks dan menyimpannya di tempat yang dapat diakses oleh skrip kita. Kita akan menggunakan penyimpanan Azure Blob untuk ini. Buka repositori buku dan unduh berkas "reviews.zip". Di portal Azure Anda, buka penyimpanan, pilih akun penyimpanan yang telah Anda siapkan sebelumnya, dan buat dua kontainer baru: "texts" dan "tables" seperti yang ditunjukkan pada Gambar di bawah ini. Kontainer "texts" akan menyimpan berkas teks input mentah dan kontainer "tables" akan menerima keluaran tabel dari proses ETL mini kita.



Setelah Anda membuat wadah "teks", unggah semua berkas teks individual di sini. Jangan unggah berkas zip-nya, tetapi isinya. Wadah Anda sekarang akan terlihat seperti Gambar di bawah.

Data input kita sekarang siap digunakan oleh skrip. Anda dapat mengosongkan "tabel" kontainer lainnya untuk saat ini. Kita akan mengisinya melalui skrip kita, yang akan kita lihat selanjutnya. Unduh berkas "ETL_For_Text_Analytics.ipynb" dari repositori buku. Buku catatan ini berisi semua kode yang Anda perlukan untuk menjalankan proses ETL mini. Jika Anda memiliki IDE lokal seperti Jupyter Notebook yang terpasang di komputer, Anda dapat langsung menggunakannya. Jika tidak, saya sarankan menggunakan Azure Notebooks sebagai IDE yang dihosting karena merupakan bagian dari langganan Azure Anda. Saya akan memandu Anda melalui contoh penggunaan Azure Notebooks, tetapi langkah yang sama berlaku ketika Anda menggunakan IDE lokal.





CATATAN

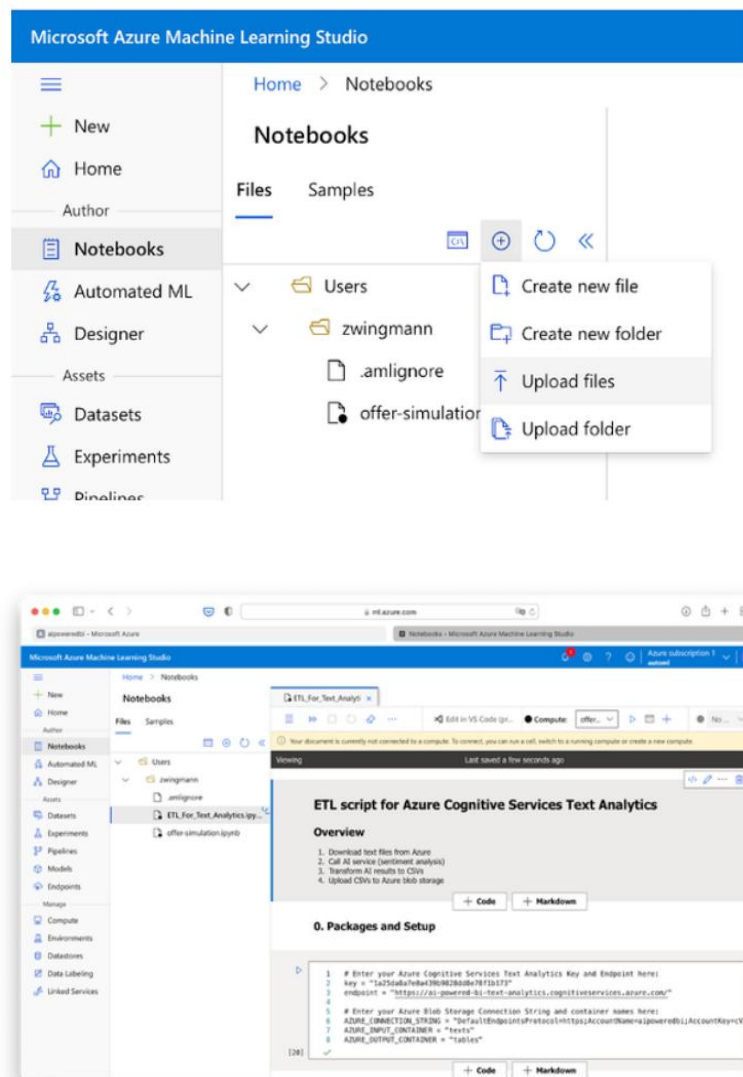
Sertakan juga versi R? Jika tidak, teks ini akan berlaku:

Kami akan menggunakan beberapa paket Azure yang diperlukan dalam kode ini untuk Autentikasi dan penyimpanan Blob yang saat ini tidak didukung di R. Meskipun alur kerja ini masih dapat dijalankan di R, prosesnya akan sangat berat. Jadi, dalam kasus ini saya hanya akan menyediakan versi Python. Semua pengguna R, silakan ikuti di sini.

Mari kita mulai dengan membuat Azure Notebook baru:

- Buka Microsoft Azure Machine Learning Studio dan pilih ruang kerja yang telah Anda gunakan di seluruh buku.
- Pilih ikon "+" dan pilih "Unggah file" seperti yang ditunjukkan pada Gambar di bawah.
- Cari "ETL_For_Text_Analytics.ipynb" di komputer Anda, centang "Saya percaya isi berkas ini", lalu klik "Unggah".
- Anda akan melihat Notebook seperti yang ditunjukkan pada Gambar di bawah.





Buku catatan akan ditampilkan di sisi kanan. Meskipun berkas ini berisi cukup banyak kode, jangan terintimidasi. Faktanya, satu-satunya hal yang perlu Anda ubah hanyalah parameter khusus untuk kredensial akses dan titik akhir layanan AI Anda.

Sebagai solusi perantara antara menjalankan seluruh buku catatan sekaligus, percaya bahwa semuanya akan berhasil, dan membahas kode baris demi baris, saya akan memberikan gambaran umum tingkat tinggi tentang apa yang dilakukan kode ini dengan menjelaskan berbagai bagian kode. Saya sarankan untuk mengikuti dan hanya mengeksekusi sel kode saat kita membahas bagian-bagian di buku ini.

Ada empat bagian utama dalam kode ini yang secara garis besar sejalan dengan empat langkah yang saya sebutkan sebelumnya selama proses ETL:

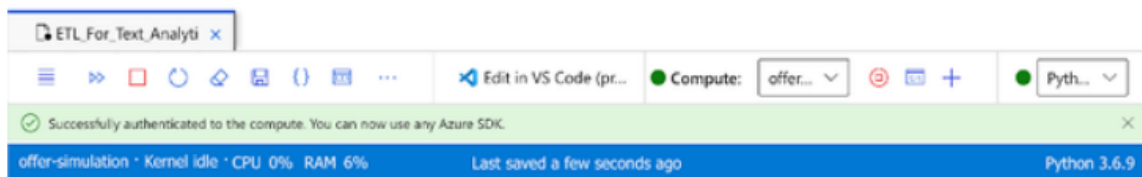
1. **Unduh berkas teks dari Azure:** Di sinilah kita akan mengunduh berkas teks mentah dari Azure ke mesin lokal yang menjalankan buku catatan.
2. **Panggil layanan AI (analisis sentimen):** Di sinilah kita mengirim berkas teks ke layanan AI dan menyimpan hasilnya.



3. **Ubah hasil AI menjadi CSV:** Di bagian ini, kita akan mengubah hasil AI dari keluaran JSON menjadi file CSV datar agar mudah digunakan oleh BI kita.
4. **Unggah CSV ke penyimpanan blob Azure:** Terakhir, kita akan mengunggah kembali file-file tersebut ke penyimpanan Blob Azure yang terhubung ke BI kita.

Sebelum memulai, pastikan Notebook terhubung ke sumber daya komputasi yang sedang berjalan. Jika Anda belum membuat sumber daya komputasi, periksa kembali bagian "Perangkat Prototipe AI" di Bab 3.

Jika Anda diminta untuk mengautentikasi lagi setelah sumber daya komputasi dimulai, cukup tekan tombol yang muncul dan autentikasi akan dilakukan secara otomatis untuk Anda seperti yang ditunjukkan pada Gambar di bawah ini.



Sekarang, buka bagian "0" dari kode. Di sinilah kita akan memastikan semua paket yang kita butuhkan telah diinstal dan dimuat, dan yang lebih penting kredensial Azure pribadi Anda telah disediakan. Sebenarnya, ini satu-satunya tempat di seluruh buku catatan tempat Anda perlu mengubah sesuatu. Ganti string dummy untuk kunci, titik akhir, dan String Koneksi Azure dengan nilai kustom Anda dari portal Azure untuk Layanan Kognitif Text Analytics dan akun Storage.

Setelah Anda memperbarui kredensial pribadi, jalankan sel kode pertama. Jika Anda masih ragu tentang cara kerja Buku Catatan ini, periksa kembali Bab 3 "Perangkat Prototipe". Keluaran kode ini seharusnya relatif sederhana. Semua paket seharusnya sudah diinstal pada sumber daya komputasi Azure dan Anda seharusnya hanya melihat notifikasi "Persyaratan telah terpenuhi" seperti yang ditunjukkan pada Gambar di bawah.



```
-simulation · Kernel idle · CPU 0% RAM 8% Last saved a few seconds ago Python 3.6.9
...
(from msrest>=0.6.18->azure-storage-blob) (2021.5.30)
Requirement already satisfied: requests-oauthlib>=0.5.0 in /anaconda/envs/azureml_py36/lib/python3.6/site-
packages (from msrest>=0.6.18->azure-storage-blob) (1.3.0)
Requirement already satisfied: isodate>=0.6.0 in /anaconda/envs/azureml_py36/lib/python3.6/site-packages (from
msrest>=0.6.18->azure-storage-blob) (0.6.0)
Requirement already satisfied: pycparser in /anaconda/envs/azureml_py36/lib/python3.6/site-packages (from
cffi>=1.12->cryptography>=2.1.4->azure-storage-blob) (2.20)
Requirement already satisfied: idna<4,>=2.5; python_version >= "3" in
/anaconda/envs/azureml_py36/lib/python3.6/site-packages (from requests>=2.18.4->azure-core<2.0.0,>=1.10.0->azure-
storage-blob) (3.2)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /anaconda/envs/azureml_py36/lib/python3.6/site-packages
(from requests>=2.18.4->azure-core<2.0.0,>=1.10.0->azure-storage-blob) (1.25.11)
Requirement already satisfied: charset-normalizer<=2.0.0; python_version >= "3" in
/anaconda/envs/azureml_py36/lib/python3.6/site-packages (from requests>=2.18.4->azure-core<2.0.0,>=1.10.0->azure-
storage-blob) (2.0.3)
Requirement already satisfied: oauthlib>=3.0.0 in /anaconda/envs/azureml_py36/lib/python3.6/site-packages (from
requests-oauthlib>=0.5.0->msrest>=0.6.18->azure-storage-blob) (3.1.1)
Note: you may need to restart the kernel to use updated packages.
```

CATATAN

Meskipun kami menyimpan kredensial Azure dalam teks biasa di dalam kode, perlu diketahui bahwa ini umumnya bukan praktik pengkodean yang baik. Kami melakukan ini untuk menjaga kesederhanaan dan juga karena notebook masih dihosting di lingkungan Azure yang terlindungi. Jika Anda ingin mempelajari praktik terbaik seputar penanganan mekanisme autentikasi, saya sarankan untuk memeriksa sumber daya Microsoft ini: Ikhtisar Azure Key Vault - Azure Key Vault

Mari kita lanjutkan ke bagian pertama dari proses mini ETL kita, yaitu pengunduhan berkas.

```
offer-simulation · Kernel idle · CPU 0% RAM 8% Last saved a few seconds ago Python 3.6.9
1. Download text files from Azure
1 azure_blob_downloader = AzureBlobDownloader()
2 azure_blob_downloader.download_all_blobs_in_container()
[6] ✓ 1 min 24 sec
...
review_20211104T055808.txt
review_20211104T075408.txt
review_20211104T091808.txt
review_20211104T102408.txt
review_20211104T115808.txt
review_20211104T131508.txt
review_20211104T143108.txt
review_20211104T161108.txt
review_20211104T174308.txt
review_20211104T192408.txt
review_20211104T211208.txt
review_20211104T224408.txt
review_20211105T001108.txt
review_20211105T015508.txt
review_20211105T025908.txt
review_20211105T045008.txt
review_20211105T060808.txt
review_20211105T071808.txt
```



Sel kode pertama dari langkah 1 ini memanggil objek kelas yang telah kita definisikan di bagian pengaturan sebelumnya. Intinya, sel ini membuat koneksi ke akun penyimpanan Azure Anda menggunakan string koneksi dan mengunduh semua berkas dari AZURE_INPUT_CONTAINER yang Anda berikan. Jalankan sel ini dan prosesnya akan selesai setelah satu menit dengan mencetak beberapa keluaran seperti yang ditunjukkan pada Gambar di bawah ini.

CATATAN

Jika Anda bertanya-tanya mengapa pengunduhan ini memakan waktu begitu lama, yakinlah bahwa ada metode untuk mempercepat proses pengunduhan, misalnya dengan menjalankan beberapa unduhan sekaligus. Bahkan, Anda juga dapat melewati URL file layanan AI secara langsung tanpa mengunduhnya sama sekali. Untuk prototipe kami, metode pengunduhan yang lambat masih dapat diterima. Namun, jika Anda beralih ke versi produksi, Anda akan mempertimbangkan alternatif yang lebih cepat.

Selanjutnya, mari kita lanjutkan ke bagian 3 kode kita - "Memanggil layanan AI". Di bagian ini, kita akan mengirimkan teks ke layanan AI untuk dianalisis. Seperti yang telah disebutkan sebelumnya, kita mengirimkan 10 berkas teks (dokumen) sekaligus ke API agar kita mendapatkan hasilnya lebih cepat. Jalankan sel kode dan setelah selesai, Anda akan melihat hasil pertama yang berisi 10 hasil pertama analisis sentimen seperti yang ditunjukkan pada Gambar di bawah ini.

```
offer-simulation · Kernel idle · CPU 0% · RAM 8% · Last saved a few seconds ago · Python 3
21 # Call the AI service
22 results = []
23 for document in documents:
24     results.append(sentiment_analysis_with_opinion_mining(document, client))
25
26 # Print the first result
27 results[0]

[8] ✓ 36 sec

... [AnalyzeSentimentResult(id=review_20211012T111608.txt, sentiment=negative, warnings=[], statistics=None,
confidence_scores=SentimentConfidenceScores(positive=0.42, neutral=0.0, negative=0.58), sentences=
[SentenceSentiment(text=A bit more noise insulation would have been good, sentiment=negative,
confidence_scores=SentimentConfidenceScores(positive=0.42, neutral=0.0, negative=0.58), length=48, offset=0,
mined_opinions=[])], is_error=False),
AnalyzeSentimentResult(id=review_20211012T131108.txt, sentiment=negative, warnings=[], statistics=None,
confidence_scores=SentimentConfidenceScores(positive=0.03, neutral=0.0, negative=0.97), sentences=
[SentenceSentiment(text=A bit noisy at night., sentiment=negative,
confidence_scores=SentimentConfidenceScores(positive=0.01, neutral=0.0, negative=0.99), length=21, offset=0,
mined_opinions=[]), SentenceSentiment(text=The shower did not work very well, sentiment=negative,
confidence_scores=SentimentConfidenceScores(positive=0.05, neutral=0.0, negative=0.95), length=33, offset=22,
mined_opinions=[MinedOpinion(target=TargetSentiment(text=shower, sentiment=negative,
confidence_scores=SentimentConfidenceScores(positive=0.02, neutral=0.0, negative=0.98), length=6, offset=26),
assessments=[AssessmentSentiment(text=work, sentiment=negative,
confidence_scores=SentimentConfidenceScores(positive=0.02, neutral=0.0, negative=0.98), length=4, offset=41,
is_negated=True)]))], is_error=False),
AnalyzeSentimentResult(id=review_20211012T144008.txt, sentiment=negative, warnings=[], statistics=None,
confidence_scores=SentimentConfidenceScores(positive=0.01, neutral=0.0, negative=0.99), sentences=
[SentenceSentiment(text=A bit pricey., sentiment=negative,
confidence_scores=SentimentConfidenceScores(positive=0.01, neutral=0.0, negative=0.99), length=13, offset=0,
mined_opinions=[]), SentenceSentiment(text=The corridors to our room on the third floor needed a refresh.,
sentiment=neutral, confidence_scores=SentimentConfidenceScores(positive=0.01, neutral=0.99, negative=0.0),
length=62, offset=14, mined_opinions=[MinedOpinion(target=TargetSentiment(text=corridors, sentiment=negative,
confidence_scores=SentimentConfidenceScores(positive=0.1, neutral=0.0, negative=0.9), length=9, offset=18),
```

Seperti yang mungkin Anda perhatikan dari pratinjau keluaran ini, struktur hasilnya sangat bertingkat. Layanan AI tidak hanya memberi kita skor sentimen untuk setiap teks, tetapi juga



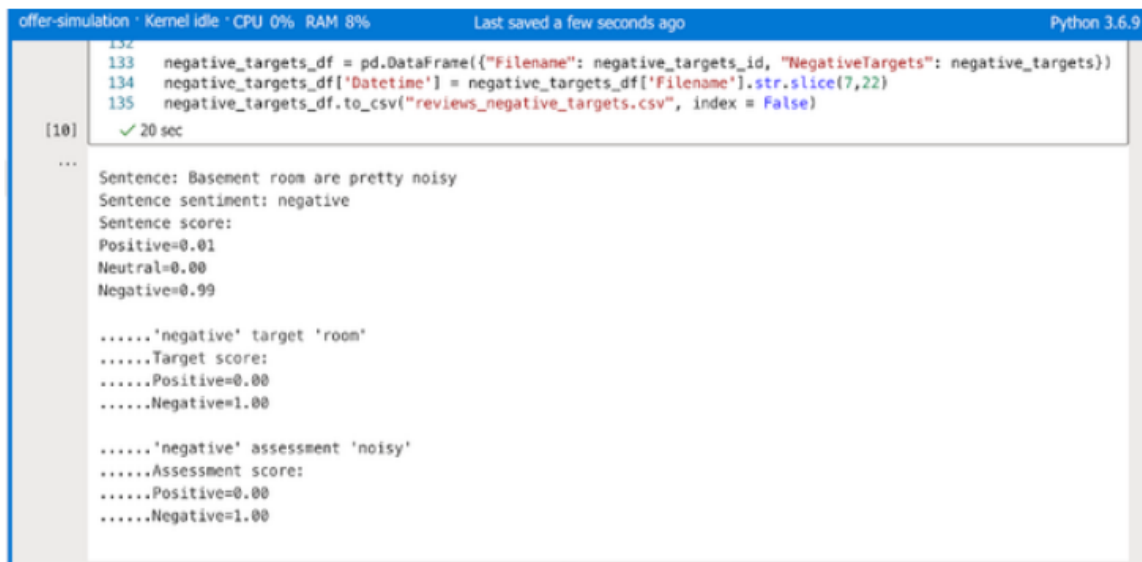
rincian opini pada tingkat kata untuk setiap teks yang dianalisis. Ini akan sangat berguna untuk interpretasi data, tetapi di sisi lain, hal ini menimbulkan kesulitan bagi kita untuk mengurai keseluruhan objek dan mengubahnya kembali menjadi tabel-tabel datar yang bagus.

Ini adalah inti dari bagian 3 kode ini. Mari kita lanjutkan!

Bagian 3 dari kode persiapan data sebenarnya adalah yang paling lengkap. Meskipun beban kerja AI utama telah selesai, masih perlu sedikit pemrosesan pasca untuk mengekstrak skor sentimen untuk setiap teks dan opini terkait yang ditemukan di setiap item.

Sekarang, jika Anda bertanya pada diri sendiri: "Bagaimana mungkin saya bisa membuat kode ini?", maka saya punya kabar baik untuk Anda: Sebagian besar kode ini sebenarnya dipinjam dari halaman dokumentasi layanan AI. Apakah Anda ingat referensi cepat di konsol Azure? Dari sinilah sebagian besar kode ini berasal. Setelah Anda memahami perilaku umumnya, akan cukup mudah untuk melakukan penyesuaian yang Anda perlukan.

Dalam kasus kami, saya melakukan tiga penyesuaian: Pertama, saya hanya mengekstrak skor sentimen tingkat tinggi per item teks dan menyimpannya sebagai tabel datar untuk kompatibilitas CSV yang lebih baik. Tabel ini, atau kerangka data, juga berisi nama berkas asli, teks asli, dan kolom tanggal dan waktu yang saya ekstrak dari nama berkas teks. Ini akan menghasilkan keluaran CSV datar yang bagus yang nantinya dapat kami tampilkan di BI kami.



```
offer-simulation Kernel Idle CPU 0% RAM 8% Last saved a few seconds ago Python 3.6.9
133 negative_targets_df = pd.DataFrame({"Filename": negative_targets_id, "NegativeTargets": negative_targets})
134 negative_targets_df["Datetime"] = negative_targets_df["Filename"].str.slice(7,22)
135 negative_targets_df.to_csv("reviews_negative_targets.csv", index = False)

[10] ✓ 20 sec

...
Sentence: Basement room are pretty noisy
Sentence sentiment: negative
Sentence score:
Positive=0.01
Neutral=0.00
Negative=0.99

.....'negative' target 'room'
.....Target score:
.....Positive=0.00
.....Negative=1.00

.....'negative' assessment 'noisy'
.....Assessment score:
.....Positive=0.00
.....Negative=1.00
```

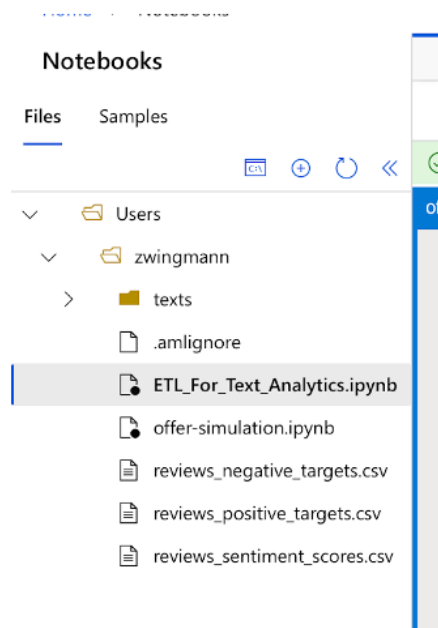
Kedua dan ketiga, saya membuat CSV datar untuk masing-masing istilah atau opini positif dan negatif yang ditemukan dalam dokumen teks, yang juga disimpan dengan referensi ke nama berkas asli dan kolom tanggal dan waktu yang diekstrak sehingga istilah-istilah ini dapat dihubungkan kembali ke konteks aslinya dan mungkin juga difilter berdasarkan waktu. Jika Anda beruntung, periksa kode baris demi baris dan lihat apa yang terjadi. Jika tidak, tidak masalah juga. Kami tidak ingin menjadi Insinyur Perangkat Lunak. Bagaimanapun, tekan



tombol jalankan untuk sel ini untuk melihat keluaran seperti yang ditunjukkan pada Gambar di bawah ini.

Demi kenyamanan, keluaran mencetak sentimen dan opini untuk setiap dokumen teks dalam format yang mudah dibaca. Misalnya, Anda dapat melihat pada Gambar di bawah ini bahwa kalimat "Ruang bawah tanah cukup berisik" meskipun terdapat kesalahan tata bahasa diidentifikasi oleh AI sebagai sentimen negatif, dan layanan AI dapat mengidentifikasi kata "ruangan" dan "berisik" sebagai pendorong utama di balik opini negatif ini. Cantik, bukan?

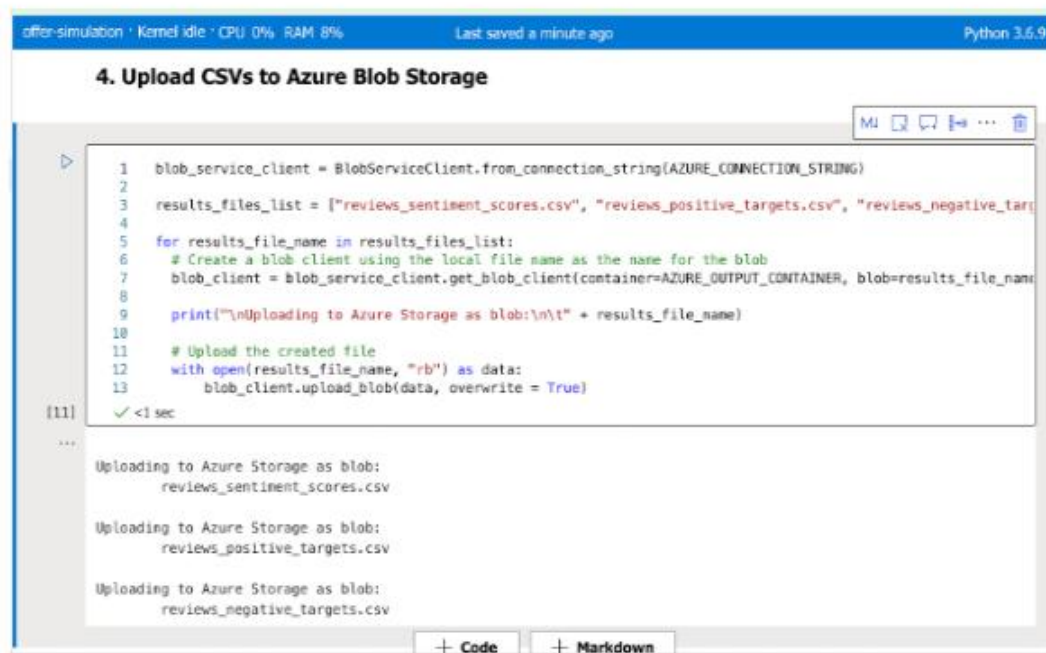
Pada akhirnya, kode ini juga menghasilkan berkas CSV yang kami inginkan. Jika Anda melihat penjelajah berkas di sebelah kiri dan menekan tombol "segarkan", Anda akan melihat tiga berkas CSV baru seperti yang ditunjukkan pada Gambar di bawah ini.



Agar berkas-berkas csv ini dapat diakses oleh BI kita, mari kita menuju ke bagian kode 4 yang membahas tentang pengunggahan berkas-berkas ini ke penyimpanan Azure Blob kita.

Dibandingkan dengan bagian 3, ini akan jauh lebih mudah.

Jalankan sel kode terakhir dan dalam sekejap Anda akan melihat output seperti yang ditunjukkan pada Gambar di bawah ini yang mengonfirmasi bahwa berkas-berkas tersebut telah berhasil diunggah.



```
offer-simulation · Kernel idle · CPU: 0% RAM: 8% Last saved a minute ago Python 3.6.9

4. Upload CSVs to Azure Blob Storage

1 blob_service_client = BlobServiceClient.from_connection_string(AZURE_CONNECTION_STRING)
2
3 results_files_list = ["reviews_sentiment_scores.csv", "reviews_positive_targets.csv", "reviews_negative_targ
4
5 for results_file_name in results_files_list:
6     # Create a blob client using the local file name as the name for the blob
7     blob_client = blob_service_client.get_blob_client(container=AZURE_OUTPUT_CONTAINER, blob=results_file_name
8
9     print("\nUploading to Azure Storage as blob:\n\t" + results_file_name)
10
11     # Upload the created file
12     with open(results_file_name, "rb") as data:
13         blob_client.upload_blob(data, overwrite = True)

[11]: ✓ <1 sec

...

Uploading to Azure Storage as blob:
    reviews_sentiment_scores.csv

Uploading to Azure Storage as blob:
    reviews_positive_targets.csv

Uploading to Azure Storage as blob:
    reviews_negative_targets.csv

+ Code + Markdown
```

Selamat! Anda telah menyelesaikan bagian tersulit dan berhasil melewati tantangan mini ETL. Sekarang, seperti halnya setiap proses ETL, proses ini dapat direproduksi.

Artinya, setiap kali Anda memperbarui berkas teks di kontainer input Blob dan berkas-berkas tersebut tetap memiliki struktur yang sama, Anda cukup menjalankan ulang seluruh alur kerja ini (sekaligus) dan berkas CSV di kontainer output akan diganti dan ditimpa dengan hasil yang baru. Namun, mari kita tinggalkan pembahasan ETL untuk saat ini dan beralih ke bagian yang menyenangkan: Memvisualisasikan dan mensintesis hasil kita di BI kita.

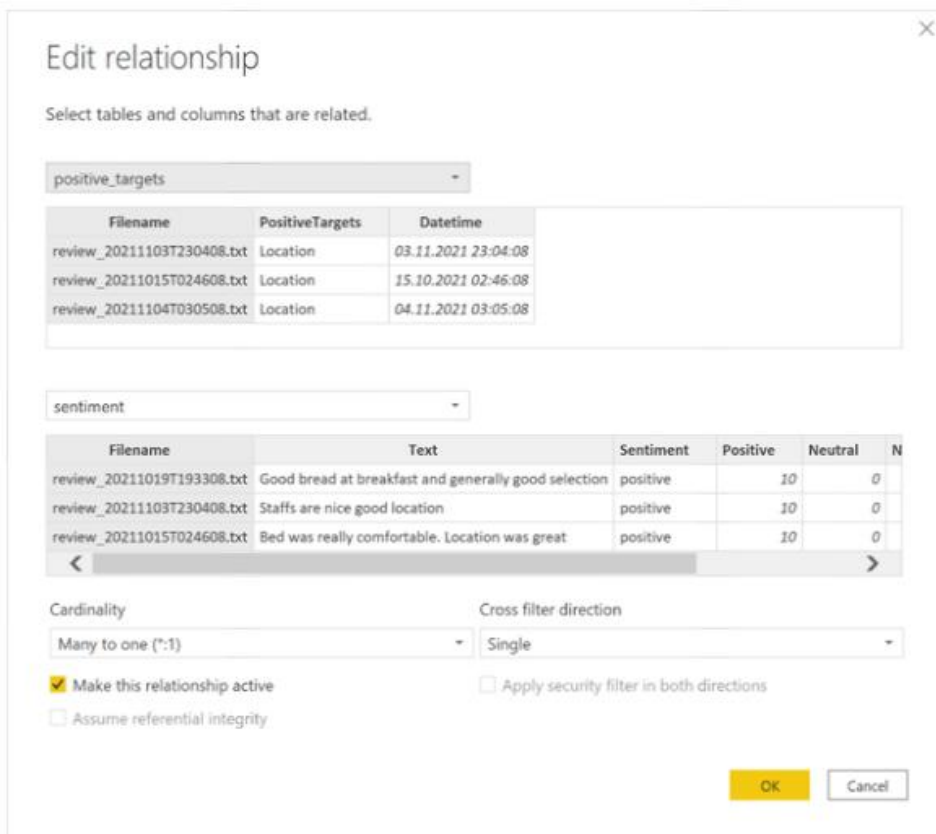
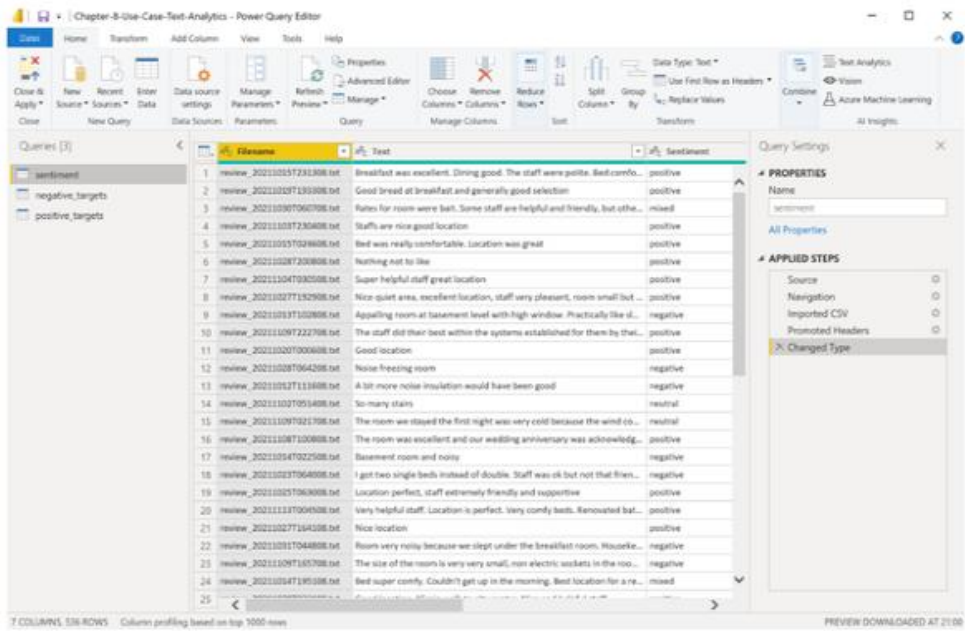
CATATAN

Jangan lupa untuk menghentikan sumber daya komputasi Anda setelah latihan ini!

