



Hubungan Antara Kecerdasan Buatan (AI) dan Kecerdasan Bisnis (BI)

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



YAYASAN PRIMA AGUS TEKNIK





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

Hubungan Antara Kecerdasan Buatan (AI) dan Kecerdasan Bisnis (BI)



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 978-634-7227-33-1 (PDF)



9

786347

227331

Hubungan antara Kecerdasan Buatan (AI) dan Kecerdasan Bisnis (BI)

Penulis :

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

ISBN : 978-634-7227-33-1

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 Yuniato, 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

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa, karena berkat rahmat dan karunia-Nya buku ini dapat diselesaikan dengan baik. Buku berjudul *Hubungan Antara Kecerdasan Buatan dan Kecerdasan Bisnis* hadir untuk memberikan pemahaman yang komprehensif mengenai keterkaitan dan integrasi antara dua disiplin ilmu yang sangat penting dalam era digital saat ini: kecerdasan buatan (Artificial Intelligence/AI) dan kecerdasan bisnis (Business Intelligence/BI).

Buku ini tersusun dalam lima bab yang dirancang secara sistematis mulai dari pengenalan konsep BI secara tradisional hingga perkembangan terkini berupa integrasi AI ke dalam BI untuk pengambilan keputusan bisnis yang lebih tepat dan efisien. Bab awal mengupas tentang evolusi BI dari pendekatan konvensional hingga era modern, diikuti dengan pembahasan mendalam mengenai kebangkitan AI dan bagaimana teknologi ini mengubah lanskap BI di perusahaan.

Selanjutnya, pembaca diajak untuk memahami penerapan praktis Enterprise BI dengan perspektif bisnis melalui berbagai studi kasus, seperti klasifikasi mutu penjualan komoditas, prediksi harga, sistem rekomendasi, dan pengenalan citra otomatis yang semuanya mengadopsi teknik AI. Bab akhir buku membahas tantangan serta peluang masa depan ketika AI dan BI bersinergi, termasuk aspek komputasi kognitif, keamanan, tata kelola, dan faktor kepercayaan dalam implementasi AI pada BI.

Kami berharap buku ini dapat menjadi sumber referensi yang bermanfaat bagi akademisi, profesional bisnis, pengambil keputusan, serta praktisi teknologi yang tertarik memahami dan memanfaatkan sinergi AI dan BI demi kemajuan organisasi dan dunia bisnis secara umum.

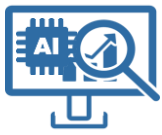
Akhir kata, saran dan kritik sangat kami harapkan untuk penyempurnaan karya ini di masa mendatang. Semoga buku ini memberikan inspirasi dan kontribusi nyata bagi perkembangan ilmu pengetahuan dan praktik bisnis berbasis teknologi.

Selamat Membaca...

Semarang, Agustus 2025

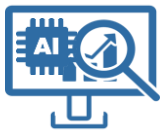
Penulis

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



DAFTAR ISI

HALAMAN JUDUL	i
KATA PENGANTAR	ii
DAFTAR ISI.....	iii
BAB 1 PENDAHULUAN	1
1.1 Pendahuluan.....	1
1.2 Cara Tradisional Mengaktifkan BI	2
1.3 Tiga Generasi BI	3
1.4 Bisnis Terintegrasi Dengan BI Untuk Pengambilan Keputusan	6
1.5 Ringkasan.....	7
BAB 2 AI DAN ANALISIS BERBASIS AI	8
2.1 Pendahuluan.....	8
2.2 Ai Dan Kebangkitannya Di Perusahaan Modern.....	8
2.3 Bagaimana AI Mengubah Lanskap Bi Perusahaan.....	9
2.4 Lingkup Analitik	9
2.5 Analisis Berbasis BI Vs. Analisis Berbasis AI	10
2.6 Dari Data Ke Keputusan Cerdas Dan Berwawasan	11
2.7 Teknik Berbasis AI Untuk BI	14
2.8 Ringkasan.....	16
BAB 3 PENGGUNAAN INDUSTRI ENTERPRISE BI PERSPEKTIF BISNIS	17
3.1 Pendahuluan.....	17
3.2 Klasifikasi Mutu Penjualan Komoditas Berdasarkan Atribut	17
3.3 Memprediksi Harga Komoditas Di Muka	26
3.4 Sistem Rekomendasi Pilihan Optimal Berdasarkan Skor	31
3.5 Pengenalan Citra Otomatis.....	31
3.6 Ringkasan.....	33
BAB 4 KASUS PENGGUNAAN ENTERPRISE BI DENGAN AI	34
4.1 Pendahuluan.....	34
4.2 Klasifikasi Kualitas Komoditas Layak Jual Berdasarkan Atribut	34
4.3 Memprediksi Harga Komoditas Di Muka	83
4.4 Sistem Rekomendasi Pilihan Optimal Berdasarkan Skor	93
4.5 Pengenalan Gambar Otomatis	166
4.6 Ringkasan.....	177
BAB 5 APA SELANJUTNYA DALAM AI BERTEMU BI?.....	178
5.1 Pendahuluan.....	178
5.2 Komputasi Kognitif Bertenaga AI.....	178
5.3 Keamanan Dan Tata Kelola Dalam BI Berbasis AI	180
5.4 Faktor Kepercayaan Dalam BI Berbasis AI	181
5.5 Ringkasan.....	183
DAFTAR PUSTAKA	184

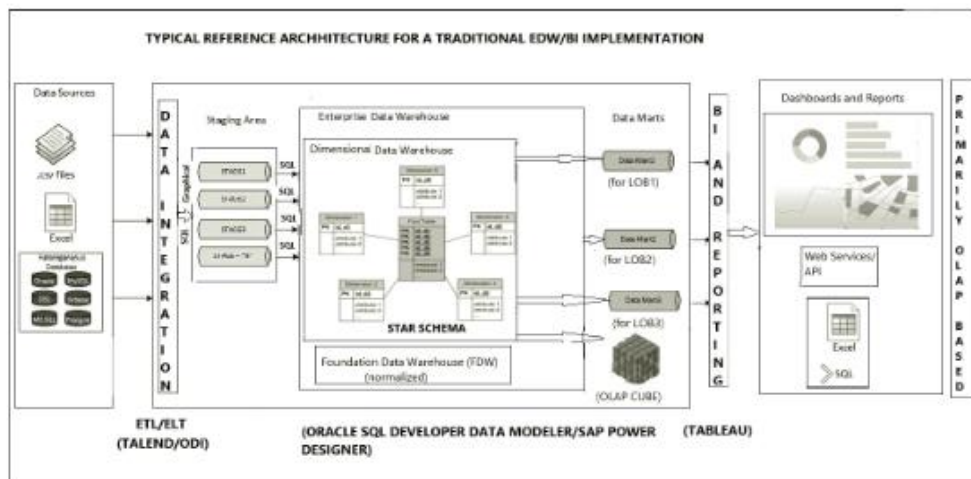


BAB 1 PENDAHULUAN

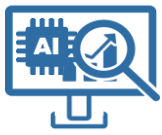
1.1 PENDAHULUAN

Bab ini dimulai dengan uraian metode konvensional untuk mengimplementasikan BI di perusahaan dan menyoroti bagaimana BI telah berevolusi menjadi analitik swalayan dan, yang terbaru, BI yang didukung AI. Bab ini membahas tiga generasi BI dan bagaimana solusi DW/BI telah mengubah lanskap pengambilan keputusan bisnis menjadi lebih baik. Terakhir, bab ini diakhiri dengan catatan tentang bagaimana istilah "bisnis" yang umum digunakan selaras dan dapat diintegrasikan ke dalam solusi BI komprehensif yang tidak hanya mutakhir tetapi juga yang terus berkembang secara otomatis berdasarkan metrik atau indikator kinerja utama (KPI) yang dihasilkan oleh solusi BI, yang pada gilirannya, dapat diintegrasikan ke dalam arsitektur yang sama dengan input, yang menghasilkan keputusan yang lebih berwawasan.

Aspek utama "memasang AI dengan BI" secara langsung serta detailnya, termasuk persimpangan antara AI, ilmu data, pembelajaran mesin, pembelajaran mendalam, komputasi kognitif, dan bagaimana hal ini dapat mengarah pada otomatisasi BI dan proses pengambilan keputusan yang menghasilkan pengambilan keputusan bisnis yang lebih terinformasi. Ada banyak topik dengan fokus baru pada kemampuan kecerdasan buatan saat ini. Di permukaan, harus diakui, mungkin tidak tampak semenarik kecerdasan buatan, yang memicu minat baru, misalnya, di bidang mobil tanpa pengemudi atau robotika. Namun, mengingat bagaimana dunia korporat memengaruhi kehidupan kita sehari-hari, kecerdasan buatan yang meningkatkan kecerdasan bisnis kini tampaknya memiliki peran yang sangat penting dalam memengaruhi cara kita bekerja sehari-hari. Karena alasan inilah buku ini memberikan cakupan yang dibutuhkan untuk mendefinisikan jalur teknologi yang dipenuhi dengan metode dan alat baru yang akan membawa kita ke masa depan kecerdasan bisnis masa depan yang belum pernah tersedia secara efektif sebelumnya dan hanya dibayangkan untuk kecerdasan buatan.



Gambar 1.1 Arsitektur tipikal implementasi EDW/BI tradisional (generasi pertama).



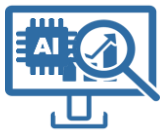
1.2 CARA TRADISIONAL MENGAKTIFKAN BI

Setiap ekosistem yang mendukung BI melibatkan integrasi semua perangkat dan teknologi mulai dari persiapan data (termasuk kualitas data) hingga integrasi data (melalui ETL/ELT atau virtualisasi data), pergudangan data (termasuk danau data yang bertindak sebagai sumber ke gudang data), penyajian data (melalui data mart, gudang data, dan/atau dasbor BI), hingga visualisasi data, dan dari dukungan keputusan hingga pengambilan keputusan yang disajikan oleh berbagai perangkat dan teknologi kecerdasan buatan saat ini untuk meningkatkan dan menyempurnakan kecerdasan bisnis. Selain itu, tata kelola data perlu diterapkan pada setiap fase ini untuk memenuhi persyaratan peraturan, dll.

Virtualisasi data dan danau data memasuki skenario EDW/BI sebagai bagian dari apa yang disebut teknologi Integrasi Data 2.0. Gambar 1.1 menunjukkan arsitektur umum solusi EDW/BI yang komprehensif. Seperti yang dapat dilihat dari gambar ini, terdapat tiga komponen utama solusi EDW/BI:

1. Integrasi Data—Ini terdiri dari pengumpulan data dari berbagai sumber dan dalam berbagai format, termasuk sebagian besar data terstruktur atau relasional yang dapat dipilah ke dalam format umum yang kemudian dapat diintegrasikan ke dalam komponen pergudangan data berikutnya.
2. Pergudangan Data—Ini terdiri dari pementasan data dari komponen integrasi data ke dalam zona pendaratan yang bertanggung jawab untuk memilah data. Berdasarkan sumber data dan jenis data, model dimensi logis dibangun yang secara jelas menjabarkan dimensi dan fakta berdasarkan proses bisnis. Ini menghasilkan skema STAR di mana suatu fakta dikelilingi oleh berbagai dimensi dalam format bintang. Model logis ini membentuk inti EDW yang menyimpan data dalam model semantik untuk pelaporan antar-hari, sebagaimana adanya, dan sebagaimana adanya. Beberapa Data Mart dibuat dari EDW untuk memenuhi kebutuhan pelaporan di tingkat departemen perusahaan. Setiap data mart dimodelkan sebagai skema STAR.
3. Komponen Kecerdasan Bisnis (BI) dan Pelaporan—Komponen ini terdiri dari dasbor dan pelaporan data (berdasarkan laporan siap pakai) menggunakan alat BI yang terintegrasi dengan EDW untuk memungkinkan analisis bisnis dan pengguna berinteraksi dengan data. Para pengguna ini diberikan akses ke data untuk menggunakan KPI yang telah ditentukan sebelumnya.

Meskipun arsitektur di atas menunjukkan EDW/BI di tingkat perusahaan, arsitektur ini memiliki keterbatasan, seperti tidak adanya pelaporan intra-hari (BI Operasional), tidak adanya kemampuan nyata untuk memungkinkan pengguna membuat KPI mereka sendiri melalui dasbor interaktif (analitik swalayan), dan pelaporan *ad hoc*. Bagian selanjutnya menguraikan bagaimana lanskap EDW/BI berevolusi hingga menghasilkan solusi EDW/BI generasi kedua dan ketiga.



1.3 TIGA GENERASI BI

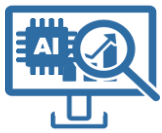
BI generasi pertama didasarkan pada solusi berbasis OLAP, yang sebagian besar mendukung format data relasional dan menggunakan ETL/ELT untuk integrasi data. EDW dirancang berdasarkan skema STAR, yang juga bersifat relasional dan dapat diperluas ke *Multidimensional Online Analytical Processing* (MOLAP). Gambar 1.1 menggambarkan arsitektur BI generasi pertama yang umum. Pelaporan dan BI generasi pertama terbatas pada pelaporan malam hari berdasarkan beban batch dan dukungan untuk integrasi data real-time, tetapi tidak ada pelaporan *ad hoc*.

Selain itu, BI operasional atau, dengan kata lain, analisis data saat data tersebut dibuat tidak memungkinkan, dan kapabilitas BI tidak diperluas untuk memungkinkan pengguna membuat KPI mereka sendiri atau analitik swalayan. Kebutuhan ini dipenuhi oleh arsitektur BI generasi kedua, yang memperkenalkan analitik sebagai bagian dari BI dan, sebagai tambahan, memanfaatkan Integrasi Data 2.0, yang hadir dengan teknologi integrasi data yang tangguh seperti virtualisasi data dan alur integrasi data serta penyempurnaan EDW seperti danau data (yang menyimpan data mentah) dan dapat digunakan sebagai sumber EDW.

Tiga kemampuan utama BI generasi kedua meliputi kueri *ad hoc*, BI operasional, dan analitik swalayan. Arsitektur ini juga dapat memanfaatkan virtualisasi data, yang memungkinkan integrasi data dalam skala besar. BI generasi kedua juga menangani BI dalam skala besar dengan memungkinkan penggunaan danau data dan kelincahan data.

Dengan kemampuan melakukan analitik pada data, perusahaan dapat menemukan alasan terjadinya sesuatu di masa lalu menggunakan data tersebut atau terkadang memprediksi masa depan menggunakan informasi ini (bagian analitik bisnis) selain mengetahui apa yang terjadi dan bagaimana kejadiannya di masa lalu berdasarkan data (bagian intelijen bisnis). Ini berarti kausalitas dan korelasi merupakan bagian dari analisis dan keduanya diperlukan untuk melakukan analisis. Selain itu, keduanya tidak sama. Generasi kedua BI mendukung intelijen bisnis dan analitik bisnis, yang menghasilkan pandangan bisnis yang holistik dan kemampuan untuk membuat keputusan yang lebih mendalam. Hal ini juga dapat membuka jalan untuk memprediksi masa depan menggunakan analitik dan intelijen ini, seperti bagaimana keputusan tertentu dapat berkembang jika dan ketika data berkembang. Selain itu, komponen kunci dari generasi kedua BI adalah menempatkan analitik bisnis di tangan analis bisnis atau pengguna akhir, sehingga mengarah pada analitik swalayan. Dengan demikian, perangkat BI disempurnakan untuk memasukkan tren-tren baru ini sebagai cara baru untuk berintegrasi ke dalam ekosistem BI yang sudah ada. Dari perspektif analisis bisnis, BI terjadi sebelum Analisis Bisnis. Meskipun BI dapat mengidentifikasi kriteria yang menyebabkan persaingan bisnis, BI memungkinkan Anda memahami mengapa kriteria tersebut menyebabkan persaingan bisnis di perusahaan Anda. Baik BI maupun BA berada di bawah payung analisis bisnis yang lebih luas.

Dengan menggunakan analitik, perusahaan dapat merancang metrik yang membantu pengambilan keputusan, yang pada gilirannya, memungkinkan persaingan juga disebut intelijen kompetitif. Sebagai perbandingan, BI menggunakan metrik yang mengukur

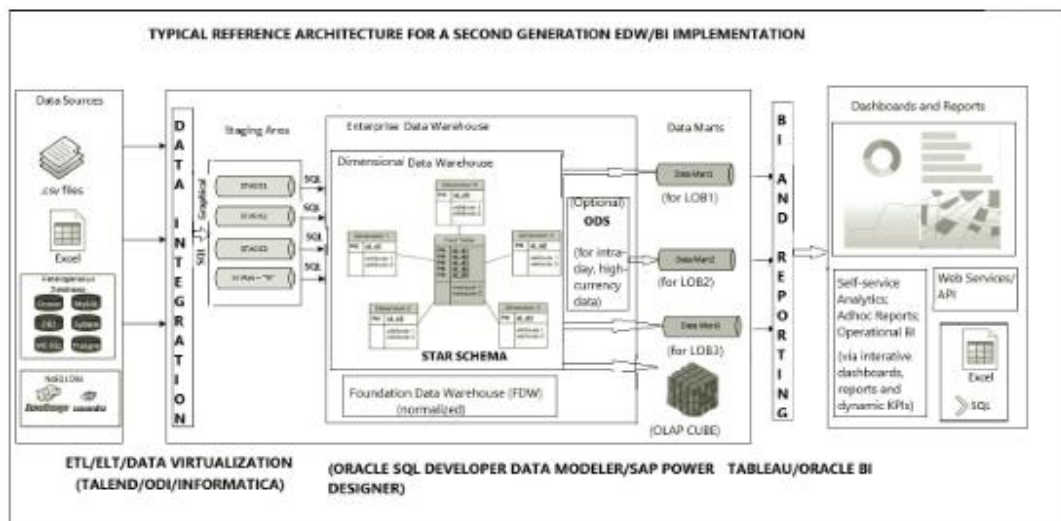


efektivitas tujuan bisnis di perusahaan Anda. Metrik ibarat ukuran atom yang melacak proses bisnis.

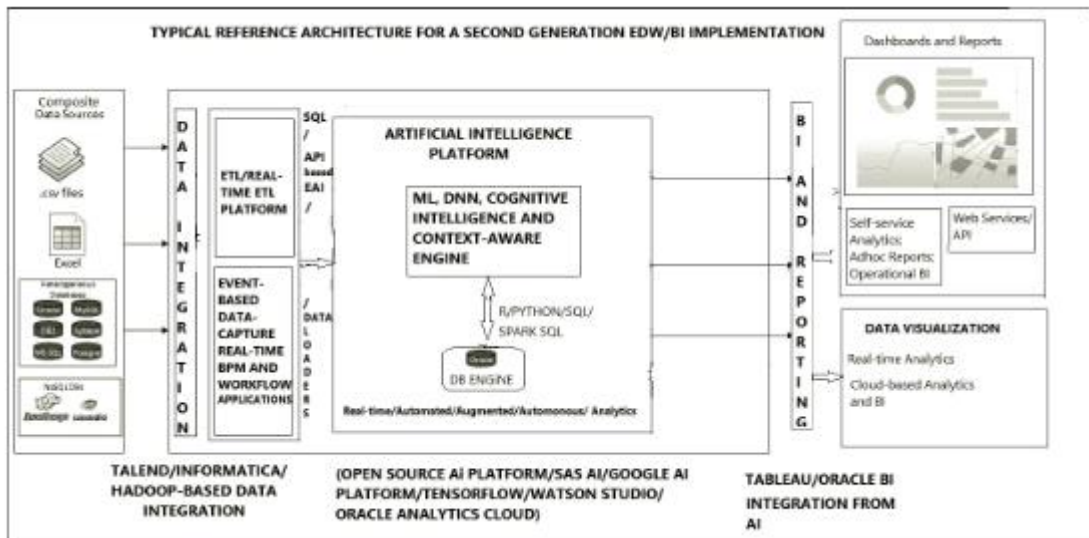
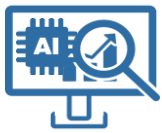
Gambar 1.2 menggambarkan arsitektur BI generasi kedua yang umum.

BI generasi ketiga didasarkan pada perpaduan AI dan BI (kecerdasan buatan dan kecerdasan bisnis), dan memungkinkan berbagai analitik bertenaga AI untuk pengambilan keputusan yang didukung BI. BI generasi ketiga unik karena memanfaatkan persimpangan AI, ilmu data, pembelajaran mesin, pembelajaran mendalam, dan komputasi kognitif, serta bagaimana teknologi ini dapat membantu dan/atau mengotomatiskan BI dan pengambilan keputusan. Aspek-aspek utama BI yang didukung AI adalah sebagai berikut:

- EDW dan BI real-time.
- BI otomatis, ditambah, dan otonom, termasuk analitik ditambah yang menggunakan kombinasi kecerdasan manusia dan mesin serta otomatisasi konten dan konteks yang terlibat untuk relevansi dan pentingnya wawasan yang didukung BI serta akurasi dan transparansinya di seluruh tingkat perusahaan. Sederhananya, ini memungkinkan analisis dan analitik data otomatis dan luas.
- Penemuan Big Data menggunakan pencarian bertenaga AI dan integrasinya ke dalam lanskap BI yang efisien dan memfasilitasi pengambilan keputusan yang terinformasi di tingkat dan skala perusahaan melalui analitik data besar yang tertransformasi.
- EDW/BI bertenaga AI telah memudahkan kemampuan untuk memanfaatkan data dalam berbagai format, seperti data terstruktur, semi-terstruktur, dan tidak terstruktur, serta data real-time. Ini juga telah menangani laju pembuatan data tersebut, serta volumenya yang besar, sekaligus memungkinkan validasi data dan menjaga kualitas data tersebut.
- Analisis tangkas.
- Skalabilitas tingkat perusahaan dan bagaimana pengalaman pelanggan dapat ditingkatkan.