Mensintesis Hasil ke dalam BI

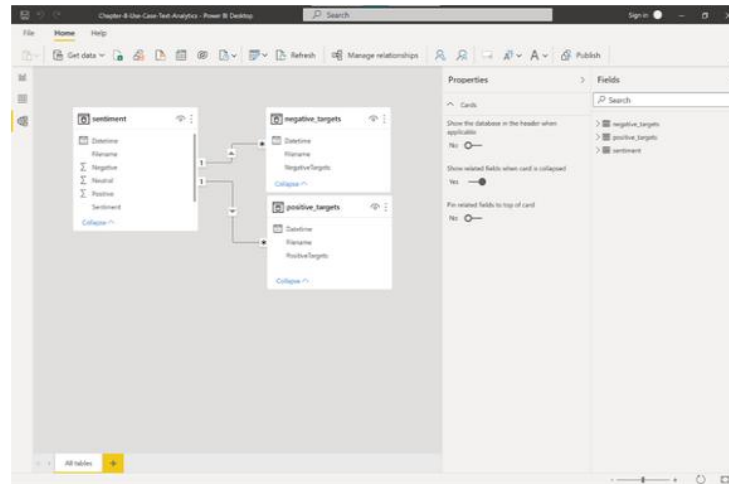
Untuk menampilkan hasil kita, kita akan kembali ke Power BI. Buat laporan baru di Power BI dan pilih "Azure Blob Storage" sebagai sumber data. Hubungkan dengan akun penyimpanan Anda dan centang kotak di samping folder "Tabel". Klik "Transformasi". Di editor Power Query, Anda akan menemukan daftar semua tabel CSV yang ada di folder tersebut. Kita perlu memeriksa ketiganya satu per satu. Pilih yang pertama dan klik tautan "Biner" untuk melihat pratinjau data Anda seperti yang ditunjukkan pada Gambar di bawah ini. Periksa kembali apakah kolom Tanggal dan Waktu telah dikonversi dengan benar ke format tanggal dan waktu. Klik "Tutup & Terapkan" dan ulangi langkah-langkah ini untuk dua file CSV lainnya.





Model data Anda sekarang seharusnya mencantumkan tiga tabel terpisah. Namun, kita tahu bahwa tabel-tabel ini sebenarnya saling terkait dan kita ingin Power BI juga mengetahui hal ini. Klik kanan pada tabel mana pun dan pilih "Edit hubungan". Di editor yang muncul, buat hubungan antara tabel "positive_targets" dan sentimen berdasarkan kolom "Filename".

Kardinalitasnya harus banyak berbanding satu seperti yang ditunjukkan pada Gambar di bawah ini. Klik OK dan ulangi pengaturan ini untuk tabel "negative_targets". Setelah itu, tabel-tabel dalam model data Anda harus dihubungkan seperti yang ditunjukkan pada Gambar di bawah ini.

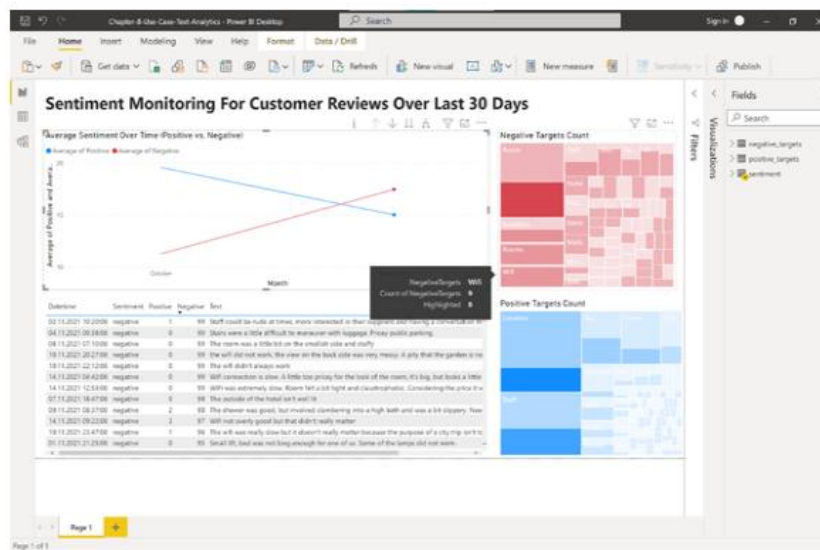
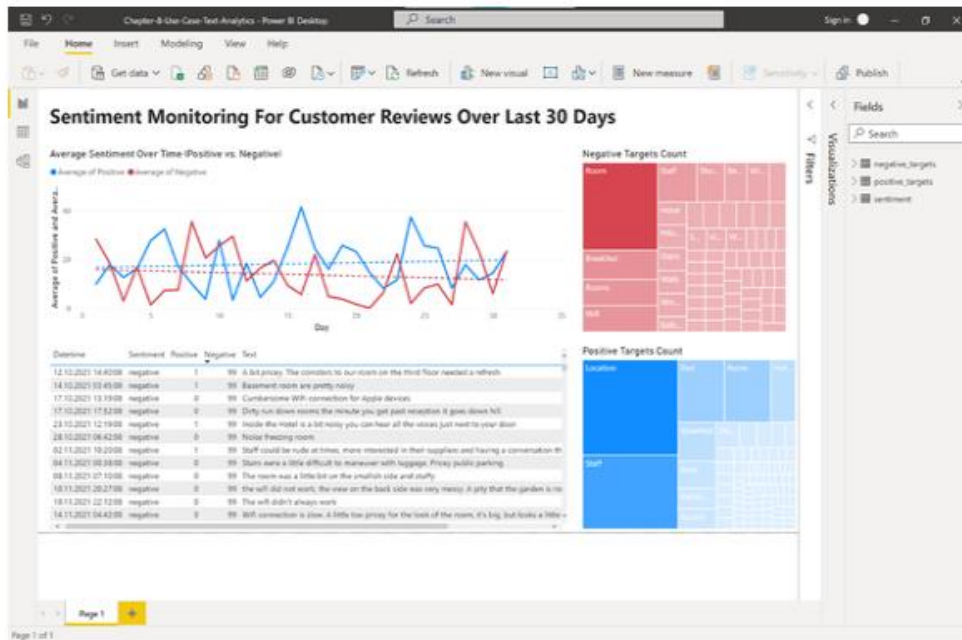


Setelah model data kita selesai, saatnya beralih ke penanganan visual yang sebenarnya. Mari kita rekap situasi secara singkat: Manajemen ingin tahu apakah ada sesuatu yang sedang terjadi yang perlu diperhatikan. Untuk mendapatkan gambaran umum tentang ulasan pelanggan, kita memerlukan empat elemen:

1. Perkembangan sentimen pelanggan dari waktu ke waktu untuk memeriksa tren
2. Daftar item yang dikeluhkan pelanggan (target negatif)
3. Daftar item yang disukai pelanggan (target positif)
4. Referensi kembali ke data asli agar kita mendapatkan lebih banyak konteks

Ada banyak cara berbeda untuk menyampaikan informasi ini. Saya memutuskan untuk menyusun laporan yang terdiri dari diagram garis yang berisi tren keseluruhan, dua peta pohon yang menyoroti target positif dan negatif, dan tabel sederhana yang mencantumkan semua umpan balik pelanggan dengan teks biasa. Anda dapat melihat hasil akhirnya pada Gambar di bawah.





Cobalah membuat ulang dasbor ini sendiri di Power BI atau buat solusi Anda sendiri. Atau, Anda dapat mengunduh berkas lengkap "Chapter-8-Use-Case-Text-Analytics.pbix" dari repositori buku dan menghubungkannya ke penyimpanan blob Azure Anda sendiri.

Jadi, apa yang ditunjukkan dasbor ini? Untuk pertama kalinya, kita dapat melihat bahwa sentimen positif dan negatif tampaknya memiliki tren yang cukup stabil dengan naik turun harian. Jika kita menggabungkan visual ke tingkat bulanan, gambarannya menjadi sedikit lebih jelas seperti yang terlihat pada Gambar di bawah ini. Dari perspektif bulanan, sentimen negatif tampaknya telah meningkat pesat. Jika kita menelusuri alasannya, kita dapat mengetahui bahwa kebanyakan orang sebenarnya mengeluh tentang kamar, sarapan, dan Wi-Fi. Meskipun kamar mungkin sulit diperbaiki dalam jangka pendek, sarapan dan Wi-Fi memberikan tindakan yang jelas yang dapat segera ditangani oleh manajemen.



Selain itu, karena kita telah membuat hubungan antar tabel, Power BI memungkinkan kita memfilter titik data tertentu: Misalnya, jika kita memilih tanggal dari diagram garis, sisa dasbor akan diperbarui sesuai kebutuhan dan hanya menampilkan data relevan yang dikumpulkan pada hari tersebut, dan sebaliknya. Apa sebenarnya yang dikeluhkan pelanggan terkait kamar? Cukup klik "Kamar" di Peta Pohon Target Negatif dan lihat apa yang dikatakan pelanggan dengan menelusuri teks asli di bagian bawah.

Ini menyediakan cara yang bagus untuk menjelajahi kumpulan data dan mencari tahu lebih lanjut tentang umpan balik pelanggan yang sebenarnya. Meskipun dasbor ini hanya menyentuh puncak gunung es analitik teks, saya harap Anda telah melihat betapa hebatnya menganalisis data teks dalam skala besar. Berkat layanan AI, hal ini dapat dilakukan dengan beberapa baris kode dan juga terintegrasi dalam sistem BI apa pun.

9.2 KASUS PENGGUNAAN: MENGURAI DOKUMEN DENGAN AI

Pernahkah Anda melihat bisnis yang tidak menggunakan formulir? Formulir bagi bisnis sama seperti API bagi programmer. Formulir memiliki keunggulan besar. Formulir biasanya berfungsi sangat baik untuk bertukar informasi antarmanusia. Dengan formulir tertentu dan beberapa deskripsi, Anda seharusnya cukup cepat memahami cara membaca formulir atau cara mengisinya. Kelemahannya sering kali muncul ketika formulir harus diproses oleh komputer. Seringkali, formulir masih berupa dokumen kertas atau versi elektroniknya - PDF.

Jika Anda ingin memprosesnya dalam skala besar, Anda seringkali tidak punya pilihan lain selain memeriksanya secara manual dan menyalin/menempel data darinya. Kwitansi juga dapat dianggap sebagai formulir kwitansi menyediakan struktur data berulang yang mudah dibaca manusia, tetapi sulit diinterpretasikan oleh mesin. Dalam kasus penggunaan ini, kita akan mempelajari bagaimana AI akan membantu kita membaca dokumen dalam skala besar dan mengekstrak nilai darinya untuk ditampilkan dalam sistem BI.

Pernyataan Masalah

Dalam skenario kita berikutnya, bayangkan kita bekerja untuk perusahaan menengah yang menyediakan layanan konsultasi. Perusahaan ini berlokasi di Jerman di mana perwakilan penjualan sering bepergian ke calon pelanggan untuk mempersiapkan atau menutup transaksi. Perjalanan tersebut sebagian besar dilakukan dengan kereta api berkecepatan tinggi yang menghubungkan kota-kota besar di Jerman, yang disebut *InterCity Express (ICE)*.





Departemen manajemen perjalanan kami mengendalikan keseluruhan biaya. Proses perjalanan bisnis terlihat seperti ini: Tenaga penjualan dapat memesan tiket kereta mereka melalui portal swalayan dari operator kereta api (Deutsche Bahn). Tiket ini dibayar menggunakan kartu kredit perusahaan. Tim manajemen perjalanan mengawasi penagihan kartu kredit di akhir setiap bulan untuk mendapatkan gambaran umum tentang biaya perjalanan.

Namun, laporan kartu kredit tidak memberikan detail lebih lanjut kecuali jumlah yang dibelanjakan dan tanggal terjadinya. Untuk mengoptimalkan biaya perjalanan dan juga memahami lebih baik perjalanan bisnis mana yang paling banyak menimbulkan biaya, manajemen perjalanan menginginkan lebih banyak wawasan tentang perjalanan tersebut, termasuk perincian rute perjalanan populer (asal kereta dan tujuan kereta).

Informasi ini dapat ditemukan dalam tanda terima pemesanan yang diterima tenaga penjualan secara otomatis melalui email setelah setiap pemesanan berhasil. Secara khusus, tim tertarik untuk mengekstrak informasi berikut dari formulir ini: Tanggal pemesanan, asal perjalanan, tujuan perjalanan, dan harga tiket. Manajemen perjalanan ingin mengetahui apakah ada cara untuk mengekstrak informasi ini secara otomatis dan idealnya melaporkannya

menggunakan BI yang ada. Mereka telah memberi kami sampel 162 struk dari seorang perwakilan penjualan untuk mengerjakan prototipe pertama.

Ikhtisar Solusi

Apa yang akan kami lakukan dalam skenario ini adalah mengekstrak informasi dari dokumen menggunakan teknik yang disebut pengenalan karakter optik (OCR). OCR bukanlah hal baru dan telah ada sejak lama. Namun, setidaknya ada dua lapisan dalam masalah ini di mana AI akan membantu kami: Lapisan pertama adalah untuk meningkatkan OCR yang sebenarnya, yaitu mengenali karakter teks tunggal dan mengubahnya menjadi bentuk yang dapat dibaca mesin (string teks biasa).

Bagian kedua adalah memahami karakter-karakter tersebut, yaitu untuk mengetahui karakter mana yang termasuk dalam sebuah kata, kalimat, atau bahkan mengenali struktur yang lebih kompleks seperti tabel. Untuk mengatasi masalah ini, kami akan mengikuti pendekatan yang serupa seperti pada kasus penggunaan sebelumnya:

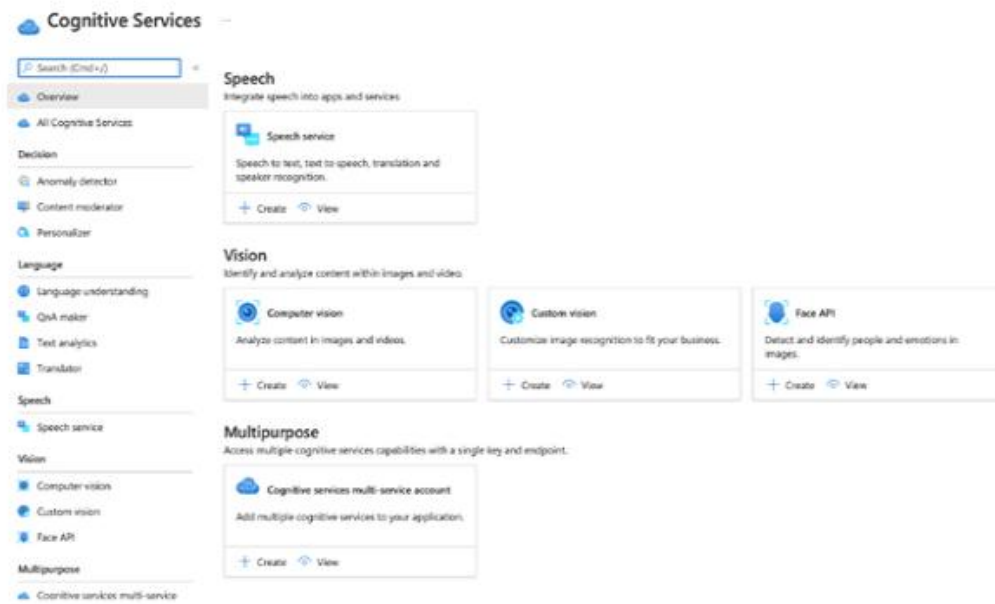
1. Kami menerapkan layanan AI siap pakai di Microsoft Azure, dalam hal ini Layanan Kognitif untuk Visi Komputer
2. Kami akan memuat data ke dalam area staging, dalam hal ini juga menggunakan penyimpanan blob Azure
3. Kami akan menyiapkan skrip ETL kecil yang memuat data dari area staging, menerapkan layanan AI, dan mengubahnya menjadi CSV datar.
4. Kami akan mengunggah berkas CSV ke lokasi yang mudah diakses dan divisualisasikan dengan alat BI kami.

Ayo!

Menyiapkan layanan AI

Buka portal Azure Anda dan cari "Layanan Kognitif" di bilah pencarian. Pilih Layanan Kognitif dari daftar saran dan lanjutkan ke halaman sumber daya yang sesuai. Gulir ke bawah ke "Visi". Anda akan melihat layanan yang tersedia seperti yang ditunjukkan pada Gambar di bawah ini:





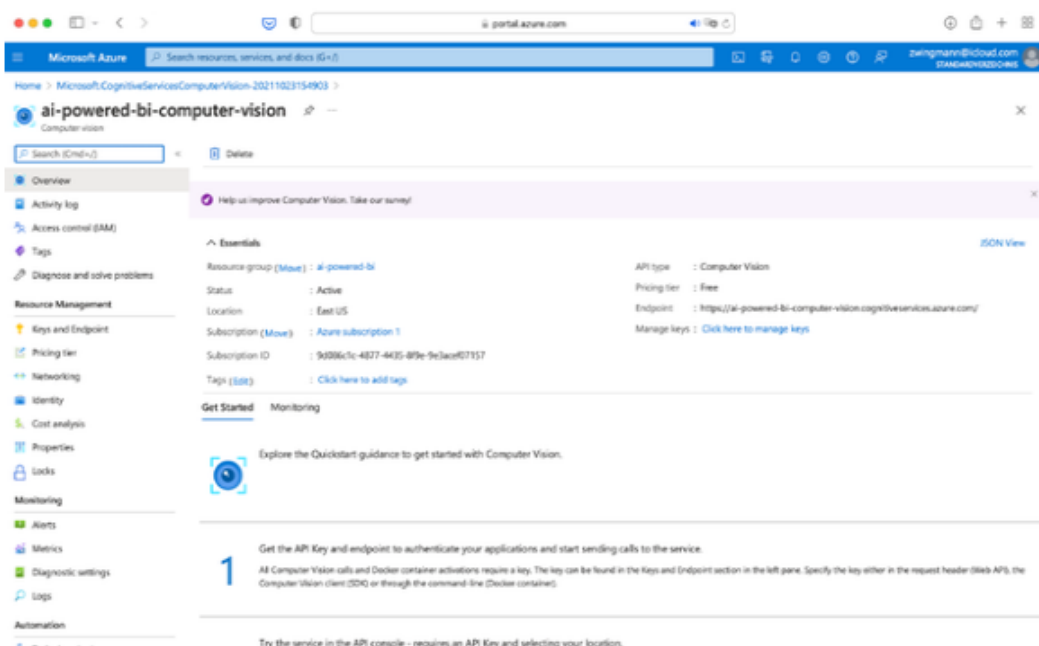
Pilih "+ Buat" dari kotak Visi Komputer dan Anda akan disambut dengan formulir yang sekarang sudah umum untuk menyiapkan sumber daya visi komputer. Sekali lagi, pilih grup sumber daya, beri nama sumber daya Anda, dan pilih tingkat harga F0 gratis. Jangan lupa untuk mencentang kotak di bagian bawah yang menyatakan bahwa Anda menyetujui "Persyaratan AI yang Bertanggung Jawab".

Luangkan waktu sejenak untuk menyetujui persyaratan ini. Pada dasarnya, ini berarti Anda tidak melanggar hak pribadi dengan menggunakan AI. Ingat: Setiap gambar atau dokumen yang Anda kirim ke API akan diproses oleh layanan yang dimiliki dan dioperasikan oleh Microsoft. Jadi, terutama jika Anda berurusan dengan data pribadi, Anda harus memastikan bahwa Anda memiliki persetujuan dan izin untuk melakukannya. Dalam kasus penggunaan kami, kami berurusan dengan data fiktif atau domain publik sehingga tidak ada risiko yang terlibat. Namun, dalam lingkungan bisnis, Anda akan berpikir dua kali sebelum mulai mengirim dokumen pelanggan atau karyawan Anda ke Layanan AI jarak jauh.

Klik "Tinjau + buat" setelah Anda mengisi formulir dan "buat" setelah validasi otomatis selesai. Setelah beberapa menit, layanan Anda akan diterapkan dan Anda dapat mengaksesnya dengan mengklik tautan notifikasi atau dengan mencari nama sumber daya Anda melalui bilah pencarian di portal Azure. Jika Anda membuka halaman sumber daya, Anda akan melihat layar yang terlihat seperti pada Gambar di bawah ini.

Panduan mulai cepat ini terlihat sedikit berbeda dari layanan AI sebelumnya, tetapi pada dasarnya berisi informasi yang sama: Anda akan menemukan tautan ke kunci akses Anda dan beberapa contoh kode untuk memulai dengan cepat. Untuk saat ini, Anda dapat membuka "Kunci dan titik akhir" untuk menemukan titik akhir HTTP sumber daya dan kunci akses rahasia. Biarkan halaman ini terbuka, karena kita akan membutuhkan kunci-kunci ini sebentar lagi. Pengaturan AI telah selesai, mari kita lanjutkan untuk memuat dan memproses berkas PDF.





CATATAN

Tunggu sebentar, mungkin Anda bertanya. Mengapa kita tidak menggunakan layanan Azure khusus "Pengenalan Formulir" untuk ini?

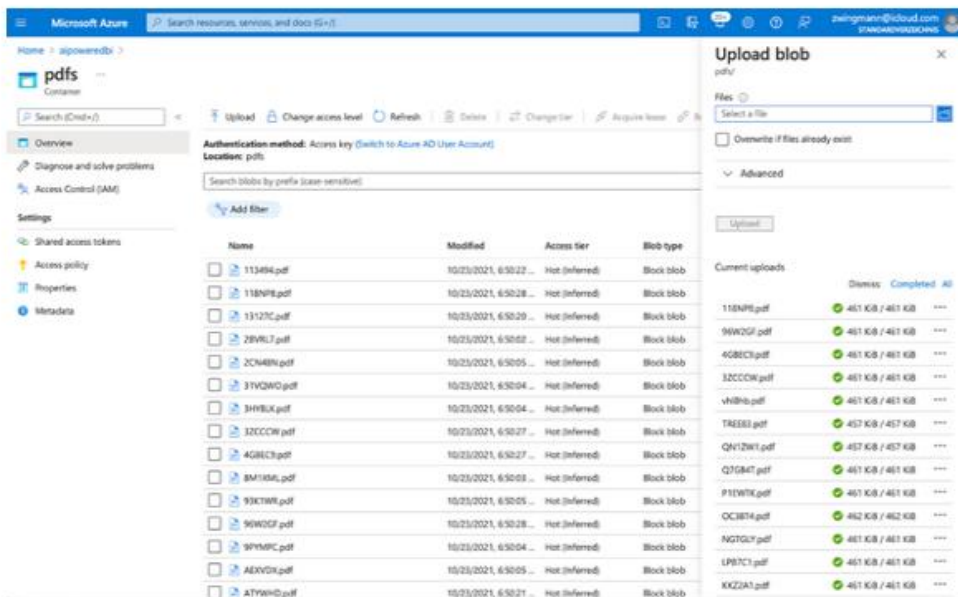
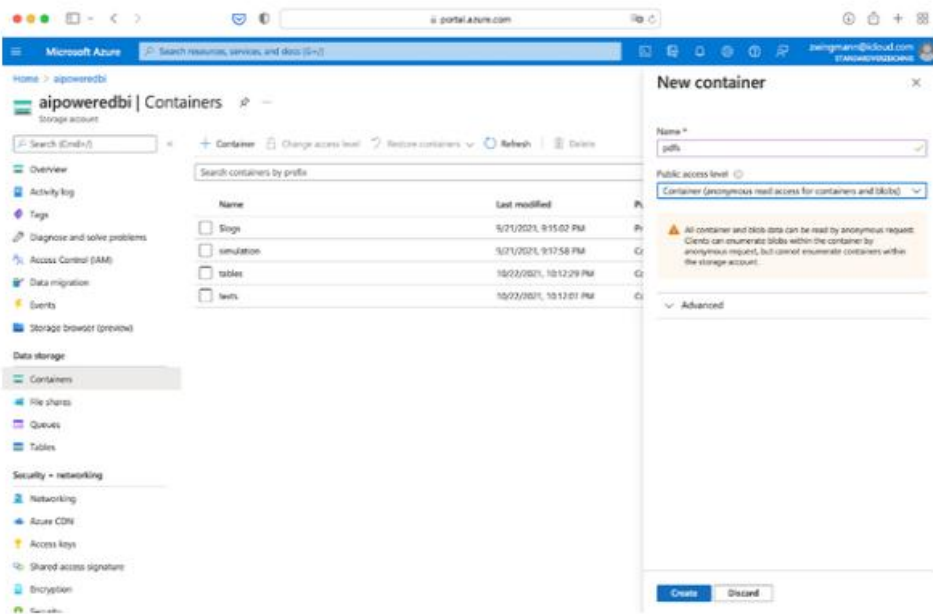
Dua alasan:

Form Recognizer adalah layanan yang lebih terspesialisasi yang menyesuaikan visi komputer dengan pengenalan formulir dan ekstraksi nilai. Layanan ini lebih terfokus, tetapi juga membutuhkan pelatihan dan pengaturan khusus tertentu sebelum berfungsi, dan ini bukanlah layanan yang siap pakai. Di sisi lain, API visi komputer adalah layanan multiguna yang bekerja seperti plug-and-play. Lebih mudah diimplementasikan, tetapi Anda akan mencapai batas ketika tugas menjadi terlalu terspesialisasi. Layanan ini juga akan berguna bagi Anda dalam skenario selain ekstraksi formulir. Rekomendasi saya adalah memulai dengan visi komputer umum terlebih dahulu dan beralih ke layanan yang lebih khusus/disesuaikan nanti, jika diperlukan. Jika Anda ingin mempelajari lebih lanjut tentang Form Recognizer, Anda dapat menemukan detail selengkapnya di sini: [Form Recognizer - Azure Applied AI Services](#)

Menyiapkan Alur Data

Pertama-tama, mari kita unggah file PDF kita ke kontainer penyimpanan Azure Blob agar kita dapat mengaksesnya dengan mudah. Buka jendela Portal Azure baru dan navigasikan ke penyimpanan Azure Blob menggunakan bilah pencarian atau cari "item terbaru" di dasbor Anda. Navigasikan ke "Kontainer" dan buat kontainer baru bernama "pdf" seperti yang ditunjukkan pada Gambar di bawah ini.





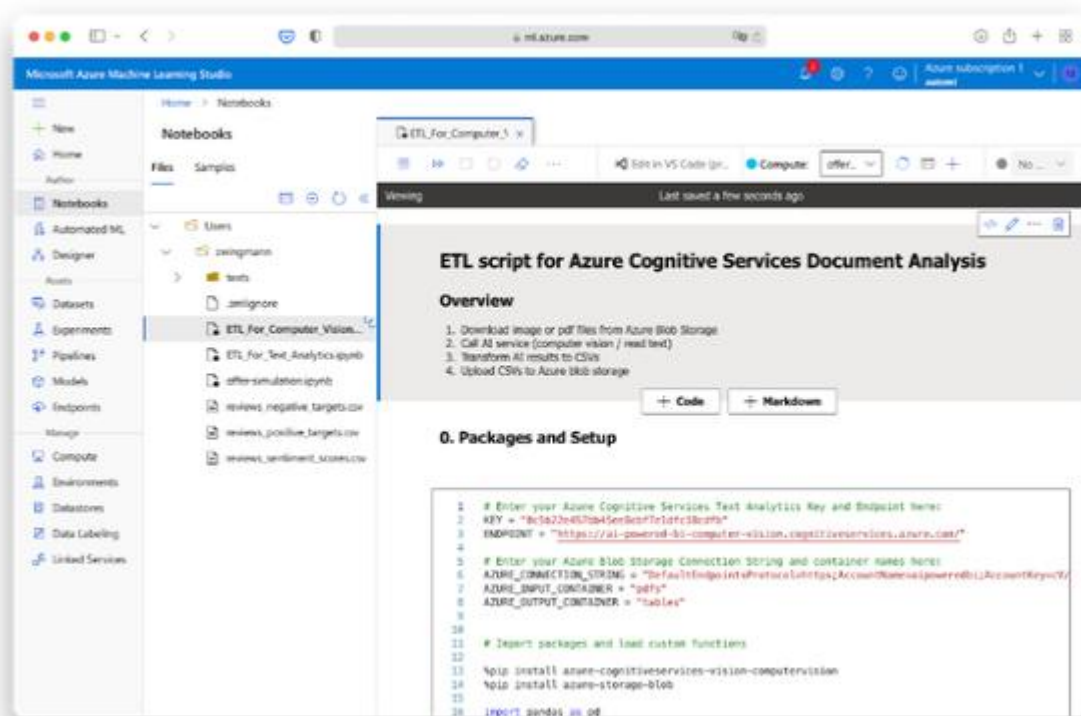
Sekarang, unduh berkas "pdfs.zip" dari repositori buku dan ekstrak berkas tersebut di komputer lokal Anda. Unduh semua konten dari folder "pdfs" lokal Anda ke container di penyimpanan blob Azure. Setelah itu, semua berkas PDF harus disimpan dalam container "pdfs" di penyimpanan blob Azure tanpa subfolder atau berkas zip apa pun.

Setelah file siap dianalisis, kita dapat melanjutkan ke proses ETL. Seperti pada contoh sebelumnya, saya akan memandu Anda melalui kode langkah demi langkah dan kita akan menjalankan skrip di Azure Notebooks. Tentu saja Anda juga dapat menggunakan IDE lokal jika diinginkan.

Pertama-tama, unduh file "ETL_For_Document_Analysis.ipynb" dari repositori buku ke komputer lokal Anda. Buka Azure Machine Learning Studio, pilih ruang kerja yang Anda



inginkan, dan klik "Mulai". Arahkan ke "Notebooks" dan pilih "Upload files" seperti yang ditunjukkan pada Gambar di bawah. Pilih file "ETL_For_Document_Analysis.ipynb" dan unggah ke Azure Notebooks. Setelah diunggah, notebook akan muncul di sisi kanan layar. Jangan lupa untuk menghubungkan notebook Anda ke sumber daya komputasi seperti yang kita lakukan pada kasus penggunaan sebelumnya. Setelah semuanya siap, layar Anda akan terlihat seperti Gambar di bawah ini.



Sekarang saya akan memandu Anda melalui kode dan menjelaskan apa yang terjadi bagian demi bagian. Silakan ikuti dengan mengeksekusi sel kode sesuai urutan yang kita bahas di buku ini.

Pertama, bagian "0. Paket dan Pengaturan" sekali lagi merupakan tempat Anda perlu memasukkan String Koneksi Azure kustom serta kunci dan titik akhir kustom untuk layanan visi komputer Anda. Apakah jendelanya masih terbuka? Bagus, kalau begitu Anda cukup menyalin dan menempelkan kedua nilai di sini. Pada bagian selanjutnya, kode memastikan bahwa semua paket yang diperlukan telah diimpor dan kita mendefinisikan beberapa fungsi kustom. Saya akan menjelaskan fungsi-fungsi ini secara lebih rinci setelah kita akan menggunakannya. Sekarang, jalankan sel kode pertama.

Bagian 1 membahas pengunduhan berkas dari penyimpanan blob Azure. Bagian ini cukup mudah dan identik dengan yang kita lakukan pada kasus penggunaan sebelumnya. Jalankan juga sel kode ini. Hasilnya, Anda akan melihat folder di peramban berkas bernama "pdfs". Jika tidak, klik ikon penyegaran kecil di peramban berkas.



Bagian 2 sekali lagi sangat singkat. Di sini, kita hanya memanggil layanan AI dan mengambil hasil mentah dari analisis. Dua hal penting yang perlu diperhatikan: Pertama, Anda mungkin memperhatikan perintah `time.sleep(6)` dalam kode. Saya menyertakan baris ini untuk memastikan kita tidak melebihi batas kuota gratis 20 permintaan per menit. Jadi mengapa kita menunggu 6 detik, bukan hanya 3 detik, karena $3 \text{ detik} * 20 \text{ permintaan} = 60 \text{ detik}$? Ini karena dalam kasus visi komputer, kita menjalankan apa yang disebut "operasi asinkron".

Ingat bahwa dalam contoh sebelumnya untuk analisis sentimen, kita hanya mengirim teks ke API dan langsung menerima respons beserta hasil AI. Dalam kasus visi komputer, operasinya lebih kompleks dan hasilnya bisa memakan waktu cukup lama (seperti detik). Oleh karena itu, untuk setiap dokumen, kita perlu membuat dua panggilan API: Satu permintaan POST untuk mengirim dokumen ke layanan AI. Dan satu permintaan GET untuk mengambil hasilnya. Jika Anda perhatikan dengan saksama fungsi `document_analysis` di bagian 0, Anda akan melihat kedua panggilan API tersebut. Faktanya, kita menunggu 2 detik setiap kali sebelum membuat permintaan GET untuk meningkatkan kemungkinan hasil tersedia untuk setiap panggilan yang kita buat. Jika Anda beralih ke paket berbayar, Anda dapat menghapus baris `time.sleep(6)` dari loop di Langkah 2 dengan aman.

Namun, saya sarankan untuk memberikan jeda singkat antara permintaan POST pertama dan GET, jika tidak, Anda akan mengalami banyak panggilan API yang kosong (tetapi tetap ditagih). Mengingat keterbatasan paket gratis, analisis keseluruhan untuk 162 dokumen PDF akan memakan waktu sekitar 30 menit. Jalankan sel kode. Dan sekarang saatnya rehat kopi sebelum kita melanjutkan pekerjaan kotor!

Memanggil layanan AI sejauh ini cukup mudah. Namun, mengekstrak informasi dari respons bisa jadi rumit. Untuk memahami alasannya, mari kita lihat seperti apa sebenarnya hasil dari AI. Untuk melakukan ini, kita tidak bisa begitu saja mencetak objek hasil. Objek hasil adalah struktur bersarang yang perlu kita uraikan. Untungnya, dokumentasi dengan contoh kode Python menunjukkan kepada kita cara melakukan ini:

```
# Inspect the first result
read_result = results[0]
# Print the detected text, line by line
if read_result.status == OperationStatusCodes.succeeded:
    for text_result in read_result.analyze_result.read_results:
        for line in text_result.lines:
            print(line.text)
            print(line.bounding_box)
```

Kode di atas akan menghasilkan keluaran berikut (dipotong demi singkatnya):



```
DB
[0.6349, 0.5273, 1.0222, 0.5165, 1.0222, 0.764, 0.6349, 0.764]
Online-Ticket
[3.0799, 0.5784, 4.6416, 0.5784, 4.6416, 0.7642, 3.0799, 0.7642]
ICE Fahrkarte
[0.5206, 1.1069, 1.4109, 1.1069, 1.4109, 1.2117, 0.5206, 1.2117]
...
Betrag
[0.5614, 3.5237, 0.873, 3.5237, 0.873, 3.6278, 0.5614, 3.6278]
92,00€
[1.109, 3.5237, 1.455, 3.5237, 1.455, 3.6197, 1.109, 3.6197]
...
Datum
[0.5621, 3.6571, 0.8733, 3.6571, 0.8733, 3.7384, 0.5621, 3.7384]
02.02.2018
...
Halt
[0.5207, 4.6438, 0.7449, 4.6438, 0.7449, 4.7352, 0.5207, 4.7352]
Datum
[2.4111, 4.6438, 2.7826, 4.6438, 2.7826, 4.7352, 2.4111, 4.7352]
Zeit
[2.9165, 4.6429, 3.1326, 4.6429, 3.1326, 4.7352, 2.9165, 4.7352]
Gleis
[3.4701, 4.6429, 3.7657, 4.6429, 3.7657, 4.7358, 3.4701, 4.7358]
Produkte
[4.104, 4.6438, 4.6302, 4.6438, 4.6302, 4.7352, 4.104, 4.7352]
Reservierung
[4.852, 4.6429, 5.6354, 4.6429, 5.6354, 4.7605, 4.852, 4.7605]
Hamburg Hbf
[0.5214, 4.8136, 1.2506, 4.8136, 1.2506, 4.932, 0.5214, 4.932]
02.02.
[2.4063, 4.8167, 2.7382, 4.8167, 2.7382, 4.9068, 2.4063, 4.9068]
ab 16:36 5
[2.9179, 4.8148, 3.5288, 4.8148, 3.5288, 4.9068, 2.9179, 4.9068]
ICE 1527
[4.1059, 4.8209, 4.6115, 4.8209, 4.6115, 4.9153, 4.1059, 4.9153]
1 Sitzplatz, Wg. 38, Pl. 12, 1 Fenster,
[4.8551, 4.8209, 6.8913, 4.8209, 6.8913, 4.9404, 4.8551, 4.9404]
Leipzig Hbf
[0.5213, 4.9636, 1.1351, 4.9636, 1.1351, 5.082, 0.5213, 5.082]
...
Seite 1 / 1
[7.1081, 11.2414, 7.5931, 11.2414, 7.5931, 11.3253, 7.1081,
11.3253]
```

Untuk memahami hal ini, mari kita bandingkan output dengan dokumen asli seperti yang ditunjukkan pada Gambar di bawah ini:



DB Online-Ticket

ICE Fahrkarte
 Fahrtantritt am 02.02.2018
 Flexpreis (Einfache Fahrt)
 Klasse: 1
 Erw: 1, mit 1 BC50
 Hinfahrt: Hamburg+City Leipzig+City, mit ICE
 Über: VIA: (LWL*WBE/UE*SDL)*BSP*(WB/P*KOET)
 Umtausch/Erstattung kostenlos bis 1 Tag vor Reiseantritt (Hinfahrt).

Zahlungspositionen und Preis

Positionen	Preis	MwSt (D) 19%	MwSt D: 7%
ICE Fahrkarte 1	92,00€	92,00€	14,69€
Reservierung 1	0,00€		
Summe	92,00€	92,00€	14,69€

Kreditkartenzahlung
 Betrag 92,00€ VU-Nr 4556895619 Transaktions-Nr 787489
 Datum 02.02.2018 Gen-Nr NBJNEB

Hinfahrt: 113494
 Gültig ab: 02.02.2018
 Zangenabdruck
 Herr Jens walter
 Auftragsnummer: 113494

Ihre Reiseverbindung und Reservierung Hinfahrt am 02.02.2018

Halt	Datum	Zeit	Gleis	Produkte	Reservierung
Hamburg Hbf	02.02.	ab 16:36	5	ICE 1527	1 Sitzplatz, Wg. 38, Pl. 12, 1 Fenster,
Leipzig Hbf	02.02.	an 19:42	11		Panorama / Lounge, Nichtraucher, Ruhebereich, Res.Nr. 8091 3012 3081 12

Wichtige Nutzungshinweise:

- Ihre Fahrkarte gilt nur zusammen mit einem amtlichen Lichtbildausweis (z.B. Personalausweis) oder Ihrer BahnCard.
- Mit Ihrer Fahrkarte zum Flexpreis können Sie jeden Zug der gewählten Verbindung nutzen: mit einer IC/EC-Fahrkarte alle IC- und EC-Züge, mit einer ICE-Fahrkarte auch alle anderen Züge.
- Das Online-Ticket gilt nur für den unter "Fahrkarte" angegebenen Reiseabschnitt. Die Übersicht "Ihre Reiseverbindung" enthält gegebenenfalls Reiseinformationen zu Teilstrecken (z.B. Bus oder Straßenbahn), für die eine weitere Fahrkarte erforderlich sein kann.
- Wenn Ihr Ticket den Zusatz "+City" oder "City mobil" zeigt, gilt dieser nur am Tag der Hinfahrt bzw. am Tag der Rückfahrt.
- Es gelten die nationalen und internationalen Beförderungsbedingungen der DB AG. Innerhalb von Verkehrsverbänden und Tarifgemeinschaften gelten deren Bedingungen. Alle Bedingungen finden Sie unter: www.bahn.de/agb und www.diebefoerderer.de.

Ihre Reisedaten können sich kurzfristig durch Bauarbeiten oder andere erforderliche Fahrplananpassungen ändern.
 Bitte informieren Sie sich kurz vor Ihrer Reise über mögliche Änderungen Ihrer Reisedaten unter www.bahn.de/reiseplan oder mobil über die App DB Navigator. Achten Sie auch auf Informationen und Ansagen im Zug und am Bahnhof. Wir danken Ihnen für Ihre Buchung und wünschen Ihnen eine angenehme Reise!

bahn.bonus
 Das Bonusprogramm der Bahn

113494

Seite 1 / 1

Pada kedua gambar, saya telah menyorot elemen-elemen dokumen yang kita minati (Tanggal, Harga, Asal, Tujuan).

Jika kita membandingkan keluaran AI dengan dokumen asli, kita dapat melihat bahwa AI mengurai dokumen baris demi baris. Dimulai dengan "DB" dan "Online-Tiket" dari kiri atas dan diakhiri dengan "Halaman 1/1" di kanan bawah.

Setiap baris teks merupakan elemen dari objek hasil, dan angka-angka yang mengikutinya merupakan kotak pembatas untuk baris-baris ini, yang menunjukkan posisi baris tersebut dalam dokumen. Keenam angka tersebut sesuai dengan koordinat x dan y seperti yang ditunjukkan pada Gambar di bawah dengan sistem koordinat yang dimulai dari kiri atas dokumen dengan koordinat (x = 0, y = 0). Setiap daftar koordinat dipetakan sebagai [x1, y1, x2,

y2, x3, y3, x4, y4]. Untuk mengekstrak informasi yang kita inginkan dari dokumen, kita perlu mengandalkan asumsi bahwa dokumen diurai dalam urutan bacaan barat (dari kiri ke kanan, atas ke tombol ini sebenarnya dapat disesuaikan untuk layanan AI jika diperlukan).



Mari kita mulai dengan informasi tanggal dengan menjalankan sel kode berikutnya yang berlabel "bagian 1" dari bagian 3. Dalam sel kode ini, sebagian besar pekerjaan dilakukan oleh fungsi kustom `get_text_by_keyword(result, "Datum")`. Fungsi yang didefinisikan di bagian 0 ini mengambil argumen kata kunci tertentu, dalam hal ini "Datum" (bahasa Jerman untuk "Tanggal") dan mengembalikan item yang mengikutinya tepat setelahnya dalam respons AI. Karena AI mengurai teks dari kiri ke kanan, hasilnya akan menjadi tanggal sebenarnya yang kita cari. Kita melakukan beberapa pemrosesan pasca untuk mengubahnya menjadi format tanggal yang lebih konvensional, dan selesai. Mengekstrak nilai formulir yang memiliki label formulir yang berbeda di sebelahnya cukup mudah.

Beralih ke bagian 2, mengekstrak informasi asal dan tujuan. Bagian ini sedikit lebih rumit karena informasinya tidak tercantum pada baris yang sama, melainkan pada baris yang berbeda dalam dokumen, mirip dengan struktur tabel. Lebih rumit lagi, daftar ini bisa lebih panjang jika terdapat kereta penghubung di antara asal dan tujuan akhir. Bagaimana kita mengatasi hal ini? Kita akan menggunakan pendekatan gabungan pencarian kata kunci dan filter kotak pembatas. Mari kita lihat sebelum menjalankan sel.

Pertama, kita memanggil fungsi `get_text_between_keywords(result, "Halt", "Wichtige Nutzungshinweise:")` yang akan memberikan kita semua elemen teks di antara penanda awal dan akhir. Dalam kasus kita, ini adalah tajuk kolom pertama tabel ("Halt") dan objek teks pertama yang mengikuti setelah tabel ("Wichtige Nutzungshinweise"). Fungsi ini akan mengekstrak semua elemen teks dalam tabel seperti yang ditunjukkan pada Gambar di bawah ini.

Ihre Reiseverbindung und Reservierung Hinfahrt am 02.02.2018

Halt	Datum	Zeit	Gleis	Produkte	Reservierung
Hamburg Hbf	02.02.	ab 16:36	5	ICE 1527	1 Sitzplatz, Wg. 38, Pl. 12, 1 Fenster,
Leipzig Hbf	02.02.	an 19:42	11		Panorama / Lounge, Nichtraucher, Ruhebereich, Res.Nr. 8091 3012 3081 12

Wichtige Nutzungshinweise:
- Ihre Fahrkarte gilt nur zusammen mit einem amtlichen Lichtbildausweis (= R. Personalausweis) oder Ihrer BahnCard

Namun, kita tidak menginginkan semua elemen teks, melainkan hanya item dari kolom pertama. Bagaimana cara mendapatkannya? Ada banyak cara untuk melakukannya, tetapi



menurut saya cara termudah adalah dengan mencari informasi ini menggunakan kotak pembatas masing-masing. Idenya adalah, dari fragmen teks yang baru saja kita kumpulkan, kita cukup mengekstrak bagian yang ditunjukkan oleh garis putus-putus pada Gambar di bawah ini.

Ihre Reiseverbindung und Reservierung Hinfahrt am 02.02.2018

Halt	Datum	Zeit	Gleis	Produkte	Reservierung
Hamburg Hbf	02.02.	ab 16:36	5	ICE 1527	1 Sitzplatz, Wg. 38, Pl. 12, 1 Fenster,
Leipzig Hbf	02.02.	an 19:42	11		Panorama / Lounge, Nichtraucher, Ruhebereich, Res.Nr. 8091 3012 3081 12

Wichtige Nutzungshinweise:

Jadi, bagaimana kita mendapatkan koordinat untuk kotak putus-putus ini? Jawabannya adalah kita cukup mengambil sudut kiri bawah tajuk kolom pertama "Halt" dan mendefinisikannya sebagai sudut kiri atas kotak putus-putus kita. Untuk mendapatkan lebar kotak, kita ambil koordinat x dari sudut kiri bawah teks "Datum", karena kita tahu ini akan selalu menjadi kolom kedua. Ini akan menjadi nilai x2 dari kotak putus-putus kita. Dan terakhir, untuk mendapatkan tinggi kotak putus-putus, kita cari koordinat y3 tertinggi di area teks yang telah kita ekstrak.

Koordinat ini saja yang kita butuhkan untuk menggambar persegi panjang. Setelah kita mendefinisikan kotak "filter", kita tinggal menyimpan hasil teks di mana kotak pembatas berada di dalam batas filter kita. Inilah yang ditangani oleh fungsi `get_text_by_position(result, filter_box)`. Fungsi ini akan mengembalikan daftar bersih dengan semua perhentian kereta. Sekarang, kita hanya perlu mengambil item pertama dan terakhir dari daftar ini dan voilà nilai asal dan tujuan kita sudah ada. Setelah memahami konsep yang terjadi di sini, jalankan sel kode untuk bagian 2.

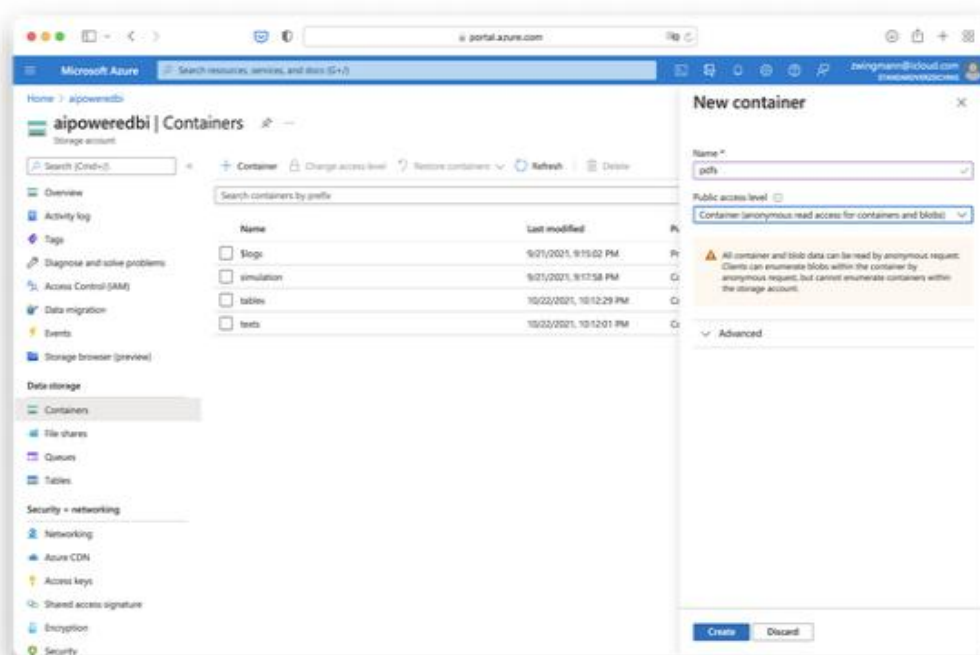
Bagian 3 sedikit lebih sederhana. Dalam kasus ini, kita mengekstrak informasi harga berdasarkan identifikasi kata kunci "Betrag" dalam dokumen, sangat mirip dengan cara kita mengekstrak tanggal. Satu-satunya perbedaan adalah kali ini kita tidak hanya mengekstrak item berikutnya setelah kata kunci, tetapi juga dua item berikutnya setelah kata kunci tersebut, seperti yang dapat Anda lihat pada parameter "2" dalam fungsi `get_text_by_keyword(result, "Betrag", 2)`. Mengapa demikian? Jawaban singkatnya adalah: Hal ini membuat skrip kita sedikit lebih tangguh. Kita dapat mengidentifikasi informasi harga dengan aman berdasarkan tanda "€" dan kita tahu bahwa tanda tersebut seharusnya berada tepat setelah label "Betrag", tetapi tidak harus item pertama, bisa juga item kedua. Lanjutkan dan jalankan sel tersebut.

Setelah menjalankan bagian ketiga, hanya tersisa satu langkah untuk transformasi data, yaitu menggabungkan hasil kita dan menulis tabel CSV. Sekarang Anda seharusnya memiliki 4 objek daftar: tanggal, harga, asal, dan tujuan, semuanya memiliki panjang yang sama. Pada langkah terakhir ini, kita akan menggabungkannya ke dalam kerangka data dan mengekspornya sebagai berkas csv. Jalankan sel terakhir ini dan Anda akan melihat berkas baru `public-transportation-costs.csv` di penjelajah berkas di sebelah kiri setelah penyegaran cepat. Tabel hasil ditunjukkan pada Gambar di bawah ini.



	Date	Origin	Destination	Cost	Receipt
0	2017-10-29	München Hbf	Hamburg Hbf	122.00	KKZ2A1.pdf
1	2018-06-08	Leipzig Hbf	München Hbf	101.25	8M1XML.pdf
2	2018-05-21	Leipzig Hbf	Hamburg Hbf	92.00	SYGLKK.pdf
3	2018-07-21	Bonn-Oberkassel Nord	Frankfurt(M) Flughafen Fernbf	56.10	bahn-B3QNOU.pdf
4	2018-04-06	Hamburg Hbf	Leipzig Hbf	89.50	OG4EB6.pdf
...
157	2020-01-11	Leipzig Hbf	Hamburg Hbf	81.90	bahn-17B6DU.pdf
158	2018-10-02	Konstanz	Bonn Hbf (tief)	109.10	bahn-UKW9RT.pdf
159	2018-11-18	Leipzig Hbf	Hamburg Hbf	89.50	bahn-G4SUFW.pdf
160	2017-10-22	Leipzig Hbf	Hamburg Hbf	87.75	GBEOH3.pdf
161	2017-11-12	Bonn Hbf	Hamburg Hbf	77.75	vh18hb.pdf

162 rows × 5 columns



Terakhir, mari unggah hasil kita ke kontainer bernama "tabel" di Azure Blob Storage. Sel kode terakhir di bagian 4 akan menanganinya untuk Anda. Jalankan sel ini dan beri catatan kecil di bagian belakang: Bagian terbesar dari pekerjaan telah selesai. Sekarang kita dapat melanjutkan dan memvisualisasikan hasilnya di BI kita.

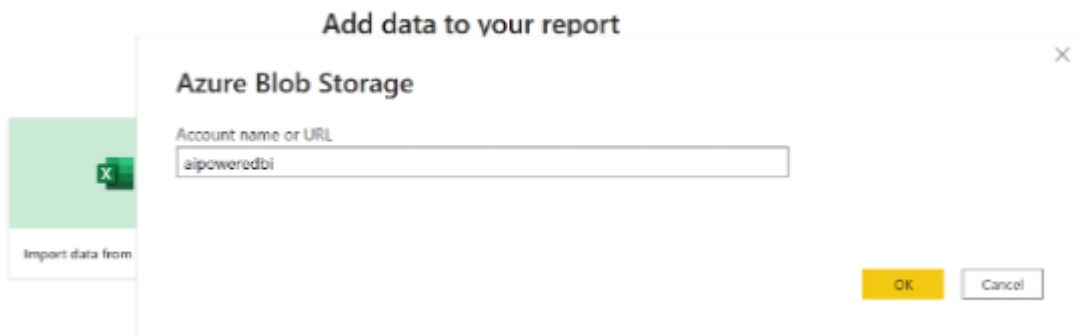
CATATAN

Jangan lupa untuk menjeda atau menghapus sumber daya komputasi Anda dari notebook!

Memvisualisasikan Hasil di BI

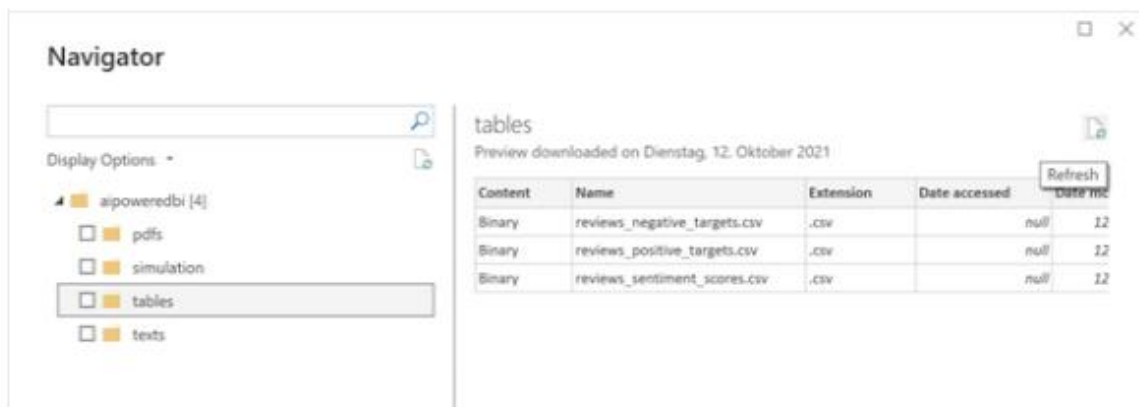
Mari kita lanjutkan ke bagian yang menyenangkan. Buka BI favorit Anda dan muat file CSV yang baru saja kita buat untuk mendapatkan lebih banyak wawasan. Saya akan memandu Anda melalui prosesnya lagi menggunakan contoh Power BI.

Di Power BI, buat laporan baru dan pilih "Azure Blob Storage" dari "Dapatkan data". Berikan nama akun Azure Blob Storage Anda seperti yang ditunjukkan pada Gambar di bawah ini.



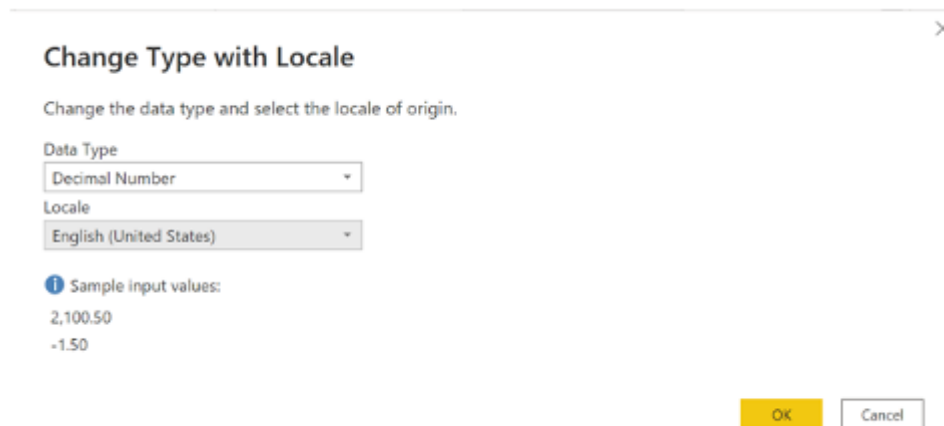
Anda akan melihat daftar file penyimpanan blob Anda. Buka folder "tables" dan klik "Refresh" seperti yang terlihat pada Gambar di bawah jika Anda masih tidak dapat melihat file baru `public-transportation-costs.csv`.

Centang kotak pada folder tables dan klik "Transform Data" yang akan membuka Power Query. Klik tautan "Binary" di samping nama file CSV "`public-transportation-costs.csv`".

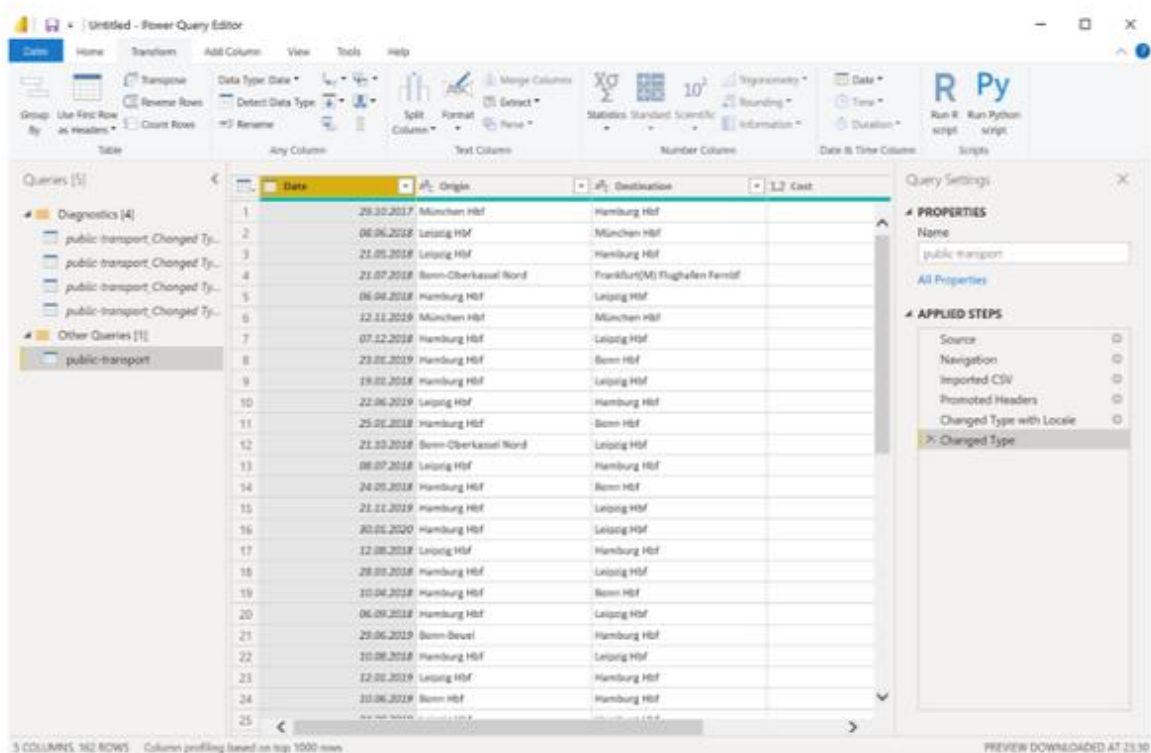


Ada dua hal yang perlu Anda pastikan di sini: Pertama, kolom tanggal diidentifikasi dengan benar sebagai tipe data "Tanggal", dan kedua, harga benar-benar dikonversi menjadi nilai numerik. Jika Power BI tidak secara otomatis mengenali tipe data yang benar, klik kanan pada kolom "Tanggal" dan pilih "Tanggal" seperti yang ditunjukkan pada Gambar di bawah ini.

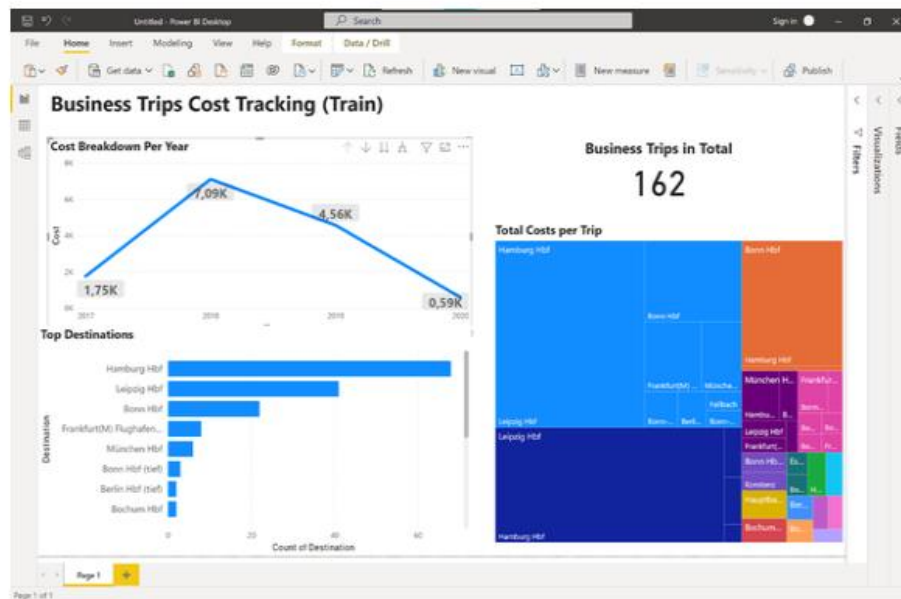




Terakhir, set data Anda akan terlihat seperti Gambar di bawah. Tutup Power Query dengan memilih "Tutup & Terapkan".



Buka laporan dan buat beberapa visual baru. Saya memutuskan untuk menampilkan total biaya perjalanan per tahun dalam diagram garis, jumlah total perjalanan bisnis sebagai metrik, destinasi teratas sebagai diagram batang horizontal, dan terakhir, semua rute (kombinasi asal dan tujuan) sebagai peta pohon dengan ukuran yang sama dengan jumlah uang yang dikeluarkan seperti yang ditunjukkan pada Gambar di bawah ini.



Dengan visual ini, tim manajemen perjalanan dapat melihat dengan jelas tren biaya keseluruhan dan jumlah perjalanan absolut. Mereka juga dapat mengidentifikasi rute mana yang paling berkontribusi terhadap total biaya perjalanan. Dalam kasus kami, kami dapat melihat bahwa koneksi dari Hamburg ke Leipzig dan sebaliknya menyumbang proporsi biaya perjalanan terbesar. Kami juga dapat melihat bahwa sebagian besar perjalanan sebenarnya dimulai di Hamburg, dengan Bonn dan Frankfurt menjadi tujuan terpopuler kedua dan ketiga dari titik asal ini.

Silakan buat ulang dasbor ini sendiri atau lihat apakah Anda dapat menemukan cara yang lebih baik untuk menampilkan data. Jika Anda ingin melihat dasbor final yang saya buat, Anda dapat mengunduh berkas [Chapter-8-Use-Case-Document-Analytics.pbix](#) dari repositori buku.

9.3 KASUS PENGGUNAAN: MENDETEKSI OBJEK DALAM GAMBAR

Pada kasus penggunaan sebelumnya, kita hanya menyentuh kemampuan visi komputer yang luas. Dalam skenario berikut, saya ingin menunjukkan kepada Anda bagaimana Anda dapat menggunakan infrastruktur yang ada lebih jauh, yaitu untuk mendeteksi objek dalam gambar. Untuk itu, mari kita lihat lebih dekat permasalahan yang ada.

Pernyataan Masalah

Kami bekerja untuk sebuah badan usaha milik pemerintah yang bergerak di bidang transportasi dan jalan raya yang berupaya meningkatkan manajemen lalu lintas dan arus lalu lintas secara keseluruhan untuk meminimalkan kemacetan jalan. Salah satu alat penting adalah mengendalikan batas kecepatan di jalan raya. Untuk menetapkan batas kecepatan, badan usaha tersebut membutuhkan pengukuran arus lalu lintas yang idealnya konstan di jalan raya.



Meskipun jalan baru dilengkapi dengan sensor dan teknologi pengukuran yang sesuai, jalan lama seringkali hanya dilengkapi dengan kamera CCTV yang digunakan oleh petugas lalu lintas untuk inspeksi lalu lintas manual. Tim operasional menghubungi kami untuk memeriksa apakah mungkin untuk menghitung lalu lintas menggunakan infrastruktur kamera yang ada dan telah memberikan beberapa contoh rekaman CCTV untuk keperluan pengujian. Gambar di bawah ini menunjukkan contoh rekaman CCTV.



Citra CCTV untuk kasus penggunaan ini disediakan melalui kumpulan data di Kaggle (citra rekaman CCTV Jalan Raya).

Ikhtisar Solusi

Untuk mengatasi masalah yang dihadapi, kami membangun layanan AI visi komputer yang sudah ada dari kasus penggunaan sebelumnya. Alih-alih operasi "Baca", kami sekarang memanggil layanan yang disebut "Deteksi Objek". Dalam hal ini, kami ingin menganalisis data secara langsung.

Jadi, alih-alih memuat kumpulan data ke penyimpanan Azure Blob, kami menggunakan rekaman CCTV langsung dari URL web dan akan memasukkan URL web ini ke layanan AI. Kami masih akan memiliki tugas ETL kecil yang menggunakan URL http, memanggil layanan AI, dan menulis output sebagai file CSV datar ke penyimpanan Azure Blob sehingga kami dapat menggunakannya dengan alat BI kami. Mari kita mulai!

Menyiapkan Layanan AI

Kami menggunakan layanan AI yang sama yang kami terapkan sebelumnya dalam kasus penggunaan untuk mengurai dokumen. Jika Anda belum menyelesaikannya, buka bagian sebelumnya, selesaikan langkah-langkah yang tercantum di bawah "Menyiapkan layanan AI", dan kembali ke sini. Jika Anda sudah melakukannya, ikuti saja langkah-langkahnya - tidak ada lagi yang perlu Anda lakukan. Ternyata, visi komputer memiliki banyak kasus penggunaan berbeda yang berjalan di bawah layanan yang sama. Dengan layanan visi komputer yang baru saja Anda terapkan, Anda tidak hanya dapat mengekstrak teks atau mendeteksi objek, tetapi juga mengenali nama merek, landmark populer, menandai gambar, dan banyak lagi. Untuk panduan lengkap, lihat dokumentasi lengkap Visi Komputer untuk Azure Cognitive Services.

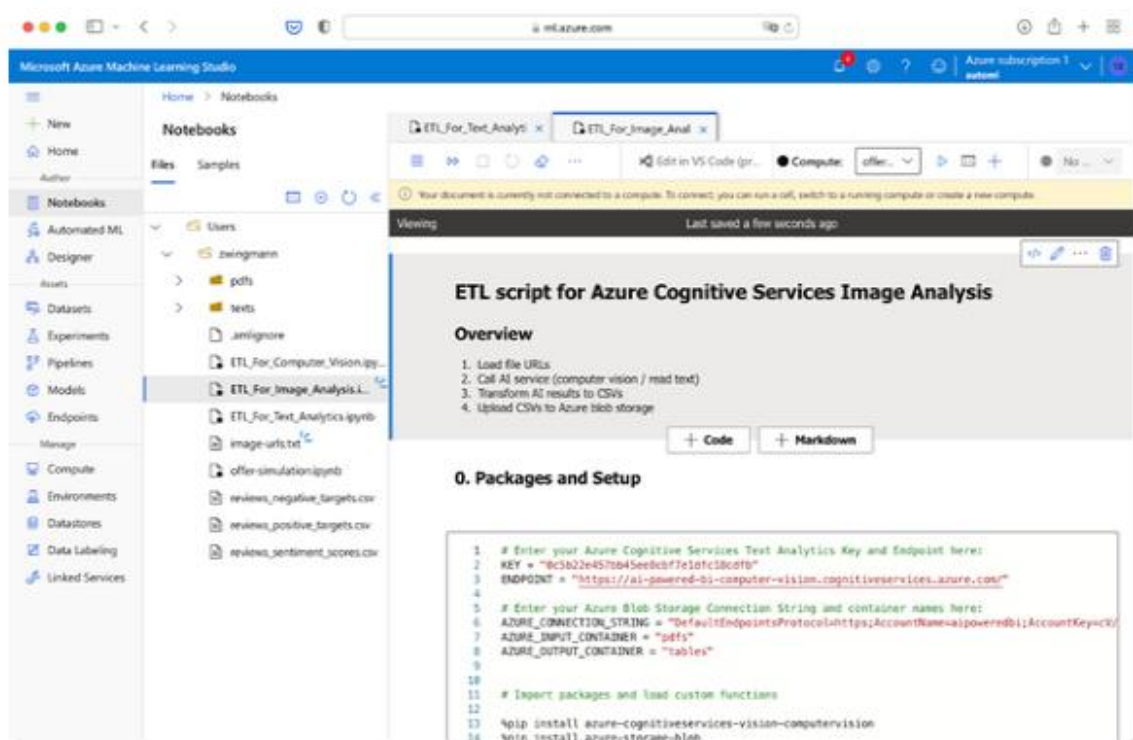
Menyiapkan Alur Data

Unduh berkas "ETL_For_Image_Analysis.ipynb" dari repositori buku. Sama seperti sebelumnya, buka berkas tersebut di IDE lokal Anda atau unggah ke Azure Notebooks. Selain



berkas ini, Anda juga memerlukan berkas “image-urls.txt” yang harus ditempatkan di direktori yang sama dengan berkas notebook.