Gambar 1.2 Arsitektur EDW/BI generasi kedua yang umum.

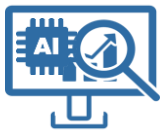


Gambar 1.3 Perpaduan antara AI generasi ketiga dan arsitektur BI.

- Solusi EDW/BI holistik yang dapat digunakan oleh pengguna teknis maupun non-teknis di semua tingkatan perusahaan.
- Otomatisasi cerdas proses bisnis dan operasional.
- Penyediaan BI sebagai layanan untuk mobilitas dan analitik pelanggan saat bepergian.
- Monetisasi data di tingkat perusahaan.
- Akselerasi dan peningkatan BI yang kompetitif. Sebagai contoh, BI dapat memiliki solusi jaringan saraf tiruan dalam yang tertanam dalam tumpukan implementasinya untuk mewujudkan hasil yang kompetitif. Kasus penggunaan implementasi semacam itu mencakup solusi penandaan gambar dan pengenalan gambar.

Penggunaan AI untuk BI memanfaatkan cara menggunakan model AI dalam konteks bisnis dengan merekomendasikan model AI/algoritma pembelajaran mesin tertentu berdasarkan masukan dan keluaran kasus penggunaan. Gambar 1.3 merinci aspek-aspek yang belum dieksplorasi secara mendalam dalam lanskap BI, seperti relevansi dan pentingnya wawasan berbasis BI serta akurasi dan transparansinya di seluruh tingkat industri perusahaan/korporat.

Gambar 1.3 menggambarkan arsitektur BI generasi ketiga yang umum didukung oleh AI. BI generasi ketiga memberikan keuntungan khusus bagi analis bisnis, analis data, ilmuwan data, dan insinyur data, serta pengguna akhir. Sebagaimana dinyatakan di awal bagian ini, BI telah berevolusi dari lanskap awalnya yang berupa pelaporan dan analisis berbasis OLAP (BI generasi pertama) menjadi kueri *ad hoc*, BI operasional, dan analitik swalayan (BI generasi kedua), hingga analitik bisnis bertenaga AI dan penemuan big data (BI generasi ketiga). Dan banyak vendor BI terkemuka di industri ini, seperti Tableau®, Microsoft®, Oracle®, IBM®, dll., telah mulai menawarkan kapabilitas bertenaga AI untuk memungkinkan semua jenis pengguna



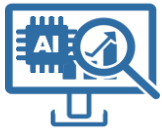
(teknis dan non-teknis) memanfaatkan teknologi dan tren baru ini dan memperoleh wawasan cerdas yang dapat menghasilkan BI yang lebih baik, lebih cerdas, dan lebih cepat.

1.4 BISNIS TERINTEGRASI DENGAN BI UNTUK PENGAMBILAN KEPUTUSAN

Setiap perusahaan memiliki lanskap bisnis yang memiliki tujuan atau sasaran tertentu untuk dicapai agar tetap kompetitif di industri. Untuk tujuan ini, lanskap bisnis tersebut terdiri dari sejumlah proses bisnis yang menentukan bagaimana perusahaan tersebut diorganisasikan untuk mencapai tujuannya. Misalnya, perusahaan manufaktur produk memiliki tujuan untuk memaksimalkan penjualan, laba, dan rantai pasokannya, yang membantunya tetap unggul dalam persaingan. Dan masing-masing tujuan ini terdiri dari satu atau lebih proses bisnis yang, pada gilirannya, diukur dengan metrik yang melacak statusnya. Selain itu, terdapat KPI yang melacak dan mengukur seberapa baik perusahaan memenuhi tujuan bisnis tertentu. KPI ini mengukur efektivitas tujuan bisnis dan merupakan komponen yang disebut sebagai kecerdasan bisnis (*business intelligence*) perusahaan. BI membutuhkan data terkini maupun historis untuk menentukan pengukuran KPI berdasarkan analisis yang dilakukan, yang pada gilirannya menentukan aspek "apa yang terjadi" dan "bagaimana terjadinya".

Kedua aspek ini juga menentukan apakah dan seberapa baik perusahaan telah mencapai tujuannya. Untuk tujuan ini, visualisasi data berdasarkan data ini memungkinkan pengguna bisnis dan/atau pengguna akhir untuk melakukan analisis "apa" dan "bagaimana" yang pada akhirnya membantu menentukan seberapa baik tujuan bisnis tertentu tercapai dan memberikan gambaran tentang apa yang dapat dilakukan untuk lebih meningkatkan KPI yang mengarah pada pertumbuhan perusahaan. Sebagai contoh, bagan visual yang menggambarkan analisis tren dan dasbor yang mengelompokkan bagan berdasarkan kombinasi data dan kemampuan drill-up/drill-down/drill-across dapat membantu pengguna bisnis untuk memahami korelasi antar efek data. Namun, BI sendiri tidak dapat menentukan alasan di balik analisis data yang sangat penting untuk benar-benar membuat keputusan berdasarkan alasan tersebut. Di sinilah analitik bisnis berperan. Sebagai aspek kedua dari analisis bisnis, analitik bisnis membantu proses pengambilan keputusan dengan membantu bagian preskriptif (tindakan bisnis yang akan diambil) dari suatu solusi, serta, terkadang, bagian prediktif dari solusi tersebut. Baik keputusan bisnis yang dibuat maupun tindakan bisnis yang diambil membantu menentukan apakah keputusan yang dibuat dan tindakan selanjutnya yang diambil dapat membantu.

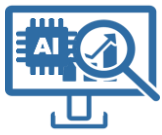
Lebih lanjut, memungkinkan pengguna bisnis untuk membuat analitik mereka sendiri berdasarkan data yang disajikan sangat membantu dalam pengambilan keputusan yang mendalam yang mendorong perusahaan menuju kesuksesan kompetitif. Kembali ke contoh perusahaan manufaktur produk, analitik bisnis dapat membantu menentukan keputusan apa yang dapat diambil ketika produk tertentu berevolusi dengan cara memperkirakan pertumbuhan penjualan dalam skenario seperti itu. Dalam skenario analitik prediktif inilah AI dapat membuat perbedaan melalui dukungan keputusan yang cerdas.



Bisnis dan pengguna bisnis memainkan peran penting dengan menjelaskan tujuan dan proses bisnis perusahaan, KPI, dan metrik yang diperlukan untuk BI serta keputusan dan tindakan selanjutnya yang akan diambil. Bisnis juga dapat menentukan jenis visualisasi data yang diperlukan untuk analisis serta visualisasi yang meningkatkan proses pengambilan keputusan.

1.5 RINGKASAN

Bab ini membahas metode tradisional pergudangan data dan BI, kemudian menguraikan tiga generasi BI melalui arsitektur tipikal yang didasarkan pada masing-masing generasi. Terakhir, bab ini membahas bagaimana "bisnis" cocok dengan lanskap analitik lahan BI. Bab selanjutnya menjelaskan AI dan analitik bertenaga AI serta perannya dalam pemberdayaan BI.



BAB 2

AI DAN ANALISIS BERBASIS AI

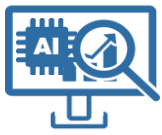
2.1 PENDAHULUAN

Beberapa tahun terakhir telah terjadi peningkatan yang luar biasa dalam AI dan adopsi AI di perusahaan. Dari penemuan data cerdas hingga standarisasi data hingga model berbasis AI yang diterapkan pada data, AI telah memberdayakan lanskap pengguna, mulai dari pengguna teknis dan ilmuwan data hingga analis bisnis, ilmuwan warga, pengguna berpengalaman, dan pengguna akhir, yang memungkinkan mereka untuk membuat analisis dari konteks bisnis secara real-time dan berdasarkan data saat data tersebut dibuat. Bab ini membahas detail tersebut dan bagaimana AI dan analitik berbasis AI telah menghasilkan solusi BI yang memberikan wawasan cerdas dari segala jenis data, sekaligus merupakan solusi generasi berikutnya, skalabel, relevan secara kontekstual, serta dapat disesuaikan dan dipersonalisasi dari on-premise hingga on-the-go.

2.2 AI DAN KEBANGKITANNYA DI PERUSAHAAN MODERN

Kebangkitan AI telah menguntungkan perusahaan dalam hal-hal berikut:

- ❖ AI telah memungkinkan ilmu data dan penerapannya dalam memecahkan masalah bisnis perusahaan.
- ❖ AI telah menyediakan area di mana AI dan komputasi kognitif dapat bekerja sama, menghasilkan platform kecerdasan otonom, augmented, dan adaptif yang optimal.
- ❖ Melalui AI, komputasi kognitif dapat terdiri dari teknologi-teknologi berikut: pemrosesan bahasa alami (NLP), pembelajaran mesin dengan jaringan saraf tiruan dan jaringan saraf tiruan dalam, algoritma yang belajar dan beradaptasi dengan AI, pembelajaran mendalam, pengenalan gambar, penalaran dan otomatisasi keputusan, serta kecerdasan emosional.
- ❖ AI telah menjadi konvergensi dari tiga teknologi canggih, yaitu pembelajaran mesin, pemrosesan bahasa alami, dan komputasi kognitif. Ke sinilah masa depan AI sedang menuju.
- ❖ Lebih lanjut, pragmatik dapat didefinisikan melalui kognisi dan merekomendasikan pilihan yang "optimal", sehingga meningkatkan persaingan. Hal ini mengintegrasikan "kognisi" ke dalam model dan melatihnya menggunakan model rekomendasi berbasis komputasi kognitif untuk pengambilan keputusan dalam hal menghasilkan pilihan yang "paling optimal" untuk digunakan, serta "memahami" mengapa pilihan tertentu optimal untuk digunakan (yaitu, tindakan terbaik berikutnya) mulai dari dukungan keputusan hingga pengambilan keputusan yang berwawasan.
- ❖ Tingkatan analitik data yang umum meliputi analitik deskriptif, analitik diagnostik, analitik prediktif, analitik preskriptif, dan komputasi kognitif. Jenis analitik data kelima komputasi kognitif menjawab pertanyaan seperti, Apa tindakan terbaik berikutnya



yang dapat diambil? Ini melibatkan konteks komputasi sehubungan dengan kasus penggunaan bisnis.

2.3 BAGAIMANA AI MENGUBAH LANSKAP BI PERUSAHAAN

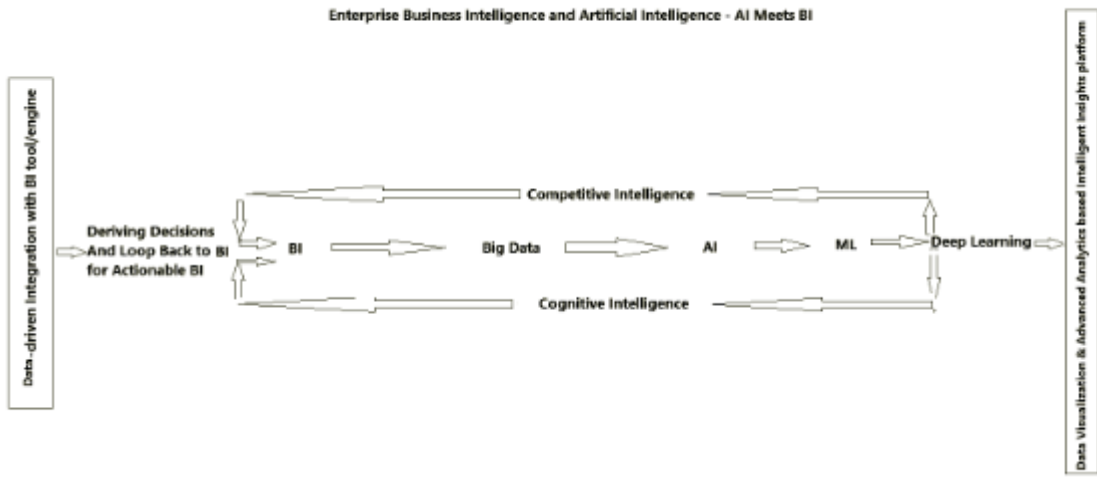
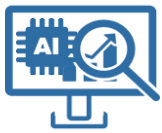
AI mengubah lanskap perusahaan dalam kecerdasan bisnis (BI) melalui kecerdasan adaptif di mana aplikasi dan analitik dapat diintegrasikan bersama dengan menerapkan dan menanamkan teknik berbasis AI/ML. Mulai dari menyimpan semua data (sebaiknya dalam katalog), melakukan penemuan data, menerapkan kualitas dan tata kelola data pada data tersebut, dan menggunakan metadata untuk rekomendasi berbasis AI/ML hingga membangun platform pembelajaran mesin untuk semua data, menambah data dengan menanamkan ML (misalnya, menambahkan konteks yang relevan ke data seperti menambahkan data etnis dan demografi berdasarkan etnis pada nama seseorang), perusahaan dapat memperoleh keunggulan kompetitif dalam industri tertentu.

Dengan memungkinkan pengayaan data, rekomendasi visual, analitik prediktif, dan analitik personal dari on-premise hingga on-the-go, solusi BI telah berkembang pesat dalam mempercepat wawasan berbasis data, sehingga mewujudkan perpaduan AI dan BI. Mempercepat pengambilan keputusan dengan AI dan ML menggunakan penemuan data cerdas, visualisasi dan dasbor interaktif, pengayaan data cerdas, AI yang dapat disesuaikan untuk membuat bagan berbasis model prediktif, dan menentukan outlier lebih cepat tanpa perlu coding atau memerlukan keahlian atau alat tambahan, adalah semua cara analitik bertenaga AI telah dan sedang mengubah lanskap BI perusahaan. Semua ini tanpa kehilangan fleksibilitas, kelangsungan bisnis, skalabilitas, kinerja, dan kolaborasi, serta dilakukan secara otomatis.

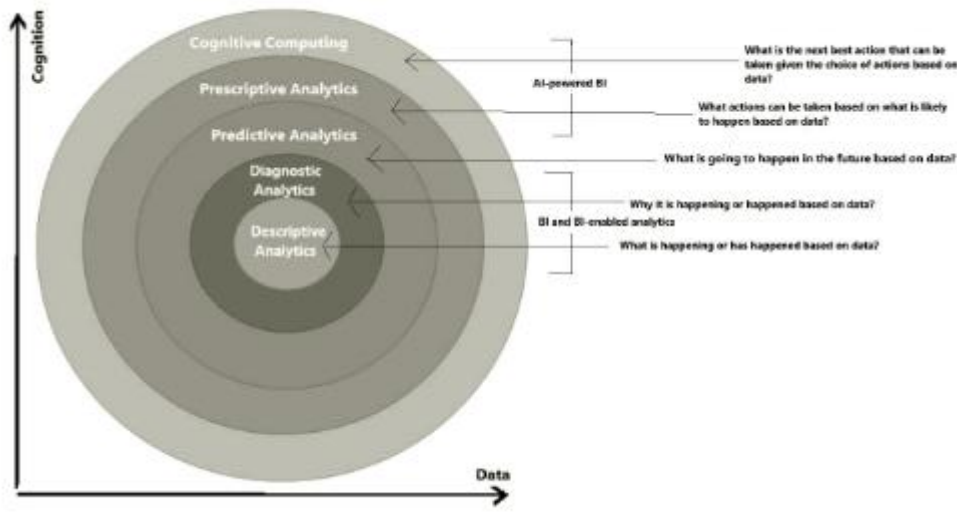
Ditambah dengan kemampuan bahasa alami seperti kueri bahasa alami (NLQ), pemrosesan bahasa alami (NLP), dan pembangkitan bahasa alami (NLG), kini kita memiliki solusi BI yang memiliki fitur-fitur ini, memberikan analitik yang menceritakan kisah berdasarkan data secara langsung, relevan, dan berkinerja tinggi. Gambar 2.1 menunjukkan sistem BI berbasis AI yang umum.

2.4 LINGKUP ANALITIK

Lingkup analitik yang disebut mengandung analitik deskriptif pada intinya dan berkembang ke luar dalam lingkaran-lingkaran melingkar yang mencakup analitik diagnostik, analitik prediktif, analitik preskriptif, dan komputasi kognitif. Jenis analitik generasinya berikutnya yang disebut analitik tertambah yang terdiri dari kecerdasan mesin dan intuisi manusia merupakan bagian dari komputasi kognitif dan AI adaptif. Dan ketika AI adaptif terintegrasi dengan BI, dunia BI baru terbuka yang dapat menyelaraskan proses bisnis untuk menghasilkan wawasan cerdas. Hal ini membantu perusahaan memperoleh kecerdasan kompetitif yang, pada gilirannya, memungkinkan bisnisnya tetap terdepan dalam industri. Gambar 2.2 menggambarkan lingkup analitik yang umum, beserta pendorong utama setiap jenis analitik.



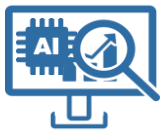
Gambar 2.1 Sistem BI berbasis AI yang umum.



Gambar 2.2 Lingkup analitik yang umum.

2.5 ANALISIS BERBASIS BI VS. ANALISIS BERBASIS AI

Kecerdasan bisnis memiliki kemampuan untuk memberikan hasil berdasarkan laporan siap pakai dan analisis visual (BI generasi pertama), analisis mandiri, dan penemuan data (BI generasi kedua). Teknik-teknik ini juga memungkinkan penerapan tren dan proyeksi berbasis tren berdasarkan data, tetapi tidak memiliki kemampuan untuk membuat prediksi, seperti kemungkinan terjadinya peristiwa tertentu misalnya, apa yang akan terjadi pada penjualan jika suatu produk berevolusi seiring waktu, dll. atau analisis yang lebih mendalam, seperti mengapa segmen pelanggan tertentu membatalkan langganan mereka untuk suatu produk tertentu. Dengan diperkenalkannya analisis berbasis AI ke dalam BI, gelombang baru BI disebut BI generasi ketiga muncul yang dapat menyediakan analisis prediktif dan banyak lagi.



Jadi, jawaban seperti apa yang dapat diberikan oleh sistem BI dan jawaban seperti apa yang dapat diberikan oleh sistem BI berbasis AI yang dapat dimanfaatkan perusahaan untuk meningkatkan persaingan bisnis? Perlu dicatat bahwa analitik berbasis BI masih diperlukan untuk semua jenis analisis bagi perusahaan, dan analitik bertenaga AI memberikan ekstensibilitas pada analitik ini melalui integrasi dengan AI, yang menghasilkan analitik yang lebih mendalam, lebih cerdas, lebih cepat, dan adaptif, serta dengan cara yang lebih baik, otomatis, dan otonom.

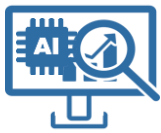
Tabel 2.1 memberikan gambaran perbandingan dan perbedaan antara analitik berbasis BI dan analitik bertenaga AI.

2.6 DARI DATA KE KEPUTUSAN CERDAS DAN BERWAWASAN

Ketika diproses dan dimasukkan ke dalam perangkat BI, data dapat membantu dalam pengambilan keputusan yang pada gilirannya dapat membantu perusahaan mencapai tujuan mereka. Misalnya, memproyeksikan pertumbuhan penjualan suatu produk berdasarkan tren penjualan produk di masa lalu dapat membantu bisnis dalam mengambil keputusan seperti memperluas basis operasinya dan/atau meningkatkan lini produknya. Pengambilan keputusan semacam ini bersifat deskriptif karena seolah-olah keputusan yang dibuat didasarkan pada fakta atau peristiwa yang telah terjadi di masa lalu.

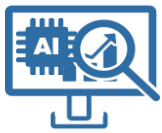
Dalam contoh penjualan produk, yang dimaksud adalah tren penjualan. Proyeksi di sini memberikan gambaran tentang bagaimana masa depan terlihat dalam hal pertumbuhan produk. Meskipun bersifat prediktif, proyeksi tidak dapat menjelaskan mengapa proyeksi yang dibuat dapat meningkatkan penjualan di masa mendatang, atau mengapa tidak. Menerapkan analitik bisnis dapat menjawab pertanyaan tentang mengapa peristiwa yang terjadi di masa lalu. Dengan mengambil contoh penjualan produk dalam konteksnya, proyeksi dapat menjelaskan mengapa tren penjualan baik pada titik waktu tertentu. Namun, pertanyaan seperti mengapa lini produk tidak mengalami pertumbuhan di wilayah tertentu lebih sulit dijawab, karena alasan berikut: Pertama, relevansi dengan jawaban yang diperoleh tidak dapat dipersonalisasi atau disesuaikan dari waktu ke waktu. Kedua, tindakan apa yang harus diambil selanjutnya dan mengapa tindakan tersebut perlu diambil selanjutnya untuk melihat peningkatan pertumbuhan penjualan tidak dapat dijawab dengan jelas. Di sinilah pengambilan keputusan yang cerdas berperan.

Analisis yang didukung BI dapat membantu dalam menyesuaikan relevansi pada jangka waktu tertentu. Ia dapat menjawab pertanyaan tentang apa dan kapan peristiwa masa lalu terjadi berdasarkan data, dan bagian analitik bisnis dapat menjawab pertanyaan tentang mengapa peristiwa tersebut terjadi berdasarkan data masa lalu, tetapi bukan pertanyaan tentang apa, kapan, dan mengapa selanjutnya, dan, lebih spesifik lagi, mengapa selanjutnya berdasarkan prediksi. Ini mencakup deskriptif, diagnostik, komparatif (sampai batas tertentu), dan prediktif (pada tingkat yang lebih rendah terutama proyeksi ke masa depan berdasarkan data dan peristiwa masa lalu).