Jika Anda membuka berkas ETL_For_Image_Analysis.ipynb, Anda akan melihat bahwa berkas tersebut terlihat sangat mirip dengan yang kita gunakan sebelumnya seperti yang Anda lihat pada Gambar di bawah ini. Saya akan memandu Anda melalui langkah-langkah utamanya secara singkat.



Di bagian 0, perbarui kredensial Azure Anda. Ini sama seperti yang Anda gunakan pada kasus penggunaan sebelumnya. Setelah Anda memasukkannya, jalankan sel kode pertama.

Di bagian 1, jalankan sel kode untuk mengimpor berkas txt dengan URL gambar. Pastikan Anda mengunduh berkas dari repositori buku dan meletakkannya di folder yang sama dengan berkas .ipynb.

Bagian 2 memanggil layanan AI visi komputer. Berbeda dengan kasus penggunaan sebelumnya, deteksi objek adalah operasi sinkron, jadi kita sebenarnya hanya memerlukan satu panggilan API per gambar dan dapat mengurangi waktu tunggu menjadi 3 detik setelah setiap gambar agar tetap berada dalam batas 20 gambar/menit untuk tingkat gratis. Jika Anda menggunakan tingkat berbayar, Anda dapat menghapus baris `time.sleep(3)` dengan aman dari skrip. Jalankan sel kode. Sambil menunggu hasilnya, ini adalah waktu yang tepat untuk memperhatikan beberapa batasan atau prasyarat untuk layanan AI deteksi objek dan ini juga berlaku kurang lebih untuk penyedia lain, tidak hanya untuk Microsoft Azure. Seperti yang dapat kita baca dalam dokumentasi Azure, batasan berikut berlaku:

Penting untuk memperhatikan keterbatasan deteksi objek agar Anda dapat menghindari atau mengurangi efek negatif palsu (objek yang terlewat) dan detail yang terbatas.

- Objek umumnya tidak terdeteksi jika ukurannya kecil (kurang dari 5% gambar).
- Objek umumnya tidak terdeteksi jika disusun berdekatan (misalnya, tumpukan piring).
- Objek tidak dibedakan berdasarkan merek atau nama produk (misalnya, berbagai jenis soda di rak toko). Namun, Anda bisa mendapatkan informasi merek dari sebuah gambar dengan menggunakan fitur Deteksi Merek.

```
1 # Print a sample result
2 results[0].as_dict()
✓
{'metadata': {'format': 'Jpeg', 'height': 400, 'width': 900},
 'model_version': '2021-04-01',
 'objects': [{'confidence': 0.862,
  'object_property': 'car',
  'parent': {'confidence': 0.872,
  'object_property': 'Land vehicle',
  'parent': {'confidence': 0.872, 'object_property': 'Vehicle'}},
  'rectangle': {'h': 89, 'w': 107, 'x': 521, 'y': 185}},
 {'confidence': 0.729,
  'object_property': 'car',
  'parent': {'confidence': 0.744,
  'object_property': 'Land vehicle',
  'parent': {'confidence': 0.744, 'object_property': 'Vehicle'}},
  'rectangle': {'h': 135, 'w': 122, 'x': 336, 'y': 263}}],
 'request_id': 'a0248e40-98a8-4dd0-bd4e-0d86e3f82a1e'}
```

Yang terpenting, kita harus mempertimbangkan poin pertama. Pada dasarnya, kita harus memastikan bahwa objek yang diinginkan cukup besar dibandingkan dengan ukuran keseluruhan gambar. Jika Anda memperhatikan URL gambar dalam berkas teks, Anda akan melihat bahwa kami sebenarnya tidak meneruskan rekaman mentah ke layanan AI, melainkan memotong gambar-gambar ini secara langsung menggunakan jaringan pengiriman konten (CDN). Dengan cara ini, kita dapat berfokus pada bagian gambar yang benar-benar penting bagi kita dan memastikan bahwa objek mobil potensial mencakup lebih dari 5% area gambar. Jadi, jika Anda melihat hasil yang buruk dengan layanan AI deteksi objek, cobalah memotong gambar sebelumnya atau membaginya menjadi beberapa bagian.

Sekarang, layanan AI seharusnya sudah berfungsi dan Anda dapat menjalankan sel kode berikut untuk mencetak contoh hasil seperti yang ditunjukkan pada Gambar di bawah. Seperti yang Anda lihat, layanan AI tidak hanya memberi kita nama objek yang terdeteksi, tetapi juga konteks tambahan, seperti posisi objek dalam gambar dan beberapa informasi semantik tambahan.

Mari kita lanjutkan ke bagian 3 transformasi data, di mana kita ingin mengurai hasil ini untuk mendapatkan jumlah kendaraan sebenarnya dalam gambar. Pendekatannya cukup sederhana. Kita memanggil fungsi `count_objects(result, "Mobil", 0.7)` yang memfilter keluaran berdasarkan dua kriteria: nama properti objek dan skor keyakinan deteksi.



Saya memilih 0.7 yang diterjemahkan menjadi sesuatu seperti "layanan AI 70% yakin bahwa objek yang terdeteksi adalah mobil". Silakan bereksperimen dengan nilai ini dan cari tahu titik optimal untuk kasus penggunaan Anda. Jalankan sel kode ini dan Anda akan melihat daftar berisi jumlahnya. Satu-satunya tugas yang tersisa adalah menyusunnya ke dalam format tabel yang baik, yang juga menyertakan URL berkas gambar asli untuk referensi kembali ke sumber data, serta indeks yang memudahkan pengurutan data. Jalankan sel kode kedua di bagian 3, dan Anda akan melihat keluaran yang mirip dengan Gambar di bawah.

```
1 # Create the final data frame for CSV export
2 output_df = pd.DataFrame({"Count": counts, "URL": urls})
3 output_df.to_csv("traffic_counts.csv", index = True)
4 output_df
```

	Count	URL
0	2	https://ucarecdn.com/de191530-3136-4b1d-818e-9...
1	2	https://ucarecdn.com/bb436a7e-b384-439b-857e-b...
2	2	https://ucarecdn.com/f3c419a2-5a31-48ef-9945-9...
3	1	https://ucarecdn.com/77deacb2-b5f4-439f-a1fe-b...
4	3	https://ucarecdn.com/cf8e7f6c-c172-4110-8c9f-8...
...
90	1	https://ucarecdn.com/0828690d-7c57-469b-85aa-9...
91	3	https://ucarecdn.com/5b2b1766-ac27-4081-a7e0-7...
92	2	https://ucarecdn.com/eb7efb73-5748-4061-a089-f...
93	4	https://ucarecdn.com/9af57430-1cba-4b7c-ad37-0...
94	3	https://ucarecdn.com/ccdc77ba-4946-43a6-a298-b...

95 rows x 2 columns

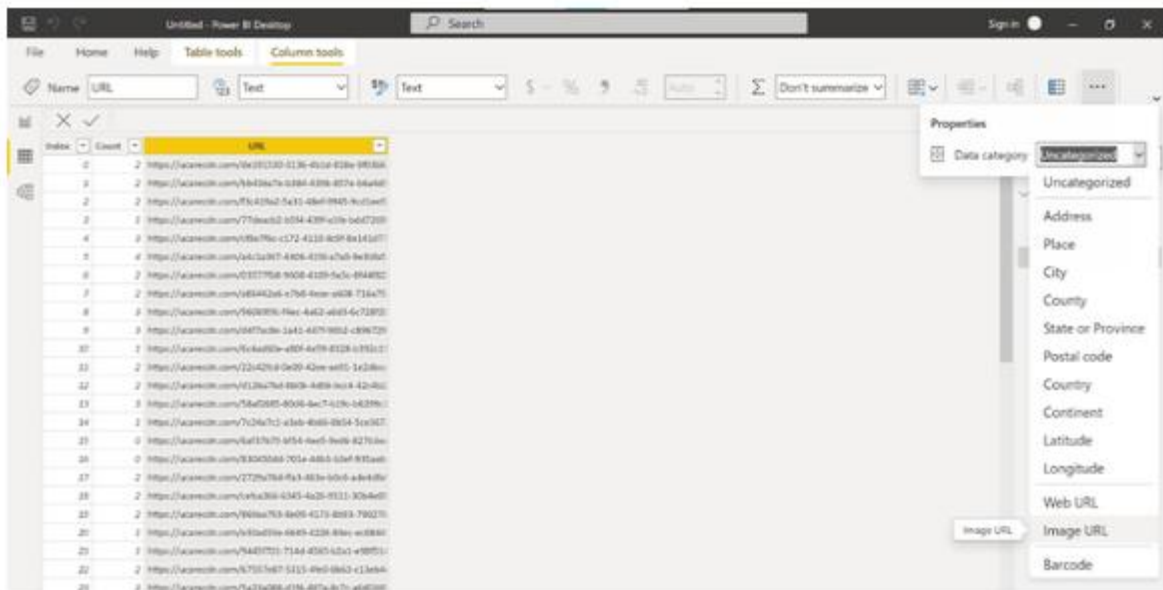
Terakhir, jalankan sel kode terakhir di bagian 4 berikutnya untuk mengunggah berkas CSV ke penyimpanan Azure Blob kita. Dan hanya itu yang perlu kita lakukan untuk prototipe pekerjaan ETL kita. Mari kita lanjutkan memvisualisasikan hasilnya di BI!

Memvisualisasikan Hasilnya di BI

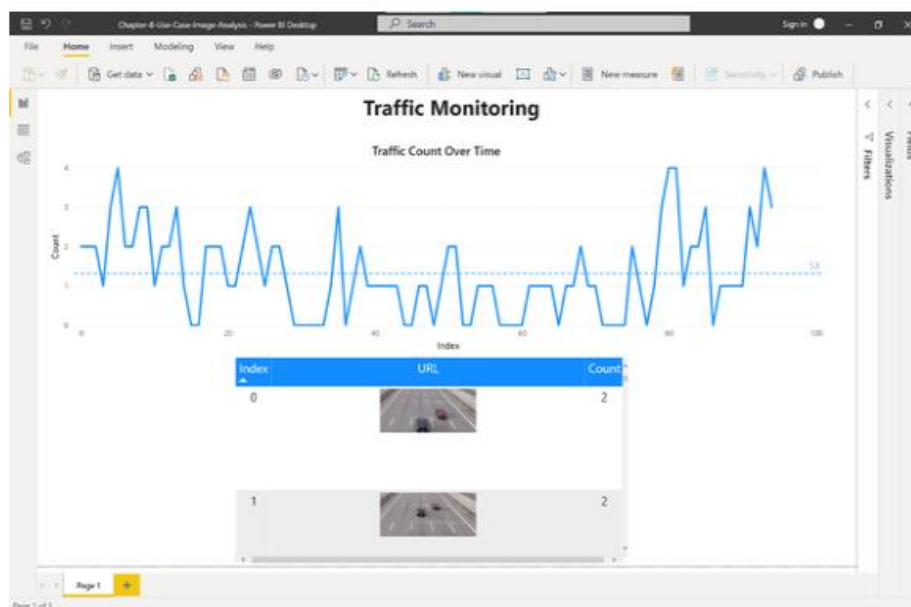
Karena kita telah menyimpan hasil AI dalam format CSV datar, alat BI apa pun dapat menggunakan data ini. Saya akan menunjukkan lagi bagaimana saya melakukannya dengan Power BI dan akan ada sedikit kejutan!

Pertama, buat berkas Power BI baru dan pilih Dapatkan Data → Azure Blob Storage, masukkan nama akun penyimpanan Anda, lalu pilih kontainer bernama "tabel". Klik "Transformasi" dan klik tautan "Biner" di baris tempat nama tabel baru "traffic_counts.csv" ditampilkan. Jika Anda tidak dapat melihat berkas ini, segarkan pratinjau di Power Query.





Sekarang, mari kita beralih ke halaman laporan. Dalam kasus saya, saya hanya memilih dua komponen: diagram garis untuk penghitungan dari waktu ke waktu (dalam kasus kami, ini adalah indeks) dan tabel yang menampilkan penghitungan, indeks, dan URL gambar. Karena kita telah memformat kolom URL sebagai tipe data "URL Gambar", Power BI akan mengambil gambar dari URL ini dan menampilkannya di dasbor kita. Luar biasa, bukan? Lihat Gambar di bawah ini untuk melihat dasbor akhirnya.



Tentu saja, tabel ini mewarisi interaktivitas penuh yang Anda harapkan dari visual Power BI. Jadi, ketika Anda memilih, misalnya, satu titik dengan jumlah lalu lintas tertinggi pada indeks



80, tabel tersebut akan menampilkan gambar yang relevan untuk titik data ini seperti yang ditunjukkan Gambar di bawah ini. Dengan cara ini, kita juga dapat memastikan bahwa deteksi AI benar-benar akurat dengan mendeteksi 4 mobil pada gambar.



Silakan buat ulang dasbor ini sendiri atau modifikasi sesuka Anda.

Anda juga dapat menemukan versi finalnya di repositori buku dengan mengunduh berkas "Chapter-8-Use-Case-Image-Analysis.ipynb"

9.4 KESIMPULAN

Bab ini baru menyentuh permukaan tentang apa yang dapat dilakukan layanan AI dengan sumber data yang bukan berformat CSV atau Excel biasa. Saya harap Anda bersenang-senang mencoba layanan AI ini dan saya lebih berharap Anda mendapatkan lebih banyak inspirasi tentang cara menggunakan alat-alat ini untuk kasus penggunaan Anda sendiri. Bab ini menyimpulkan fondasi layanan AI yang dapat memberdayakan BI Anda. Di bab selanjutnya, kita tidak akan memperkenalkan alat baru, tetapi kita akan melihat bagaimana kita dapat menggabungkan apa yang telah kita pelajari untuk membangun dasbor BI bertenaga AI yang berfungsi penuh dan terintegrasi.



BAB 10

MEMBANGUN DASBOR ANALISIS PELANGGAN BERTENAGA AI

10.1 PERNYATAAN MASALAH

Dalam skenario ini, kami adalah bagian dari tim analitik data sebuah penyedia telekomunikasi. Kepala penjualan dan pemasaran divisi konsumen telah memulai sebuah proyek yang seharusnya membahas lebih lanjut topik churn pelanggan. Sesuai definisi bisnis, churn terjadi ketika pelanggan membatalkan kontrak mereka, terlepas dari berapa lama sisa durasi kontrak (perusahaan menawarkan kontrak bulanan dan 24 bulan).

Bisnis ini saat ini menghadapi tantangan berikut:

- Tingkat churn telah diukur, tetapi pemasaran dan penjualan belum dapat memahaminya. Metrik churn tampaknya naik turun secara sporadis dan mereka kesulitan mendapatkan wawasan yang berarti dari metrik ini.
- Sebagai langkah efektif untuk mengatasi churn, bisnis ini telah mengidentifikasi penawaran balasan sebagai strategi yang layak. Penawaran diberikan kepada pelanggan yang memutuskan kontrak sebagai cara untuk menarik mereka kembali atau mencegah mereka berhenti sama sekali. Penawaran tersebut ternyata efektif, tetapi mahal. Oleh karena itu, bisnis ini ingin mengetahui pelanggan mana yang paling menguntungkan untuk ditargetkan oleh penawaran balasan churn tersebut dan idealnya ingin memprediksi churn sebelum terjadi untuk memastikan kesesuaian yang baik antara jenis penawaran dan segmen pelanggan.
- Bisnis ini menjalankan survei rutin di antara pelanggan yang mencakup banyak umpan balik dalam bentuk teks terbuka. Dengan melihat beberapa sampel data survei, tim survei menunjukkan bahwa jawaban dalam bentuk teks dapat memberikan sinyal penting untuk pemodelan churn.
- Bisnis ini mengharapkan tingkat interaktivitas tertentu untuk menyisir data guna mendapatkan gambaran tentang apa yang terjadi terkait churn pelanggan di berbagai segmen pelanggan.

Kami sekarang menjadi bagian dari tim proyek dan diharapkan dapat mengidentifikasi cara-cara yang tepat untuk menyajikan informasi yang diminta kepada para pemangku kepentingan bisnis tentu saja sesegera mungkin dan dengan keterbatasan anggaran yang besar.

10.2 IKHTISAR SOLUSI

Untuk mengatasi masalah di atas, kami akan menggunakan kombinasi data BI kami yang sudah ada (yang dapat berasal dari gudang data, tetapi dalam kasus kami dikirimkan sebagai file CSV).



Data BI tersebut berisi informasi dimensional tentang pelanggan seperti demografi, layanan, metode pembayaran, dll., serta tabel fakta dengan informasi seperti masa berlaku pelanggan, biaya bulanan, dan jenis kontrak.

Tabel fakta yang kami sediakan mencakup rentang waktu selama 5 bulan terakhir dan sudah menyertakan tanda churn yang telah dihitung oleh analisis internal kami. Selain label churn, tabel fakta juga berisi tanda apakah pelanggan menerima counter-churn atau tidak di masa lalu. Terkait data survei, tim telah mengirimkan kepada kami dump CSV berisi hasil survei terbaru dari beberapa bulan terakhir, termasuk jawaban teks terbuka dan ID pelanggan terkait.

Dengan data ini, kami akan mencoba membangun prototipe dasbor BI yang menyediakan informasi sesuai permintaan tim penjualan dan pemasaran. Inti dari dasbor ini adalah kami akan menggunakan model prediktif yang menghitung risiko churn individual untuk setiap pelanggan berdasarkan data historis.

Untuk memahami jawaban teks terbuka, kami akan menjalankan analisis sentimen pada umpan balik pelanggan dan melihat apakah label sentimen ini akan membantu kami meningkatkan kinerja model prediktif. Untuk mengujinya, kami akan membangun dua model satu yang menyertakan data sentimen dan satu yang tidak, sehingga kami dapat membandingkan hasilnya.

Alih-alih menangani alur kerja secara manual, kami akan menggunakan perangkat lunak bernama Azure ML Designer yang merupakan bagian dari langganan Azure kami. ML Designer adalah platform AI tanpa kode yang memungkinkan kami membangun dan menerapkan alur kerja pembelajaran mesin terkelola di cloud Azure dengan seminimal mungkin hambatan teknis. Terakhir, kami akan menggabungkan hasilnya di alat BI kami. Dalam kasus kami, itu lagi-lagi Power BI, tetapi kami akan berusaha melakukan pekerjaan berat di luar Power BI sebisa mungkin agar kompatibel dengan platform BI lainnya.

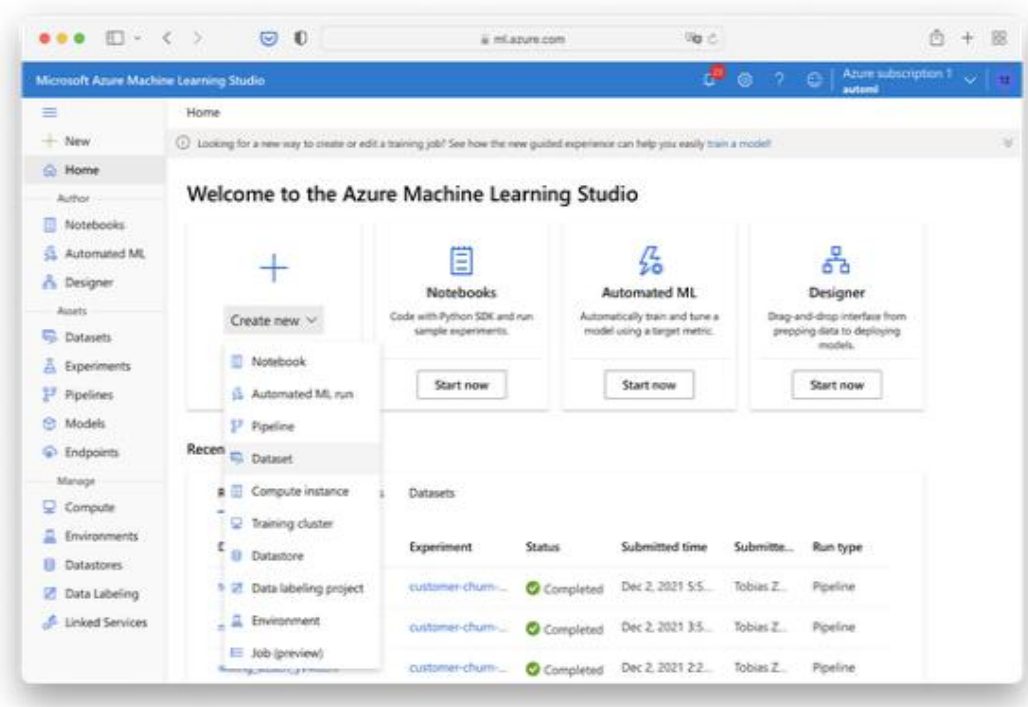
10.3 MEMPERSIAPKAN SET DATA

Unduh berkas berikut dari repositori buku:

- `customers_factTable_Jan-May_2022.csv` - Berkas berisi informasi tentang metrik pelanggan dan label churn untuk Januari - Mei 2022.
- `survey_responses_Jan-May_2022.csv` - Survei pelanggan terhadap sampel pelanggan yang dilakukan antara Januari dan Mei 2022 yang berisi umpan balik pelanggan.
- `customers_factTable_June_2022.csv` - Berkas berisi fakta pelanggan untuk bulan Juni 2022 saat ini.
- `survey_responses_June_2022.csv` - Data survei baru dari Juni 2022.

Buka Microsoft Azure Machine Learning Studio, pilih ruang kerja yang Anda inginkan, dan Anda akan melihat dasbor seperti yang ditunjukkan pada Gambar di bawah ini.





Gambar 10.1. Dasbor Azure Machine Learning Studio

Pertama, kita akan mengunggah keempat file CSV sebagai Dataset ke Azure ML Studio satu per satu agar dapat diakses melalui ML Designer nanti.

Klik ikon "+" dan pilih Dataset atau pilih Dataset dari menu di sebelah kiri.

Untuk setiap file CSV, pilih "Buat dataset" → dari file lokal. Berikan nama untuk dataset seperti pada nama file, misalnya "customers_factTable_Jan-May_2022" untuk file customers_factTable_Jan-May_2022.csv. Ikuti formulirnya, unggah file CSV, dan periksa pratinjau data jika semuanya sudah sesuai. Pembatas di semua file harus berupa koma dan pengodeannya adalah UTF-8.

Untuk skema, Anda perlu melakukan beberapa penyesuaian. Pastikan skema untuk tabel fakta pelanggan adalah sebagai berikut (Catatan: Kolom Churn dan OfferAccepted tidak muncul dalam berkas dari bulan Juni karena ini mewakili bulan terbaru dan informasi churn belum tersedia itulah yang akan kita prediksi sebentar lagi):

- customerID: String
- tenure: Integer
- Contract: String
- Monthly Charges: Desimal (titik)
- Churn: String
- Don't include: OfferAccepted
- Don't include: Month

Untuk kedua berkas survei, pastikan kolomnya sesuai dengan skema berikut:

- id: String
- text: String

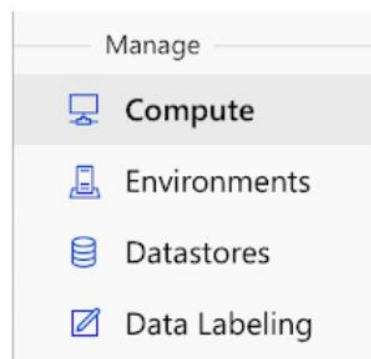
Setelah ini, Anda akan melihat empat set data berikut di dalam Machine Learning Studio:

- customers_factTable_Jan-May_2022
- survey_responses_Jan-May_2022
- customers_factTable_June_2022
- survey_responses_June_2022

Mengalokasikan Sumber Daya Komputasi

Setelah data tersedia, kita perlu memastikan bahwa ada Beberapa sumber daya tersedia di mana komputasi aktual dapat dilakukan. Untuk menambahkan sumber daya komputasi, pilih Compute dari menu di sebelah kiri Azure Machine Learning Studio seperti yang ditunjukkan pada Gambar 10.2.

Di sini, Anda seharusnya masih melihat sumber daya komputasi yang telah Anda gunakan sebelumnya di seluruh bab. Jika Anda melihat sumber daya tersebut, tetapi belum dimulai, pilih sumber daya tersebut dan klik Mulai. Jika Anda belum memiliki sumber daya atau telah menghapusnya sebelumnya, klik Baru dan buat sumber daya komputasi baru dengan pengaturan default dan jenis mesin termurah (STANDARD_DS1_V2) yang akan memadai untuk studi kasus ini. Setelah Anda memiliki setidaknya satu instans komputasi yang aktif dan berjalan, Anda dapat melanjutkan ke ML Designer.

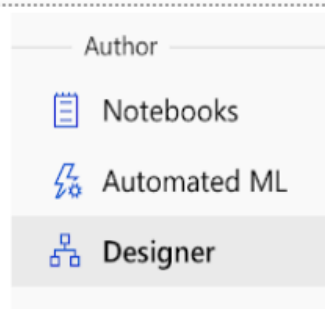


Gambar 10.2. Sumber Daya Komputasi di Azure ML Studio

Membangun Alur Kerja ML

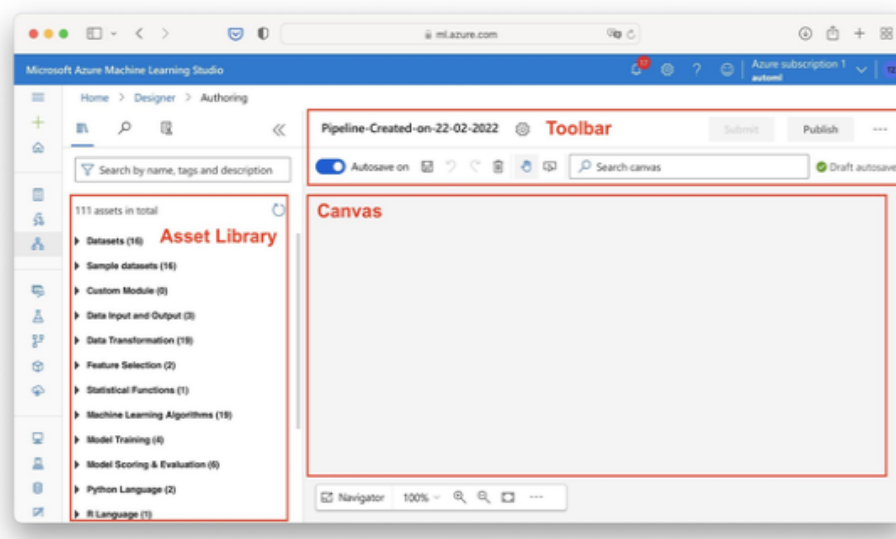
Lanjutkan dengan memilih Designer dari menu di sebelah kiri seperti yang terlihat pada Gambar 10.3. ML Studio Designer akan menjadi antarmuka seret dan lepas untuk membangun apa yang disebut alur kerja pembelajaran mesin. Kita dapat melatih model kita sendiri, tetapi kita juga dapat melakukan beberapa prapemrosesan data dasar dan menjalankan skrip khusus.





Gambar 10.3. ML Designer di Azure ML Studio

Buat pipeline pertama Anda dengan mengklik "Pipeline baru". Anda sekarang akan melihat kanvas kosong untuk ML Studio Designer seperti yang ditunjukkan pada Gambar 10.4. ML Studio Designer memiliki komponen-komponen utama: Pustaka aset di sebelah kiri yang berisi modul siap pakai, kanvas di sebelah kanan untuk membangun pipeline Anda, dan bilah alat di atas untuk Menyimpan, Mengedit, dan Mempublikasikan seluruh pipeline.



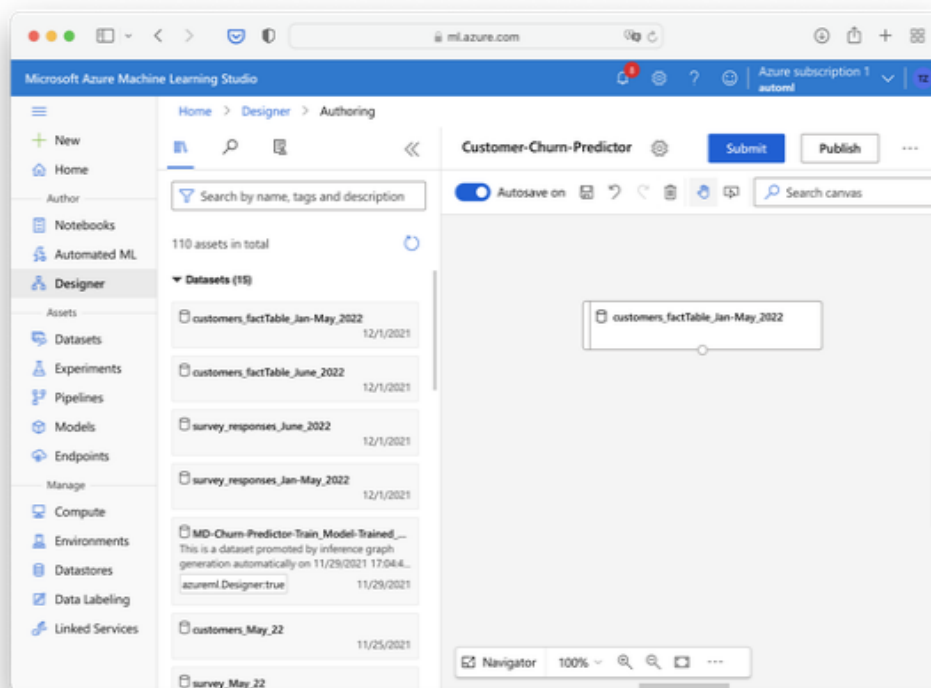
Gambar 10.4. Antarmuka Azure ML Designer

Sebelum memulai, tetapkan instans komputasi yang sedang berjalan ke sesi kita. Jika jendela pengaturan tidak muncul secara otomatis, Anda dapat mengaksesnya dengan mengklik ikon roda gigi di samping nama pipeline pada bilah alat. Untuk nama pipeline, klik dua kali judul pipeline dan ganti namanya menjadi "Customer-Churn-Predictor".

Sekarang kita dapat mulai mengisi kanvas kosong. Intinya adalah Anda membangun alur kerja kustom menggunakan modul bawaan di sebelah kiri dan menggabungkannya di kanvas. Setiap modul dapat memiliki port input dan output serta menyediakan pengaturan yang berbeda. Pilih Dataset dari bilah alat di sebelah kiri yang berisi pustaka aset dan pilih dataset "customers_factTable_Jan-May_2022". Seret dan letakkan ke kanvas seperti yang ditunjukkan pada Gambar 10.5.

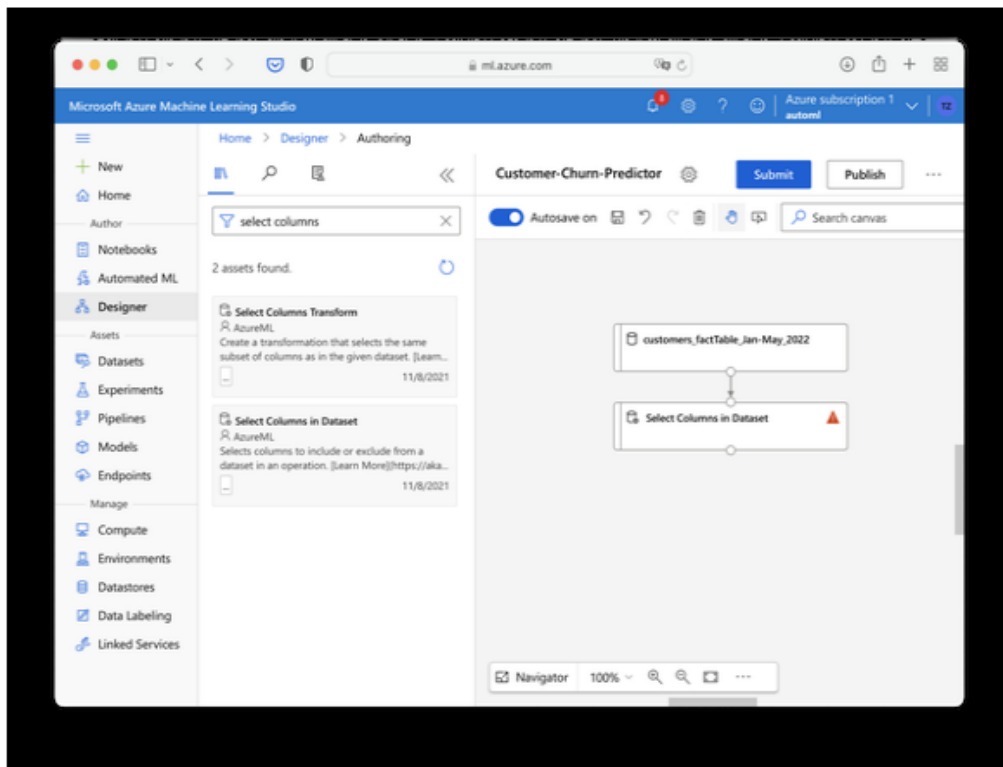


Pipeline akan dijalankan dari atas ke bawah. Jadi, semuanya akan dimulai dengan memuat dataset ini. Jika mau, Anda dapat melihat pratinjau data dengan mengklik kanan modul dan memilih "Pratinjau data". Melihat pratinjau keluaran masing-masing modul di kanvas merupakan cara intuitif untuk melakukan debug, terutama di awal saat Anda baru mengenal antarmukanya. Selanjutnya, mari kita lakukan sesuatu dengan data tersebut. Tujuan kita adalah membangun pipeline machine learning yang sangat sederhana yang akan memprediksi kolom "Churn" berdasarkan fitur input kita, mirip dengan kasus penggunaan Auto ML.