Tabel 2.1 Analitik Berbasis BI versus Berbasis AI

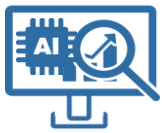
Analitik Berbasis BI	Analitik Bertenaga AI
Menyarankan faktor deskriptif berbasis garis keturunan data dan menghasilkan analitik berdasarkan faktor tersebut. Di sini, analisis ini tidak menyarankan data baru yang dapat dimanipulasi untuk analisis yang lebih baik (hanya analisis).	Menyarankan segala hal tentang data hanya dengan melihat sumbernya, serta menyarankan data baru yang dapat digunakan untuk menghasilkan analitik. Selain itu, AI dapat menghasilkan data baru berdasarkan data yang sudah ada melalui teknik augmentasi data (analisis dan sintesis).
Responsif dan visual data dapat digunakan untuk menyarankan metrik berdasarkan dasbor dan jenis komponen visual lainnya. Selain itu, alasan data dapat diperoleh berdasarkan respons yang diperoleh dari bagaimana peristiwa berbasis data dukungan pengambilan keputusan.	Proaktif dan dapat divisualisasikan secara dinamis berdasarkan hasil proaktif untuk wawasan mendalam pengambilan keputusan.
Bersifat non-percakapan membantu dalam proses pengambilan keputusan bisnis melalui analisis dan visualisasi interaktif.	Baik analitik percakapan maupun visual yang pertama memungkinkan pengajuan pertanyaan dalam bahasa alami, dan yang terakhir mengintegrasikan hasil ke dalam visualisasi instan yang dibuat secara otomatis untuk analitik lebih lanjut, namun tetap mempertahankan konteksnya.
Menggunakan analisis berbasis kueri yang bekerja pada jenis data tertentu. Hal ini mungkin menjadi kendala saat melakukan eksplorasi data yang mendalam bagi pengguna bisnis.	Memungkinkan analisis pada semua jenis kumpulan data dengan membawa algoritma ke data, sehingga membantu dalam eksplorasi data tanpa batasan berlapis bagi pengguna bisnis.
Kueri yang mendukung BI sebagian besar menggunakan rangkaian baris yang dikembalikan oleh kueri untuk melakukan analisis, memperoleh analisis, dan mempelajarinya.	Eksplorasi berbasis AI terutama menggunakan kumpulan kueri inklusi dan eksklusif untuk belajar dari maupun menghasilkan analitik baru. Selain itu, jenis pembelajaran ini memungkinkan studi korelasi yang terlibat dan menggunakannya untuk menentukan kepentingan variabel (pentingnya penggerak bisnis utama).
Fleksibilitas analitis merupakan fungsi dari penggerak bisnis, indikator kinerja utama (KPI), dan analitik swalayan. Sebagai contoh, analitik yang didukung BI dapat menentukan Margin Kotor dan Pendapatan Kotor, beserta trennya dari waktu ke waktu. Namun, menghubungkan keduanya untuk menentukan analitik khusus yang paling relevan bagi setiap jenis pengguna tidak mudah	Sebaliknya, analitik bertenaga AI tidak hanya memperhitungkan konteks di balik keduanya, tetapi juga korelasi di antara keduanya, seiring waktu, dengan cara menyarankan data baru dan menghasilkan data baru untuk membantu dalam menghitung konteks (analitik tambahan) dan analisis komparatif. Selain itu, analitik ini mempelajari apa yang terjadi dan apa yang tidak terjadi dengan mempelajari dan



<p>dicapai hanya dengan BI bahkan ketika visualisasi tersedia.</p>	<p>mengeksplorasi set inklusi dan eksklusi. Terakhir, analitik ini mempelajari perubahan yang dibuat setiap jenis pengguna pada visualisasi praktik terbaik serta dari perubahan pilihan dalam visualisasi. Hal ini memberikan kemudahan dan fleksibilitas dalam menentukan analitik khusus yang disesuaikan untuk setiap jenis pengguna.</p>
<p>Analisis lanjutan dan integrasi analisis khusus yang melampaui kemampuan mesin BI tertentu dapat menjadi tantangan tersendiri.</p>	<p>Dapat dengan mudah diintegrasikan dengan mesin pihak ketiga lain yang menyediakan analitik mendalam, analitik prediktif, dan model AI/ML lainnya.</p>
<p>Analisis tertanam terbatas pada batasan alat BI.</p>	<p>Analisis tertanam yang luas dengan fitur penyempurnaan yang mencakup berbagai konteks bisnis, menghasilkan wawasan baru dan dengan demikian memungkinkan BI adaptif.</p>
<p>Analitik berbasis BI dapat membantu dalam menyesuaikan relevansi pada rentang waktu tertentu. Analitik ini dapat menjawab pertanyaan tentang apa dan kapan peristiwa masa lalu terjadi berdasarkan data, dan bagian analitik bisnis dapat menjawab mengapa peristiwa tersebut terjadi berdasarkan data masa lalu, tetapi bukan pertanyaan tentang apa, kapan, dan mengapa selanjutnya, dan lebih spesifik lagi, mengapa selanjutnya berdasarkan prediksi. Ini mencakup deskriptif, diagnostik, komparatif (sampai batas tertentu), dan prediktif (pada tingkat yang lebih rendah terutama proyeksi ke masa depan berdasarkan data dan peristiwa masa lalu).</p>	<p>Analisis berbasis AI dapat digunakan untuk menyesuaikan dan meningkatkan relevansi seiring waktu, sehingga memberikan wawasan tentang apa, mengapa, dan ke mana langkah selanjutnya. Aspek "mengapa" dari apa dan ke mana langkah selanjutnya merupakan aspek unik yang dapat ditentukan oleh AI. Ini mencakup analisis naratif, visual, diagnostik, komparatif, dan prediktif.</p>
<p>Temukan produk berdasarkan fiturnya klasifikasi deklaratif menggunakan KPI dan/atau analisis deskriptif.</p>	<p>Temukan produk dengan menganalisis klasifikasi foto-gambar menggunakan pembelajaran mendalam.</p>

Analisis berbasis AI dapat digunakan untuk menyesuaikan dan meningkatkan relevansi seiring waktu, sehingga memberikan wawasan tentang apa, mengapa, dan ke mana langkah selanjutnya. Bagian "mengapa" dari apa dan ke mana langkah selanjutnya merupakan aspek unik yang dapat ditentukan oleh AI. Ini mencakup analisis naratif, visual, diagnostik, komparatif, dan prediktif.

Algoritma pembelajaran mesin dan model AI prediktif dapat diintegrasikan ke dalam platform BI untuk menghitung relevansi kontekstual menggunakan berbagai teknik yang pada gilirannya dapat membantu menjawab pertanyaan-pertanyaan seperti di atas, yang lebih penting dalam



hal pendorong bisnis sekaligus relevan dengan konteks yang dimaksud. Perjalanan dari data menuju keputusan yang berwawasan melalui analitik berbasis AI biasanya terdiri dari langkah-langkah berikut:

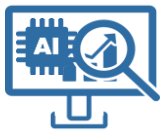
1. Menemukan masalah dan pendorong bisnis utama yang terkait dengannya
2. Melakukan impor data mandiri dari semua sumber data yang relevan
3. Memperkaya data dengan memanfaatkan pembelajaran mesin berbasis AI dan analitik tingkat lanjut (terintegrasi ke dalam platform BI)
4. Melakukan eksplorasi data tambahan dengan AI tertanam dan mengubah visualisasi menjadi narasi, jika diperlukan, menggunakan teknik NLG
5. Menemukan wawasan berbasis data yang didukung oleh AI/ML dan pembelajaran mendalam yang paling selaras dengan pendorong bisnis utama dalam konteksnya. Gunakan augmented analytics yang dikombinasikan dengan AI/ML untuk menghasilkan wawasan yang lebih cerdas.
6. Buat visual secara otomatis untuk wawasan berbasis data menggunakan AI/ML dan simpan sebagai komponen yang dapat digunakan kembali, misalnya, format PDF atau grafik.
7. Buat visual berbasis suara menggunakan NLP dan tambahkan narasi ke visual menggunakan NLG.
8. Otomatiskan analitik bertenaga AI di atas melalui alur kerja yang dapat diterapkan dan dijadwalkan, atau dengan cara lain yang didukung oleh platform BI yang tertanam AI/ML.
9. Aktifkan hal-hal di atas untuk solusi BI bertenaga AI yang dapat diakses melalui perangkat seluler.
10. Perkaya solusi untuk pembelajaran dan eksplorasi lebih lanjut berdasarkan umpan balik pengguna yang dapat dimasukkan ke dalam solusi dan dimanfaatkan menggunakan AI/ML yang melekat di dalamnya.

Langkah-langkah di atas dapat diikuti sebagai cetak biru untuk solusi BI bertenaga AI pada umumnya yang memungkinkan semua jenis data yang menceritakan kisah secara real-time, dapat divisualisasikan, dan berwawasan.

2.7 TEKNIK BERBASIS AI UNTUK BI

Terdapat banyak teknik berbasis AI yang dapat disematkan ke dalam platform BI atau diintegrasikan dengan sistem AI pihak ketiga. Dimulai dengan analitik tertanam dan analitik terpandu hingga analitik yang lebih canggih seperti analitik tertambah dan yang berbasis pembelajaran mendalam hingga AI percakapan, platform BI saat ini dimodernisasi untuk mengatasi berbagai masalah bisnis di seluruh spektrum analitik. Bagian ini menguraikan beberapa teknik berbasis AI yang relevan dengan empat kasus penggunaan industri. Berikut daftarnya:

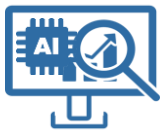
1. Algoritma pembelajaran mesin klasifikasi menggunakan hutan acak dan pohon keputusan digunakan untuk mengklasifikasikan tingkat penjualan komoditas berdasarkan atributnya (analitik deskriptif).



2. Teknik berbasis pembelajaran mendalam (menggunakan jaringan saraf tiruan) untuk memprediksi harga komoditas sebelumnya (analitik prediktif).
3. Algoritma pembelajaran mesin berbasis jaringan saraf tiruan untuk sistem rekomendasi guna menyarankan pilihan optimal berdasarkan skor (analitik preskriptif dan prediktif). Sebagai contoh, praktik penggunaan jaringan saraf dalam untuk membuat rekomendasi dari sudut pandang bisnis dapat mencakup rekomendasi sumber produk yang layak dengan memprediksi "skor" setiap sumber sebuah solusi analitik preskriptif dan memprediksi kinerja produk di pasar berdasarkan sumber yang direkomendasikan sebuah solusi analitik prediktif. Keluaran rekomendasi dapat diintegrasikan secara visual ke dalam dasbor BI untuk mendapatkan wawasan yang dapat ditindaklanjuti (misalnya, Tableau® Dashboard, Oracle® BI, Microsoft® BI, dll.).
4. Pengenalan gambar otomatis (untuk klasifikasi gambar).

Pertimbangkan contoh sistem BI yang menemukan dan merekomendasikan perangkat wearable berdasarkan pilihan pribadi pelanggan. Ada dua pendekatan yang dapat dilakukan, yaitu:

1. Dapatkan detail pelanggan beserta detail pakaian dan perangkat wearable lain yang baru-baru ini dibeli pelanggan dan bandingkan dengan metrik serupa setiap bulan selama enam bulan terakhir. Kemudian, rekomendasikan perangkat wearable yang serupa atau yang lebih baik dalam kategori yang sama melalui analitik deskriptif. Hasilnya dapat dikirim ke perangkat seluler pelanggan beserta deskripsi rekomendasi dan gambar perangkat wearable terkait.
2. Identifikasi pelanggan menggunakan pengenalan wajah serta pakaian dan perangkat wearable lainnya dengan melihat gambar pelanggan yang sama saat memasuki toko dan mengenali fitur wajah, melakukan pengenalan gambar untuk mengidentifikasi perangkat wearable orang tersebut seperti yang terlihat pada gambar, dan menghubungkan hasil keduanya. Kemudian, gabungkan hasil pengenalan gambar dengan data penjualan pelanggan secara terintegrasi, lakukan analisis mendalam, dan buat rekomendasi. Kemudian, kirimkan hasilnya kembali ke pelanggan yang sama menggunakan gambar wajah yang dikenali dan detail yang dapat dikenakan (kecerdasan mesin menggunakan AI/ML terintegrasi—pengenalan wajah, pemrosesan gambar, penemuan cerdas, asosiasi produk, dan analisis visual yang dipersonalisasi). Bagian penting dari analisis berbasis pembelajaran mendalam ini yang pertama kali dilakukan adalah pengenalan wajah dari gambar pelanggan yang diambil di dalam toko. Langkah ini sendiri melibatkan setidaknya empat sub-langkah, seperti yang tercantum di bawah ini:
 - i. Ekstraksi fitur—fitur apa yang harus diekstraksi dari wajah untuk membentuk set fitur yang dapat membantu mengidentifikasi wajah tertentu.
 - ii. Pemilihan fitur—fitur apa dari set yang diekstraksi di atas yang harus digunakan untuk mengidentifikasi "wajah" dengan sebaik-baiknya. Dengan



kata lain, fitur apa saja yang terbaik untuk digunakan dari set fitur yang diekstraksi.

- iii. Rekayasa fitur—bagaimana mengimprovisasi fitur yang dipilih pada langkah (ii) untuk mengidentifikasi gambar lain yang akan diklasifikasikan sebagai "wajah".
- iv. Citra Klasifikasi—cara mengklasifikasikan "wajah" yang terdeteksi sebagai milik orang tertentu.

Pendekatan kedua (didorong oleh AI) memiliki keunggulan dalam memberikan hasil berbasis data yang lebih cerdas secara real-time, relevansi yang lebih baik, rekomendasi yang disesuaikan, dan kemampuan untuk menjustifikasi mengapa rekomendasi tersebut dapat ditindaklanjuti untuk situasi yang saling menguntungkan.

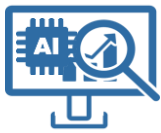
Beberapa aplikasi bisnis dari model pembelajaran mendalam berbasis AI adalah sebagai berikut:

- Otomatisasi TI
- Analisis sentimen
- Kecerdasan sosial dan rekayasa sosial
- Analisis preskriptif merekomendasikan produk berdasarkan "peringkat pengguna" mereka
- Pembuatan dan kueri bahasa alami
- Pengenalan gambar seperti pada penandaan gambar Facebook™
- Analisis visual
- Prediksi cuaca
- Pemrosesan ucapan-ke-teks kontekstual
- Aplikasi genetik

BI dapat memiliki solusi jaringan saraf dalam (DNN) yang tertanam dalam tumpukan implementasinya untuk mewujudkan hasil yang kompetitif. Misalnya, untuk aplikasi berbasis penandaan gambar dan pengenalan gambar, model jaringan saraf konvolusional (CNN) dapat dimanfaatkan hingga tingkat tersebut.

2.8 RINGKASAN

Bab ini berfokus pada penjelasan bagaimana analitik bertenaga AI menggunakan model pembelajaran mesin dapat diimplementasikan dalam platform BI untuk memungkinkan pengambilan keputusan yang cepat dan mendalam, yang dapat dicapai untuk tujuan BI berdasarkan hasil berbasis data yang diperoleh. Bab ini menguraikan berbagai aspek tentang bagaimana solusi AI yang memenuhi BI dapat diimplementasikan dalam perusahaan modern sebuah solusi yang memberikan wawasan waktu nyata untuk semua jenis pengguna, mulai dari TI hingga pengguna tingkat lanjut hingga pengguna akhir. Bab selanjutnya membahas empat kasus penggunaan dunia nyata yang mencakup analitik deskriptif, diagnostik, prediktif, dan preskriptif dari perspektif bisnis, secara detail, termasuk bagaimana masing-masing kasus tersebut cocok dengan lanskap BI, tetapi diimplementasikan menggunakan analitik bertenaga AI.



BAB 3

PENGGUNAAN INDUSTRI ENTERPRISE BI PERSPEKTIF BISNIS

3.1 PENDAHULUAN

Saat ini, kita memasuki generasi ketiga kecerdasan bisnis (BI), yang didukung oleh kecerdasan buatan (AI). Mulai dari persiapan dan pengolahan data hingga pengambilan keputusan hingga memonetisasi model berbasis AI untuk proses bisnis, BI yang didukung AI telah merevolusi pengambilan keputusan yang cerdas. Analisis bisnis, analisis data, pakar industri, dan pengguna akhir kini memiliki kemampuan untuk menggunakan semua jenis data perusahaan yang mereka miliki untuk menghasilkan data baru, menyimpulkan pendorong bisnis utama dan anomali dari data yang dihasilkan, dan menceritakan kisah secara visual yang sulit dipahami.

Lebih jauh lagi, BI yang didukung AI memungkinkan bisnis untuk membuat model khusus dengan kecerdasan bawaan yang cukup cerdas untuk mendeteksi faktor pendorong persaingan yang unggul dan memasukkannya ke dalam analisis dan dasbor analitik. Hal ini secara langsung berperan dalam memberikan pengalaman pelanggan yang mutakhir, yang didorong oleh solusi AI dan BI. Bab ini menjelaskan empat kasus industri dari perspektif bisnis dan sudut pandang BI yang didukung AI, serta bagaimana analisis ini dapat membantu dalam BI perusahaan (termasuk pengambilan keputusan yang cerdas).

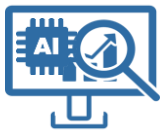
3.2 KLASIFIKASI MUTU PENJUALAN KOMODITAS BERDASARKAN ATRIBUT

Bagian ini menjelaskan kasus penggunaan klasifikasi daya jual komoditas sebagai mutu berdasarkan biaya yang dikurangkan dari kualitas dan atribut lainnya. Faktor-faktor utama yang secara langsung memengaruhi mutu, anomali, dan metrik kinerja utama (nilai F1, beserta presisi dan perolehan kembali) digambarkan secara visual. Poin penting yang perlu diperhatikan di sini adalah bahwa model pembelajaran mesin berbasis AI digunakan untuk memprediksi mutu berdasarkan data baru, dan matriks positif palsu versus positif benar ditampilkan yang dapat digunakan oleh pengguna bisnis untuk membuat keputusan yang tepat dalam menentukan biaya penjualan komoditas berdasarkan mutunya.

Analisis Deskriptif

Tujuan Kasus Penggunaan

- Sasaran
Untuk mengklasifikasikan mutu penjualan komoditas (anggur) berdasarkan kualitas dan atribut lainnya. Hal ini memungkinkan pengambilan keputusan, dari aspek kualitatif, terkait biaya dan penjualan anggur. Dalam kasus penggunaan khusus ini, kami menggunakan dua nilai untuk grade, yaitu 'SaleableCheap' dan 'SaleableCostly'.
- Perspektif Bisnis
Untuk memvisualisasikan distribusi data, faktor-faktor kunci yang memengaruhi grade, anomali grade, dan kinerja klasifikasi model berbasis AI dan ML yang dapat digunakan



untuk menentukan biaya anggur secara kuantitatif dan kualitatif berdasarkan grade yang diprediksi dan, dengan demikian, menentukan daya jual anggur.

- Set Data Masukan

Set data grade anggur putih dalam file .csv yang berisi grade dan 11 variabel lain yang terkait sebagai kolom.

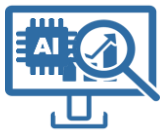
Pengantar Hutan Acak dan Pohon Keputusan dalam Pembelajaran Mesin

Untuk setiap data, dengan sekumpulan elemen data atau atribut yang menyusunnya, dan model bertenaga AI yang diterapkan pada data ini yang pada akhirnya dapat digunakan untuk mengklasifikasikan atau memprediksi hasil bisnis, setidaknya terdapat enam aspek yang dapat dianalisis. Berikut daftarnya dan apa yang diwakilinya:

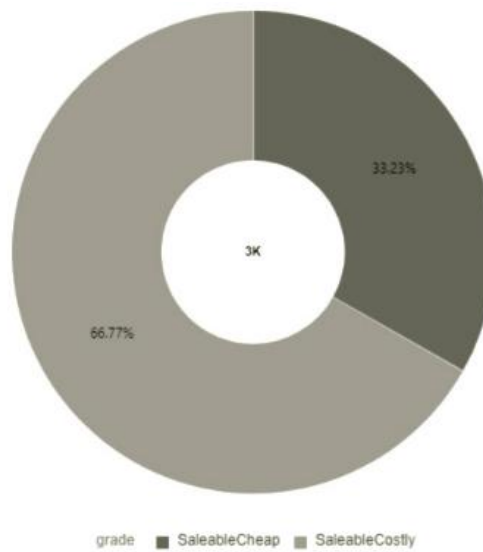
1. Nilai dan jenis nilai elemen data atau atribut, serta bagaimana elemen data tersebut saling terkait. Jika atribut bersifat kategorikal, nilai dalam atribut yang dapat diklasifikasikan, serta distribusi kelasnya, juga penting. Dalam contoh kasus penggunaan ini, ini berarti menentukan nilai grade dan distribusinya di seluruh dataset.
2. Penggerak utama atribut tertentu dalam dataset. Dengan kata lain, atribut apa dalam dataset yang paling baik menjelaskan atribut tersebut dan, oleh karena itu, paling memengaruhi pentingnya fitur. Dalam contoh kasus penggunaan ini, pertanyaannya adalah, Apa penggerak utama variabel grade dan bagaimana variabel tersebut dapat diandalkan untuk memberikan skor dalam keputusan bisnis yang membutuhkan grade sebagai faktor kunci?
3. Persentase akurasi keseluruhan untuk memprediksi nilai atribut, yaitu grade dalam contoh kasus penggunaan ini.
4. Kelompok dalam data yang dapat dianggap sebagai outlier.
5. Jenis model bertenaga AI yang dibangun atau diterapkan pada data pelatihan (yang dapat berupa subset dari dataset) dan model yang dibangun yang dapat bekerja pada data pelatihan ini.
6. Performa model bertenaga AI yang dibangun pada Langkah 5 ketika diterapkan pada data perusahaan dunia nyata dan integrasi serta analisis BI yang sesuai, termasuk otomatisasi analitik yang terlibat dan kemampuan kognitif model tersebut untuk memberikan konteks terbaik dari keputusan berbasis BI.