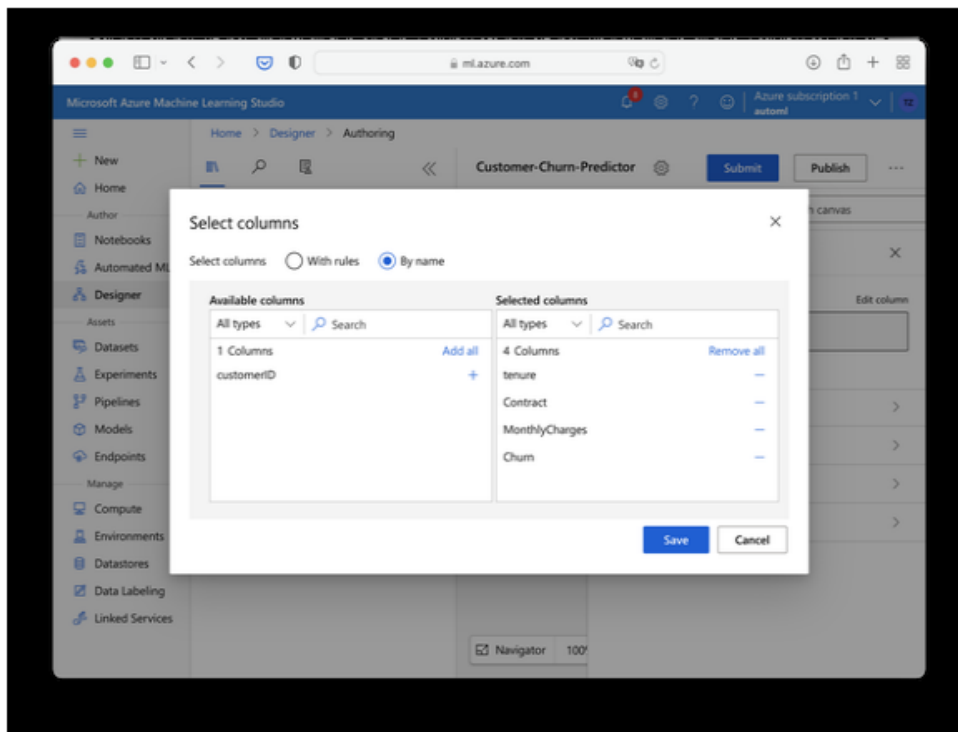


Gambar 10.5. Menambahkan Dataset ke Kanvas

Gunakan bilah pencarian di panel modul di sebelah kiri, cari "Pilih Kolom di Dataset". Tujuan modul ini adalah untuk menghapus kolom yang tidak kita perlukan untuk prediksi, seperti ID Pelanggan. Seret modul ke kanvas di bawah modul Dataset. Tutup pengaturan untuk saat ini, jika muncul secara otomatis. Hubungkan port keluaran dari modul Dataset ke modul Select Columns in Dataset seperti terlihat pada Gambar 10.6.



Gambar 10.6. Menambahkan modul "Select Columns in Dataset"



Gambar 10.7. Memfilter kolom

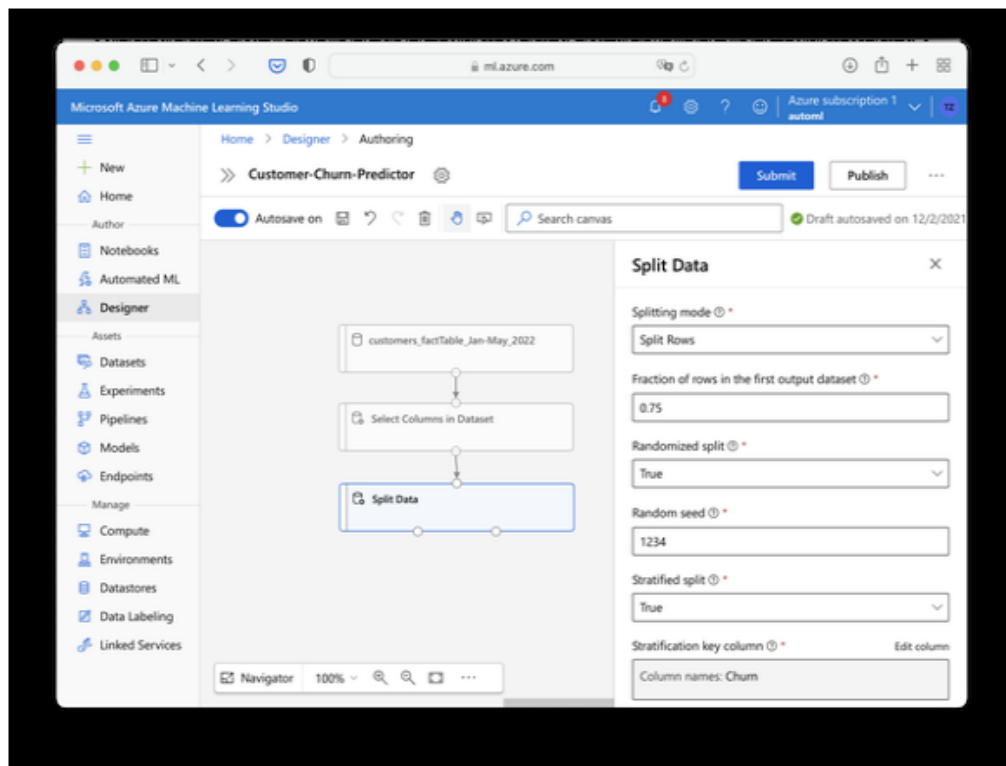
Menggambar koneksi antara dua modul memungkinkan metadata mengalir ke hilir dalam alur kerja kita. Artinya, modul "Select Columns in Dataset" kita sekarang akan mengetahui nama



kolom yang tersedia. Klik modul pada kanvas untuk membuka pengaturan sekali lagi. Klik "Edit columns" dan pilih "Select columns by name" seperti yang terlihat pada Gambar 10.8.

Pilih semua kecuali kolom CustomerID. Konfirmasikan dengan mengklik "Simpan". Tutup panel pengaturan modul untuk memberi ruang lebih bagi kanvas. Selanjutnya, kita ingin membagi data menjadi set pelatihan dan set pengujian. Cari "Pisahkan Data" di bilah pencarian modul dan seret modul ke kanvas. Dalam pengaturan modul, tentukan parameter pemisahan berikut:

- Mode pemisahan: Pisahkan Baris
- Fraksi baris pada keluaran pertama: 0,75
- Pemisahan acak: Benar
- Seed acak: 1234 (untuk reproduktifitas)
- Pemisahan bertingkat: Benar
- Kolom kunci stratifikasi: Churn



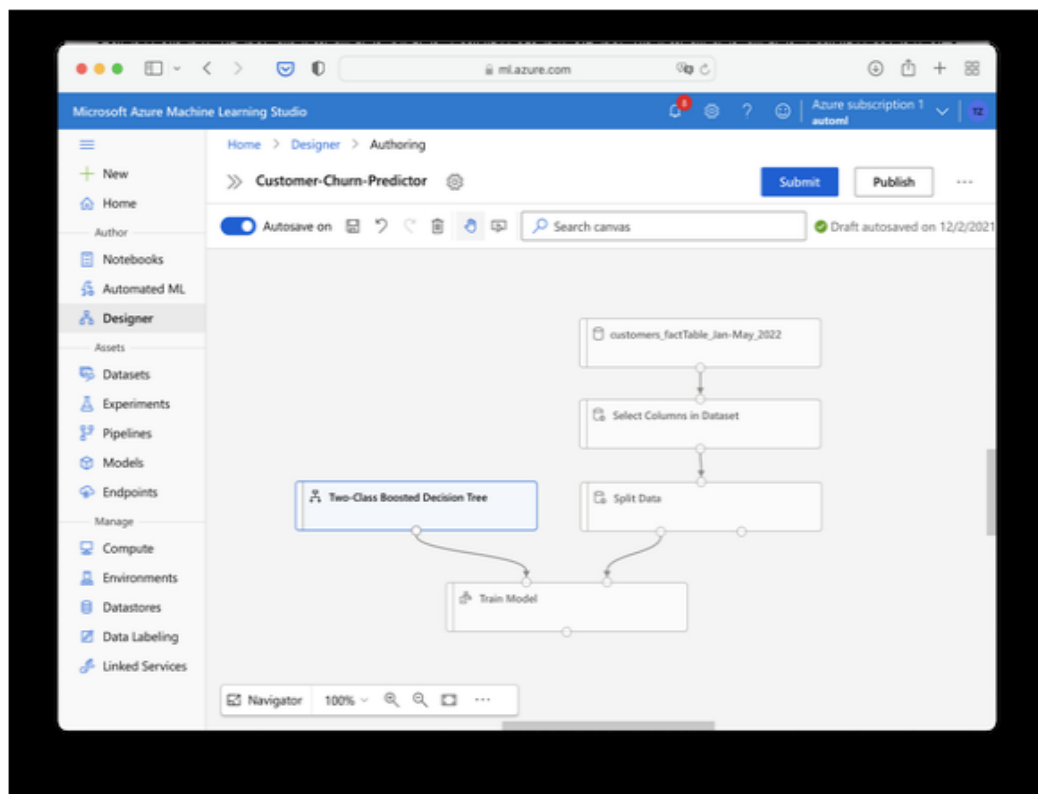
Gambar 10.8. Menambahkan Komponen Data Terpisah

Dengan pengaturan ini, kita melakukan pemisahan pelatihan-pengujian-stratifikasi dengan 75% data untuk pelatihan dan 25% untuk pengujian. Stratifikasi berarti distribusi kelas untuk label Churn akan sama untuk set pelatihan dan pengujian. Tutup pengaturan dan hubungkan port keluaran modul "Select Columns in Dataset" ke port masukan modul "Split Data". Seperti yang Anda lihat, modul Split Data kini memiliki dua keluaran, satu untuk dataset pelatihan (kiri) dan satu untuk dataset pengujian (kanan).



Sekarang kita dapat menambahkan komponen Train Model. Cari komponen tersebut di panel modul dan seret ke kanvas. Anda akan menyadari bahwa modul ini memiliki dua masukan: masukan kiri mengharapkan algoritma pelatihan dan masukan kanan mengharapkan dataset pelatihan. Hubungkan simpul keluaran kiri modul Data Terpisah ke port masukan kanan modul Train Model. Kita masih perlu memberi tahu komponen Train Model kolom mana yang sebenarnya ingin kita prediksi. Jadi, klik komponen tersebut untuk membuka pengaturan dan pilih "Churn" sebagai kolom Label.

Sekarang kitalah yang menentukan algoritma pelatihan yang sebenarnya. Berbeda dengan Auto ML, kita harus menentukan sendiri algoritma pelatihannya. Buku ini memang bukan buku Dasar-Dasar Pembelajaran Mesin, tetapi dalam sebagian besar skenario pembelajaran terawasi, Anda akan mencapai hasil yang cukup baik dengan memilih metode ensemble seperti Decision Forest atau Boosted Trees sebagai baseline pertama.



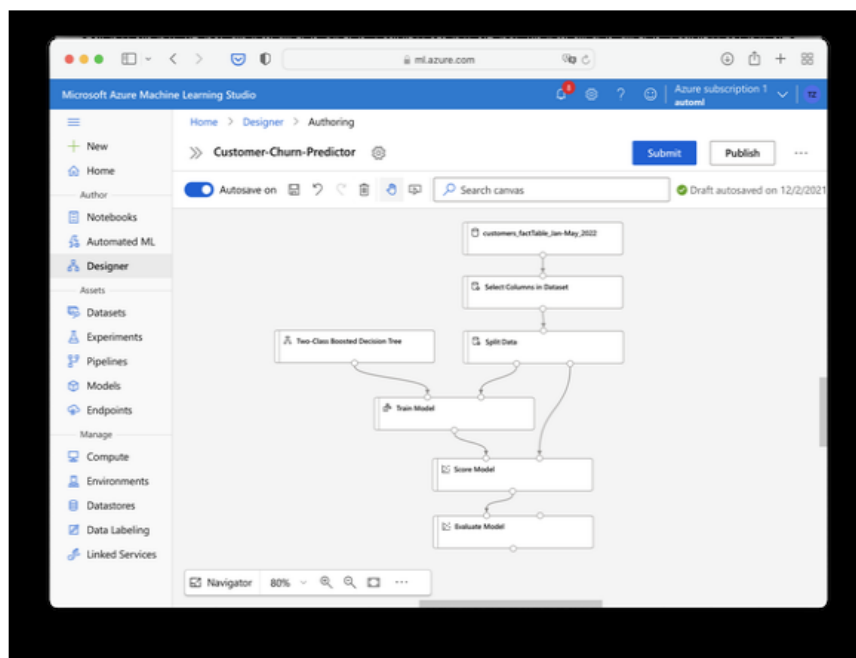
Gambar 10.9. Menambahkan Komponen Model dan Melatih Model

Model-model ini mencoba menggabungkan beberapa pembelajar yang lemah seperti pohon keputusan menjadi algoritma pembelajaran yang kuat yang mencoba meminimalkan kesalahan prediksi pada dataset pelatihan Anda. Model-model ini berfungsi untuk masalah klasifikasi dan regresi, sehingga merupakan algoritma yang baik dan menyeluruh, meskipun agak rumit. Keunggulan ML Designer adalah Anda dapat dengan mudah mencoba berbagai algoritma yang lebih sederhana, seperti regresi linier atau logistik, dan melihat kinerjanya pada set data Anda.



Untuk latihan kita, cari "Two-Class Boosted Decision Tree" dari panel modul dan seret ke kanvas. Kita dapat membiarkan semua pengaturannya default untuk saat ini. Hubungkan modul Decision Tree ke komponen Train Model dan pipeline Anda akan terlihat seperti pada Gambar 10.10.

Kita hampir selesai! Hanya ada satu hal yang kurang, yaitu menambahkan komponen yang menilai beberapa prediksi aktual berdasarkan model yang telah dilatih dan menghitung beberapa metrik evaluasinya. Cari komponen "Score Model" dan seret ke kanvas. Hubungkan port keluaran dari komponen Train Model ke port masukan kiri dari komponen Score Model.



Gambar 10.10. Alur Kerja End-to-End Pertama di ML Designer

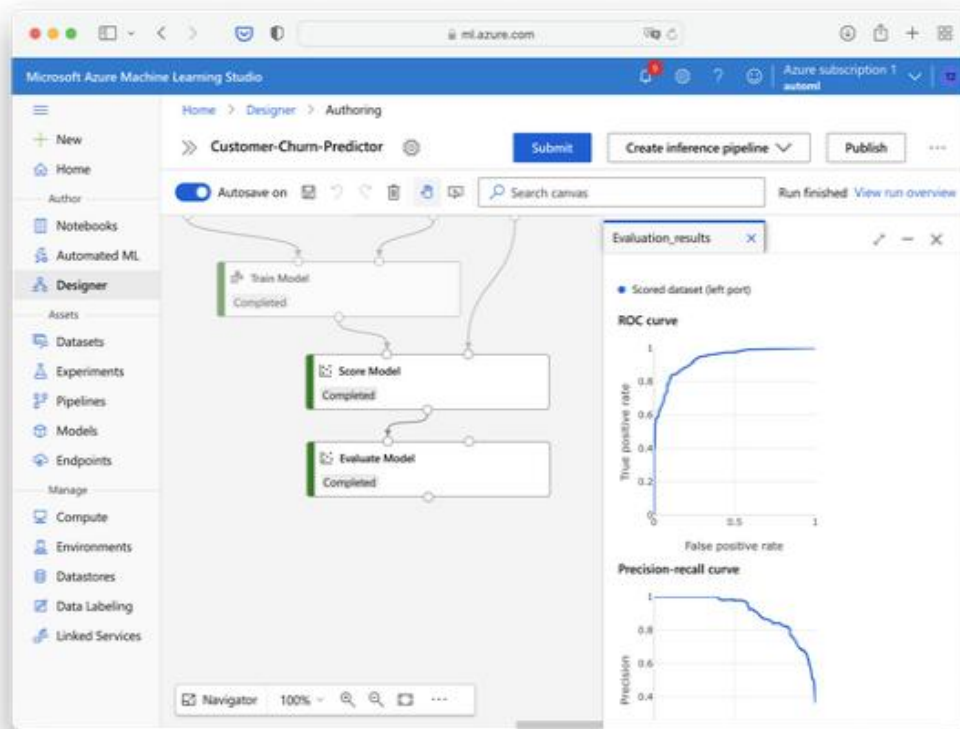
Kemudian, hubungkan port keluaran kanan dari komponen Split Data ke port masukan kanan dari komponen Score Model. Ini akan memungkinkan kita menghitung prediksi untuk dataset uji menggunakan model yang telah dilatih pada dataset pelatihan. Terakhir, temukan modul Evaluate Model, seret ke kanvas, dan hubungkan port keluaran Score Model ke port masukan kiri dari komponen Evaluate Model. Pipeline akhir Anda sekarang akan terlihat seperti Gambar 10.11.

Selamat! Anda baru saja membangun alur Machine Learning pertama Anda dari awal. Tekan tombol "Kirim" untuk menjalankan alur tersebut. Semua proses akan dikumpulkan dalam apa yang disebut eksperimen, seperti yang biasa kita lakukan dalam skenario AutoML. Saya sarankan untuk membuat eksperimen baru untuk alur ini agar pekerjaan Anda tetap teratur. Buat eksperimen baru dan berikan nama yang menarik, misalnya "customer-churn-prediction-training". Tekan Kirim dan bersantailah sementara Azure ML Studio bekerja di balik layar.

Anda dapat mengikuti proses pelatihan karena Designer akan menunjukkan langkah mana yang sedang berjalan dan langkah mana yang telah selesai. Meskipun proses gagal



karena suatu alasan atau terganggu, langkah-langkah yang telah berhasil dijalankan hingga saat ini akan disimpan. Selama Anda tidak mengubah modul sebelumnya dalam alur, output dari alur akan disimpan. Seluruh proses akan memakan waktu sekitar 10-15 menit.



Gambar 10.11. Metrik evaluasi untuk alur kerja pertama

Alasan mengapa proses ini memakan waktu lama adalah karena Azure sebenarnya menjalankan proses terpisah untuk setiap modul, yang menciptakan beberapa overhead. Keuntungannya adalah antarmuka ini juga akan berfungsi dengan baik untuk kumpulan data yang sangat besar karena setiap modul berjalan secara independen. Itulah sebabnya menjalankan perancang untuk beberapa kumpulan data yang agak kecil seperti dalam kasus kami akan membutuhkan waktu lebih lama dibandingkan dengan menjalankannya di notebook lokal Anda. Namun, semakin besar kumpulan data, semakin banyak keuntungan kinerja yang akan Anda dapatkan.

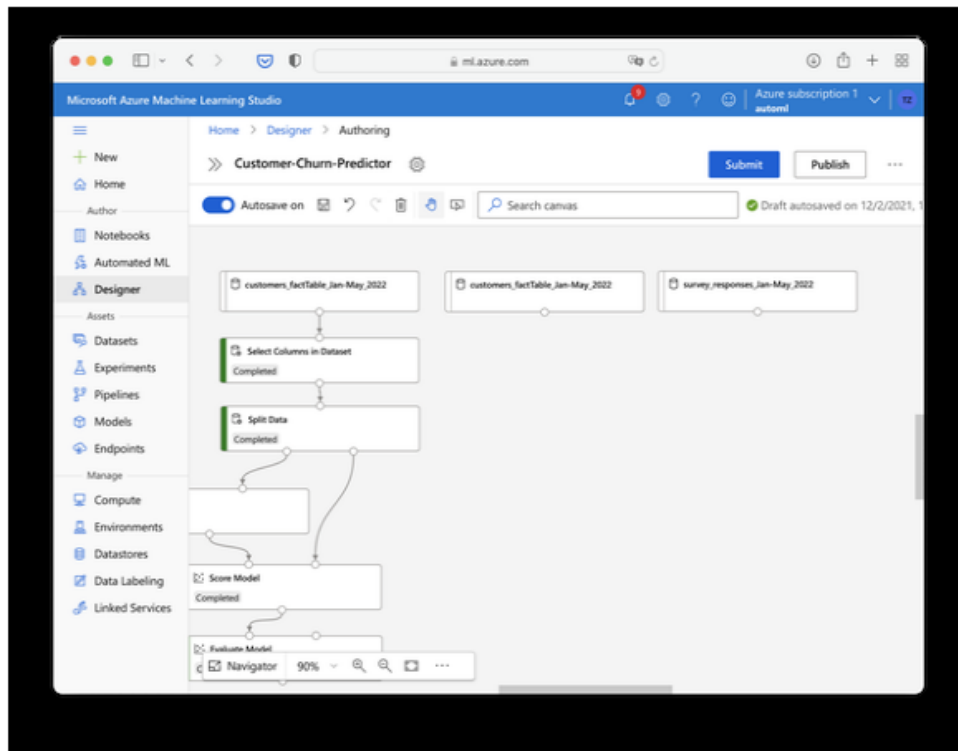
Setelah proses selesai, Anda akan melihat lencana "Selesai" pada modul Evaluasi Model terakhir. Dalam hal ini, klik kanan modul dan pilih "Pratinjau Data → Hasil Evaluasi" dari menu konteks. Ini akan memberikan Anda output seperti yang terlihat pada Gambar 10.12. Klik ikon Perbesar untuk menambah ruang dan seluruh antarmuka akan terlihat sangat mirip dengan yang telah Anda lihat di bab-bab sebelumnya. Pada percobaan pertama, kami mencapai Akurasi 86,6% dan Skor F1 80,9%. Hasil ini cukup baik untuk percobaan pertama, tetapi mari kita lihat apakah kita dapat meningkatkan metrik ini lebih jauh dengan menggabungkan dataset pelatihan kita dengan beberapa data tambahan.

Tutup metrik evaluasi dan kembali ke kanvas. Kali ini, kita ingin menambahkan analisis umpan balik pelanggan dan melihat apakah hal itu membantu meningkatkan prediksi churn



pelanggan kita. Pendekatan kita adalah dengan membangun alur kerja kedua di samping alur kerja yang sudah ada dan memasukkan hasilnya ke port input yang tepat di modul Evaluate Model. Ini akan memberi kita perbandingan langsung dari kedua pendekatan pemodelan.

Mari kita bahas ini selangkah demi selangkah. Pertama, seret dan lepas modul Dataset `customers_factTable_Jan-May_2022` dari panel modul di sebelah kiri ke kanvas. Selain itu, cari modul Dataset untuk `survey_responses_Jan-May_2022` dan seret modul tersebut ke sebelahnya. Kanvas Anda sekarang akan terlihat seperti pada Gambar 10.12.



Gambar 10.12. Menambahkan lebih banyak sumber data ke perancang ML

Untuk lebih memahami data survei, klik kanan modul dan pilih "Pratinjau data". Dataset hanya berisi dua kolom, ID pelanggan dan respons teks, keduanya merupakan nilai yang berasal dari hasil survei pelanggan asli.

Sejauh ini, model pembelajaran mesin kami belum dapat menangani input teks mentah dengan baik sebagai fitur prediktif. Agar lebih mudah diakses, kami akan membaca jawaban teks terbuka dari dataset survei, mengirimkannya ke Azure Cognitive Service, dan mendapatkan nilai sentimen untuk setiap teks. Klasifikasi "negatif" atau "positif" kemudian akan menjadi fitur tambahan untuk model klasifikasi kami. Untuk tujuan ini, kami akan menggunakan Azure Cognitive Services untuk Analisis Sentimen yang telah kami terapkan di Bab 8.

Mari kita hadirkan komponen layanan AI ini ke kanvas! Percaya atau tidak, tetapi (setidaknya pada saat penulisan ini) tidak ada modul bawaan untuk Azure Cognitive Services yang dapat Anda seret ke kanvas di Azure ML Designer. Namun, yang bisa kita lakukan hanyalah menambahkan modul yang menjalankan kode Python atau R untuk kita. Dan itulah



pendekatan yang akan kita ambil: Kita akan mengambil respons dari survei, mengirimkannya ke API layanan AI jarak jauh menggunakan Python atau R, dan memasukkan hasilnya kembali ke dalam alur perancang ML. Untuk memulai, cari "Execute Python Script" dan seret ke kanvas. Hubungkan port output dari Dataset `survey_responses` ke port input pertama dari modul Execute Python Script.

Cara kerja modul skrip adalah dengan mengharapkan hingga dua kerangka data (tabel) yang kemudian dapat dimodifikasi dalam fungsi yang disebut `azureml_main`. Hasilnya akan dikembalikan lagi sebagai satu atau dua kerangka data. Runtime Python yang mendasarinya hadir dengan sejumlah paket terbatas. Untuk mengetahui lebih lanjut, periksa sumber daya Execute Python Script di perancang - Azure Machine Learning.

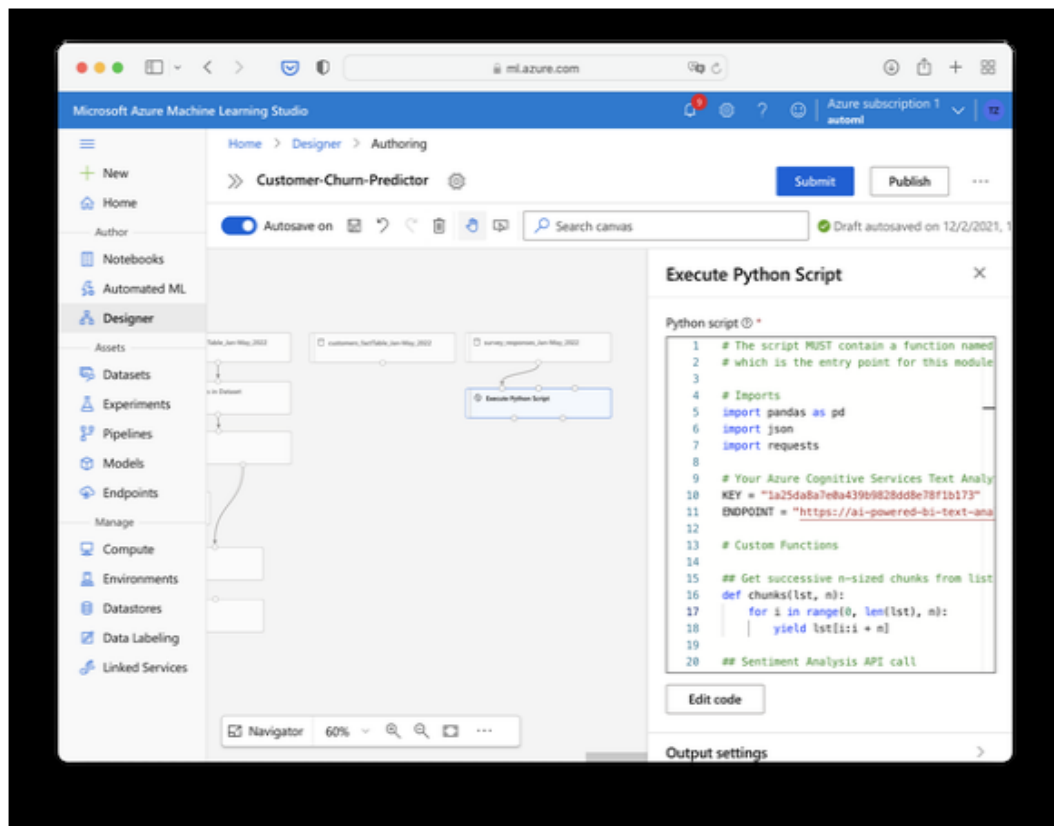
Yang perlu kita lakukan sekarang adalah mengganti kode demo pra-pembuatan dengan kode kustom kita untuk tugas yang ingin kita lakukan. Buka berkas `ml-designer.py` dari repositori buku di editor teks. Ganti kunci dan titik akhir dengan parameter kustom Anda seperti yang terlihat di Bab 8. Sekarang, pilih semua skrip, lalu salin. Ganti semua kode contoh modul Execute Python Script di Azure ML Designer dengan isi clipboard Anda. Modul skrip sekarang seharusnya hanya berisi kode dari berkas `ml-designer.py` dan tidak ada yang lain seperti yang terlihat pada Gambar 10.13.

Kode ini pada dasarnya akan membaca kerangka data sebagai input, mengubahnya ke format yang dapat ditangani oleh layanan API. Kerangka data asli akan dikembalikan dengan prediksi sentimen terlampir. Sekarang, klik "Kirim" untuk menjalankan bagian baru dari alur kerja ini dan pastikan semuanya berjalan sesuai harapan.

Perhatikan bahwa semua langkah yang kita jalankan sebelumnya di sisi kiri grafik tidak akan dijalankan ulang karena kita tidak mengubah apa pun. Sebaliknya, proses ini hanya akan membaca dataset baru dan memasukkannya ke dalam modul Skrip. Setelah beberapa menit, Anda akan melihat bahwa skrip berhasil dijalankan. Jika muncul kesalahan, klik modul Skrip dan lihat berkas log. Pesan kesalahan di sana akan membantu Anda melakukan debug, misalnya jika kunci sumber daya Anda tidak valid atau kuota gratis Anda telah terlampaui.

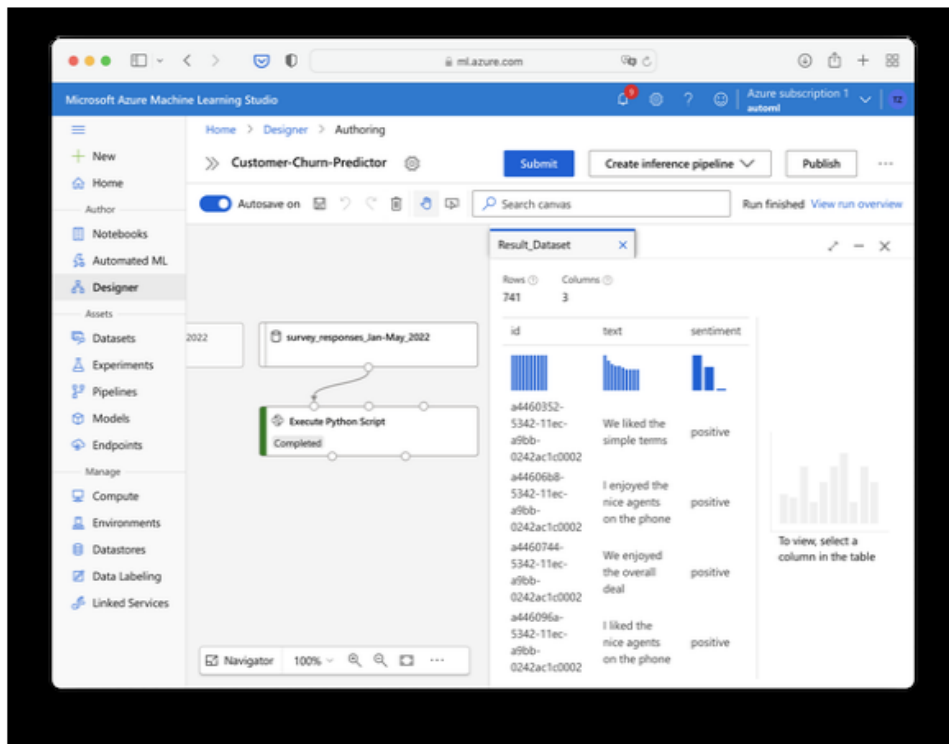
Setelah proses selesai, klik kanan modul Skrip dan pilih "Pratinjau data → Dataset hasil". Seperti yang ditunjukkan pada Gambar 10.13, Anda akan melihat bahwa setiap umpan balik pelanggan kini telah mendapatkan label sentimen.