Sebagaimana ditunjukkan di bagian sebelumnya, daya jual grade diklasifikasikan sebagai SaleableCheap atau SaleableCostly. Gambar 3.1 menunjukkan distribusi grade ke dalam kelas-kelas ini.

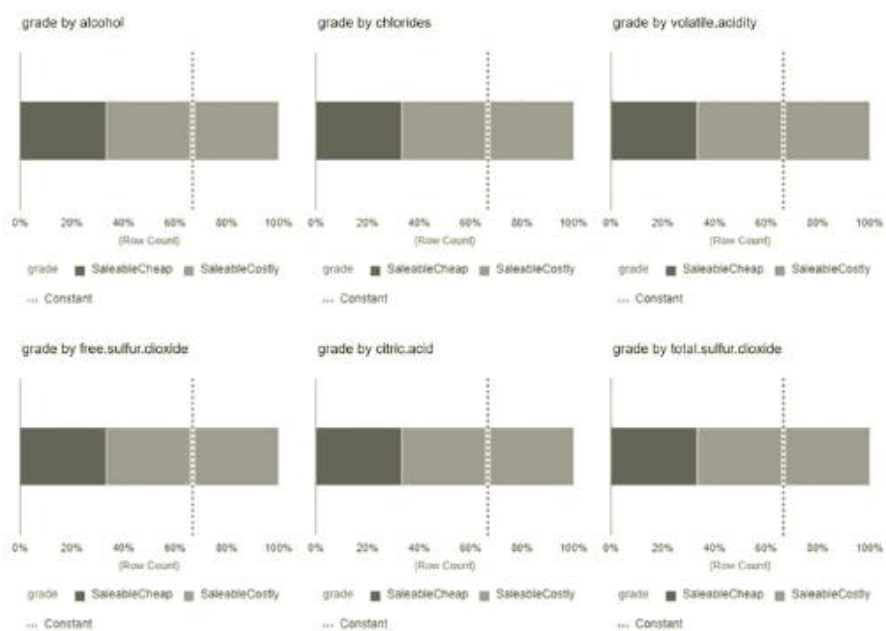
Berikut ini adalah snapshot data pelatihan:



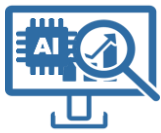
	fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides	free.sulfur.dioxide		
	total.sulfur.dioxide	density	pH	sulphates	alcohol	grade		
1	7.2	0.31	0.35	7.2	0.046			45
	178	0.99550	3.14	0.53	9.7	SaleableCheap		
2	7.0	0.20	0.33	4.7	0.030			25
	76	0.99202	2.88	0.54	10.5	SaleableCostly		
3	6.6	0.26	0.28	9.4	0.028			13
	121	0.99254	3.17	0.34	12.1	SaleableCostly		
4	7.3	0.19	0.27	13.9	0.057			45
	155	0.99807	2.94	0.41	8.8	SaleableCostly		
5	7.2	0.23	0.38	6.1	0.067			20
	90	0.99496	3.17	0.79	9.7	SaleableCheap		
6	7.9	0.35	0.24	15.6	0.072			44
	229	0.99785	3.03	0.59	10.5	SaleableCostly		



Gambar 3.1 Distribusi mutu anggur menjadi dua kelas: SaleableCheap dan SaleableCostly.



Gambar 3.2 Pendorong utama mutu anggur.



Kriteria kunci lainnya adalah pendorong utama mutu. Gambar 3.2 menunjukkan hal ini dalam hal atribut anggur apa yang memengaruhi proses mutu anggur.

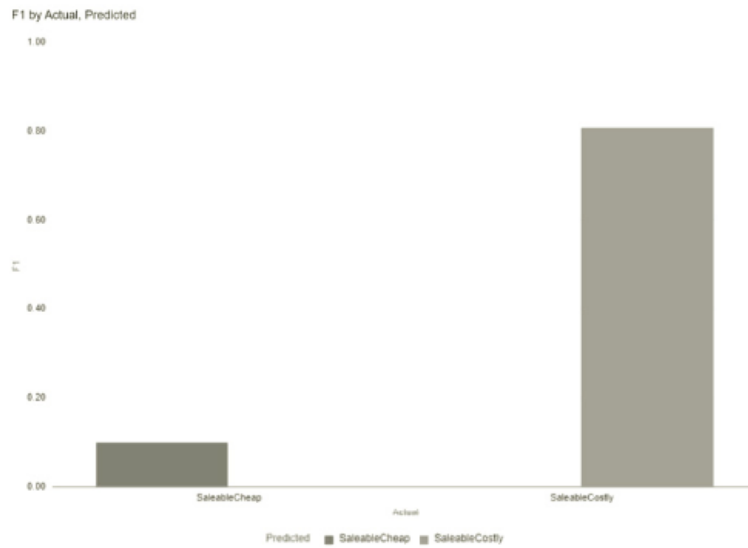
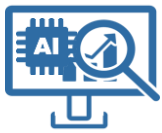
Mengenai aspek ketiga (sebagaimana diuraikan di atas), persentase keseluruhan untuk memprediksi hasil mutu adalah 65% di 7 kelompok. Kelompok-kelompok ini, beserta persentase akurasi dan keyakinannya, digambarkan pada Gambar 3.3a dan 3.3b. Selanjutnya, penting untuk menentukan anomali mutu. Hal ini ditunjukkan pada Gambar 3.4. Model AI/ML yang digunakan untuk kasus penggunaan yang dimaksud adalah klasifikasi hutan acak. Setelah membangun model pada data pelatihan, metrik kinerja yang diperoleh seperti yang digambarkan pada Gambar 3.5 hingga 3.9.

Selama pembuatan model AI-ML hutan acak, 30% data pelatihan diindikasikan untuk digunakan untuk validasi model. Ini berarti bahwa dari 3.428 observasi dalam set data pelatihan, 2.400 observasi digunakan untuk membangun model dan 1.028 observasi digunakan untuk memvalidasi model yang dibangun dan, dalam prosesnya, meningkatkan akurasi model. Proses validasi yang terlibat disebut validasi k-fold, di mana $k \geq 2$, dan dalam fase validasi, model akan mengulangi proses pembangunan dan validasi sehingga nilai variabel target optimal dan tingkat kesalahan diminimalkan.

Gambar 3.5 menunjukkan matriks kebingungan (tabel silang jumlah observasi di seluruh kelas aktual dan yang diprediksi). Ini juga memberikan % akurasi model serta tingkat positif palsu. Hal lain yang perlu diperhatikan dalam gambar ini adalah angka 1.028. Ini mewakili 30% dari data pelatihan yang digunakan untuk validasi, yang, ketika dihitung sebagai 30% dari 3.428 (jumlah total observasi), menghasilkan 1.028. Dan nilai prediksi berasal dari proses validasi k-fold yang terjadi selama proses pembangunan model.



Gambar 3.3a % Akurasi Keseluruhan untuk memprediksi hasil nilai -1.



Gambar 3.6 Metrik Performa Model AI-ML F1 berdasarkan Aktual dan Prediksi pada data pelatihan (validasi).

Variabel respons dalam kasus penggunaan ini adalah grade dengan nilai SaleableCheap dan SaleableCostly. Kedua kelas ini perlu dipertimbangkan dalam menentukan akurasi prediksinya. Dari Gambar 3.1 terlihat adanya ketidakseimbangan kelas. Seperti yang terlihat pada Gambar 3.5 dan 3.6, dalam kasus ini, pengklasifikasi kami memberikan skor F1 yang tinggi untuk kelas SaleableCostly (0,8) dalam hal Prediksi.

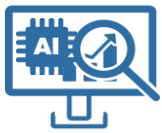
Selain itu, tingkat kesalahan klasifikasi untuk SaleableCostly adalah 0,7% (sangat rendah). Namun, skor F1 untuk SaleableCheap sekitar 0,1, dengan tingkat kesalahan klasifikasi sebesar 94%. Skor F1 dianggap sebagai rata-rata harmonis dari recall dan presisi. Pengklasifikasi kami menunjukkan apakah suatu anggur memiliki grade SaleableCheap atau SaleableCostly. Hal ini menunjukkan bahwa nilai SaleableCostly lebih baik daripada SaleableCheap karena memiliki skor F1 yang lebih tinggi dan tingkat kesalahan klasifikasi yang lebih rendah. Namun, dalam kasus penggunaan kami, penting untuk mengevaluasi kedua nilai tersebut, terutama karena ketidakseimbangan kelas. Kami dapat melakukannya dengan menggunakan dua metrik tambahan: Presisi dan Recall. Untuk menghitung presisi dan recall, kami perlu mengetahui empat definisi, seperti yang dinyatakan di bawah ini:

- Positif benar (TP): Hasil di mana model memprediksi kelas positif dengan tepat
- Negatif benar (TN): Hasil di mana model memprediksi kelas negatif dengan tepat
- Positif salah (FP): Hasil di mana model memprediksi kelas positif secara tidak tepat
- Negatif salah (FN): Hasil di mana model memprediksi kelas negatif secara tidak tepat

Dengan menerjemahkan hal ini ke dalam model AI yang kami bangun dan nilai, serta mempertimbangkan matriks pada Gambar 3.5, kami dapat memperoleh dua tabel (lihat Tabel 3.1 dan 3.2).

Tabel 3.1 Tabel untuk Kelas SaleableCheap Menunjukkan TP, FP, FN, dan TN

Benar Positif	Positif Palsu
----------------------	----------------------



Sebenarnya: SaleableCheap Diprediksi: SaleableCheap	Aktual: SaleableCostly Diprediksi: SaleableCheap
Negatif Palsu Sebenarnya: SaleableCheap Diprediksi: SaleableCostly	Benar Negatif Aktual: SaleableCostly Diprediksi: SaleableCostly

Tabel 3.2 Tabel untuk Kelas SaleableCostly Menunjukkan TP, FP, FN, dan TN

Benar Positif Sebenarnya: SaleableCostly Diprediksi: SaleableCostly	Positif Palsu Aktual: SaleableCheap Diprediksi: SaleableCostly
Negatif Palsu Sebenarnya: SaleableCostly Diprediksi: SaleableCheap	Benar Negatif Aktual: SaleableCheap Diprediksi: SaleableCheap

Dari perspektif bisnis, metrik Presisi dan Ingat cenderung menjawab pertanyaan-pertanyaan berikut:

- Untuk presisi, berapa persen prediksi positif yang benar-benar akurat? Dalam kasus penggunaan yang dibahas, hal ini mengarah pada identifikasi yang tepat antara mutu murah dan mahal (dan sebaliknya).
- Untuk ingat, berapa persen positif aktual yang dikenali dengan benar? Dalam kasus penggunaan yang dibahas, hal ini mengarah pada identifikasi setiap kelas mutu dari kedua kelas tersebut.

Dengan mempertimbangkan kedua pernyataan di atas, model yang kita bangun harus berkinerja baik dalam hal presisi maupun ingat.

Sekarang, mari kita hitung presisi dan perolehan model kita untuk masing-masing dari dua kelas. Berikut ini menggambarkan cara melakukannya:

$$Precision (p) = TP / (TP + FP)$$

$$Recall (r) = TP / (TP + FN)$$

Dan dari uraian di atas, kita dapat menghitung skor F1 sebagai berikut:

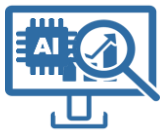
$$F1 - score = 2 / (1/r + 1/p)$$

Dengan mengganti nilai-nilai tersebut ke dalam rumus-rumus ini untuk setiap kelas dari Gambar 3.5 dan Tabel 3.1 dan 3.2, kita memperoleh presisi dan perolehan kembali serta skor F1 untuk masing-masing dari dua kelas, seperti yang diberikan di bawah ini.

$$Precision_{SaleableCheap} = TP / (TP + FP) = 18 / (18 + 5) = 0.783$$

$$Precision_{SaleableCostly} = TP / (TP + FP) = 684 / (684 + 321) = 0.68$$

These two values match with that shown in Figure 3.7.



$$\text{RecallSaleableCheap} = TP / (TP + FN) = 18 / (18 + 321) = 0.05$$

$$\text{RecallSaleableCostly} = TP / (TP + FN) = 684 / (684 + 5) = 0.993$$

These two values match with that shown in Figure 3.8.

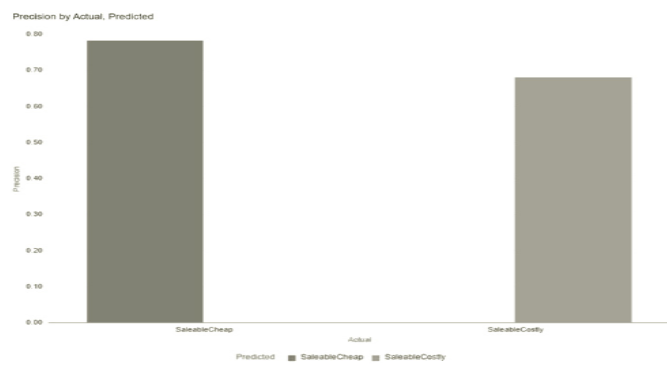
$$F1\text{SaleableCheap} = 2 / (1/\text{RecallSaleableCheap} + 1/\text{PrecisionSaleableCheap}) = 0.094$$

$$F1\text{SaleableCostly} = 2 / (1/\text{RecallSaleableCostly} + 1/\text{PrecisionSaleableCostly}) = 0.807$$

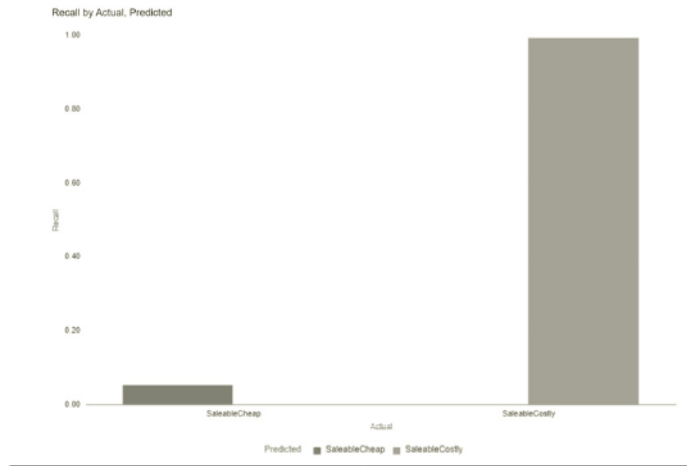
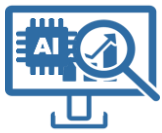
These two values match with that shown in Figure 3.6.

Presisi berkaitan dengan positif palsu (FP) dan presisi yang lebih tinggi berarti lebih sedikit positif palsu. Memiliki 0 atau lebih sedikit FP berarti prediksi model baik. Dalam kasus penggunaan kami, kami memiliki PrecisionSaleableCheap sebesar 0,783 dan PrecisionSaleableCostly sebesar 0,68, yang secara signifikan lebih tinggi dibandingkan dengan batas pengukuran 1,0.

Recall berkaitan dengan negatif palsu (FN) dan memiliki recall yang lebih tinggi berarti lebih sedikit negatif palsu. Memiliki 0 atau lebih sedikit FN berarti prediksi model baik. Dalam kasus kami, kami memiliki RecallSaleableCheap sebesar 0,05 dan RecallSaleableCostly sebesar 0,993. FN yang sesuai untuk SaleableCheap dan SaleableCostly masing-masing adalah 321 dan 5. Namun, seiring dengan peningkatan nilai Presisi, nilai yang sesuai untuk Recall menurun, dan sebaliknya untuk kedua kelas. Ini adalah perilaku yang diharapkan karena, dalam kebanyakan kasus, terdapat trade-off antara Presisi dan Recall. Dengan mengamati hal ini, model kami menunjukkan bahwa kedua kelas SaleableCheap dan SaleableCostly menjadi faktor dalam analisis dan analitik bisnis.



Gambar 3.7 Metrik Kinerja Model AI-ML Presisi berdasarkan Aktual, Diprediksi pada data pelatihan (validasi).



Gambar 3.8 Metrik Kinerja Model AI-ML yang Diingat Berdasarkan Aktual, Diprediksi pada data pelatihan (validasi).

**Confusion Matrix on test data for the AI-ML
Random Forest used to predict on test data**

Count by grade, Predictedgrade

grade	SaleableCheap	SaleableCostly
	Count	Count
SaleableCheap	26	475
SaleableCostly	7	962

Gambar 3.9 Matriks Kebingungan Performa Model AI-ML pada data uji (baru).

Ketika kami menghitung presisi dan perolehan prediksi versus aktual menggunakan data uji, hasilnya adalah sebagai berikut:

$$Precision_{SaleableCheap} = TP / (TP + FP) = 10 / (10 + 2) = 0.83$$

$$Recall_{SaleableCheap} = TP / (TP + FN) = 10 / (10 + 491) = 0.019$$

$$Precision_{SaleableCostly} = TP / (TP + FP) = 967 / (967 + 491) = 0.663$$

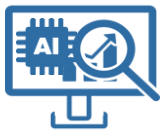
$$Recall_{SaleableCostly} = TP / (TP + FN) = 967 / (967 + 2) = 0.998$$

Dan skor F1 yang sesuai adalah sebagai berikut:

$$F1_{SaleableCheap} = 0.037$$

$$F1_{SaleableCostly} = 0.80$$

Nilai-nilai di atas menunjukkan perilaku yang serupa dengan yang terlihat pada data validasi dalam hal presisi tinggi dengan recall rendah dan sebaliknya, serta skor F1, sehingga menunjukkan bahwa kedua kelas diperhitungkan dalam analisis dan pengambilan keputusan ketika model diterapkan pada data dunia nyata.



3.3 MEMPREDIKSI HARGA KOMODITAS DI MUKA

Bagian ini menjelaskan studi kasus di bidang metalurgi dengan memprediksi harga logam mulia berdasarkan data bulanan yang tersedia. Penggerak utama yang secara langsung memengaruhi harga, anomali, dan metrik kinerja utama (Akurasi, nilai F1, beserta presisi dan recall) digambarkan secara visual. Poin penting yang perlu diperhatikan di sini adalah bahwa model pembelajaran mesin bertenaga AI digunakan untuk menghasilkan prediksi harga pada data baru, dan matriks positif palsu versus positif benar ditampilkan yang dapat digunakan oleh pengguna bisnis untuk membuat keputusan yang mendalam dalam menentukan biaya penjualan logam mulia.

Analisis Prediktif

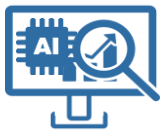
Tujuan Kasus Penggunaan

- Sasaran
Untuk memprediksi harga logam mulia berdasarkan harga bulanan yang tersedia untuk tahun 1950 hingga 2019. Hal ini memungkinkan pengambilan keputusan terkait penetapan harga emas batangan di pasar dalam berbagai mata uang dan di berbagai belahan dunia. Dalam kasus penggunaan khusus ini, kami menggunakan dua atribut 'Bulan' dan 'Tahun' dengan tanggal yang merupakan hari pertama setiap bulan.
- Perspektif Bisnis
Memvisualisasikan distribusi data, faktor-faktor pendorong utama yang memengaruhi harga, anomali harga, dan kinerja klasifikasi model berbasis AI dan ML yang dapat digunakan untuk memprediksi harga logam secara kuantitatif dan kualitatif.
- Masukan Dataset
Harga emas batangan di awal setiap bulan, dari tahun 1950 hingga 2019, dalam file .csv yang memiliki kolom Tanggal dan Harga.

Pengantar Singkat Kasus Penggunaan dari Sudut Pandang Analisis dan Analitik Prediktif

Kami menganalisis data masukan sebagai berikut:

1. Nilai dan jenis nilai elemen data atau atribut, serta bagaimana elemen data tersebut saling terkait. Jika atribut bersifat kategoris, nilai dalam atribut yang dapat diklasifikasikan dan distribusi kelasnya juga penting. Dalam kasus penggunaan contoh kami, variabel respons kami adalah harga yang kontinu. Dalam kasus penggunaan ini, visualisasi menunjukkan grafik Harga berdasarkan Tanggal pada data yang telah disiapkan, dengan Tanggal mewakili nilai tanggal dalam format MM/DD/YYYY dan DD selalu '01', yang mewakili hari pertama setiap bulan.
2. Jenis model bertenaga AI yang dibangun atau diterapkan pada data pelatihan (yang dapat berupa subset dari dataset) dan model yang dibangun yang beroperasi pada data pelatihan ini. Dalam kasus penggunaan ini, kami menggunakan tiga metode: ARIMA Musiman, ARIMA, dan ETS.
3. Tiga model bertenaga AI yang sama yang digunakan pada Langkah 5 (di atas) ketika diterapkan pada data perusahaan dunia nyata dan integrasi serta analisis BI yang sesuai, termasuk otomatisasi analitik yang terlibat dan kemampuan kognitif model tersebut untuk memberikan konteks terbaik dari keputusan berbasis BI.