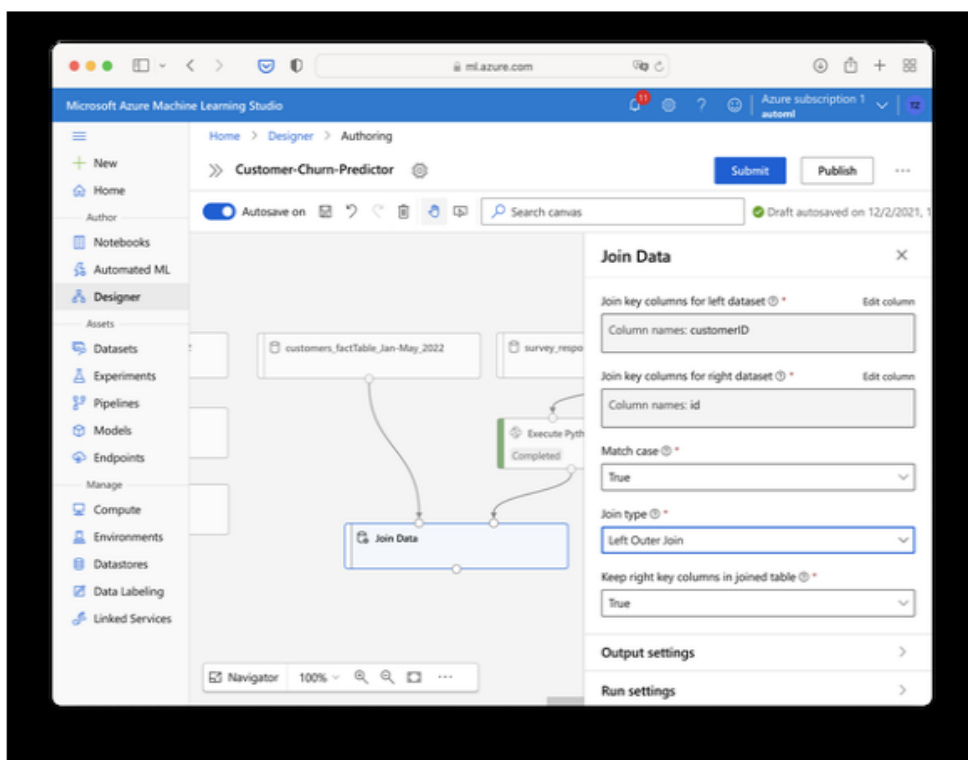


Gambar 10.13. Modul skrip Python di Azure ML Designer

Tujuan kita sekarang adalah menambahkan informasi ini ke dataset utama yang digunakan untuk melatih model churn. Yang perlu kita lakukan adalah menggabungkan tabel ini ke tabel utama tanpa kehilangan data apa pun di tabel utama perlu diingat bahwa survei ini tidak dilakukan untuk semua pelanggan, tetapi hanya untuk sampel. Proses ini disebut Full Outer Left Join, di mana tabel kiri adalah dataset utama kita dan tabel kanan adalah dataset tempat kita ingin menggabungkan informasi tambahan. Temukan modul yang sesuai dengan mencari "Join Data" di pustaka aset.



Gambar 10.14. Hasil analisis sentimen di Azure ML Designer



Gambar 10.15. Menggabungkan data dari berbagai sumber

Seret modul Join Data ke kanvas. Hubungkan port output pertama dari modul skip sebelumnya ke port input kanan modul Join Data. Kemudian, hubungkan port output



customers_factTable_Jan-May_2022 (yang belum kita gunakan) ke port input kiri modul Join Data. Pilih modul Join Data di kanvas untuk menampilkan pengaturannya.

Kita perlu menentukan kolom kunci tempat kedua set data akan digabungkan. Anda dapat mengedit kolom kunci dengan mengklik "Edit kolom" untuk set data kiri dan kanan. Untuk set data kiri, pilih kolom customerID dan untuk set data kanan, pilih kolom "id". Atur tipe gabungan ke Left Outer Join dan atur "Keep right key columns in merged table" ke False.

Jalankan pipeline dengan menekan tombol kirim untuk memverifikasi bahwa penggabungan berfungsi seperti yang diharapkan. Dataset gabungan Anda seharusnya memiliki 7 kolom dan 3.814 baris yang dapat Anda verifikasi dengan klik kanan pada modul Gabung Data → Pratinjau data → dataset hasil setelah proses selesai.

Sekarang kita harus membuat ulang sisa pipeline seperti yang telah kita lakukan sebelumnya. Saya tidak akan membahas langkah-langkah ini secara detail karena berulang. Singkatnya, tambahkan modul berikut dan terapkan pengaturan berikut:

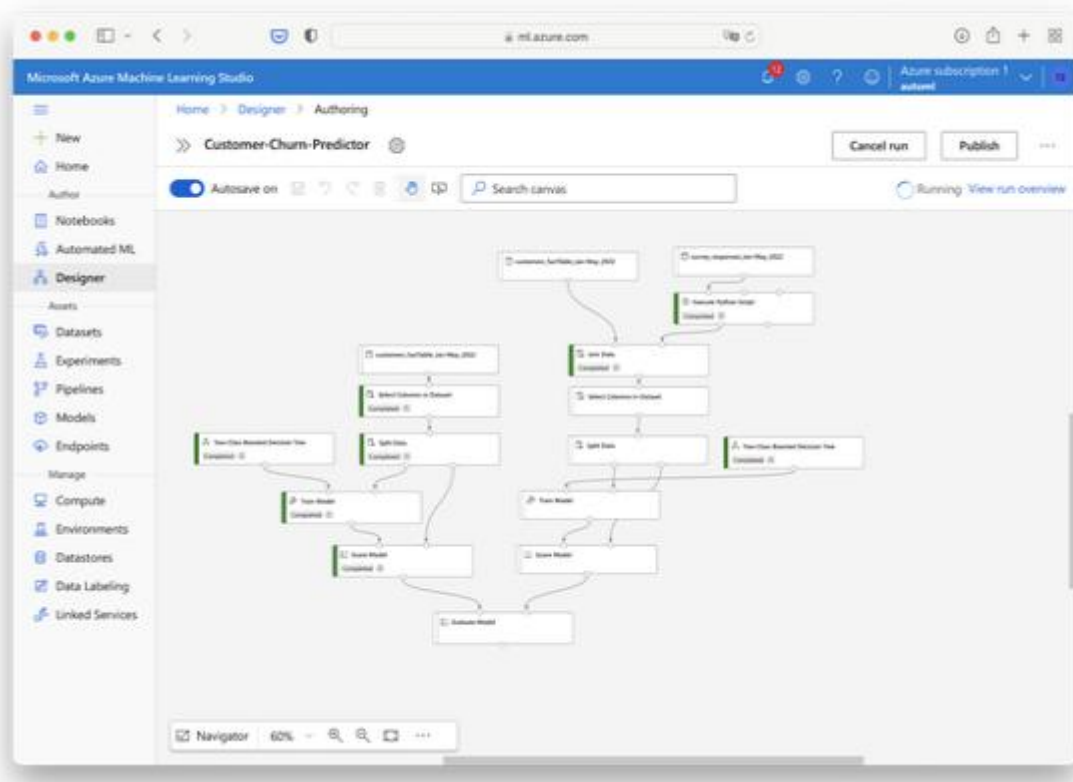
- Pilih Kolom dalam Dataset
 - Hapus kolom "teks", pertahankan kolom baru "sentimen".
- Pisahkan Data:
 - Sama seperti sebelumnya, sampel berstrata 0,75 pada kolom "Churn".
- Latih Model
 - Kolom label adalah "Churn".
- Pohon Keputusan Dua Kelas yang Ditingkatkan.
 - Biarkan pengaturan default.
- Model Skor

Hubungkan port keluaran terakhir dari komponen Model Skor ke port masukan kanan dari komponen Evaluasi Model yang telah kita buat sebelumnya. Pipeline terakhir ditunjukkan pada Gambar 10.16.

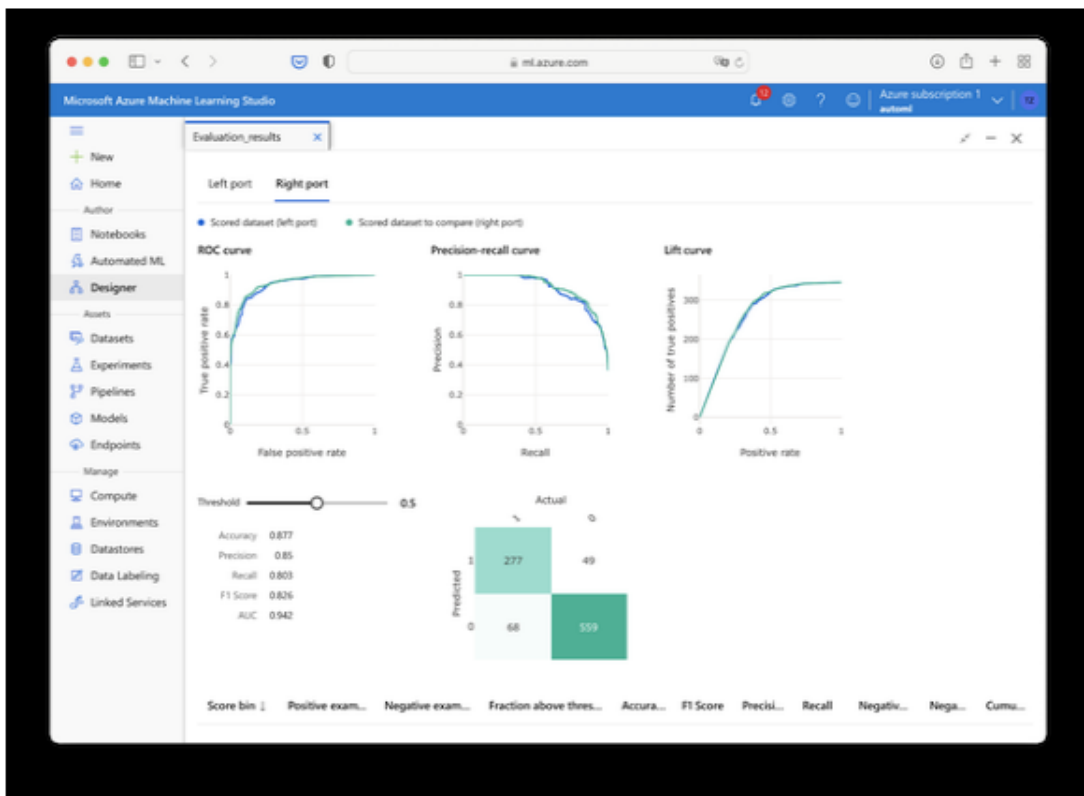
Klik Kirim untuk menjalankan seluruh grafik. Jika Anda merasa sudah waktunya untuk minum kopi lagi, Anda benar! Seluruh proses mungkin memakan waktu beberapa menit. Periksa kembali setelah proses selesai.

Setelah proses selesai, mari kita periksa modul evaluasi kita dan lihat apakah kinerja model dapat ditingkatkan. Klik kanan modul Evaluasi Model dan pilih Pratinjau Data → Hasil Evaluasi.





Gambar 10.16. Alur kerja ujung ke ujung kedua

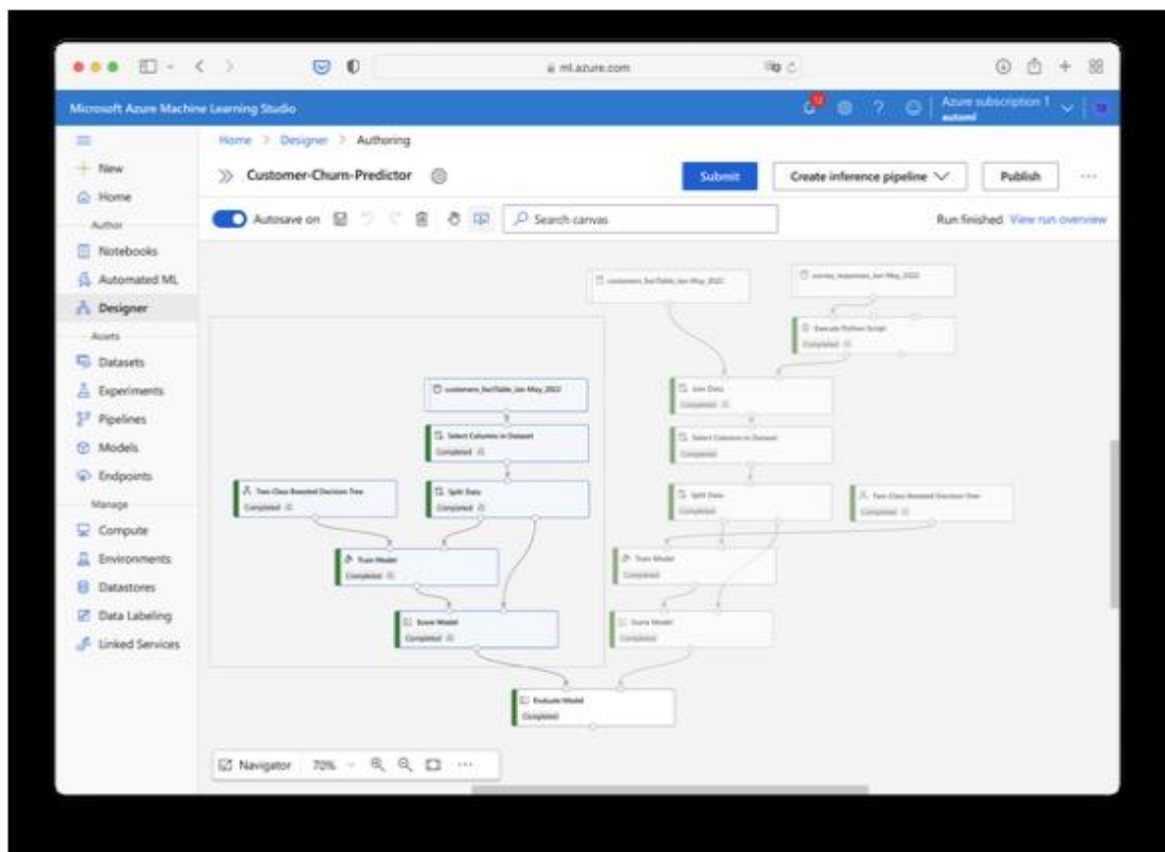


Gambar 10.17. Metrik evaluasi alur kerja kedua

Melihat hal ini, kita dapat melihat bahwa Akurasi dapat ditingkatkan satu poin persentase lagi menjadi 87,7% dan Skor F1 telah meningkat menjadi 82,6%. Ini bukanlah peningkatan kinerja yang luar biasa, tetapi tetap solid, mengingat label sentimen hanya diberikan untuk sebagian kecil data. Ini berarti bahwa informasi sentimen itu sendiri memiliki daya prediktif yang sangat tinggi untuk kasus penggunaan kita. Jadi, ada baiknya menyimpan informasi ini dalam kumpulan data kita dan menggunakannya kapan pun memungkinkan.

Sekarang, pelatihan model kita telah selesai. Selamat! Anda baru saja menggabungkan layanan AI siap pakai dengan algoritma pembelajaran mesin canggih untuk membuat model AI kustom yang canggih! Sekarang saatnya untuk menjalankan model dan membuatnya siap untuk inferensi.

Sebelum kita melanjutkan, mari kita rapikan grafik kita dengan memilih semua bagian di sebelah kiri dan menghapusnya. Karena semua proses sebelumnya telah dicatat dalam eksperimen kita, kita masih dapat mengakses hasil dari bagian pelatihan model ini nanti jika kita membutuhkannya. Untuk memilih beberapa modul, pilih alat seleksi dari bilah alat dan pilih semua modul grafik di sisi kiri grafik seperti yang ditunjukkan pada Gambar 10.18. Klik ikon hapus untuk menghapus modul.

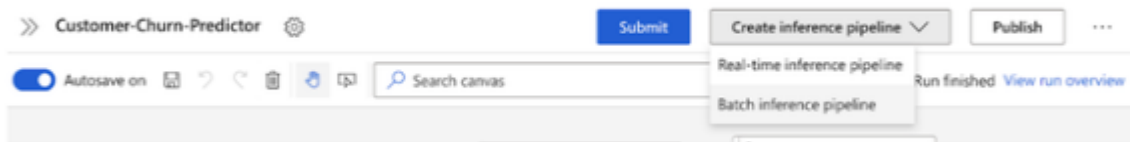


Gambar 10.18. Memilih beberapa komponen di kanvas

Terakhir, sambungkan kembali modul Model Skor ke port input pertama modul Evaluasi Model dengan memilih koneksi yang ada di antara keduanya, menghapusnya, dan menggambar ulang



ke port input pertama. Kirim alur pelatihan untuk terakhir kalinya ini akan cepat karena hanya akan menghitung ulang hasil untuk modul terakhir. Kita dapat men-deploy grafik ini sekarang sebagai alur inferensi. Yang perlu kita lakukan adalah memilih Buat alur Inferensi dari menu atas seperti yang terlihat pada gambar 10.19, lalu pilih Alur inferensi batch.



Gambar 10.19. Membuat alur inferensi batch

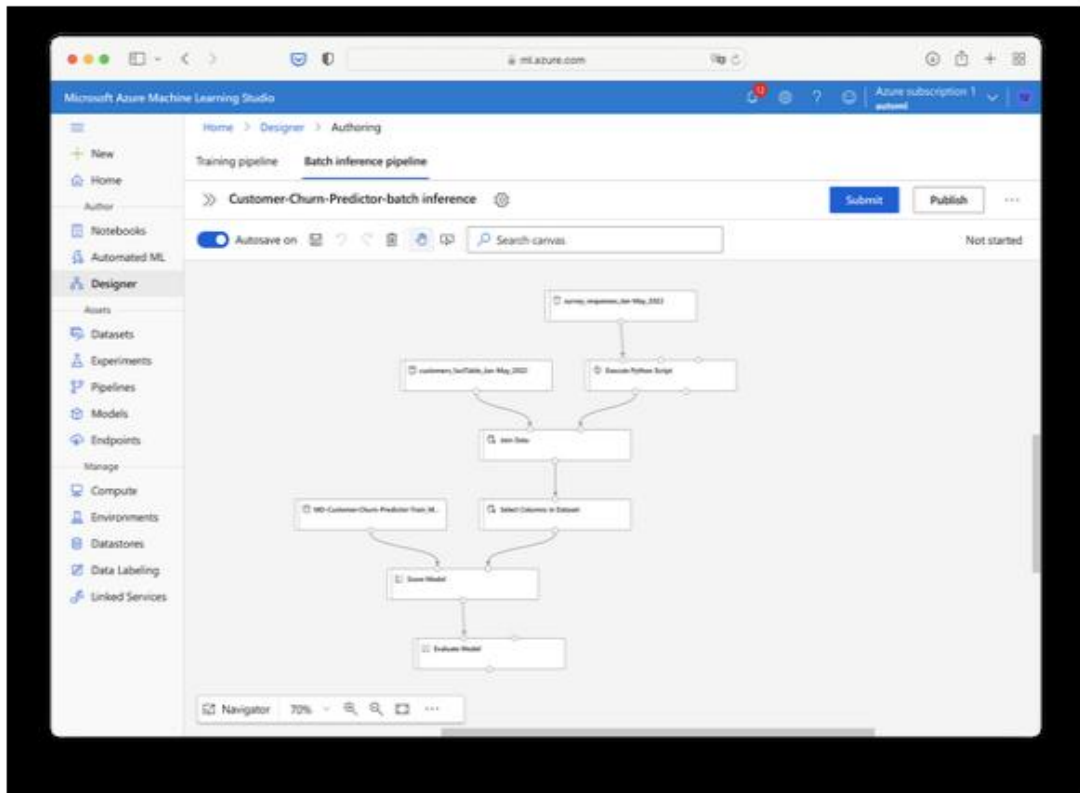
Perbedaan utama antara alur inferensi online dan batch dalam kasus ini adalah alur batch menggunakan dataset blob sebagai input, sedangkan alur online menggunakan data yang dikirimkan melalui API online.

Bagaimanapun, alur pelatihan akan diubah menjadi alur inferensi. Jika Anda perhatikan dengan saksama header ML Designer, Anda akan melihat tab baru bernama "Batch inference pipeline" seperti yang ditunjukkan pada Gambar di bawah.

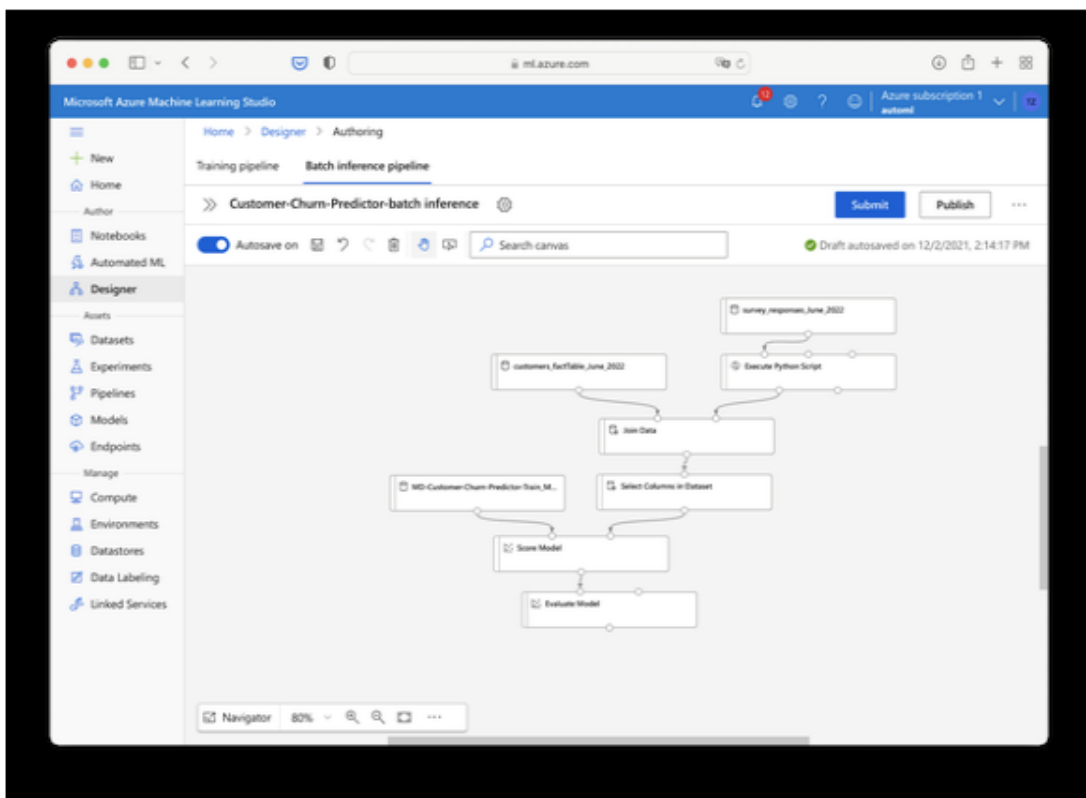
Perbedaan utama antara alur pelatihan dan inferensi adalah alur inferensi tidak berisi komponen pelatihan apa pun. Perhatikan bagaimana model Boosted Decision Tree dan modul Train model telah digantikan oleh modul baru bernama ML-Customer-Churn-Predictor-Train_Model_Trained_model seperti yang terlihat pada Gambar di bawah. Modul ini akan secara otomatis mereferensikan model terlatih kita sehingga dapat digunakan untuk inferensi. Sisa alur persiapan data, termasuk skrip Python kita, akan tetap sama.

Untuk menjalankan inferensi data baru, ganti dataset input dengan customers_factTable_June_2022 dan survey_responses_June_2022. Kedua dataset ini mewakili data baru dari pelanggan yang belum diketahui apakah mereka akan berhenti berlangganan atau tidak. Untuk mengganti dataset, cukup hapus dataset yang ada dari kanvas dan ambil dataset baru dari pustaka aset. Kanvas Anda akan terlihat seperti Gambar di bawah ini.





Gambar 10.20. Alur kerja inferensi batch (mentah)

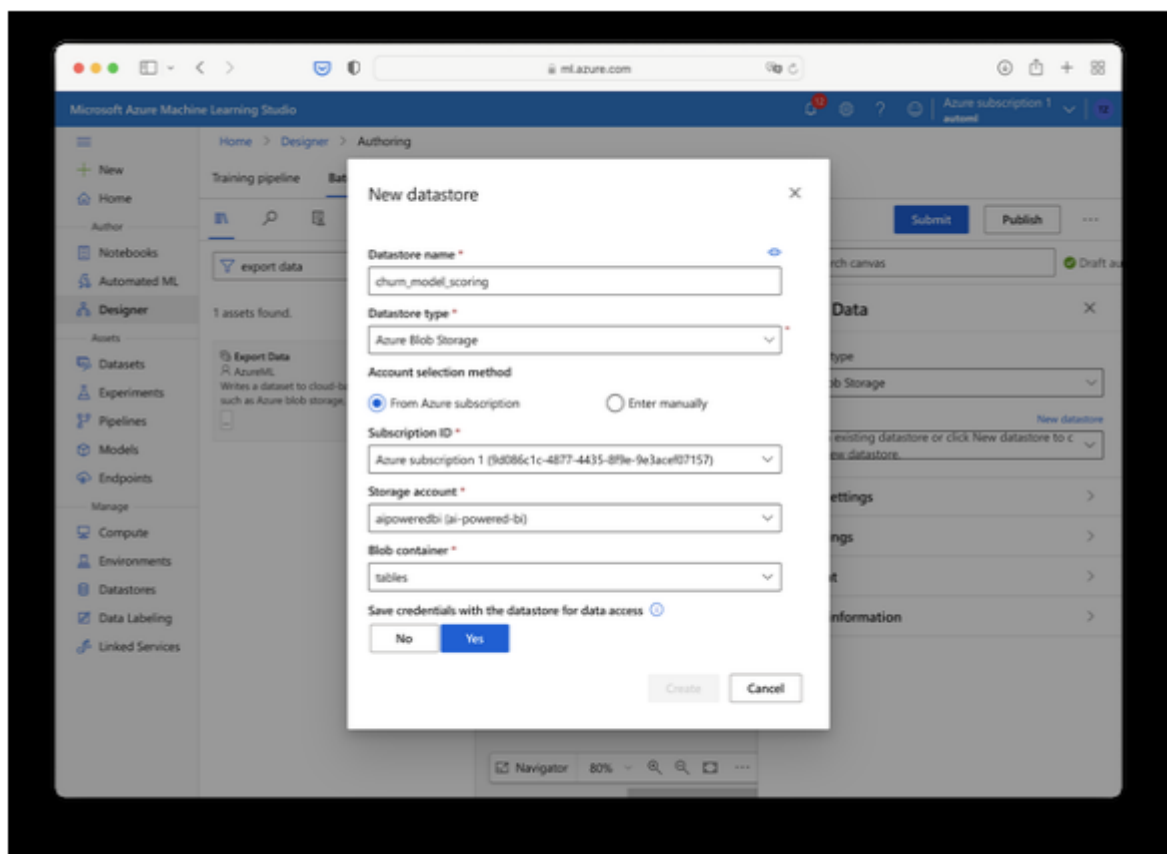


Gambar 10.21. Alur kerja inferensi batch (dipangkas)

Sebelum kita dapat menjalankan inferensi, kita masih perlu melakukan beberapa penyesuaian kecil: Cari Module Select Columns in Dataset di kanvas. Kanvas tersebut masih berisi kolom Churn yang kita gunakan untuk pelatihan. Kita tidak memiliki kolom ini selama inferensi (kolom itulah yang ingin kita prediksi!). Jadi, edit pengaturan dan hapus kolom Churn dari modul. Selain itu, pertahankan kolom customerID, karena kita akan membutuhkannya nanti untuk mencocokkan prediksi churn dengan data pelanggan.

Selain itu, jika Anda melompat ke akhir alur kerja kita, Anda masih akan melihat modul Evaluate Model sebagai komponen terakhir di sini. Namun, kita tidak memerlukan komponen ini pada tahap ini, karena kita tidak memiliki ground truth untuk dibandingkan. Sebagai gantinya, kita ingin mengeksport data yang telah diberi skor ke suatu lokasi. Untuk melakukannya, hapus modul Evaluate Model dari kanvas dan seret modul "Ekspor Data" dari pustaka aset (Data Input dan Output) ke kanvas.

Di pengaturan, pilih penyimpanan blob Azure sebagai tipe data. Untuk Datastore, klik New Datastore dan beri nama seperti churn_model_scoring. Untuk akun penyimpanan, pilih akun penyimpanan yang sebelumnya Anda gunakan di Bab dan pilih "tabel" di bawah "Blob container". Pengaturan Anda akan terlihat seperti Gambar 10.22.

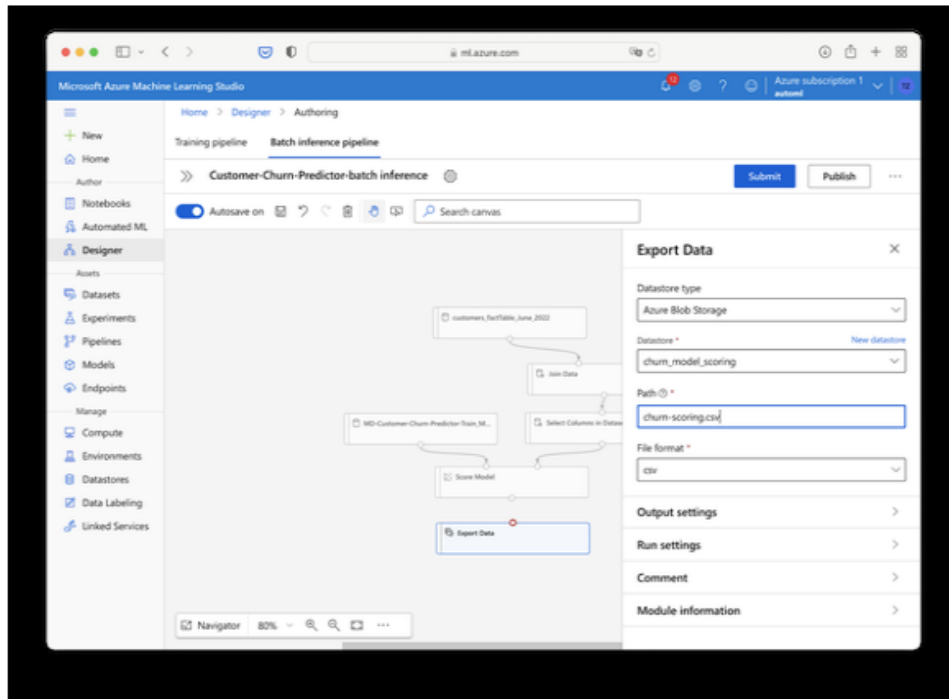


Gambar 10.22. Membuat penyimpanan data baru

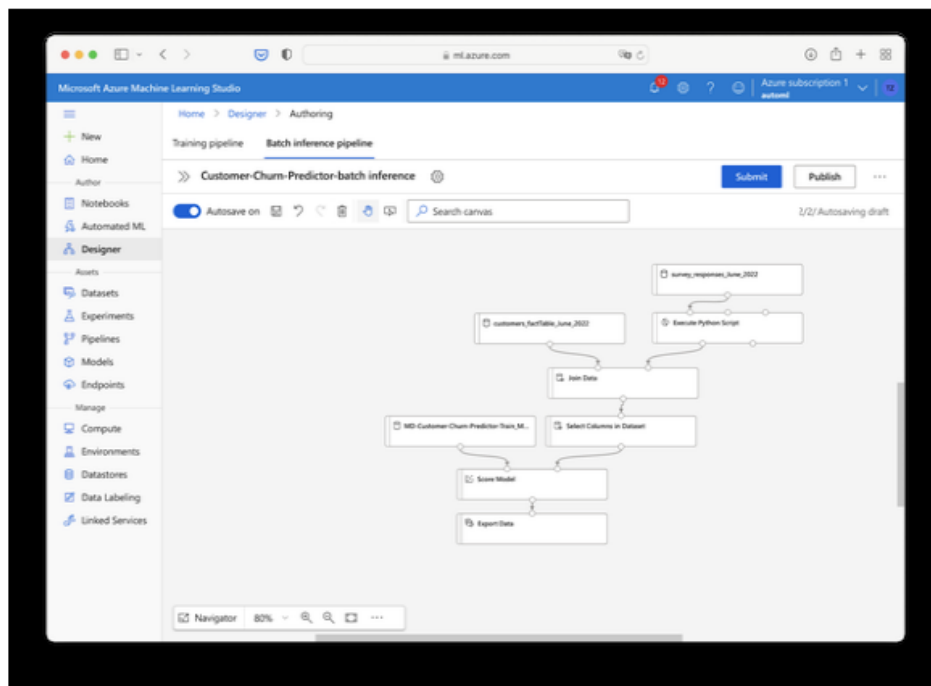
Gulir ke bawah dan tempel Kunci Akun Anda dari sumber daya Akun Azure Storage ke kolom yang sesuai di jendela pengaturan ini. Klik Buat dan Anda akan kembali ke pengaturan



Ekspor Data. Periksa kembali apakah penyimpanan data yang baru Anda buat telah dipilih dan beri nama file output Anda di kolom Jalur, misalnya churn-scoring.csv. Terakhir, atur format file ke CSV. Gambar di bawah menunjukkan contoh pengaturan ekspor.



Gambar 10.23. Tambahkan modul Data Ekspor dan pengaturannya

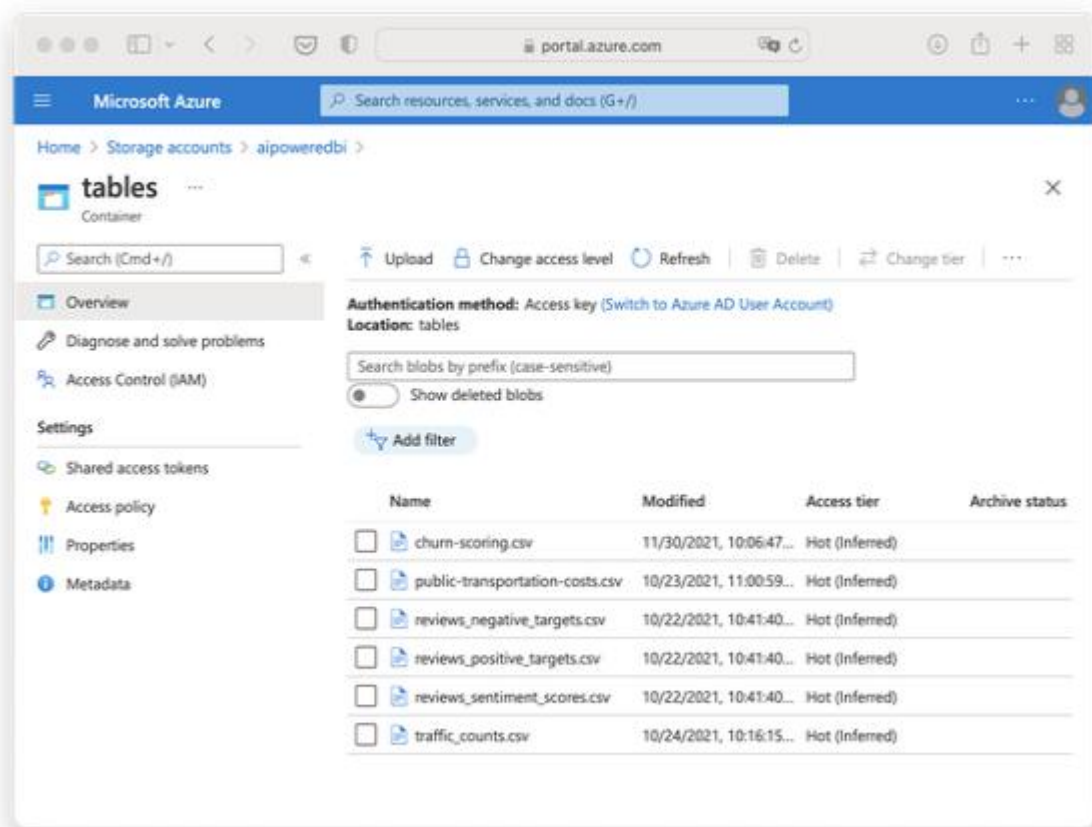


Gambar 10.24. Alur inferensi akhir



Terakhir, hubungkan modul Model Skor ke modul Data Ekspor. Alur akhir ditunjukkan pada Gambar di bawah.