Berikut adalah cuplikan data input:

	Tanggal	Harga
1	1950-01	34.73
2	1950-02	34.73
3	1950-03	34.73
4	1950-04	34.73
5	1950-05	34.73
6	1950-06	34.73

Berikut ini adalah sebagian data pelatihan yang telah disiapkan:

	Tanggal	Harga
1	1950-01-01	34.73
2	1950-02-01	34.73
3	1950-03-01	34.73
4	1950-04-01	34.73
5	1950-05-01	34.73
6	1950-06-01	34.73

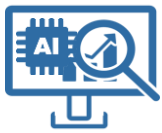
Daftar berikut menjelaskan secara singkat tiga model peramalan yang digunakan:

1. ARIMA merupakan Rata-Rata Bergerak Terintegrasi Auto-Regresif, yang dominan digunakan ketika data masa lalu secara keseluruhan cukup memadai untuk memproyeksikan masa depan.
2. ARIMA Musiman—Model ini dapat digunakan ketika terdapat pola perubahan yang berulang pada data selama periode waktu tertentu. Model ini dapat dianggap sebagai model yang tepat untuk meramalkan harga logam mulia di mana terdapat pola perubahan yang berulang selama beberapa tahun (berdasarkan perubahan harga selama beberapa bulan pada tahun-tahun tersebut).
3. Pemulusan Tiga Eksponensial (ETS)—Model standar ini lebih sering digunakan untuk memodelkan data deret waktu.

Untuk ketiga model ini, jumlah periode mewakili jumlah hari yang dianggap sebagai satu periode. Gambar 3.10 hingga 3.16 menggambarkan visualisasi Prakiraan Sebelum dan Sesudah yang diterapkan pada set data pelatihan menggunakan ketiga jenis model peramalan di atas.

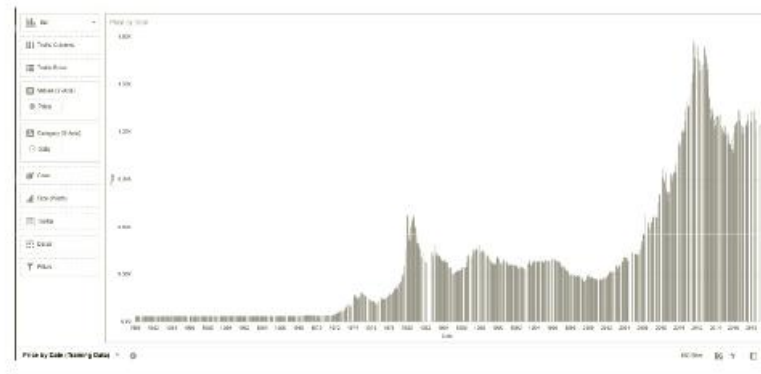
Grafik pada Gambar 3.10 hingga 3.19 menunjukkan hal-hal berikut:

- *Metrik yang diramalkan adalah "Harga" logam mulia pada awal setiap bulan dalam tahun tertentu.*
- *Deret waktu adalah periode waktu di mana model ML prakiraan sedang dibangun. Dalam kasus penggunaan kami, rentangnya adalah dari Januari 1950 hingga Januari 2019.*
- *Kolom dimensi adalah grain yang menjadi dasar prakiraan. Dalam kasus penggunaan kami, ini adalah hari pertama setiap bulan dalam tahun tertentu.*
- *Dalam grafik yang menunjukkan prakiraan, keluaran ML prakiraan adalah harga yang diprediksi, (bulan, tanggal, tahun) harga yang diprediksi, batas bawah nilai interval,*

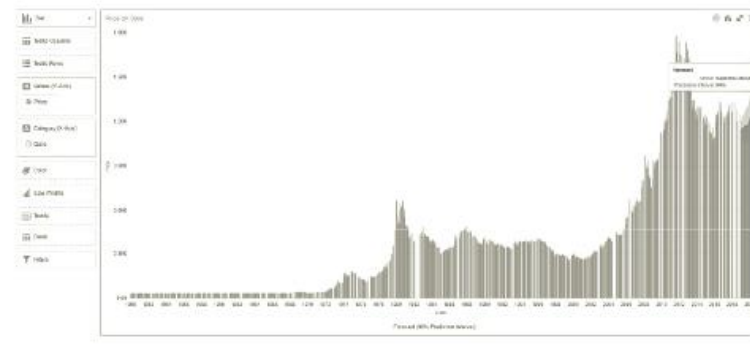


batas atas interval prediksi, dan nilai interval prediksi keyakinan yang digunakan untuk model prakiraan ML.

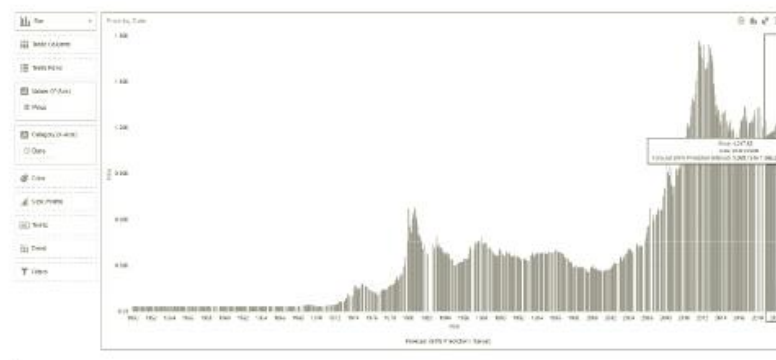
Gambar 3.17 hingga 3.19 menggambarkan visualisasi pada data uji sebelum perkiraan dan setelah perkiraan menggunakan model ARIMA Musiman.



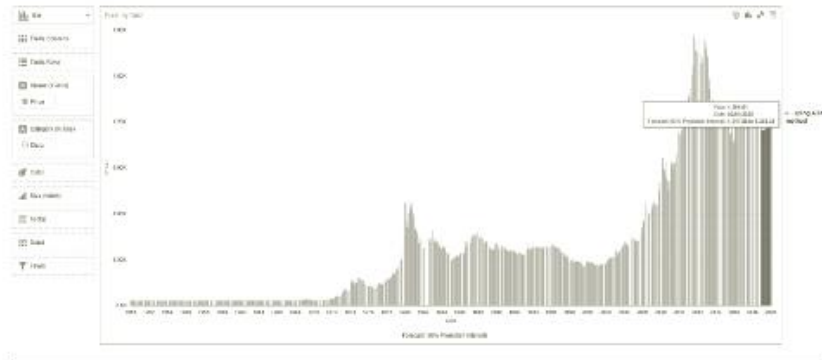
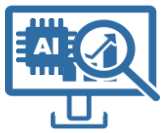
Gambar 3.10 Data pelatihan Harga berdasarkan Tanggal—Sebelum Prakiraan.



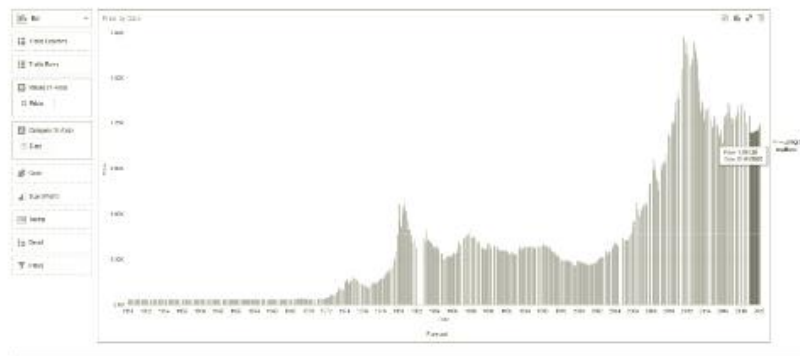
Gambar 3.11 Prakiraan (Harga berdasarkan Tanggal) menggunakan metode ARIMA Musiman (data pelatihan).



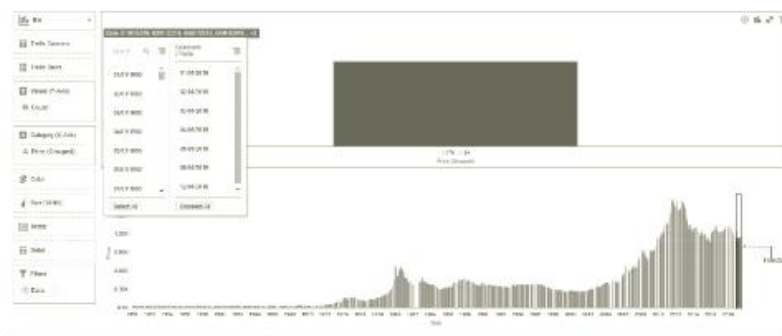
Gambar 3.12 Prakiraan (Harga berdasarkan Tanggal) menggunakan metode ARIMA Musiman (data pelatihan).



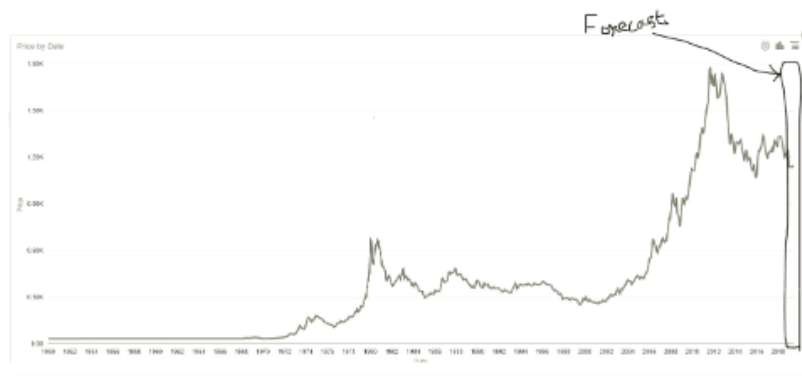
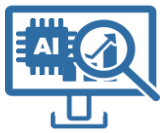
Gambar 3.13 Prakiraan (Harga berdasarkan Tanggal) menggunakan metode ARIMA (data pelatihan) yang menunjukkan Harga prakiraan untuk Februari 2020 (titik data baru).



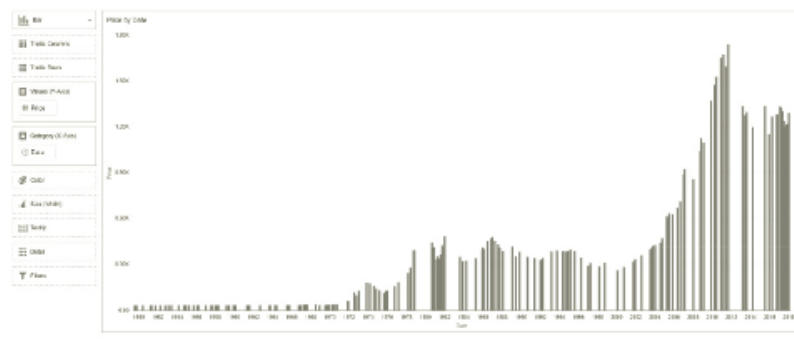
Gambar 3.14 Prakiraan (Harga berdasarkan Tanggal) menggunakan metode ETS (data pelatihan) yang menunjukkan Harga prakiraan untuk Januari 2020 (titik data baru).



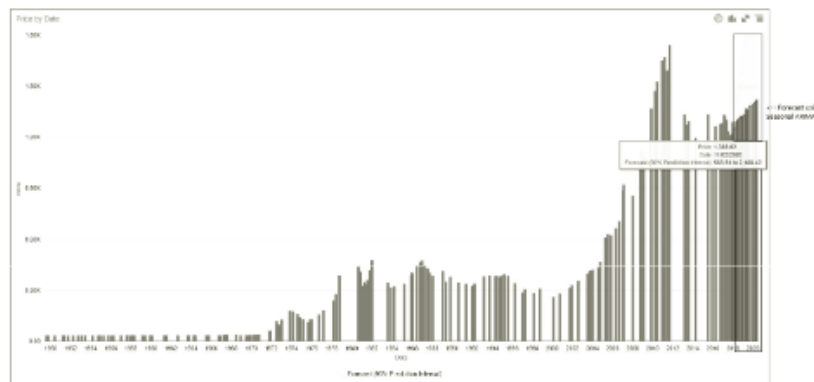
Gambar 3.15 Visualisasi Perkiraan Harga berdasarkan Tanggal pada data pelatihan menunjukkan tampilan terperinci nilai perkiraan.



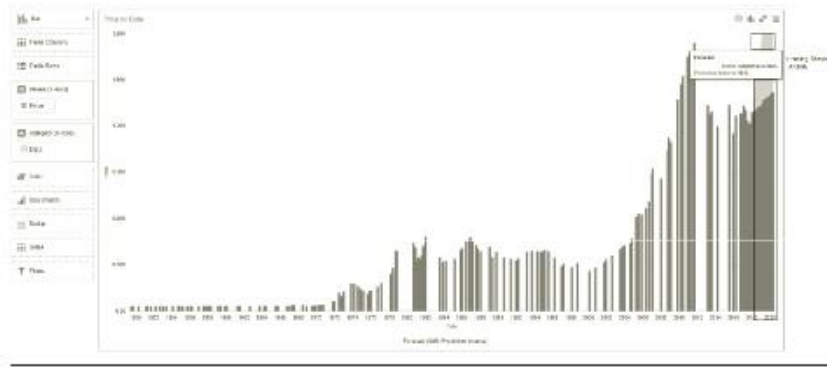
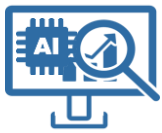
Gambar 3.16 Visualisasi Perkiraan Harga berdasarkan Tanggal pada data pelatihan menggunakan grafik kontinu.



Gambar 3.17 Data uji Harga berdasarkan Tanggal—Sebelum Perkiraan.



Gambar 3.18 Prakiraan (Harga berdasarkan Tanggal) menggunakan ARIMA Musiman pada data uji.



Gambar 3.19 Prakiraan (Harga berdasarkan Tanggal) menggunakan ARIMA Musiman pada data uji.

Seperti yang dapat dilihat dari grafik dengan nilai prakiraan, periode waktu bagian prakiraan meluas hingga tahun 2021, dalam beberapa kasus, dan terdapat beberapa overfitting dalam hal harga yang diprediksi. Ini merupakan perilaku yang diharapkan untuk model ML yang akan digeneralisasi pada data dunia nyata perusahaan. Data prakiraan dari visualisasi ini dihitung sebagai data augmented untuk data perusahaan lain yang, ketika digabungkan dengan data perusahaan tersebut, dapat dianalisis lebih lanjut melalui integrasi dengan dasbor BI untuk berfungsi sebagai analitik augmented, sehingga menghasilkan solusi AI yang bertemu dengan BI solusi yang meningkatkan standar wawasan kompetitif.

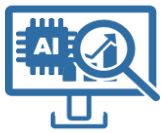
3.4 SISTEM REKOMENDASI PILIHAN OPTIMAL BERDASARKAN SKOR

Kasus penggunaan ini dibahas dari sudut pandang bisnis secara mendetail berdasarkan visualisasi berorientasi implementasi dan perspektif BI di Bab 4, Bagian 4.4.

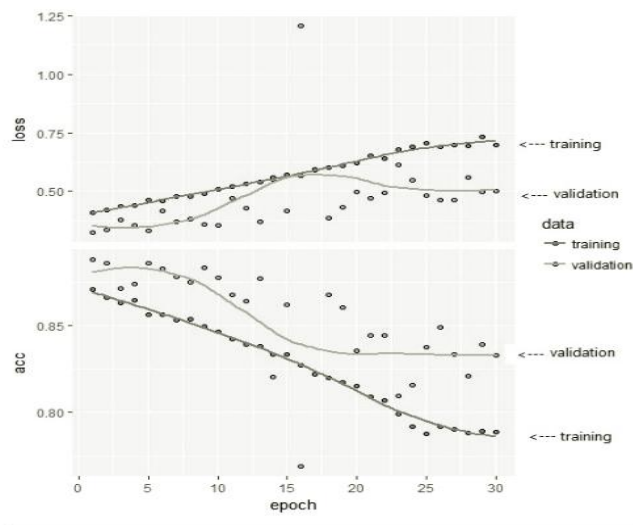
3.5 PENGENALAN CITRA OTOMATIS

Kasus penggunaan ini didasarkan pada pengenalan otomatis kelas citra masukan sebagai bagian dari serangkaian kelas yang telah ditentukan sebelumnya. Hal ini digunakan dalam banyak aplikasi di dunia nyata, seperti penandaan citra oleh Facebook™, dll. Kasus ini menggunakan jaringan saraf tiruan konvolusional (CNN) berbasis pembelajaran mendalam yang melakukan tugas-tugas berikut pada citra masukan menggunakan apa yang disebut lapisan jaringan saraf tiruan:

1. Lapisan masukan yang terdiri dari citra masukan sebagai larik piksel berukuran $n \times n$.
2. Lapisan ekstraksi fitur yang mengenali fitur-fitur tertentu dalam citra masukan. Contohnya adalah mata, telinga, hidung, dll., pada citra wajah. Menggunakan lapisan aktivasi untuk memicu simpul-simpul dalam CNN guna menghitung keluaran pada setiap simpul berdasarkan fungsi aktivasi.
3. Lapisan Max Pooling untuk reduksi dimensionalitas.
4. Lapisan Flatten untuk membuat larik 1D piksel yang telah diproses dari larik multi-D piksel dari citra masukan guna memastikan piksel-piksel ini berada dalam bentuk yang tepat untuk dikirim ke lapisan berikutnya, yang disebut lapisan padat.
5. Lapisan Dropout untuk mengurangi overfitting.



6. Lapisan Dense untuk menentukan simpul mana yang akan diaktifkan dan seberapa besar berdasarkan korelasi antara larik 1D dalam Flatten dan Dense. Disebut padat karena setiap lapisan dalam Dense terhubung ke setiap simpul di lapisan berikutnya— dengan kata lain, terhubung sepenuhnya.
7. Lapisan Softmax untuk mengklasifikasikan kelas mana yang dimiliki citra masukan. (Di sini, jumlah kelas diasumsikan lebih besar dari 2.)
8. Lapisan keluaran yang mengembalikan label yang sesuai dengan simpul-simpul yang intensitasnya paling tinggi diaktifkan.



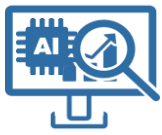
Gambar 3.20 Kurva kerugian dan akurasi model CNN pada citra pelatihan dan citra validasi.

Model CNN pertama-tama berlatih pada data pelatihan, kemudian mengklasifikasikan gambar dalam data uji. Dengan demikian, model ini mengenali gambar dalam data uji dengan menerapkan model berbasis AI. Keluaran dari klasifikasi (dan dengan demikian pengenalan) adalah penentuan label gambar yang tepat untuk setiap gambar yang diberikan dan pemeriksaan tampilan raster dari gambar yang sama dalam set uji.

Performa CNN sering diukur dengan mempelajari kurva kerugian dan akurasi model. Contoh kurva kerugian dan akurasi untuk kasus penggunaan kami dalam mengenali gambar dari dataset Fashion MNIST ditunjukkan pada Gambar 3.20. Pengamatan berikut dapat dilakukan dari gambar ini:

1. Seiring bertambahnya jumlah epoch batchwise atau iterasi pelatihan, kerugian meningkat dan akurasi menurun. Akurasi menurun sebesar $\sim 0,05$ dan kerugian meningkat sebesar $\sim 0,25$.
2. Namun, berdasarkan data validasi, akurasinya konstan pada 0,825 atau 82,5%. Dan kerugian pada data validasi konstan pada 0,5.

Akurasi klasifikasi sebesar 82,5 mengklasifikasikan sepatu ankle (dalam set uji) dengan tepat sebagai milik label 9, seperti yang ditunjukkan pada Gambar 3.21.



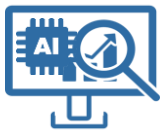
Gambar 3.21 Label yang diprediksi dalam set pengujian termasuk dalam label 9, yang cocok dengan nilai aktual 9.

Dengan memeriksa nilai label (Prediksi versus Aktual) untuk setiap label yang diberikan, misalnya, `pred_test[1]` dan `test y-part[1]`, diamati bahwa setiap nilai termasuk dalam kelas 9. Selain itu, dengan memplot citra raster dari `test x-part` asli [1, ,], kita melihat bahwa itu adalah sepatu semata kaki. (Lihat citra raster yang ditunjukkan pada Gambar 3.21.) BI dapat memiliki solusi DNN yang tertanam dalam tumpukan implementasinya untuk mencapai hasil yang kompetitif. Misalnya, untuk aplikasi berbasis penandaan gambar dan pengenalan gambar, model CNN dapat dimanfaatkan secara optimal.

3.6 RINGKASAN

Bab ini berfokus pada pembahasan empat kasus penggunaan solusi berbasis AI dan ML dari perspektif bisnis dan perspektif AI bertemu BI. Kasus penggunaan dunia nyata yang dibahas adalah mengklasifikasikan Komoditas Kelas Layak Jual Berdasarkan Atributnya (solusi analitik deskriptif), Memprediksi Harga Komoditas di Muka (solusi analitik prediktif), Sistem Rekomendasi untuk Menyarankan Pilihan Optimal Berdasarkan Skor (solusi analitik preskriptif dan prediktif), dan Pengenalan Gambar Otomatis (solusi analitik preskriptif dan augmented). Deskripsi setiap kasus penggunaan disajikan dengan visualisasi yang menggambarkan dan membantu dalam analisis dan memperoleh analitik yang berguna untuk analis bisnis, ilmuwan data, dan pengguna akhir.

Dalam kasus tertentu, metrik yang berpusat pada kinerja dan membantu dalam pengambilan keputusan juga dibahas. Bab berikut merinci spesifikasi implementasi dari solusi kasus penggunaan ini menggunakan model berbasis AI dan ML, yang berdasarkan klasifikasi menggunakan hutan acak dan pohon keputusan, dan jaringan saraf berbasis pembelajaran mendalam. Ini menunjukkan bagaimana solusi AI dapat diimplementasikan menggunakan model pembelajaran mesin dan kode R bersama dengan bagaimana pengambilan keputusan dapat dicapai untuk tujuan BI berdasarkan hasil yang diperoleh.



BAB 4

KASUS PENGGUNAAN ENTERPRISE BI DENGAN AI

4.1 PENDAHULUAN

Sejak AI dan solusi berbasis AI, termasuk untuk kecerdasan bisnis, dipraktikkan, telah banyak kasus penggunaan di dunia nyata yang menunjukkan implementasinya. Praktisi AI dan, baru-baru ini, ilmuwan data telah menggunakan solusi pembelajaran mesin dan pembelajaran mendalam berbasis AI untuk mengimplementasikan solusi yang telah mencapai hasil tak terbayangkan di dunia nyata dan enterprise BI. Bab ini menjelaskan empat kasus penggunaan tersebut, sebagaimana diuraikan dalam Bab 3, serta implementasinya dan bagaimana hal itu dapat membantu dalam enterprise BI (termasuk pengambilan keputusan cerdas). Implementasi ini menggunakan kode R dan berbagai algoritma serta model pembelajaran mesin untuk menghasilkan solusi yang diinginkan dan perannya dalam proses BI, menggunakan beberapa grafik dan kurva kinerja.

4.2 KLASIFIKASI KUALITAS KOMODITAS LAYAK JUAL BERDASARKAN ATRIBUT

Bagian ini menjelaskan algoritma hutan acak dan kinerja klasifikasi relatifnya dengan pohon keputusan menggunakan dataset tidak seimbang berdasarkan metrik seperti Area Under (ROC) Curve (AUC), pengukuran Kappa, dan statistik F1, serta mengapa metrik akurasi tidak sepenting pengukuran lain untuk dataset tidak seimbang.

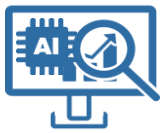
Metodologi AI Analisis Deskriptif Menggunakan Algoritma Pembelajaran Mesin Hutan Acak dan Membandingkannya dengan Pohon Keputusan

Tujuan Kasus Penggunaan

- Sasaran
Untuk mengklasifikasikan kualitas komoditas (anggur) berdasarkan atributnya. Hal ini memungkinkan pengambilan keputusan, dari aspek kualitatif, terkait penjualan anggur.
- Teknologi
AI dan ML menggunakan hutan acak dan pohon keputusan serta membandingkan kinerja klasifikasi relatifnya.
- Implementasi
Menggunakan *rf* dan *rpart of train()* dari pustaka pembelajaran mesin di R.
- Set Data Input
Set data kualitas anggur putih dalam berkas .csv yang berisi kualitas dan 11 variabel lain yang terkait sebagai kolom.

Pengantar Singkat Hutan Acak dan Pohon Keputusan Berbasis Pembelajaran Mesin

Hutan acak adalah kombinasi pohon keputusan tunggal berdasarkan sampel acak data pelatihan. Ini juga disebut ansambel pohon keputusan. Subset sampel yang digunakan untuk setiap pohon keputusan berbeda dan dipilih secara acak. Jumlah total pohon dapat ditentukan pada waktu pembuatan model. Pohon-pohon tersebut kemudian diagregasi untuk menghasilkan keputusan klasifikasi berdasarkan fitur terpenting yang difaktorkan ke dalam



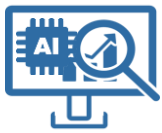
model. Hutan acak menunjukkan akurasi yang lebih tinggi dibandingkan dengan pohon keputusan.

Tabel 4.1 merangkum konsep dasar hutan acak dan pohon keputusan. Untuk menghindari overfitting, kumpulan fitur yang tepat perlu dipilih untuk membangun model klasifikasi menggunakan hutan acak. Jumlah fitur yang digunakan untuk menghasilkan setiap pohon juga penting. Fitur-fitur ini dipilih secara acak dengan nilai default $\sqrt{\text{jumlah total fitur}}$ (tetapi dapat ditentukan melalui parameter `mtry`). Jumlah fitur ini digunakan untuk pemisahan di setiap node.

Pengenalan yang sangat baik tentang pohon keputusan dan hutan acak untuk klasifikasi dan regresi dapat ditemukan di "Pohon Keputusan dan Hutan Acak untuk Klasifikasi dan Regresi pt.1" (<https://towardsdatascience.com/decision-trees-and-random-forests-for-classification-and-regression-pt-1-dbb65a458df>) dan "Pohon Keputusan dan Hutan Acak untuk Klasifikasi dan Regresi pt. 2" (<https://towardsdatascience.com/decision-trees-and-random-forests-for-classification-and-regression-pt-2-2b1fcd03e342>). Selain itu, penting untuk melakukan pra-proses data, terutama ketika dataset yang digunakan tidak seimbang dengan referensi ke kelas-kelas yang terlibat dalam model klasifikasi. Pengantar yang baik untuk hal ini dapat ditemukan di "Menangani Data yang Tidak Seimbang dalam Pembelajaran Mesin" (https://shiring.github.io/machine_learning/2017/04/02/unbalanced).

Tabel 4.1 Hutan Acak versus Pohon Keputusan

Hutan Acak	Pohon Keputusan
Sekumpulan pohon keputusan.	Pohon keputusan tunggal.
Dapat digunakan untuk klasifikasi dan regresi.	Dapat digunakan untuk klasifikasi dan regresi.
Mudah diinterpretasikan.	Mudah diinterpretasikan.
Tidak memerlukan penskalaan fitur dan pengodean fitur kategoris, seperti pengodean one-hot, jika dibandingkan dengan jaringan saraf tiruan.	Tidak memerlukan penskalaan fitur dan pengkodean fitur kategoris, seperti pengkodean one-hot, jika dibandingkan dengan jaringan saraf tiruan.
Tunjukkan fitur apa saja yang berkontribusi terhadap pemilihan variabel dan tingkat kepentingan relatifnya berdasarkan lokasi kedalaman di pohon (menggunakan plot tingkat kepentingan variabel).	Tunjukkan fitur apa saja yang berkontribusi pada pemilihan variabel dan tingkat kepentingan relatifnya berdasarkan lokasi kedalaman di pohon.
Pohon RF tidak dipangkas dan mempelajari sampel acak data pelatihan untuk setiap pohon keputusan.	Pohon keputusan tunggal dipangkas pada data pelatihan.
Pamerkan bagging fitur dengan membagi pada setiap simpul pohon menggunakan serangkaian variabel fitur acak untuk mencapai celah optimal, sehingga meningkatkan akurasi. Hal ini memungkinkan peningkatan akurasi dalam properti pemilihan variabel dibandingkan dengan pohon keputusan. Hal ini dapat	Dipisahkan di setiap simpul menggunakan satu variabel fitur pada setiap pemisahan. Hal ini biasanya menggunakan variabel pertama untuk pemisahan, sehingga batas keputusan seringkali bergantung pada urutan fitur dalam set pelatihan.



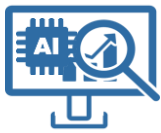
menghasilkan kelas-kelas yang dapat dipisahkan untuk klasifikasi, karena interval keputusan akan bervariasi.	
Lebih kecil kemungkinannya mengalami overfitting, karena interval keputusan seringkali simetris terhadap kelas-kelas ketika pohon-pohon tersebut digabungkan.	Overfitting terkadang dapat terjadi karena urutan fitur dalam pohon tunggal memengaruhi tugas klasifikasi atau regresi yang sedang dilakukan.
Karena alasan-alasan di atas, akurasi keputusan lebih baik dibandingkan dengan pohon keputusan, terutama ketika dataset besar dan jumlah fiturnya juga besar.	Masih cukup akurat dan baik untuk tugas klasifikasi dan regresi, terutama ketika set datanya kecil.

Untuk membangun model optimal dan menentukan kinerja hutan acak, langkah-langkah berikut dapat digunakan:

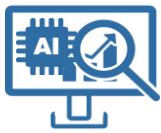
1. Gunakan kombinasi sejumlah pohon dan pemilihan fitur berdasarkan fitur terpenting untuk meminimalkan estimasi tingkat kesalahan *Out-of-Bag* (OOB), seperti yang ditunjukkan dengan mencetak model setelah dibangun. Hal ini dapat dilakukan dengan terlebih dahulu menentukan nilai optimal jumlah pohon (seperti yang diberikan oleh keluaran fungsi `trainControl()` pada set pelatihan untuk melakukan validasi silang guna menentukan nilai optimal parameter model) dan menggunakannya sebagai nilai untuk parameter `trControl()` pada fungsi `train()` beserta parameter `method = "rf"` untuk membangun hutan acak. Fungsi `trainControl()` memberikan nilai untuk `mtry` yang digunakan dalam validasi silang ini dan dapat digunakan untuk melanjutkan pembangunan model menggunakan pengklasifikasi `randomForest()` (dalam R).
2. Gunakan matriks konfusi untuk menentukan kinerja model RF dengan melihat jumlah kesalahan klasifikasi.
3. Gunakan metrik AUC untuk menentukan kinerja model RF dengan melihat kurva ROC yang diplot dan nilai Kappa, bukan nilai metrik akurasi. Pengukuran Kappa memberikan nilai yang lebih pragmatis dalam kasus dataset yang tidak seimbang.

Hal ini tercermin dalam pragmatik implementasi model yang dijelaskan secara rinci di subbagian berikutnya.

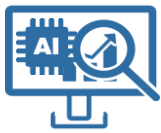
Pragmatik Implementasi Implementasi Hutan Acak

**Daftar 4.1** Membangun dan Menguji Model Hutan Acak untuk Klasifikasi

```
setwd("<directory-name>")
library(caret)
library(e1071)
# The file winequality-white.csv is in the working directory
winedata1 <- read.csv("winequality-white.csv", header=TRUE, sep=',';')
head(winedata1)
class(winedata1$quality)
anyNA(winedata1)
# Mapping Wine Quality to Saleable Grade
# <=3 NotAccept
# 4 LessThanAvg
# 5 Avg
# 6 MoreThanAvg
# 7 Good
# 8 Great
# >=9 Excellent
winedata1$grade <- ifelse(winedata1$quality <= 3, 'NotAccept',
                          ifelse(winedata1$quality == 4, 'LessThanAvg',
                                  ifelse(winedata1$quality == 5, 'Average',
                                          ifelse(winedata1$quality == 6, 'MoreThanAvg',
                                                  ifelse(winedata1$quality == 7, 'Good',
                                                          ifelse(winedata1$quality == 8, 'Great',
                                                                  ifelse(winedata1$quality >= 9, 'Excellent',
                                                                          '')))))))))
winedata1$grade <- as.factor(winedata1$grade)
set.seed(12345)
sample_size <- 0.70 * nrow(winedata1)
sampledata1 <- sample(seq_len(nrow(winedata1)), sample_size)
training_data1 <- winedata1[sampledata1, ]
test_data1 <- winedata1[-sampledata1, ]
nrow(training_data1) # 3428
nrow(test_data1) # 1470
training_data1 <- training_data1[,-12]
summary(training_data1$grade)
summary(test_data1$grade)
summary(winedata1$grade)
head(training_data1)
trctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 10,
savePredictions = TRUE, verboseIter = FALSE)
set.seed(12345)
```



```
model_rf <- train(grade ~ ., data = training_data1, method = "rf", preProcess =
c("scale", "center"), trControl = trctrl)
print(model_rf)
model_rf$finalModel$confusion
# Feature importance
imp2 <- model_rf$finalModel$importance
dim(imp2)
class(imp2)
imp2
imp2[order(imp2, decreasing = TRUE), ]
# estimate variable importance
importance2 <- varImp(model_rf, scale = TRUE)
plot(importance2)
# plot(imp2[order(imp2, decreasing = TRUE), ])
# predicting on test data
head(test_data1)
test_data_new1 <- test_data1[,-12]
head(test_data_new1)
test_data_new1 <- test_data_new1[,-12]
head(test_data_new1)
head(training_data1)
model_rf_pred <- predict(model_rf, test_data_new1)
cm_model_rf <- confusionMatrix(model_rf_pred, test_data1$grade)
cm_model_rf
# Measure model performance with ROC and AUC
library(ROCR)
# Calculate the probability of new observations belonging to each class
# grade_pred_ROC_AUC will be a matrix with dimensions (data_set_size x number_of_classes)
grade_pred_ROC_AUC <- predict(model_rf, test_data_new1, type="prob")
# Use colors for roc curves for the classes (one for each grade measure):
colors <- c("green", "orange", "purple", "yellow", "blue", "black", "red")
# Specify the different classes
ROC_AUC_classes <- levels(test_data1$grade)
# For each class
for (i in 1:7)
{
  # Define observations belonging to each class i.e., class[i]
  Actualgrade_values <- ifelse(test_data1$grade==ROC_AUC_classes[i],1,0)
  # Assess the performance of classifier for class[i]
  pred <- prediction(grade_pred_ROC_AUC[,i], Actualgrade_values)
  perf <- performance(pred, "tpr", "fpr")
  pdf(paste("Random_Forest_ROC_Curve_for_", ROC_AUC_classes[i], ".pdf"))
  plot(perf, main=paste("Random Forest ROC Curve for ",
ROC_AUC_classes[i]), col=colors[i])
  dev.off()
  # Calculate the AUC and print it to screen
```



```
auc.perf <- performance(pred, measure = "auc")
print(paste("Random_Forest_AUC for ",ROC_AUC_classes[i],":",auc.perf@y.values))
}
results_rf <- data.frame(Actual = test_data1$grade, Predicted = predict(model_rf,
test_data_new1, type = "prob"))
head(results_rf)
results_rf$predicted_grade <- ifelse(results_rf$Predicted.Average > 0.5, "Average",
  ifelse(results_rf$Predicted.Excellent > 0.5, "Excellent",
    ifelse(results_rf$Predicted.Good > 0.5, "Good",
      ifelse(results_rf$Predicted.Great > 0.5, "Great",
        ifelse(results_rf$Predicted.LessThanAvg > 0.5, "LessThanAvg",
          ifelse(results_rf$Predicted.MoreThanAvg > 0.5, "MoreThanAvg",
            ifelse(results_rf$Predicted.NotAccept > 0.5, "NotAccept",
              ')))))))))
results_rf$predicted_grade <- factor(results_rf$predicted_grade,
levels=levels(results_rf$Actual))
cm_original <- confusionMatrix(results_rf$predicted_grade, test_data1$grade)
cm_original
pairs(table(model_rf_pred, test_data1$grade), main="Wine Quality Classification")
```

Daftar 4.2 Keluaran Hutan Acak (dari kode dalam Daftar 4.1)

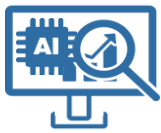
```
> setwd("<directory-name>")
> library(caret)
Loading required package: lattice
Loading required package: ggplot2

Attaching package: 'ggplot2'

The following object is masked from 'package:randomForest':

  margin

Warning messages:
1: package 'caret' was built under R version 3.4.4
2: package 'ggplot2' was built under R version 3.4.4
3: As of rlang 0.4.0, dplyr must be at least version 0.8.0.
x dplyr 0.7.8 is too old for rlang 0.4.2.
i Please update dplyr to the latest version.
i Updating packages on Windows requires precautions:
  <https://github.com/jennybc/what-they-forgot/issues/62>
> library(e1071)
Warning message:
package 'e1071' was built under R version 3.4.4
> # The file winequality-white.csv is in the working directory
> winedata1 <- read.csv("winequality-white.csv", header=TRUE, sep=';')
> head(winedata1)
```



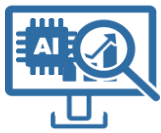
```
fixed.acidity volatile.acidity citric.acid residual.sugar chlorides free.sulfur.dioxide
1          7.0          0.27          0.36          20.7          0.045          45
2          6.3          0.30          0.34          1.6          0.049          14
3          8.1          0.28          0.40          6.9          0.050          30
4          7.2          0.23          0.32          8.5          0.058          47
5          7.2          0.23          0.32          8.5          0.058          47
6          8.1          0.28          0.40          6.9          0.050          30

total.sulfur.dioxide density    pH sulphates alcohol quality
1          170  1.0010  3.00      0.45    8.8      6
2          132  0.9940  3.30      0.49    9.5      6
3           97  0.9951  3.26      0.44   10.1     6
4          186  0.9956  3.19      0.40    9.9      6
5          186  0.9956  3.19      0.40    9.9      6
6           97  0.9951  3.26      0.44   10.1     6

> class(winedata1$quality)
[1] "integer"
> anyNA(winedata1)
[1] FALSE

      > # Mapping Wine Quality to Saleable Grade
      > # <=3 NotAccept
      > # 4 LessThanAvg
      > # 5 Avg
      > # 6 MoreThanAvg
      > # 7 Good
      > # 8 Great
      > # >=9 Excellent

> wine_data1$grade <- ifelse(wine_data1$quality <= 3, 'NotAccept',
+                             ifelse(wine_data1$quality == 4, 'LessThanAvg',
+                                     ifelse(wine_data1$quality == 5, 'Average',
+                                             ifelse(wine_data1$quality == 6, 'MoreThanAvg',
+                                                   ifelse(wine_data1$quality == 7, 'Good',
+                                                         ifelse(wine_data1$quality == 8, 'Great',
+                                                               ifelse(wine_data1$quality >= 9, 'Excellent',
+                                                                     ')))))))))
> wine_data1$grade <- as.factor(wine_data1$grade)
> set.seed(12345)
> sample_size <- 0.70 * nrow(wine_data1)
> sampled_data1 <- sample(seq_len(nrow(wine_data1)), sample_size)
> training_data1 <- wine_data1[sampled_data1, ]
> test_data1 <- wine_data1[-sampled_data1, ]
> nrow(training_data1) # 3428
[1] 3428
> nrow(test_data1) # 1470
[1] 1470
> training_data1 <- training_data1[,-12]
> summary(training_data1$grade)
```



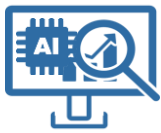
```
Average    Excellent    Good    Great    LessThanAvg    MoreThanAvg    NotAccept
      1014            4      618      126        110        1541          15
> summary(test_data1$grade)
  Average    Excellent    Good    Great    LessThanAvg    MoreThanAvg    NotAccept
      443            1      262        49         53         657          5
> summary(winedata1$grade)
  Average    Excellent    Good    Great    LessThanAvg    MoreThanAvg    NotAccept
     1457            5      880      175        163        2198          20
> head(training_data1)
  fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
3531          7.2            0.31         0.35           7.2         0.046
4289          7.0            0.20         0.33           4.7         0.030
3726          6.6            0.26         0.28           9.4         0.028
4338          7.3            0.19         0.27          13.9         0.057
2235          7.2            0.23         0.38           6.1         0.067
815           7.9            0.35         0.24          15.6         0.072
  free.sulfur.dioxide total.sulfur.dioxide density    pH sulphates alcohol    grade
3531                45                178 0.99550 3.14    0.53    9.7    Average
4289                25                76 0.99202 2.88    0.54   10.5 MoreThanAvg
3726                13                121 0.99254 3.17    0.34   12.1 MoreThanAvg
4338                45                155 0.99807 2.94    0.41    8.8    Great
2235                20                90 0.99496 3.17    0.79    9.7    Average
815                 44                229 0.99785 3.03    0.59   10.5 MoreThanAvg
> trctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 10,
savePredictions = TRUE, verboseIter = FALSE)
> set.seed(12345)
> model_rf <- train(grade ~ ., data = training_data1, method = "rf", preProcess =
c("scale", "center"), trControl = trctrl)
> print(model_rf)
Random Forest

3428 samples
 11 predictor
 7 classes: 'Average', 'Excellent', 'Good', 'Great', 'LessThanAvg', 'MoreThanAvg',
'NotAccept'

Pre-processing: scaled (11), centered (11)
Resampling: Cross-Validated (10 fold, repeated 10 times)
Summary of sample sizes: 3084, 3085, 3086, 3088, 3082, 3086, ...
Resampling results across tuning parameters:

 mtry Accuracy    Kappa
  2  0.6617835  0.4694894
  6  0.6580509  0.4671895
 11  0.6545221  0.4625398

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 2.
> model_rf$finalModel$confusion
```



```

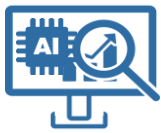
Average Excellent Good Great LessThanAvg MoreThanAvg NotAccept class.error
Average      661         0   9   0           6         338         0  0.3481262
Excellent    0         0   2   0           0          2         0  1.0000000
Good         10         0 330   2           0        276         0  0.4660194
Great        0         0  32  47           0         47         0  0.6269841
LessThanAvg  53         0   1   0          26         30         0  0.7636364
MoreThanAvg 214         0  91   0           0       1236         0  0.1979234
NotAccept    7         0   0   0           0          8         0  1.0000000

> # Feature importance
> imp2 <- model_rf$finalModel$importance
> dim(imp2)
[1] 11 1
> class(imp2)
[1] "matrix"
> imp2

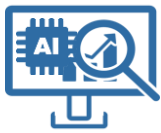
                                MeanDecreaseGini
fixed.acidity                    176.0452
volatile.acidity                  230.2272
citric.acid                       192.6936
residual.sugar                    203.8580
chlorides                         206.5411
free.sulfur.dioxide               214.6507
total.sulfur.dioxide              211.3666
density                           239.4751
pH                                198.1514
sulphates                         185.8240
alcohol                           254.5991
> imp2[order(imp2, decreasing = TRUE), ]
      alcohol      density  volatile.acidity  free.sulfur.dioxide
      254.5991      239.4751           230.2272           214.6507
total.sulfur.dioxide  chlorides  residual.sugar           pH
      211.3666      206.5411           203.8580           198.1514
      citric.acid      sulphates  fixed.acidity
      192.6936      185.8240           176.0452

> # estimate variable importance
> importance2 <- varImp(model_rf, scale = TRUE)
> plot(importance2)
> # plot(imp2[order(imp2, decreasing = TRUE), ])
> # predicting on test data
> head(test_data1)
      fixed.acidity  volatile.acidity  citric.acid  residual.sugar  chlorides  free.sulfur.dioxide
5          7.2         0.23         0.32         8.5         0.058         47
7          6.2         0.32         0.16         7.0         0.045         30
8          7.0         0.27         0.36        20.7         0.045         45
9          6.3         0.30         0.34         1.6         0.049         14
12         8.6         0.23         0.40         4.2         0.035         17
14         6.6         0.16         0.40         1.5         0.044         48