Sekarang, klik "Kirim" untuk menjalankan alur inferensi Anda. Agar semuanya tetap teratur, buat eksperimen baru seperti "customer-churn-inference" yang memisahkan proses inferensi Anda dari proses pelatihan. Proses inferensi akan memakan waktu beberapa menit. Sekali lagi, kita melakukan ini pada sampel data yang relatif kecil, tetapi prosesnya sendiri juga dapat diskalakan dengan cukup baik ke kumpulan data yang sangat besar tanpa penskalaan yang proporsional seiring waktu. Setelah proses inferensi selesai, Anda akan melihat file baru churn-scoring.csv di folder tabel di wadah penyimpanan Anda seperti yang terlihat pada Gambar di bawah.



Gambar 10.25. Output tabel di penyimpanan blob Azure

Saatnya beralih ke BI kita!

Menggabungkan Berbagai Layanan AI dalam BI

Setelah kita membangun model prediktif dan mengubah teks tak terstruktur menjadi data kategorikal yang mudah dianalisis, mari kita gabungkan semuanya dengan membangun Dasbor Pelanggan Berbasis AI. Ingat, tujuan kita adalah membangun dasbor untuk pemasaran dan penjualan yang memberikan gambaran umum tingkat tinggi tentang perkembangan churn pelanggan, apakah semuanya 'masih normal', dan seberapa efektif tindakan penanggulangan



yang telah diambil. Dasbor ini juga akan menyediakan beberapa fungsi tambahan untuk menggali lebih dalam wawasan khusus.

Berbeda dengan bab-bab sebelumnya, saya tidak akan memberikan instruksi langkah demi langkah yang mendetail tentang cara pembuatan dasbor, tetapi akan berfokus pada konsep-konsep utama dan berbagai komponen yang bekerja di sini. Anda dapat mengunduh berkas PBIX Bab-9-customer-dashboard.pbix atau membuatnya sendiri. Saya menggunakan langkah-langkah berbeda yang telah dijelaskan sebelumnya.

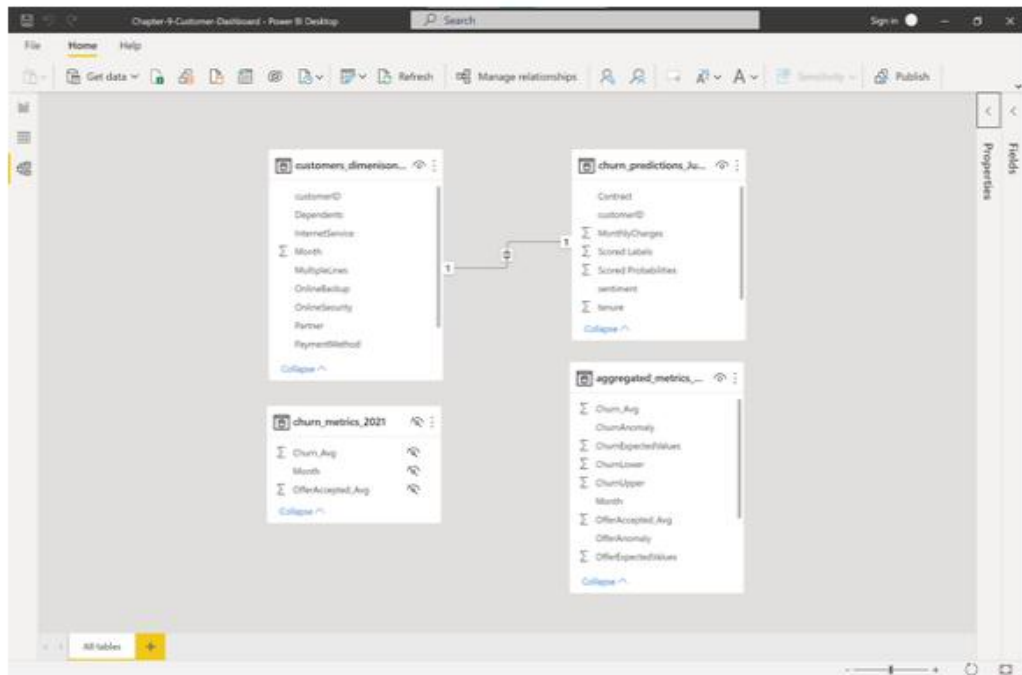
CATATAN

Untuk tutorial video langkah demi langkah tentang cara membuat dasbor, kunjungi www.aipoweredbi.com/customer-dashboard untuk tutorial video gratis.

Untuk membangun dasbor pelanggan, kita memerlukan model data seperti yang terlihat pada Gambar X, yang didasarkan pada empat tabel berikut:

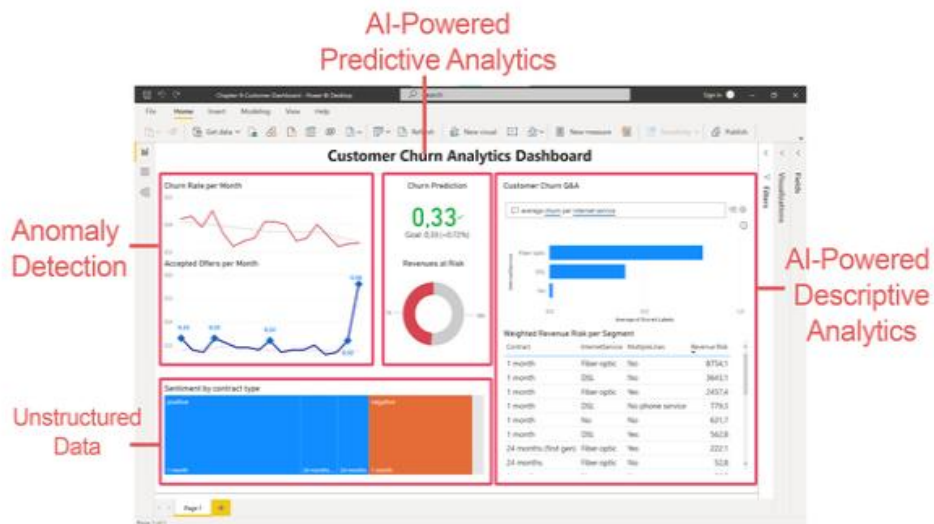
- `churn_predictions_June_2022`: Tabel ini berisi kumpulan data churn yang dinilai yang kita peroleh dari Azure ML Designer.
- `customer_dimensions_June_2022`: Tabel ini berisi data BI lebih lanjut untuk pelanggan seperti masa kerja, layanan, informasi demografis, dll. Tabel ini memiliki relasi 1 banding 1 dengan tabel `churn_predictions_June_2022` berdasarkan kolom ID pelanggan.
- `Churn_Metrics_2021`: Tabel yang mencakup tingkat churn agregat dan tingkat penerimaan penawaran per bulan dari bulan-bulan sebelumnya. Data ini dapat berasal dari gudang data SQL, dalam kasus kami, data ini disediakan sebagai file CSV dari repositori buku.
- `Aggregated_metrics_2022`: Tabel ini dibuat oleh Power BI dengan mengambil kumpulan data `churn_metrics_2021` dan menambahkan prediksi tingkat churn untuk Juni 2022. Power BI akan mengambil deret waktu ini dan mengirimkannya ke deteksi Azure Anomaly untuk menyimpulkan batas bawah dan atas yang diharapkan untuk metrik tersebut dan menetapkan tanda untuk anomali yang ditemukan.





Gambar 10.26. Model data di Power BI

Dengan data ini, kita dapat membuat dasbor, yang tampilannya seperti pada Gambar 10.27:



Gambar 10.27. Berbagai komponen dasbor akhir

Mari kita bahas masing-masing komponen.

Deteksi Anomali

Di kiri atas, Anda akan melihat dua diagram garis yang menunjukkan tingkat churn dan tingkat penerimaan penawaran per bulan. Anda dapat melihat bagian dasbor ini pada Gambar di bawah. Meskipun diagram garis itu sendiri 'hanya' mewakili analitik deskriptif dasar ("Apa yang terjadi?"), kami memperkaya diagram dengan menggabungkan dua informasi tambahan

di sini. Yang pertama adalah kami menunjukkan garis abu-abu yang mewakili nilai yang diharapkan dari masing-masing metrik berdasarkan data yang telah kita lihat sejauh ini. Yang kedua adalah kami menyoroti titik data yang menunjukkan lonjakan positif abnormal (baik tingkat churn yang 'terlalu' tinggi maupun tingkat penerimaan tawaran yang 'terlalu' tinggi). Anomali ini digambarkan oleh penanda data persegi. Dengan informasi tambahan ini, kita dapat melihat bahwa tren churn secara keseluruhan tampak negatif (yang merupakan hal yang baik!), dan pada saat yang sama, kami tidak benar-benar mengamati outlier yang jelas meskipun garisnya tampak agak bergelombang.

Untuk tawaran, kita dapat dengan jelas mengamati bahwa lonjakan tinggi pada bulan lalu terdeteksi sebagai anomali, tetapi begitu pula beberapa titik data sebelumnya. Oleh karena itu, tren keseluruhan tampak cukup stabil, kecuali untuk beberapa outlier. Semua informasi ini berasal dari layanan AI deteksi anomali yang kami panggil dengan skrip R atau Python langsung dari Power BI.

→ Menilai apakah perubahan Tingkat Churn dan Penerimaan Tawaran normal atau abnormal (anomali)



Gambar 10.28. Metrik Churn dan Penerimaan Penawaran per Bulan

Analisis Prediktif

Bagian ini akan menggabungkan informasi historis dengan prediksi AI dari atas. Ini mencakup prediksi tingkat churn pelanggan untuk bulan ini, serta terjemahannya terhadap pendapatan yang berisiko. Prediksi tingkat churn pelanggan untuk bulan ini adalah KPI yang kami hitung sebagai metrik ringkasan berdasarkan prediksi AI individual yang telah kami lihat dari model yang kami gunakan di ML Designer. Kami dapat membandingkan metrik ini dengan tujuan yang dalam hal ini didefinisikan sebagai tingkat churn 0,33 - kriteria yang kemungkinan terpenuhi di bulan ini. Agar informasi ini lebih mudah diakses oleh pengguna bisnis, kami menggabungkan metrik ini dengan diagram donat yang disebut "Pendapatan yang berisiko" yang menjumlahkan semua pendapatan bulan ini dari pelanggan yang ditandai dengan tanda churn = 1.



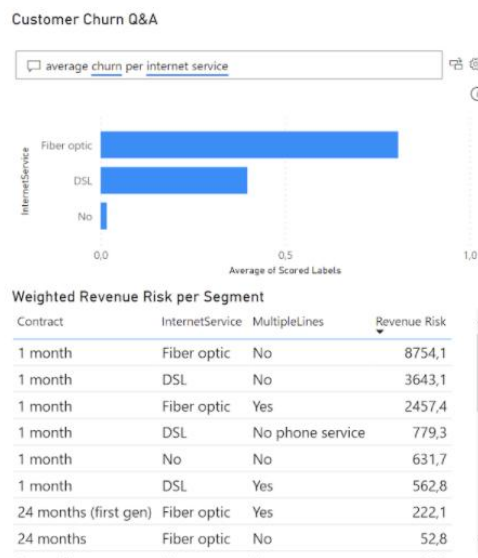
Perlu dicatat bahwa metrik ini agak sederhana karena tidak memperhitungkan nilai umur pelanggan. Kita bisa menambahkan informasi ini jika mau, tetapi demi demo ini, saya memutuskan untuk tetap menggunakan dasar-dasarnya. Intinya adalah kita dapat menggabungkan prediksi AI dengan data BI historis atau transaksional agar prediksi tersebut lebih relevan bagi bisnis. Kita akan melihat contoh lain yang lebih detail di komponen berikutnya.



Gambar 10.29. Metrik KPI prediksi churn

Analisis Deskriptif Berbasis AI

Bagian dasbor ini adalah tempat AI bekerja paling lancar, namun mungkin juga paling intuitif bagi pengguna. Bagian deskriptif berbasis AI dari analisis ini terdiri dari dua grafik seperti yang ditunjukkan pada Gambar 10.30. Bagian atas adalah alat tanya jawab dan bagian bawah adalah tabel visual.



Gambar 10.30. Komponen analitik deskriptif

Mari kita jelajahi alat Tanya Jawab terlebih dahulu. Alat Tanya Jawab menggabungkan skor churn dengan data BI untuk memungkinkan kueri seperti: Berapa rata-rata tingkat churn yang diprediksi di segmen xyz? Visual ini dapat digunakan untuk menjalankan kueri terhadap model data kita menggunakan bahasa alami. Berkat hubungan antara hasil churn yang diprediksi dan data BI dimensional dari gudang data, pengguna dapat dengan mudah memadukan prediksi bertenaga AI dengan properti tambahan tentang pelanggan.

Seperti yang ditunjukkan pada contoh di atas, pengguna dapat menanyakan "Rata-rata churn per layanan internet" dan melihat bahwa tingkat churn untuk Serat optik jauh lebih tinggi daripada kategori layanan internet lainnya.

Satu-satunya pengaturan yang perlu kita buat untuk ini adalah mendefinisikan sinonim di alat Tanya Jawab sehingga "rata-rata churn" dapat dicocokkan dengan kolom data "label skor" sebagaimana disebut awalnya di tabel data sumber. Bagian kedua dalam bentuk tabel di bawah ini adalah contoh lain bagaimana pengguna dapat memanfaatkan kombinasi antara data BI historis dan prediksi AI.

Pengguna dapat menghitung pendapatan berisiko dengan menggabungkan prediksi churn dengan pendapatan saat ini. Untuk melakukan ini, tabel menggunakan kolom baru bernama "Risiko Pendapatan" yang dihasilkan di Power BI sebagai "pengukuran cepat" yang mengalikan pendapatan bulanan pelanggan dengan probabilitas churn masing-masing. Jika kita menjumlahkannya di berbagai kategori pelanggan, kita dapat dengan mudah menemukan segmen pelanggan yang paling banyak kehilangan uang, sehingga angka churn agregat menjadi lebih mudah ditindaklanjuti untuk penjualan dan pemasaran.

Jika kita melihat grafik di atas, kita dapat dengan jelas melihat bahwa segmen pelanggan yang memiliki kontrak bulanan dengan layanan internet serat optik dan tanpa banyak saluran memberikan leverage terbesar untuk mempertahankan pendapatan dengan risiko pendapatan sebesar Rp 87.540.000 per bulan. Jika pemasaran mempertimbangkan insentif penawaran counter-churn yang dipersonalisasi, kategori ini akan menarik untuk diteliti lebih lanjut.

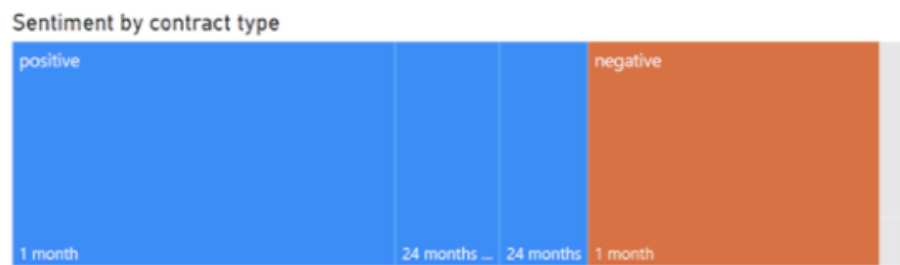
Data tidak Terstruktur

Dan terakhir, kami memberikan lebih banyak wawasan tentang skor sentimen yang dikombinasikan dengan data dimensional. Analisis sentimen dari data tidak terstruktur pada akhirnya menjadi bagian dari model prediksi churn kami, tetapi juga dapat berdiri sendiri untuk memberikan wawasan lebih lanjut. Visual peta pohon seperti yang ditunjukkan pada Gambar di bawah membandingkan label sentimen dengan dimensi pelanggan yang kami inginkan - dalam hal ini jenis kontrak.

Dari visual ini, kita dapat melihat bahwa meskipun sebagian besar pelanggan sebenarnya puas dengan sentimen komentar survei mereka, mayoritas pelanggan yang memberikan umpan balik negatif menggunakan paket bulanan. Mengingat tingginya kemungkinan churn pelanggan dalam kategori ini, prioritas pertama adalah menyelidiki kategori ini untuk mengetahui apa yang pada dasarnya salah di sini.

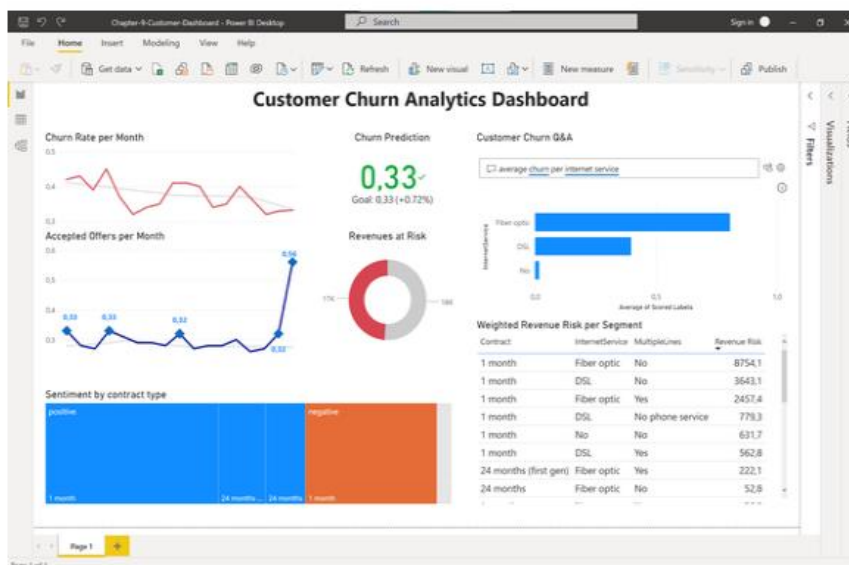


Jika kita menggabungkan semuanya dan melihat dasbornya sendiri seperti yang terlihat pada Gambar 10.32, kita dapat menyediakan dasbor yang berpusat pada pelanggan untuk Pemasaran dan Penjualan yang menggunakan AI di balik layar, bukan untuk pamer, tetapi dengan satu tujuan: Membuat data lebih mudah ditindaklanjuti dan lebih bermakna bagi para pemangku kepentingan bisnis.



Gambar 10.31. Analisis Sentimen

Cobalah sendiri, buka berkas Power BI yang disertakan di komputer Anda dan bereksperimen dengan berbagai komponen di dalamnya. Informasi mana yang akan Anda tambahkan, ubah, atau hapus seluruhnya?



Gambar 10.32. Dasbor Akhir

10.4 KESIMPULAN

Saya harap bab terakhir ini telah menunjukkan kepada Anda bahwa menggunakan layanan AI dalam BI Anda bukanlah keputusan "salah satu" - "atau". Faktanya, manfaat terbesar datang dari menggabungkan beberapa layanan AI. Namun, jangan biarkan teknologi ini menyeret Anda ke dalam lubang kelinci di mana fungsionalitas BI dipenuhi dengan komponen AI yang bekerja. Ingatlah bahwa setiap layanan AI yang Anda tambahkan juga menambah lapisan utang teknis lainnya seseorang pada akhirnya harus mengurus model, alat

Tanya Jawab, integrasi data, dan sebagainya. Namun, menggabungkan berbagai layanan AI merupakan cara yang bagus untuk memvalidasi apakah bisnis mendapatkan manfaat dari wawasan lebih lanjut atau menemukan apa yang sebenarnya dicari bisnis.

Dengan tumpukan analitik fleksibel Anda yang mencakup mulai dari mengakses data tidak terstruktur, analitik deskriptif hingga analitik preskriptif dari dalam platform yang sama, Anda akan berada di posisi yang tepat untuk menemukan titik optimal antara menambahkan kompleksitas teknis dan memberikan nilai bisnis yang sebenarnya.

Selamat, Anda telah berhasil membangun prototipe pertama Anda yang berisi tidak kurang dari 4 layanan AI yang berbeda! Lalu apa selanjutnya? Seperti yang telah disebutkan, ini masih berupa prototipe dan masih merupakan proses yang baik sebelum kami dapat mengirimkan dasbor ini ke lingkungan yang produktif. Di bab selanjutnya, kita akan mempelajari lebih lanjut tentang apa yang diperlukan untuk beralih dari prototipe ke produksi dan cara mengatasi kompleksitas yang menyertainya.



BAB 11

LANGKAH BERIKUTNYA: DARI PROTOTYPE KE PRODUKSI

11.1 PENEMUAN VS. PENGIRIMAN

Namun, perlu diingat bahwa apa yang telah dilakukan sejauh ini "hanya" pembuatan prototipe. Anda melakukan apa yang disebut manajer produk sebagai proses "penemuan". Penemuan adalah tentang memvalidasi nilai, kegunaan, kelayakan atau viabilitas suatu produk, atau kasus penggunaan.

Perlu dicatat bahwa sebagian besar prototipe kemungkinan besar tidak akan bertahan dalam fase penemuan. Itulah sifat permainannya: Anda ingin belajar dengan cepat dan gagal dengan cepat. Tidak masalah. Saat ini, Anda seharusnya tidak memiliki terlalu banyak sumber daya yang diinvestasikan, dan idealnya Anda memiliki ide-ide lain dalam daftar tunggu Anda untuk dikembangkan. Tetapi apa yang terjadi jika prototipe Anda benar-benar bertahan dalam fase validasi? Apa yang terjadi jika prototipe tersebut berhasil? Katakanlah pengguna uji BI awal Anda menyukai fitur baru atau prediksi model. Semua orang antusias dengan solusi baru Anda dan beberapa orang bahkan mungkin menyebarkannya di perusahaan.

Bagaimana Anda menskalakan prototipe Anda untuk produksi? Jawaban sederhananya adalah: Anda tidak menskalakannya sama sekali. Dalam contoh pembuatan prototipe kami, kami melakukan semuanya sendiri: kami membuat mesin virtual untuk pekerjaan komputasi, mengelola kunci akses untuk penyimpanan, dan bahkan menetapkan kebijakan keamanan untuk kontainer kami.

Anda mungkin sudah menebaknya: ini bukan yang seharusnya Anda lakukan dalam skenario produksi. Platform cloud Azure (dan semua platform lainnya) mungkin terlihat sangat intuitif dan mudah digunakan setelah beberapa pelatihan. Namun kenyataannya, platform tersebut merupakan lanskap yang sangat kompleks yang membutuhkan pengetahuan ahli untuk mengelola dan mengendalikannya secara aman dan efektif. Jika tidak, Anda berisiko mengalami pelanggaran data atau lonjakan tagihan sumber daya, atau keduanya.

Jika sebuah prototipe ternyata berhasil dan lulus uji dampak dan kelayakan, saatnya untuk melepaskannya dan beralih ke tahap yang berbeda: Pengiriman. Di perusahaan, ini lebih dari sekadar membuat versi "bersih" dari kode dan dasbor Anda. Saat Anda melanjutkan ke Pengiriman, Anda harus memprioritaskan hal-hal berikut:

- Skalabilitas
- Keandalan
- Performa
- Kemudahan Pemeliharaan

Selain itu, prototipe biasanya memiliki beberapa kerentanan keamanan karena tidak diuji dengan benar pada tahap pengembangan ini. Oleh karena itu, Anda sebaiknya tidak langsung memindahkan prototipe ke tahap produksi, tetapi pikirkan kembali dimensi-dimensi ini. Mari kita bahas konsep-konsep ini secara singkat:



- Skalabilitas mengacu pada tantangan teknis dan non-teknis dalam menyediakan solusi Anda bagi lebih banyak pengguna. Secara teknis, ini tentang batas atas berapa banyak pengguna yang dapat mengakses solusi secara bersamaan, yang dibatasi oleh infrastruktur Anda. Dalam istilah non-teknis, Anda ingin mengetahui bagaimana proses dapat menghambat penerapan aplikasi. Misalnya, jika Anda harus menyiapkan dasbor secara manual untuk setiap pengguna, ukuran maksimum solusi Anda akan dibatasi oleh jumlah orang yang dapat Anda tugaskan. Jika Anda menerapkan dasbor secara otomatis atau menggunakan server BI khusus, Anda dapat menjalankan solusi Anda dalam skala yang jauh lebih besar.
- Keandalan berarti bahwa solusi Anda melakukan apa yang diharapkan. Ketika terjadi kegagalan, apakah dapat diperbaiki dengan mudah dan dengan sedikit atau tanpa waktu henti? Cepat atau lambat, Anda perlu menanggapi pertanyaan pengguna dan sering kali menyediakan bantuan daring. Seberapa cepat masalah ini dapat diperbaiki?
- Performa mengacu pada kecepatan aplikasi Anda merespons permintaan pengguna. Seiring bertambahnya jumlah pengguna, hal ini menjadi faktor penting. Kesan pertama sangatlah penting, dan aplikasi yang lambat dimuat atau interaksi yang lambat dapat menghambat pertumbuhan. Lagipula, Anda ingin pengguna dapat menyelesaikan tugas mereka secepat mungkin tanpa mengalami penundaan.
- Kemudahan pemeliharaan (*maintainability*) berkaitan dengan seberapa mudahnya melakukan perubahan setelah solusi berjalan. Apakah fitur ditambahkan atau diubah tanpa memerlukan programmer? Apakah kode Anda didokumentasikan dengan baik sehingga tidak terjadi kesalahan tak terduga saat pengguna baru mengerjakannya?

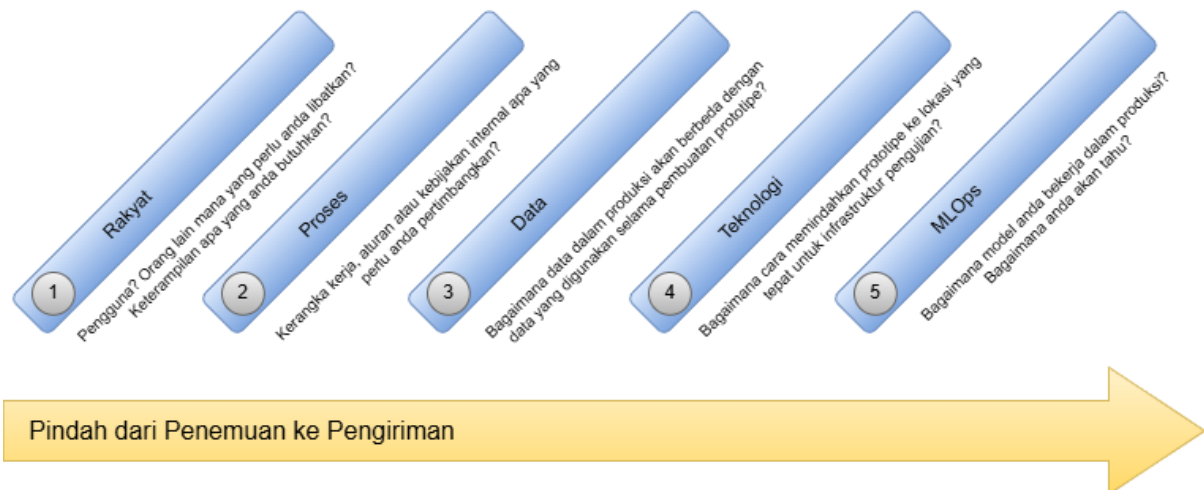
Aplikasi produksi Anda harus memiliki skalabilitas, keandalan, performa, dan kemudahan pemeliharaan yang terintegrasi sejak awal. Prioritas sebenarnya bergantung pada kebutuhan bisnis Anda. Jika Anda ingin menyediakan solusi Anda kepada banyak pengguna, skalabilitas sangatlah penting. Di sisi lain, jika aplikasi perusahaan dirancang hanya untuk sejumlah pengguna tertentu, tidak masalah untuk menurunkan prioritas tujuan skalabilitas dan mendokumentasikan batasan ini dalam proyek pengiriman Anda.

Apa pun pilihannya: Dalam kebanyakan kasus, tumpukan prototipe Anda tidak memenuhi prinsip-prinsip pengiriman ini karena Anda telah mengorbankannya demi kecepatan dan kenyamanan pengembang. Itulah mengapa Anda sebaiknya tidak mencoba menskalakan prototipe, tetapi membangunnya kembali dengan benar selangkah demi selangkah.

11.2 KRITERIA KEBERHASILAN PENGIRIMAN PRODUK AI

Jadi, bagaimana Anda mencapai transisi yang sukses dari fase Penemuan ke Fase Pengiriman? Untuk mengembangkan produk AI perusahaan yang sukses dalam konteks BI, Anda perlu mempertimbangkan 5 dimensi: Orang, Proses, Data, Teknologi, dan MLOps, seperti yang ditunjukkan pada Gambar 11.1.





Gambar 11.1. Kriteria Keberhasilan Pengiriman Produk AI dalam Konteks BI

Mari kita bahas lebih detail:

Orang

Orang menentukan keberhasilan atau kegagalan produk AI Anda.

Selama fase penemuan, tim Anda mungkin tidak melampaui aturan 2 pizza. Artinya: semua orang dalam tim proyek merasa puas dengan dua pizza besar. Biasanya 3-5 orang.

Setelah Anda menargetkan pengiriman dalam skenario perusahaan, hal ini akan berubah. Proyek AI perusahaan pada dasarnya adalah proyek perangkat lunak perusahaan. Alih-alih dua pizza, pertimbangkan untuk memesan prasmanan besar, karena ada berbagai macam orang yang mungkin terlibat dalam proyek pengembangan perangkat lunak baru.

Ini mungkin termasuk, sebagai contoh:

- Keamanan TI
- Manajemen aplikasi TI
- Manajemen infrastruktur TI
- Kantor tata kelola data
- Pimpinan bisnis dari berbagai departemen lain
- Pengacara
- Dewan pekerja
- ...

Berbagai fungsi yang perlu Anda sertakan sangat bergantung pada bagaimana perusahaan Anda diorganisasikan dan seberapa besar perusahaan tersebut. Orang-orang ini seringkali sejalan dengan kerangka kerja proses yang perlu Anda ikuti (lihat poin berikutnya).

Meskipun saya tidak bisa memberikan daftar lengkap orang-orang yang harus Anda ajak bicara, saya secara umum merasa bermanfaat untuk mengorganisasikan berbagai pemangku kepentingan Anda dalam matriks pemangku kepentingan yang memetakan orang-orang berdasarkan dua dimensi luas: Kekuasaan atas proyek dan minat mereka terhadap pekerjaan tersebut.

Tujuannya adalah mengidentifikasi pemangku kepentingan di setiap kategori untuk menentukan strategi yang tepat dalam berkomunikasi dengan mereka. Ada lima kategori:



Kelola secara ketat, Jaga kepuasan, Jaga informasi, dan Pantau, yang akan kita bahas secara singkat.

- ❖ **Kelola secara ketat:** Kategori ini mencakup orang-orang yang memiliki kekuasaan besar dan minat tinggi terhadap proyek Anda. Mereka adalah pemangku kepentingan terpenting Anda. Anda perlu sering berbicara dengan mereka, mengidentifikasi kekhawatiran sejak dini, dan terlibat agar mereka tetap sepaham dan mengelola ekspektasi mereka.

Contoh: Sponsor, manajemen puncak, direktur, Manajer Proyek.

- ❖ **Jaga kepuasan:** Mereka adalah orang-orang yang memiliki kekuasaan besar tetapi kurang tertarik pada proyek Anda. Mereka seringkali tidak terdeteksi, tetapi sangat penting karena mereka memiliki kekuasaan untuk memutuskan apakah proyek harus dilanjutkan atau tidak, tergantung pada hasil atau kemajuan saat ini.

Contoh: Manajemen program, perencanaan keuangan

- ❖ **Jaga informasi:** Orang-orang yang memiliki banyak kepentingan tetapi sedikit kekuasaan terhadap proyek Anda perlu terus diberi informasi secara berkala. Mereka biasanya adalah orang-orang yang pada akhirnya akan menggunakan apa yang sedang dibangun. Meskipun kekuasaan formal mereka kecil, Anda harus sering berkomunikasi dengan mereka. Mereka begitu terlibat sehingga cenderung memengaruhi pemangku kepentingan lain (yang lebih berkuasa).

Contoh: Pengguna akhir, klien

- ❖ **Pemantau:** Orang-orang dengan sedikit kekuasaan dan minat dalam proyek Anda. Orang-orang ini mungkin bukan prioritas bagi Anda, dan Anda tidak ingin terlalu mengganggu mereka. Namun, mereka tetap perlu dipantau dan diinformasikan secara sporadis, meskipun kecil kemungkinannya mereka akan menimbulkan masalah besar.