```



```
total.sulfur.dioxide density pH sulphates alcohol quality grade
5 186 0.9956 3.19 0.40 9.9 6 MoreThanAvg
7 136 0.9949 3.18 0.47 9.6 6 MoreThanAvg
8 170 1.0010 3.00 0.45 8.8 6 MoreThanAvg
9 132 0.9940 3.30 0.49 9.5 6 MoreThanAvg
12 109 0.9947 3.14 0.53 9.7 5 Average
14 143 0.9912 3.54 0.52 12.4 7 Good
> test_data_new1 <- test_data1[,-12]
> head(test_data_new1)
fixed.acidity volatile.acidity citric.acid residual.sugar chlorides free.sulfur.dioxide
5 7.2 0.23 0.32 8.5 0.058 47
7 6.2 0.32 0.16 7.0 0.045 30
8 7.0 0.27 0.36 20.7 0.045 45
9 6.3 0.30 0.34 1.6 0.049 14
12 8.6 0.23 0.40 4.2 0.035 17
14 6.6 0.16 0.40 1.5 0.044 48
total.sulfur.dioxide density pH sulphates alcohol grade
5 186 0.9956 3.19 0.40 9.9 MoreThanAvg
7 136 0.9949 3.18 0.47 9.6 MoreThanAvg
8 170 1.0010 3.00 0.45 8.8 MoreThanAvg
9 132 0.9940 3.30 0.49 9.5 MoreThanAvg
12 109 0.9947 3.14 0.53 9.7 Average
14 143 0.9912 3.54 0.52 12.4 Good
> test_data_new1 <- test_data_new1[,-12]
> head(test_data_new1)
fixed.acidity volatile.acidity citric.acid residual.sugar chlorides free.sulfur.dioxide
5 7.2 0.23 0.32 8.5 0.058 47
7 6.2 0.32 0.16 7.0 0.045 30
8 7.0 0.27 0.36 20.7 0.045 45
9 6.3 0.30 0.34 1.6 0.049 14
12 8.6 0.23 0.40 4.2 0.035 17
14 6.6 0.16 0.40 1.5 0.044 48
total.sulfur.dioxide density pH sulphates alcohol
5 186 0.9956 3.19 0.40 9.9
7 136 0.9949 3.18 0.47 9.6
8 170 1.0010 3.00 0.45 8.8
9 132 0.9940 3.30 0.49 9.5
12 109 0.9947 3.14 0.53 9.7
14 143 0.9912 3.54 0.52 12.4
> head(training_data1)
fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
3531 7.2 0.31 0.35 7.2 0.046
4289 7.0 0.20 0.33 4.7 0.030
3726 6.6 0.26 0.28 9.4 0.028
4338 7.3 0.19 0.27 13.9 0.057
2235 7.2 0.23 0.38 6.1 0.067
815 7.9 0.35 0.24 15.6 0.072
```



```

free.sulfur.dioxide total.sulfur.dioxide density  pH sulphates alcohol  grade
3531                45                    178 0.99550 3.14    0.53   9.7   Average
4289                25                    76 0.99202 2.88    0.54  10.5 MoreThanAvg
3726                13                    121 0.99254 3.17    0.34  12.1 MoreThanAvg
4338                45                    155 0.99807 2.94    0.41   8.8    Great
2235                20                    90 0.99496 3.17    0.79   9.7   Average
815                 44                    229 0.99785 3.03    0.59  10.5 MoreThanAvg

```

```

> model_rf_pred <- predict(model_rf, test_data_new1)
> cm_model_rf <- confusionMatrix(model_rf_pred, test_data1$grade)
> cm_model_rf

```

Confusion Matrix and Statistics

Prediction	Reference						
	Average	Excellent	Good	Great	LessThanAvg	MoreThanAvg	NotAccept
Average	302	0	6	0	33	79	1
Excellent	0	0	0	0	0	0	0
Good	3	0	143	13	1	42	0
Great	0	0	1	17	0	1	0
LessThanAvg	1	0	0	0	1	0	0
MoreThanAvg	137	1	112	19	18	535	4
NotAccept	0	0	0	0	0	0	0

Overall Statistics

```

Accuracy : 0.6789 # This is less reliable as dataset is imbalanced
95% CI : (0.6544, 0.7027)
No Information Rate : 0.4469
P-Value [Acc > NIR] : < 2.2e-16

```

```

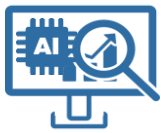
Kappa : 0.4964 # This is a more reliable performance metric
McNemar's Test P-Value : NA

```

Statistics by Class:

	Class: Average	Class: Excellent	Class: Good	Class: Great
Sensitivity	0.6817	0.0000000	0.54580	0.34694
Specificity	0.8841	1.0000000	0.95116	0.99859
Pos Pred Value	0.7173	NaN	0.70792	0.89474
Neg Pred Value	0.8656	0.9993197	0.90615	0.97795
Prevalence	0.3014	0.0006803	0.17823	0.03333
Detection Rate	0.2054	0.0000000	0.09728	0.01156
Detection Prevalence	0.2864	0.0000000	0.13741	0.01293
Balanced Accuracy	0.7829	0.5000000	0.74848	0.67277

	Class: LessThanAvg	Class: MoreThanAvg	Class: NotAccept
Sensitivity	0.0188679	0.8143	0.000000
Specificity	0.9992943	0.6421	1.000000
Pos Pred Value	0.5000000	0.6477	NaN
Neg Pred Value	0.9645777	0.8106	0.996599
Prevalence	0.0360544	0.4469	0.003401



```

Detection Rate          0.0006803          0.3639          0.000000
Detection Prevalence    0.0013605          0.5619          0.000000
Balanced Accuracy       0.5090811          0.7282          0.500000
> results_rf <- data.frame(Actual = test_data1$grade, Predicted = predict(model_rf,
test_data_new1, type = "prob"))
> head(results_rf)
      Actual Predicted.Average Predicted.Excellent Predicted.Good Predicted.Great
5  MoreThanAvg          0.142             0.000          0.024          0.004
7  MoreThanAvg          0.782             0.000          0.012          0.002
8  MoreThanAvg          0.232             0.000          0.038          0.006
9  MoreThanAvg          0.252             0.000          0.012          0.002
12 Average              0.188             0.000          0.066          0.006
14      Good           0.134             0.002          0.304          0.214
      Predicted.LessThanAvg Predicted.MoreThanAvg Predicted.NotAccept
5              0.004             0.826             0.000
7              0.012             0.190             0.002
8              0.002             0.722             0.000
9              0.032             0.702             0.000
12             0.048             0.662             0.030
14             0.000             0.344             0.002
> results_rf$predicted_grade <- ifelse(results_rf$Predicted.Average > 0.5, "Average",
+   ifelse(results_rf$Predicted.Excellent > 0.5, "Excellent",
+   ifelse(results_rf$Predicted.Good > 0.5, "Good",
+   ifelse(results_rf$Predicted.Great > 0.5, "Great",
+   ifelse(results_rf$Predicted.LessThanAvg > 0.5, "LessThanAvg",
+   ifelse(results_rf$Predicted.MoreThanAvg > 0.5, "MoreThanAvg",
+   ifelse(results_rf$Predicted.NotAccept > 0.5, "NotAccept",
+   '))))))
> results_rf$predicted_grade <- factor(results_rf$predicted_grade,
levels=levels(results_rf$Actual))
> cm_original <- confusionMatrix(results_rf$predicted_grade, test_data1$grade)
> cm_original

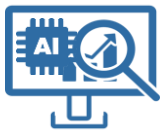
```

Confusion Matrix and Statistics

Prediction	Reference						
	Average	Excellent	Good	Great	LessThanAvg	MoreThanAvg	NotAccept
Average	261	0	2	0	23	52	0
Excellent	0	0	0	0	0	0	0
Good	1	0	115	5	0	18	0
Great	0	0	0	16	0	1	0
LessThanAvg	0	0	0	0	1	0	0
MoreThanAvg	88	0	63	7	7	405	2
NotAccept	0	0	0	0	0	0	0

Overall Statistics

Accuracy : 0.7479
95% CI : (0.7207, 0.7737)



```
No Information Rate : 0.4461
P-Value [Acc > NIR] : < 2.2e-16
```

```
Kappa : 0.6027
```

```
McNemar's Test P-Value : NA
```

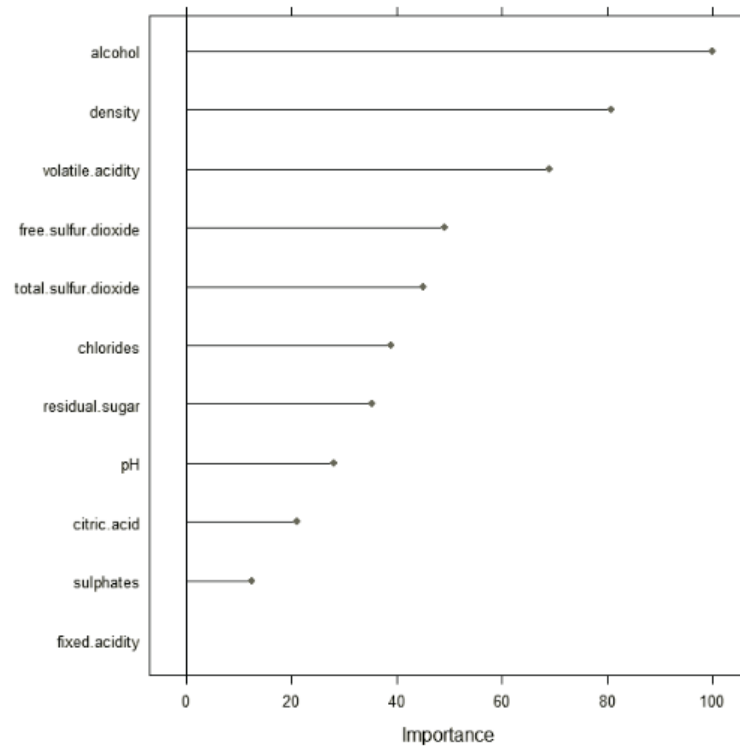
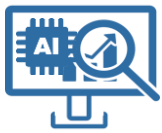
```
Statistics by Class:
```

```
                Class: Average Class: Excellent Class: Good Class: Great
Sensitivity          0.7457                NA      0.6389      0.57143
Specificity          0.8926                 1      0.9729      0.99904
Pos Pred Value       0.7722                NA      0.8273      0.94118
Neg Pred Value       0.8779                NA      0.9300      0.98857
Prevalence           0.3280                 0      0.1687      0.02624
Detection Rate       0.2446                 0      0.1078      0.01500
Detection Prevalence 0.3168                 0      0.1303      0.01593
Balanced Accuracy    0.8192                NA      0.8059      0.78523

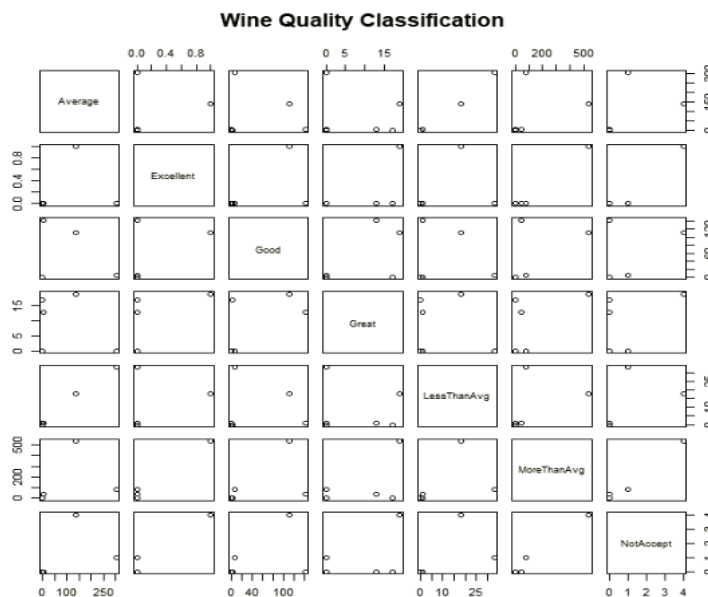
                Class: LessThanAvg Class: MoreThanAvg Class: NotAccept
Sensitivity          0.0322581            0.8508            0.000000
Specificity          1.0000000            0.7174            1.000000
Pos Pred Value       1.0000000            0.7080                NaN
Neg Pred Value       0.9718574            0.8566            0.998126
Prevalence           0.0290534            0.4461            0.001874
Detection Rate       0.0009372            0.3796            0.000000
Detection Prevalence 0.0009372            0.5361            0.000000
Balanced Accuracy    0.5161290            0.7841            0.500000
```

```
> pairs(table(model_rf_pred, test_data1$grade), main="Wine Quality Classification")
>
```

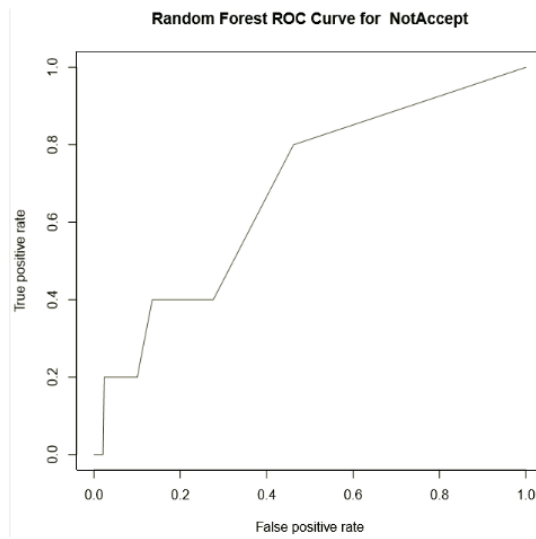
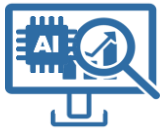
Plot kepentingan variabel, plot pasangan, dan plot ROC untuk mutu anggur dari implementasi hutan acak klasifikasi mutu anggur yang dapat dijual ditunjukkan pada Gambar 4.1–4.9.



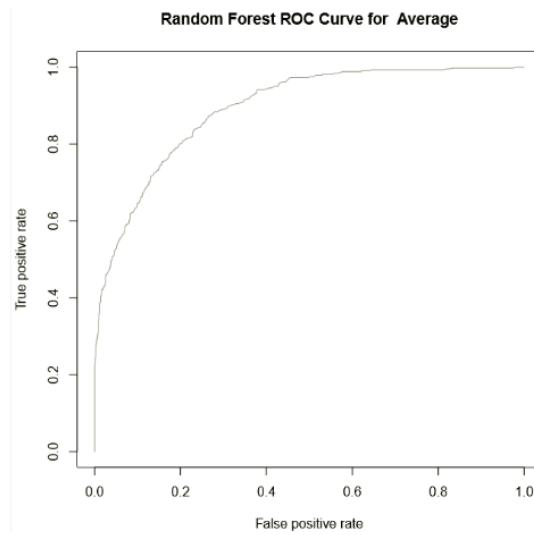
Gambar 4.1 Plot kepentingan variabel dari hutan acak. Plot kepentingan menunjukkan kontribusi fitur tertentu terhadap klasifikasi model. Plot ini menunjukkan dampak fitur terhadap akurasi model, sehingga memungkinkan model untuk mengeliminasi (atau mempertahankan) fitur dengan dampak paling kecil (atau paling besar) (juga disebut pemilihan fitur). Hal ini mengurangi bias yang salah yang seharusnya muncul pada fitur dengan dampak paling kecil.



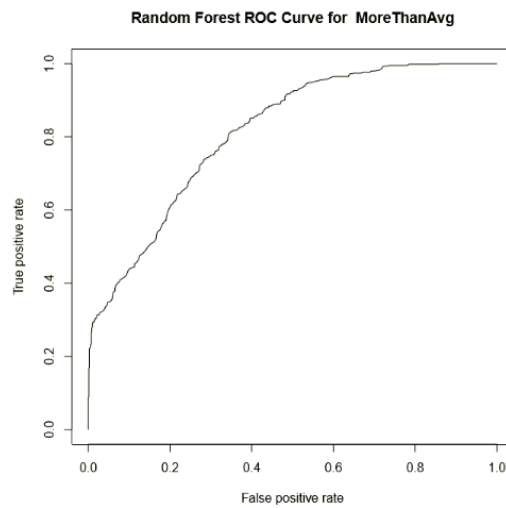
Gambar 4.2 Plot berpasangan Klasifikasi Mutu Anggur menggunakan Random Forest.



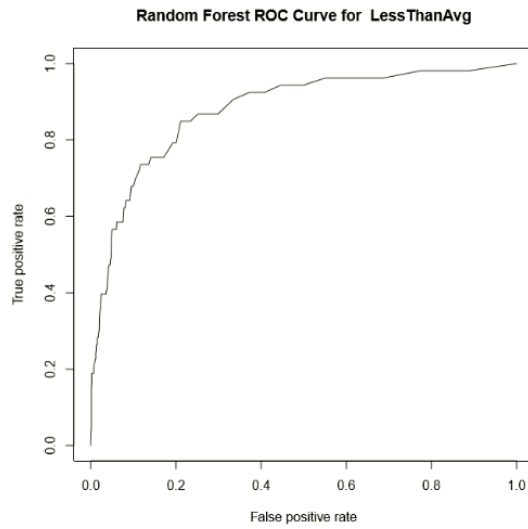
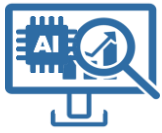
Gambar 4.3 Kurva ROC Hutan Acak untuk NotAccept.



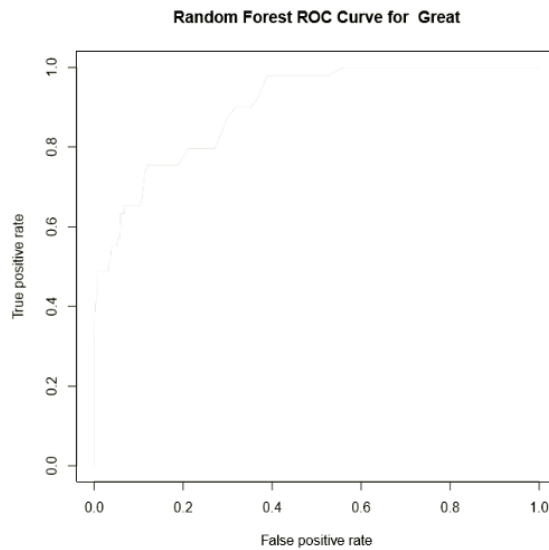
Gambar 4.4 Kurva ROC Hutan Acak untuk Rata-rata.



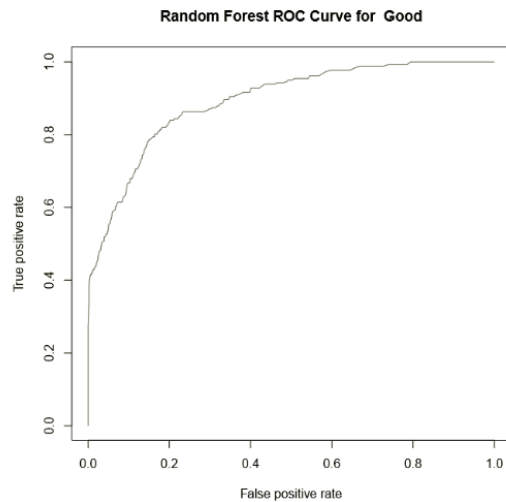
Gambar 4.5 Kurva ROC Hutan Acak untuk MoreThanAvg.



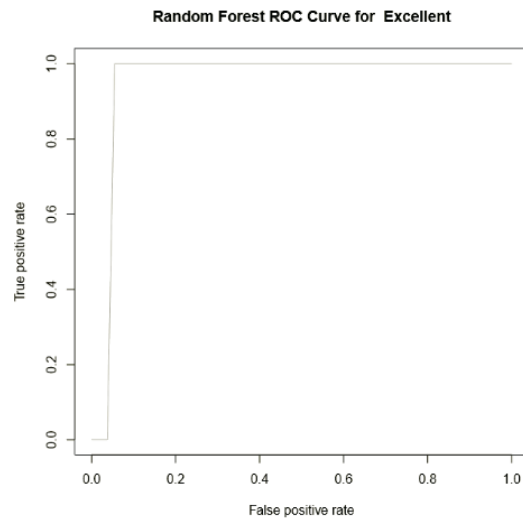
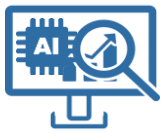
Gambar 4.6 Kurva ROC Hutan Acak untuk LessThanAvg.



Gambar 4.7 Kurva ROC Hutan Acak untuk Hebat.