Contoh: Karyawan dari departemen lain

Setelah Anda mulai memetakan berbagai orang ke dalam kategori-kategori ini, Anda akan mendapatkan gambaran tentang kompleksitas proyek Anda dan Anda dapat menyesuaikan matriks ini seiring perkembangan proyek Anda.

Proses

Karena proyek AI pada dasarnya adalah proyek perangkat lunak, Anda perlu memperhatikan kerangka kerja dan proses di perusahaan Anda yang mengatur proses pengembangan artefak perangkat lunak. Tergantung pada organisasi Anda, mungkin terdapat aturan yang berbeda dan tidak mungkin untuk mencantumkan semuanya dalam buku ini. Namun, jika Anda umumnya tidak familiar dengan bagaimana proyek perangkat lunak ditangani, saya akan memberikan beberapa istilah kunci yang perlu diperhatikan untuk melihat apakah istilah-istilah tersebut mungkin berlaku untuk organisasi Anda:

- **ISO/IEC:** Seri standar ISO/IEC 25000 juga dikenal sebagai SQuaRE (Persyaratan dan Evaluasi Kualitas Sistem dan Perangkat Lunak). Standar-standar ini bertujuan untuk menyediakan kerangka kerja untuk mengevaluasi kualitas produk perangkat lunak. Jika organisasi Anda menggunakan standar bersertifikat ISO, proyek Anda mungkin termasuk dalam aturan ini dan perlu disiapkan sesuai dengan itu.



- Scrum: Scrum adalah kerangka kerja pengembangan perangkat lunak yang digunakan oleh perusahaan untuk pengembangan perangkat lunak yang gesit. Meskipun tujuannya adalah untuk memungkinkan proses pengembangan yang ringan dan iteratif, Scrum memiliki beban kerja yang relatif ketat dan peran yang tetap, misalnya Pemilik Produk dan Master Scrum. Pastikan ada kerangka kerja seperti Scrum di organisasi Anda karena ini akan menentukan struktur proyek.
- BPM: Konsep "Manajemen Proses Bisnis" (BPM) umumnya digunakan di organisasi besar untuk mengelola proses bisnis utama dan masing-masing pemiliknya. BPM juga digunakan dalam pengembangan perangkat lunak dan Anda mungkin perlu mendokumentasikan proyek Anda sesuai dengan itu.
- PMO: Kantor Manajemen Proyek (PMO) adalah departemen internal tempat standar proyek khusus perusahaan ditetapkan dan dipelihara. Bergantung pada ukuran proyek dan cara kerja perusahaan Anda, Anda mungkin setidaknya harus mematuhi aturan mereka dan terkadang bahkan melaporkannya secara berkala.

Setelah prototipe Anda memiliki kesempatan untuk memasuki fase pengiriman, saya sangat menyarankan Anda untuk menghubungi departemen TI/DevOps Anda (jika Anda belum melakukannya selama pembuatan prototipe) untuk memastikan bahwa pengaturan proyek Anda memenuhi semua persyaratan formal yang diperlukan untuk proyek pengembangan perangkat lunak.

Data

Data ini seharusnya tidak lagi mengejutkan Anda. Selama pembuatan prototipe, Anda seharusnya sudah dapat mengidentifikasi jebakan data terbesar, dan jika Anda berhasil mencapai tahap pengiriman, Anda entah bagaimana telah menemukan cara untuk mengatasinya. Satu-satunya pertanyaan yang tersisa adalah: Akankah pendekatan yang Anda ambil berhasil dalam produksi? Apakah pendekatan tersebut aman dan andal? Apakah skalabilitasnya memadai? Adakah hal lain yang perlu Anda pertimbangkan?

Berasal dari BI, Anda seharusnya memiliki pengetahuan yang kuat tentang faktor-faktor yang dapat Anda gunakan dalam organisasi Anda untuk memindahkan data dan apa saja persyaratan atau kendala yang ada. Berikut beberapa area yang perlu Anda perhatikan secara khusus:

- ✓ Jalur data: Ingat bagaimana kita dulu mengunggah file CSV secara manual ke Azure ML Studio dan menyimpan hasilnya di Azure Blob Storage? Kemungkinan besar, pendekatan ini tidak akan berhasil dalam skenario produksi karena terlalu banyak hal yang dapat rusak. Idealnya, Anda mengambil data dari gudang data dan menyimpan hasil ekspornya sesuai skema yang konsisten di tempat yang dapat diakses langsung oleh aplikasi produksi Anda bahkan bisa berupa Azure Blob Storage.
- ✓ Pembersihan dan manajemen data: Setelah Anda mengotomatiskan alur data dan membuat skrip ETL yang sebenarnya, Anda perlu memastikan keduanya dapat dipelihara dan diotomatisasi (Anda tidak akan dapat melakukan pembersihan data manual). Tergantung pada ukuran proyek, bagian pembersihan data ini saja dapat menjadi kendala utama bagi keseluruhan inisiatif.



- ✓ Akses data: Saya harap kasus penggunaan ini telah menunjukkan kepada Anda pentingnya kemudahan mengakses dan berbagi data di berbagai tahap proses pengembangan. Jika Anda menyimpan semua data di gudang data cloud terpusat, mengaksesnya "hanya" masalah hak akses, bukan kendala teknis. Saya tahu bahwa banyak perusahaan masih berjuang untuk menghilangkan silo data mereka dan meningkatkan pembagian data di berbagai unit bisnis. Idealnya, kasus penggunaan prototipe Anda telah memberikan contoh yang jelas tentang manfaat membuat data mudah diakses dan menghubungkan aplikasi Anda ke sumber data tersebut dengan beberapa skrip sederhana. Menghosting gudang data Anda di cloud dapat memiliki kekurangan, tetapi dalam hal pengembangan yang cepat dan fleksibel, ini jelas merupakan keuntungan besar.
- ✓ Perlindungan data: Saat Anda beralih ke tahap produksi dan melayani lebih banyak pengguna, pertimbangkan apakah data Anda memiliki batasan lain dibandingkan dengan data yang Anda gunakan dalam pembuatan prototipe. Misalnya, dalam pembuatan prototipe, Anda mungkin menggunakan data pelanggan dari AS dan mengolahnya di Microsoft Azure. Dalam produksi, pengguna Anda mungkin juga berada di Uni Eropa dan Anda memerlukan izin khusus untuk memproses data tersebut jika berisi informasi pribadi. Menyimpan data di luar Uni Eropa juga dapat dengan cepat menjadi masalah bagi kelompok pengguna ini. Jika Anda memiliki kerangka kerja tata kelola data, libatkan tim tata kelola data Anda sejak dini, idealnya pada tahap pembuatan prototipe, untuk mengidentifikasi kendala utama sejak dini.

Teknologi

Dalam 99% kasus, tumpukan teknologi yang Anda gunakan untuk pembuatan prototipe tidak akan sama dengan tumpukan teknologi yang Anda gunakan dalam produksi. Dan jika ya selamat Anda telah memahami apa itu transformasi digital. Menggunakan infrastruktur yang sama untuk pembuatan prototipe cepat dan pengujian fitur baru, serta menjalankan beban kerja produksi dalam skala besar, bukanlah hal yang umum. Alasannya adalah sistem produksi dioptimalkan agar skalabel, andal, mudah dipelihara, berkinerja tinggi, dan aman. Seperti yang telah kita pelajari, kita umumnya mengorbankan tujuan-tujuan ini demi pengembangan yang lebih cepat dan fleksibel saat membuat prototipe.

Jika Anda ingin meluncurkan aplikasi perangkat lunak baru atau bahkan satu fitur dalam sistem (BI) yang sudah ada, Anda harus dipandu oleh persyaratan lingkungan pengujian. Tahap pengujian biasanya akan menyerupai lingkungan produksi Anda dan merupakan tonggak pertama Anda untuk menghadirkan sesuatu yang dapat digunakan untuk pengujian.

Pertanyaan terpenting yang perlu Anda tanyakan pada diri sendiri saat ini adalah:

Bagian mana dari prototipe saya yang perlu saya buat ulang untuk mengirimkannya ke lingkungan pengujian dan mana yang dapat saya gunakan kembali? Jawaban atas pertanyaan ini bisa berada di antara dua ekstrem: A) Anda tidak perlu membangun ulang apa pun karena tumpukan prototipe sama dengan tumpukan pengujian. Dalam hal ini, Anda hanya perlu fokus untuk benar-benar meningkatkan berbagai hal dan memastikan bahwa apa yang Anda bangun memenuhi kriteria lingkungan produksi untuk skalabilitas, keandalan, kemudahan perawatan,



performa, dan keamanan. Ekstrem lainnya adalah opsi B, di mana Anda pada dasarnya harus membangun ulang semuanya.

Dalam kebanyakan kasus, Anda akan berada di antara kedua kutub ini dan Anda perlu mengidentifikasi komponen yang dapat Anda gunakan kembali. Apa pun yang tersisa perlu dibangun ulang dan akan menentukan ukuran tim pengembangan dan anggaran Anda. Semakin kompleks produk Anda, semakin tinggi biaya pengembangan dan pemeliharaannya.

Berdasarkan kasus penggunaan yang telah kami bahas dalam buku ini, berikut adalah beberapa komponen yang telah kami gunakan dalam pembuatan prototipe yang mungkin dapat Anda gunakan kembali bahkan dalam produksi:

- ✚ API Titik Akhir Model: Model yang telah kami implementasikan menggunakan Azure ML Studio umumnya memenuhi semua persyaratan beban kerja produksi. Pastikan untuk memeriksa bagian selanjutnya, "MLOps", untuk kriteria model pembelajaran mesin dalam produksi. Namun, infrastrukturnya sendiri akan kompatibel.
- ✚ Antarmuka Pengguna/Laporan: Jika Anda menggunakan Power BI dalam lanskap pelaporan yang ada, Anda mungkin dapat menggunakan kembali tata letak laporan dan dasbor yang telah kita buat dalam latihan. Cara Anda menyajikannya kepada pengguna mungkin berbeda (misalnya, menggunakan Power BI Server), tetapi tampilan dan nuansanya akan tetap sama, sehingga Anda tidak perlu membuatnya lagi.
- ✚ Alur Data: Kami menggunakan Dataset di Azure ML Studio untuk melatih model kami. Selama Anda mempertahankan skemanya, Anda seharusnya dapat menggabungkan data baru dengan cukup cepat, baik dengan mengimpor data dari file maupun dari sumber daya lain di Azure. Selama Anda berhasil memasukkan data ke dalam Dataset menggunakan skema yang sama, alur untuk persiapan data di Azure ML Studio tetap sama.
- ✚ Dokumentasi: Idealnya, Anda telah menggunakan fase pembuatan prototipe untuk membuat beberapa catatan tentang keseluruhan proses pengembangan Anda. Ini bisa berupa dokumentasi non-teknis seperti "Masalah apa yang Anda selesaikan?", "Pendekatan apa yang Anda ambil?", "Siapa saja yang Anda libatkan?", serta dokumentasi yang lebih teknis (skema data, dokumentasi API, model data Power BI). Kemungkinan besar Anda dapat menggunakan kembali komponen-komponen ini, atau setidaknya menggunakannya sebagai titik awal untuk memindahkan proyek prototipe ke fase pengiriman.

Usahakan untuk menggunakan kembali sebanyak mungkin dan membangun kembali semua yang lain dengan benar.

MLOps

Proses pengelolaan dan pengoperasian model pembelajaran mesin dalam sistem produksi merupakan area yang umum disebut sebagai MLOps. Minimal, ini membutuhkan dokumentasi API dan staf yang memadai untuk menangani model setelah diterapkan.

Dalam kasus terbaik, MLOps mencakup seluruh siklus hidup solusi pembelajaran mesin: mulai dari akuisisi data hingga kontrol versi, eksperimen, pemantauan kinerja, manajemen titik akhir, hingga penghentian model. Oleh karena itu, model pembelajaran mesin



dapat dianggap sebagai proyek perangkat lunak ujung ke ujung dan harus dikelola dengan tepat. Oleh karena itu, tidak mengherankan jika Anda akan menemukan beberapa konsep yang sama dalam MLOps yang juga diterapkan pada solusi keseluruhan yang menggunakan model pembelajaran mesin. Aspek-aspek berikut merupakan faktor keberhasilan terpenting bagi MLOps:

Manusia

Sejujurnya: Jika Anda tidak memiliki seseorang untuk setidaknya memantau model Anda dari waktu ke waktu dan mengurus pemeliharannya, lebih baik Anda tidak mengimplementasikan model pembelajaran mesin sama sekali. Ada banyak hal yang dapat terjadi setelah implementasi. Fenomena yang paling umum adalah konsep yang disebut "penyimpangan data".

Ini berarti bahwa data yang dilihat model Anda dalam produksi menjadi semakin berbeda dari data yang dilihatnya selama pelatihan dan membuat prediksi menjadi kurang akurat dan kurang andal seiring waktu. Meskipun ada alat untuk membantu mengotomatiskan proses ini, pada akhirnya, manusialah yang memutuskan apakah model baru perlu dilatih ulang dan data apa yang akan digunakan untuk itu. Selain kinerja model, debugging juga merupakan komponen yang sangat penting. Jika model menghasilkan kesalahan atau masalah lain muncul, siapa yang bertanggung jawab untuk memperbaikinya?

Otomatisasi

Salah satu tujuan utama MLOps adalah mengotomatiskan proses semaksimal mungkin. Jika Anda tidak dapat dengan mudah mereproduksi analisis atau pelatihan model, kecil kemungkinan tim Anda akan dapat mengulangi model baru dengan cukup cepat untuk memberikan nilai. Karena alasan ini, pipeline otomatis dianggap sebagai salah satu aspek terpenting dalam MLOps, terutama jika Anda menjalankan lebih dari satu atau dua model dalam produksi.

Keandalan dan Kegunaan Ulang

Jika organisasi Anda memperluas penggunaan model ML, penting bagi Anda untuk menetapkan proses standar pembuatannya yang andal dan yang pengoperasiannya dapat diverifikasi dengan mudah setiap saat. Jika Anda gagal melakukan ini, biaya pembuatan model baru atau perubahan selanjutnya pada model yang ada dapat melebihi nilai model pembelajaran mesin. Anda juga perlu memastikan bahwa ketika terjadi perubahan dalam tim, transisi berjalan lancar dan rekan kerja baru dapat mengambil alih dengan cepat.

Keamanan

Penting bagi tim MLOps Anda untuk memahami data apa yang mereka gunakan dan aturan tata kelola apa yang berlaku. Dalam beberapa kasus, data ini mencakup informasi pengguna sensitif yang mungkin tidak dapat diakses di luar wilayah geografis tertentu, mungkin tidak dapat diakses oleh semua orang, atau mungkin penggunaannya dibatasi oleh undang-undang privasi. MLOps juga memastikan bahwa semua data dan sumber daya infrastruktur dikelola melalui aturan manajemen identitas dan akses (IAM) dan pengaturan teknis yang sesuai.



Skalabilitas

Mengikuti jejak sumber daya ML yang terus berkembang merupakan perhatian utama MLOps. Salah satu cara terbaik untuk menghindari hal ini adalah dengan meningkatkan skala infrastruktur Anda sesuai kebutuhan, baik selama pelatihan maupun inferensi. Di satu sisi, Anda ingin menghindari situasi di mana Anda membatasi kemajuan pelatihan pembelajaran mesin karena tidak menyediakan infrastruktur yang cukup atau tidak tepat. Di sisi lain, Anda tidak ingin mencadangkan kluster komputer yang besar untuk pekerjaan ML yang hanya berjalan beberapa bulan sekali.

Platform pembelajaran mesin khusus seperti Azure ML Studio akan membantu Anda mengatasi beberapa masalah yang dihadapi MLOps. Namun pada akhirnya, organisasi Anda membutuhkan kerangka kerja proses dan sumber daya manusia untuk mengelola aspek-aspek ini. Pastikan untuk mempertimbangkannya sebelum memindahkan layanan ML kustom Anda sendiri ke tahap produksi. Beberapa area ini juga berlaku untuk layanan AI, meskipun dalam hal ini Anda biasanya memiliki vendor yang akan mengurus sebagian besar hal atau mendukung Anda sesuai kebutuhan.

Mulailah dengan Memberikan Peningkatan

Ya, itu banyak. Itulah mengapa Anda tidak boleh memulai perjalanan ini sendirian. Konsultasikan dengan departemen TI Anda atau pakar eksternal sesuai kebutuhan jika Anda ingin serius dan memperluas upaya AI Anda. Anda akan membutuhkan banyak dukungan, terutama di awal, untuk menghindari kesalahan serius.

Di saat yang sama, jangan terlalu terintimidasi oleh prosesnya. Sekarang, Anda seharusnya sudah yakin bahwa langkah dari penemuan hingga pengiriman benar-benar sepadan dengan semua upaya yang dikeluarkan. Saat memulai, jangan menargetkan hasil yang besar, tetapi cobalah untuk memberikannya secara bertahap. Dengan begitu, Anda akan tetap fleksibel terhadap perubahan dan memiliki kesempatan untuk merayakan keberhasilan kecil di sepanjang jalan.

11.3 KESIMPULAN

Anda berhasil!

Saya harap buku ini memberi Anda wawasan tentang dunia AI dan bagaimana AI dapat digunakan dalam BI. Ada sejumlah kasus penggunaan yang telah kami bahas di sini untuk memberi Anda gambaran tentang banyak kemungkinan yang dapat Anda jelajahi dengan aplikasi pembelajaran mesin.

Dengan bantuan buku ini, Anda seharusnya sudah berada di jalur yang tepat untuk mengembangkan aplikasi Kecerdasan Bisnis berbasis AI Anda sendiri. Jadi, apa yang telah Anda pelajari? Berikut beberapa konsep yang kami bahas:

- Cara menemukan kasus penggunaan yang berdampak pada bisnis
- Cara menilai kelayakan proyek AI
- Hal-hal dasar yang perlu diketahui tentang pembelajaran mesin
- Cara menggunakan prototipe untuk penemuan produk dan alat apa yang akan digunakan dalam konteks prototipe AI



- Kasus penggunaan praktis tentang bagaimana AI dapat digunakan untuk meningkatkan hasil BI.
- Cara beralih ke fase berikutnya untuk membawa proyek Anda dari prototipe ke produksi

Saya harap kasus penggunaan dalam buku ini telah membangkitkan minat Anda tentang bagaimana kecerdasan buatan dapat meningkatkan hasil bisnis di BI di bidang pengumpulan data, analitik deskriptif, diagnostik, prediktif, atau preskriptif.

Jika ya, jangan berhenti di sini. Langkah selanjutnya adalah membangun aplikasi Kecerdasan Bisnis Anda sendiri menggunakan teknik kecerdasan buatan. Dunia berubah dengan cepat, tetapi dengan informasi dalam buku ini, Anda akan diperlengkapi dengan baik untuk menavigasi bidang ini sendiri. Saya mendoakan yang terbaik untuk perjalanan AI Anda di masa depan dan berharap Anda akan meluncurkan banyak proyek yang sukses. Anda selalu dapat menghubungi saya jika Anda memiliki masukan, pertanyaan, atau ide untuk prototipe seputar kecerdasan bisnis bertenaga AI dan ingin sukses di salah satu bidang ini.



DAFTAR PUSTAKA

- Agbadamasi, T. O., Opoku, L. K., Adukpo, T. K., & Mensah, N. (2025). The role of business intelligence in AI ethics: Empowering US companies to achieve transparent and responsible AI. *EPRA International Journal of Economics, Business and Management Studies (EBMS)*, 12(3), 8-14.
- Airinei, D., & Homocianu, D. (2010). *The mobile business intelligence challenge*. *Economy Informatics*.
- Alhawamdeh, H., Alkhalwaldeh, B. Y., Zraqat, O., & Alhawamdeh, A. M. (2024). Leveraging business intelligence in organizational innovation: a leadership perspective in commercial banks. *International Journal of Academic Research in Accounting, Finance and Management Sciences*, 14(1), 295-309.
- Alijoyo, F. A., Pradhan, R., Vats, S., Rani, V. K., Kholmukhamedov, T., & Karthik, M. (2024, December). AI-Powered Business Intelligence for Smarter Decision-Making and Growth. In *2024 International Conference on Artificial Intelligence and Quantum Computation-Based Sensor Application (ICAIQSA)* (pp. 1-5). IEEE.
- Angle, M. (2024). *From Data to Decisions: How AI-Driven Analytics Reshapes Business Intelligence and Profitability*.
- Ao, S. I., Hurwitz, M., & Palade, V. (2025). Cognitive computing and business intelligence applications in accounting, finance and management. *Big Data and Cognitive Computing*, 9(3), 54.
- Azmi, M., Mansour, A., & Azmi, C. (2023). A context-aware empowering business with ai: Case of chatbots in business intelligence systems. *Procedia Computer Science*, 224, 479-484.
- Basole, R. C., Park, H., & Seuss, C. D. (2024). Complex business ecosystem intelligence using AI-powered visual analytics. *Decision Support Systems*, 178, 114133.
- Bharadiya, J. P. (2023). The role of machine learning in transforming business intelligence. *International Journal of Computing and Artificial Intelligence*, 4(1), 16-24.
- Biere, M. (2003). *Business intelligence for the enterprise*. Prentice Hall Professional.
- Bulusu, L., & Abellera, R. (2020). *AI meets BI: Artificial intelligence and business intelligence*. Auerbach Publications.
- Chintala, S., & Thiyagarajan, V. (2023). AI-Driven Business Intelligence: Unlocking the Future of Decision-Making. *ESP International Journal of Advancements in Computational Technology*, 1, 73-84.

- Das, B. C., Mahabub, S., & Hossain, M. R. (2024). Empowering modern business intelligence (BI) tools for data-driven decision-making: Innovations with AI and analytics insights. *Edelweiss Applied Science and Technology*, 8(6), 8333-8346.
- Deckler, G. (2019). *Learn Power BI: A beginner's guide to developing interactive business intelligence solutions using Microsoft Power BI*. Packt Publishing Ltd.
- Deshpande, N., Ahmed, S., & Khode, A. (2016). Business intelligence through patinformatics: A study of energy efficient data centres using patent data. *Journal of Intelligence Studies in Business*, 6(3).
- Edge, D., Larson, J., & White, C. (2018, April). Bringing AI to BI: enabling visual analytics of unstructured data in a modern Business Intelligence platform. In *Extended abstracts of the 2018 CHI conference on human factors in computing systems* (pp. 1-9).
- Foster, K., Smith, G., Ariyachandra, T., & Frolick, M. N. (2015). Business intelligence competency center: Improving data and decisions. *Information Systems Management*, 32(3), 229-233.
- Gowthami, K., & Kumar, M. P. (2017). Study on business intelligence tools for enterprise dashboard development. *International Research Journal of Engineering and Technology*, 4(4), 2987-2992.
- Guo, B., Liu, Y., Ouyang, Y., Zheng, V. W., Zhang, D., & Yu, Z. (2019). Harnessing the power of the general public for crowdsourced business intelligence: a survey. *IEEE Access*, 7, 26606-26630.
- Halim, K. K., & Halim, S. (2019). Business intelligence for designing restaurant marketing strategy: A case study. *Procedia Computer Science*, 161, 615-622.
- Hodinka, M., Štencl, M., Hřebíček, J., & Trenz, O. (2014). Business intelligence in Environmental reporting powered by XBRL. *Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis*, 62(2), 355-362.
- Hsu, Y. H., Cheng, C. H., Ko, P. H., Tang, C. P., Huang, C. W., & Tseng, C. W. (2025). Assessment of the impact of power business intelligence on adenoma detection rate: a prospective observational trial. *BMC gastroenterology*, 25(1), 275.
- Islam, M. M., Desai, K., Rabbani, M. M., Ahmad, S., & Snigdha, E. Z. (2025). AI-Powered Business Intelligence in IT: Transforming Data into Strategic Solutions for Enhanced Decision-Making. *The American Journal of Engineering and Technology*, 7(02), 59-73.
- Jain, A. (2024). *AI-Powered Business Intelligence Dashboards: A Cross-Sector Analysis of Transformative Impact and Future Directions*.
- Jayaraman, K. D., & Jain, S. (2024). Leveraging Power BI for advanced business intelligence and reporting. *International Journal for Research in Management and Pharmacy*, 13(11), 21-36.

- Jiang, J., Xie, H., Shen, S., Shen, Y., Zhang, Z., Lei, M., ... & Chen, P. (2025). Siriusbi: A comprehensive IIm-powered solution for data analytics in business intelligence. *Proceedings of the VLDB Endowment*, 18(12), 4860-4873.
- Kajava, E. (2018). *Improving company performance through implementation of Business Intelligence tools: Implementation of a Microsoft Power BI in a Case Study Company*.
- Khaddam, A. A., & Alzghoul, A. (2025). Artificial intelligence-driven business intelligence for strategic energy and ESG management: A systematic review of economic and policy implications. *International Journal of Energy Economics and Policy*, 15(4), 635.
- Maghsoudi, M., & Nezafati, N. (2023). Navigating the acceptance of implementing business intelligence in organizations: A system dynamics approach. *Telematics and Informatics Reports*, 11, 100070.
- Miyamoto, M. (2015). Application of competitive forces in the business intelligence of Japanese SMEs. *International Journal of Management Science and Engineering Management*, 10(4), 273-287.
- Munjala, M. B. (2025). *Harnessing the Power of Data Analytics and Business Intelligence to Drive Innovation in Biotechnology and Healthcare: Transforming Patient Outcomes through Predictive Analytics, Genomic Research, and Personalized Medicine*. *Cuestiones de Fisioterapia*, 54(3), 2222-2235.
- Nadipalli, R. (2017). *Effective business intelligence with QuickSight*. Packt Publishing Ltd.
- Natarajan, A. K., Galety, M. G., Iwendi, C., Das, D., & Shankar, A. (Eds.). (2024). *AI-Powered Business Intelligence for Modern Organizations*. IGI Global.
- Negash, S., & Gray, P. (2008). Business intelligence. In *Handbook on Decision Support Systems 2: Variations* (pp. 175-193). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Osho, G. O., Omisola, J. O., & Shiyambola, J. O. (2020). An Integrated AI-Power BI Model for Real-Time Supply Chain Visibility and Forecasting: A Data-Intelligence Approach to Operational Excellence. *Unknown Journal*.
- Panda, S. P., & Padhy, A. (2025). *Business Intelligence with Power BI and Tableau: Cloud-Based Data Warehousing, Predictive Analytics, and Artificial Intelligence-Driven Decision Support*. Deep Science Publishing.
- Paramesha, M., Rane, N., & Rane, J. (2024). Big data analytics, artificial intelligence, machine learning, internet of things, and blockchain for enhanced business intelligence. *Artificial Intelligence, Machine Learning, Internet of Things, and Blockchain for Enhanced Business Intelligence* (June 6, 2024).
- Pirna, C., Marinescu, N. I., Ghiculescu, L. D., & Ciocardia, R. C. (2017). BUSINESS INTELLIGENCE DEVELOPMENT WITH POWER BI APPLIED IN NONCONVENTIONAL TECHNOLOGIES. *Nonconventional Technologies Review/Revista de Tehnologii Neconventionale*, 21(4).

- Power, D. J. (2013). *Mobile decision support and business intelligence: an overview*. *Journal of Decision Systems*, 22(1), 4-9.
- Pradesyah, R. (2021). *Power business intelligence in the data science visualization process to forecast CPO prices*. *International Journal of Science, Technology & Management*, 2(6), 2198-2208.
- Rahman, N., Aldhaban, F., & Akhter, S. (2013). *Emerging technologies in business intelligence*. *2013 Proceedings of PICMET'13: Technology Management in the IT-Driven Services (PICMET)*, 542-547.
- Raj, R., Wong, S. H., & Beaumont, A. J. (2016, November). *Business Intelligence Solution for an SME: A Case Study*. In *8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, IC3K 2016* (pp. 41-50). SciTePress.
- Rane, N., Paramesha, M., Choudhary, S., & Rane, J. (2024). *Business intelligence and business analytics with artificial intelligence and machine learning: Trends, techniques, and opportunities*. *Techniques, and Opportunities* (May 17, 2024).
- Raza, M. H., Rind, Y. M., Javed, I., Zubair, M., Mehmood, M. Q., & Massoud, Y. (2023). *Smart meters for smart energy: a review of business intelligence applications*. *IEEE Access*, 11, 120001-120022.
- Reddy, C. S., Sangam, R. S., & Srinivasa Rao, B. (2018, November). *A survey on business intelligence tools for marketing, financial, and transportation services*. In *Smart Intelligent Computing and Applications: Proceedings of the Second International Conference on SCI 2018, Volume 2* (pp. 495-504). Singapore: Springer Singapore.
- Richardson, J., Sallam, R., Schlegel, K., Kronz, A., & Sun, J. (2020). *Magic quadrant for analytics and business intelligence platforms*. *Gartner ID G, 386610, 00041-5*.
- Rimon, S. T., Sufian, M. A., Guria, Z. M., Morshed, N., Mosaddeque, A. I., & Ahamed, A. (2024, December). *Impact of AI-powered business intelligence on smart city policy-making and data-driven governance*. In *IET Conference Proceedings CP908* (Vol. 2024, No. 30, pp. 475-481). Stevenage, UK: The Institution of Engineering and Technology.
- Selvarajan, G. P. (2023). *Augmenting Business Intelligence with AI: A Comprehensive Approach to Data-Driven Strategy and Predictive Analytics*. *International Journal of All Research Education and Scientific Methods*, 11(10), 2121-2132.
- Sharda, R., Delen, D., Turban, E., Aronson, J., & Liang, T. (2014). *Business intelligence and analytics*. *System for Decesion Support*, 398, 2014.
- Sharma, K., Shetty, A., Jain, A., & Dhanare, R. K. (2021, January). *A comparative analysis on various business intelligence (BI), data science and data analytics tools*. In *2021 International Conference on Computer Communication and Informatics (ICCCI)* (pp. 1-11). IEEE.

- Srivastava, G., S, M., Venkataraman, R., V, K., & N, P. (2022). A review of the state of the art in business intelligence software. *Enterprise Information Systems*, 16(1), 1-28.
- Talakola, S. (2022). Analytics and reporting with Google Cloud platform and Microsoft Power BI. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, 3(2), 43-52.
- Thompson, W. J., & Van der Walt, J. S. (2010). Business intelligence in the cloud. *South African Journal of Information Management*, 12(1), 1-15.
- Torres, M. (2022). Data Engineering Challenges in AI-Powered Business Intelligence Platforms. *International Journal of Emerging Trends in Computer Science and Information Technology*, 3(3), 1-9.
- Usman, M., Moinuddin, M., & Khan, R. (2024). Unlocking insights: harnessing the power of business intelligence for strategic growth. *International Journal of Advanced Engineering Technologies and Innovations*, 4(1), 97-117.
- Weng, L., Tang, Y., Feng, Y., Chang, Z., Chen, R., Feng, H., ... & Chen, W. (2025, May). Datalab: A unified platform for llm-powered business intelligence. In *2025 IEEE 41st International Conference on Data Engineering (ICDE)* (pp. 4346-4359). IEEE.
- Zakizadeh, M., & Zand, M. (2024, February). Transforming the Energy Sector: Unleashing the Potential of AI-Driven Energy Intelligence, Energy Business Intelligence, and Energy Management System for Enhanced Efficiency and Sustainability. In *2024 20th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP)* (pp. 1-7). IEEE.
- Zamil, M. H., Mohiuddin, M., & Mamun, M. N. H. (2024). Business intelligence systems in finance and accounting: A review of real-time dashboarding using Power BI & Tableau. *American Journal of Scholarly Research and Innovation*, 3(02), 52-79.
- Zohuri, B., & Moghaddam, M. (2020). From business intelligence to artificial intelligence. *Journal of Material Sciences & Manufacturing Research*, 1(1), 1-10.
- Zwingmann, T. (2022). *Ai-powered business intelligence*. " O'Reilly Media, Inc."

KECERDASAN BISNIS (BI) DIDUKUNG AI :

Meningkatkan prediksi dan pembuatan keputusan
dengan ML (Machine Learning)

Dr. Ir. Agus Wibowo, M.Kom, M.Si, MM

BIO DATA PENULIS



Penulis memiliki berbagai disiplin ilmu yang diperoleh dari Universitas Diponegoro (UNDIP) Semarang. dan dari Universitas Kristen Satya Wacana (UKSW) Salatiga. Disiplin ilmu itu antara lain teknik elektro, komputer, manajemen dan ilmu sosiologi. Penulis memiliki pengalaman kerja pada industri elektronik dan sertifikasi keahlian dalam bidang Jaringan Internet, Telekomunikasi, Artificial Intelligence, Internet Of Things (IoT), Augmented Reality (AR), Technopreneurship, Internet Marketing dan bidang pengolahan dan analisa data (komputer statistik).

Penulis adalah pendiri dari Universitas Sains dan Teknologi Komputer (Universitas STEKOM) dan juga seorang dosen yang memiliki Jabatan Fungsional Akademik Lektor Kepala (Associate Professor) yang telah menghasilkan puluhan Buku Ajar ber ISBN, HAKI dari beberapa karya cipta dan Hak Paten pada produk IPTEK. Sejak tahun 2023 penulis tercatat sebagai Dosen luar biasa di Fakultas Ekonomi & Bisnis (FEB) Universitas Diponegoro Semarang. Penulis juga terlibat dalam berbagai organisasi profesi dan industri yang terkait dengan dunia usaha dan industri, khususnya dalam pengembangan sumber daya manusia yang unggul untuk memenuhi kebutuhan dunia kerja secara nyata.



YAYASAN PRIMA AGUS TEKNIK

PENERBIT :

YAYASAN PRIMA AGUS TEKNIK
Jl. Majapahit No. 605 Semarang
Telp. (024) 6723456. Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

ISBN 978-634-7227-65-2 (PDF)



9

786347

227652