Gambar 4.8 Kurva ROC Hutan Acak untuk Baik.

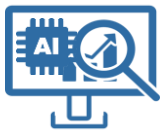


Gambar 4.9 Kurva ROC Hutan Acak untuk Unggul.

Perbandingan dengan Pohon Keputusan

Daftar 4.3 Membangun dan Menguji Model Pohon Keputusan untuk Klasifikasi

```
setwd("<directory-name>")
library(caret)
library(e1071)
library(rpart.plot)
# The file winequality-white.csv is in the working directory
winedata1 <- read.csv("winequality-white.csv", header=TRUE, sep=';')
head(winedata1)
class(winedata1$quality)
anyNA(winedata1)
# Mapping Wine Quality to Saleable Grade
# <=3 NotAccept
# 4 LessThanAvg
# 5 Avg
```

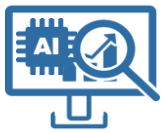


```
# 6 MoreThanAvg
# 7 Good
# 8 Great
# >=9 Excellent
winedata1$grade <- ifelse(winedata1$quality <= 3, 'NotAccept',
  ifelse(winedata1$quality == 4, 'LessThanAvg',
  ifelse(winedata1$quality == 5, 'Average',
  ifelse(winedata1$quality == 6, 'MoreThanAvg',
  ifelse(winedata1$quality == 7, 'Good',
  ifelse(winedata1$quality == 8, 'Great',
  ifelse(winedata1$quality >= 9, 'Excellent',
  ''))))))

winedata1$grade <- as.factor(winedata1$grade)
set.seed(12345)
sample_size <- 0.70 * nrow(winedata1)
sampledata1 <- sample(seq_len(nrow(winedata1)), sample_size)
training_data1 <- wineData1[sampledata1, ]
test_data1 <- wineData1[-sampledata1, ]
nrow(training_data1) # 3428
nrow(test_data1) # 1470
training_data1 <- training_data1[,-12]
summary(training_data1$grade)
summary(test_data1$grade)
summary(winedata1$grade)
head(training_data1)

trctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 10,
  savePredictions = TRUE, verboseIter = FALSE)
set.seed(12345)
#model_rf <- train(grade ~ ., data = training_data1, method = "rf", preProcess =
  c("scale", "center"), trControl = trctrl)
wine.dTree <- train(grade ~., data = training_data1, method = "rpart",
  parms = list(split = "information"),
  trControl=trctrl,
  tuneLength = 10)

# As an alternative to above call, the below statement using the rpart() can be run
# wine.dTree <- rpart(formula, data = training_data, method="class")
#summary(fit)
wine.dTree
```



```
prp(wine.dTree$finalModel) # Print Primary decision tree structure
head(test_data1)
test_data_new1 <- test_data1[,-12]
head(test_data_new1)
test_data_new1 <- test_data_new1[,-12]
head(test_data_new1)
head(training_data1)

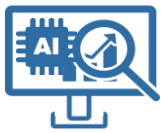
test_pred <- predict(wine.dTree, newdata = test_data_new1)
cm <- confusionMatrix(test_pred, test_data1$grade ) #check accuracy
cm

#Training the Decision Tree Classifier with gini as criterion
set.seed(12345)
wine.dTree_gini <- train(grade ~., data = training_data1, method = "rpart",
                        parms = list(split = "gini"),
                        trControl=trctrl, tuneLength = 10)

wine.dTree_gini
prp(wine.dTree_gini$finalModel)
test_pred_gini <- predict(wine.dTree_gini, newdata = test_data_new1)
cm_gini <- confusionMatrix(test_pred_gini, test_data1$grade) #check accuracy
cm_gini
table(test_pred, test_data1$grade)
table(test_pred_gini, test_data1$grade)

# Measure model performance with AUC
library(ROCR)

# Calculate the probability of new observations belonging to each class
# Source_pred_ROC_AUC will be a matrix with dimensions (data_set_size x
number_of_classes)
grade_pred_ROC_AUC_dtree <- predict(wine.dTree, test_data_new1, type="prob")
# Use colors for roc curves for the three classes (one for each Source):
colors <- c("green", "orange", "purple", "yellow", "blue", "black", "red")
# Specify the different classes
ROC_AUC_classes <- levels(test_data1$grade)
# For each class
for (i in 1:7)
{
  # Define observations belonging to each class i.e., class[i]
  Actualgrade_values <- ifelse(test_data1$grade==ROC_AUC_classes[i],1,0)
  # Assess the performance of classifier for class[i]
  pred <- prediction(grade_pred_ROC_AUC_dtree[,i], Actualgrade_values)
  perf <- performance(pred, "tpr", "fpr")
}
```



```
pdf(paste("Decision_Tree_v7_ROC_Curve_for_",ROC_AUC_classes[i],".pdf"))
plot(perf,main=paste("Decision Tree ROC Curve for ",
ROC_AUC_classes[i]),col=colors[i])
dev.off()
# Calculate the AUC and print it to screen
auc.perf <- performance(pred, measure = "auc")
print(paste("Decision Tree AUC for ",ROC_AUC_classes[i],":",auc.perf@y.values))
}

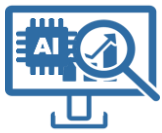
results_dTree_gini <- data.frame(Actual = test_data1$grade, Predicted = predict(wine.
dTree_gini, test_data_new1, type = "prob"))
head(results_dTree_gini)

results_dTree_gini$predicted_grade <- ifelse(results_dTree_gini$Predicted.Average >
0.5, "Average",
      ifelse(results_dTree_gini$Predicted.Excellent > 0.5, "Excellent",
            ifelse(results_dTree_gini$Predicted.Good > 0.5, "Good",
                  ifelse(results_dTree_gini$Predicted.Great > 0.5, "Great",
                        ifelse(results_dTree_gini$Predicted.LessThanAvg > 0.5, "LessThanAvg",
                              ifelse(results_dTree_gini$Predicted.MoreThanAvg > 0.5, "MoreThanAvg",
                                    ifelse(results_dTree_gini$Predicted.NotAccept > 0.5, "NotAccept",
                                            ')))))))))
results_dTree_gini$predicted_grade <- factor(results_dTree_gini$predicted_grade,
levels=levels(results_dTree_gini$Actual))
cm_original_dtree_gini <- confusionMatrix(results_dTree_gini$predicted_grade,
test_data1$grade)
cm_original_dtree_gini

pairs(table(test_pred_gini, test_data1$grade), main="Wine Quality Classification
using Decision Tree")
```

Daftar 4.4 Keluaran Pohon Keputusan (dari kode pada Daftar 4.3)

```
# Output of above code
> # decision trees v7 regular
> setwd("<directory-name>")
> library(caret)
Loading required package: lattice
Loading required package: ggplot2
Warning messages:
1: package 'caret' was built under R version 3.4.4
2: package 'ggplot2' was built under R version 3.4.4
3: As of rlang 0.4.0, dplyr must be at least version 0.8.0.
x dplyr 0.7.8 is too old for rlang 0.4.2.
i Please update dplyr to the latest version.
i Updating packages on Windows requires precautions:
  <https://github.com/jennybc/what-they-forgot/issues/62>
> library(e1071)
```



Warning message:

```
package 'e1071' was built under R version 3.4.4
```

```
> library(rpart.plot)
```

```
Loading required package: rpart
```

```
> # The file winequality-white.csv is in the working directory
```

```
> winedata1 <- read.csv("winequality-white.csv", header=TRUE, sep=';')
```

```
> head(winedata1)
```

	fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides
1	7.0	0.27	0.36	20.7	0.045
2	6.3	0.30	0.34	1.6	0.049
3	8.1	0.28	0.40	6.9	0.050
4	7.2	0.23	0.32	8.5	0.058
5	7.2	0.23	0.32	8.5	0.058
6	8.1	0.28	0.40	6.9	0.050

	free.sulfur.dioxide	total.sulfur.dioxide	density	pH	sulphates	alcohol
1	45	170	1.0010	3.00	0.45	8.8
2	14	132	0.9940	3.30	0.49	9.5
3	30	97	0.9951	3.26	0.44	10.1
4	47	186	0.9956	3.19	0.40	9.9
5	47	186	0.9956	3.19	0.40	9.9
6	30	97	0.9951	3.26	0.44	10.1

```
quality
```

1	6
2	6
3	6
4	6
5	6
6	6

```
> class(winedata1$quality)
```

```
[1] "integer"
```

```
> anyNA(winedata1)
```

```
[1] FALSE
```

```
> # Mapping Wine Quality to Saleable Grade
```

```
> # <=3 NotAccept
```

```
> # 4 LessThanAvg
```

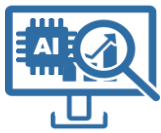
```
> # 5 Avg
```

```
> # 6 MoreThanAvg
```

```
> # 7 Good
```

```
> # 8 Great
```

```
> # >=9 Excellent
```



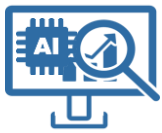
```
> winedata1$grade <- ifelse(winedata1$quality <= 3, 'NotAccept',
+                           ifelse(winedata1$quality == 4, 'LessThanAvg',
+                                   ifelse(winedata1$quality == 5, 'Average',
+                                           ifelse(winedata1$quality == 6, 'MoreThanAvg',
+                                                   ifelse(winedata1$quality == 7, 'Good',
+                                                         ifelse(winedata1$quality == 8, 'Great',
+                                                                 ifelse(winedata1$quality >= 9, 'Excellent',
+                                                                                               ')))))))))
>
>
> winedata1$grade <- as.factor(winedata1$grade)
> set.seed(12345)
> sample_size <- 0.70 * nrow(winedata1)
> sampledatal <- sample(seq_len(nrow(winedata1)), sample_size)
> training_data1 <- winedata1[sampledatal, ]
> test_data1 <- winedata1[-sampledatal, ]
> nrow(training_data1) # 3428
[1] 3428
> nrow(test_data1) # 1470
[1] 1470

> training_data1 <- training_data1[,-12]
> summary(training_data1$grade)
  Average   Excellent     Good     Great LessThanAvg MoreThanAvg
    1014         4       618     126       110       1541
NotAccept
    15

> summary(test_data1$grade)
  Average   Excellent     Good     Great LessThanAvg MoreThanAvg
    443         1       262     49       53       657
NotAccept
    5

> summary(winedata1$grade)
  Average   Excellent     Good     Great LessThanAvg MoreThanAvg
    1457         5       880     175       163       2198
NotAccept
    20

> head(training_data1)
```



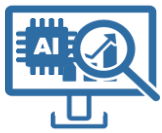
```
fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
3531          7.2             0.31         0.35           7.2         0.046
4289          7.0             0.20         0.33           4.7         0.030
3726          6.6             0.26         0.28           9.4         0.028
4338          7.3             0.19         0.27           13.9        0.057
2235          7.2             0.23         0.38           6.1         0.067
815           7.9             0.35         0.24           15.6        0.072

free.sulfur.dioxide total.sulfur.dioxide density  pH sulphates alcohol
3531                45                178 0.99550 3.14      0.53    9.7
4289                25                76 0.99202 2.88      0.54   10.5
3726                13                121 0.99254 3.17      0.34   12.1
4338                45                155 0.99807 2.94      0.41    8.8
2235                20                90 0.99496 3.17      0.79    9.7
815                 44                229 0.99785 3.03      0.59   10.5

grade
3531 Average
4289 MoreThanAvg
3726 MoreThanAvg
4338 Great
2235 Average
815 MoreThanAvg
>
> trctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 10,
savePredictions = TRUE, verboseIter = FALSE)
> set.seed(12345)
> #model_rf <- train(grade ~ ., data = training_data, method = "rf", preProcess =
c("scale", "center"), trControl = trctrl)
> wine.dTree <- train(grade ~., data = training_data, method = "rpart",
+                    parms = list(split = "information"),
+
+                    trControl=trctrl,
+                    tuneLength = 10)
> # As an alternative to above call, the below statement using the rpart() can be run
> # wine.dTree <- rpart(formula, data = training_data, method="class")
> #summary(fit)
> wine.dTree
CART

3428 samples
  11 predictor
   7 classes: 'Average', 'Excellent', 'Good', 'Great', 'LessThanAvg', 'MoreThanAvg',
'NotAccept'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 10 times)
Summary of sample sizes: 3084, 3085, 3086, 3088, 3082, 3086, ...
Resampling results across tuning parameters:
```



cp	Accuracy	Kappa
0.002543720	0.5345185	0.2621422
0.002649709	0.5337576	0.2601781
0.003179650	0.5285617	0.2464255
0.003577107	0.5245704	0.2364919
0.003709592	0.5242192	0.2349841
0.005299417	0.5159950	0.1961872
0.006006006	0.5173653	0.1901325
0.007065889	0.5189706	0.1873288
0.013778484	0.5139181	0.1815030
0.059618442	0.4864673	0.1175249

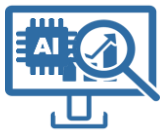
Accuracy was used to select the optimal model using the largest value. The final value used for the model was $cp = 0.00254372$.

```
> prp(wine.dTree$finalModel) # Print Primary decision tree structure
> head(test_data1)
  fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
5           7.2             0.23         0.32           8.5      0.058
7           6.2             0.32         0.16           7.0      0.045
8           7.0             0.27         0.36          20.7      0.045
9           6.3             0.30         0.34           1.6      0.049
12          8.6             0.23         0.40           4.2      0.035
14          6.6             0.16         0.40           1.5      0.044

  free.sulfur.dioxide total.sulfur.dioxide density  pH sulphates alcohol
5                   47                   186 0.9956 3.19    0.40    9.9
7                   30                   136 0.9949 3.18    0.47    9.6
8                   45                   170 1.0010 3.00    0.45    8.8
9                   14                   132 0.9940 3.30    0.49    9.5
12                  17                   109 0.9947 3.14    0.53    9.7
14                  48                   143 0.9912 3.54    0.52   12.4

  quality      grade
5       6 MoreThanAvg
7       6 MoreThanAvg
8       6 MoreThanAvg
9       6 MoreThanAvg
12      5   Average
14      7     Good

> test_data_new1 <- test_data1[,-12]
> head(test_data_new1)
```



```
fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
5          7.2          0.23          0.32          8.5          0.058
7          6.2          0.32          0.16          7.0          0.045
8          7.0          0.27          0.36          20.7         0.045
9          6.3          0.30          0.34          1.6          0.049
12         8.6          0.23          0.40          4.2          0.035
14         6.6          0.16          0.40          1.5          0.044

free.sulfur.dioxide total.sulfur.dioxide density  pH sulphates alcohol
5          47          186  0.9956  3.19          0.40          9.9
7          30          136  0.9949  3.18          0.47          9.6
8          45          170  1.0010  3.00          0.45          8.8
9          14          132  0.9940  3.30          0.49          9.5
12         17          109  0.9947  3.14          0.53          9.7
14         48          143  0.9912  3.54          0.52          12.4

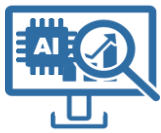
grade
5 MoreThanAvg
7 MoreThanAvg
8 MoreThanAvg
9 MoreThanAvg
12 Average
14 Good

> test_data_new1 <- test_data_new1[,-12]
> head(test_data_new1)
fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
5          7.2          0.23          0.32          8.5          0.058
7          6.2          0.32          0.16          7.0          0.045
8          7.0          0.27          0.36          20.7         0.045
9          6.3          0.30          0.34          1.6          0.049
12         8.6          0.23          0.40          4.2          0.035
14         6.6          0.16          0.40          1.5          0.044

free.sulfur.dioxide total.sulfur.dioxide density  pH sulphates alcohol
5          47          186  0.9956  3.19          0.40          9.9
7          30          136  0.9949  3.18          0.47          9.6
8          45          170  1.0010  3.00          0.45          8.8
9          14          132  0.9940  3.30          0.49          9.5
12         17          109  0.9947  3.14          0.53          9.7
14         48          143  0.9912  3.54          0.52          12.4

> test_pred <- predict(wine.dTree, newdata = test_data_new1)
> cm <- confusionMatrix(test_pred, test_data$grade ) #check accuracy
> cm

Confusion Matrix and Statistics
```



	Reference						
Prediction	Average	Excellent	Good	Great	LessThanAvg	MoreThanAvg	NotAccept
Average	206	0	10	0	31	103	1
Excellent	0	0	0	0	0	0	0
Good	3	1	68	17	0	52	0
Great	1	0	2	3	0	1	0
LessThanAvg	5	0	0	1	1	2	0
MoreThanAvg	228	0	182	28	21	499	4
NotAccept	0	0	0	0	0	0	0

Overall Statistics

Accuracy : 0.5286
 95% CI : (0.5027, 0.5544)
 No Information Rate : 0.4469
 P-Value [Acc > NIR] : 2.089e-10

Kappa : 0.2373

McNemar's Test P-Value : NA

Statistics by Class:

	Class: Average	Class: Excellent	Class: Good	Class: Great
Sensitivity	0.4650	0.0000000	0.25954	0.061224
Specificity	0.8588	1.0000000	0.93957	0.997185
Pos Pred Value	0.5869	NaN	0.48227	0.428571
Neg Pred Value	0.7882	0.9993197	0.85403	0.968558
Prevalence	0.3014	0.0006803	0.17823	0.033333
Detection Rate	0.1401	0.0000000	0.04626	0.002041
Detection Prevalence	0.2388	0.0000000	0.09592	0.004762
Balanced Accuracy	0.6619	0.5000000	0.59956	0.529205

	Class: LessThanAvg	Class: MoreThanAvg	Class: NotAccept
Sensitivity	0.0188679	0.7595	0.000000
Specificity	0.9943543	0.4305	1.000000
Pos Pred Value	0.1111111	0.5187	NaN
Neg Pred Value	0.9644079	0.6890	0.996599
Prevalence	0.0360544	0.4469	0.003401
Detection Rate	0.0006803	0.3395	0.000000
Detection Prevalence	0.0061224	0.6544	0.000000
Balanced Accuracy	0.5066111	0.5950	0.500000

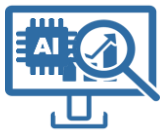
> #Training the Decision Tree Classifier with gini as criterion

> set.seed(12345)

```
> wine.dTree_gini <- train(grade ~., data = training_data1, method = "rpart",
+                           parms = list(split = "gini"),
+                           trControl=trctrl, tuneLength = 10)
```

> wine.dTree_gini

CART



```
3428 samples
  11 predictor
  7 classes: 'Average', 'Excellent', 'Good', 'Great', 'LessThanAvg', 'MoreThanAvg',
  'NotAccept'
```

No pre-processing

Resampling: Cross-Validated (10 fold, repeated 10 times)

Summary of sample sizes: 3084, 3085, 3086, 3088, 3082, 3086, ...

Resampling results across tuning parameters:

cp	Accuracy	Kappa
0.002543720	0.5343409	0.2675094
0.002649709	0.5338720	0.2656402
0.003179650	0.5324351	0.2588795
0.003577107	0.5295228	0.2522079
0.003709592	0.5295229	0.2521572
0.005299417	0.5193445	0.2253181
0.006006006	0.5147324	0.2068900
0.007065889	0.5131566	0.1882744
0.013778484	0.5072367	0.1686660
0.059618442	0.4852160	0.1150732

Accuracy was used to select the optimal model using the largest value.

The final value used for the model was cp = 0.00254372.

```
> prp(wine.dTree_gini$finalModel)
> test_pred_gini <- predict(wine.dTree_gini, newdata = test_data_new1)
> cm_gini <- confusionMatrix(test_pred_gini, test_data1$grade) #check accuracy
> cm_gini
```

Confusion Matrix and Statistics

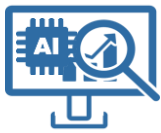
Prediction	Reference						
	Average	Excellent	Good	Great	LessThanAvg	MoreThanAvg	NotAccept
Average	246	0	14	1	30	138	1
Excellent	0	0	0	0	0	0	0
Good	7	0	93	20	1	81	0
Great	0	0	0	1	0	0	0
LessThanAvg	8	0	0	0	2	3	0
MoreThanAvg	182	1	155	27	20	435	4
NotAccept	0	0	0	0	0	0	0

Overall Statistics

```
Accuracy : 0.5286
95% CI : (0.5027, 0.5544)
```

```
No Information Rate : 0.4469
```

```
P-Value [Acc > NIR] : 2.089e-10
```



Kappa : 0.2593

McNemar's Test P-Value : NA

Statistics by Class:

	Class: Average	Class: Excellent	Class: Good	Class: Great
Sensitivity	0.5553	0.0000000	0.35496	0.0204082
Specificity	0.8208	1.0000000	0.90977	1.0000000
Pos Pred Value	0.5721	NaN	0.46040	1.0000000
Neg Pred Value	0.8106	0.9993197	0.86672	0.9673247
Prevalence	0.3014	0.0006803	0.17823	0.0333333
Detection Rate	0.1673	0.0000000	0.06327	0.0006803
Detection Prevalence	0.2925	0.0000000	0.13741	0.0006803
Balanced Accuracy	0.6881	0.5000000	0.63237	0.5102041

	Class: LessThanAvg	Class: MoreThanAvg	Class: NotAccept
Sensitivity	0.037736	0.6621	0.000000
Specificity	0.992237	0.5215	1.000000
Pos Pred Value	0.153846	0.5279	NaN
Neg Pred Value	0.964997	0.6563	0.996599
Prevalence	0.036054	0.4469	0.003401
Detection Rate	0.001361	0.2959	0.000000
Detection Prevalence	0.008844	0.5605	0.000000
Balanced Accuracy	0.514986	0.5918	0.500000

> table(test_pred, test_data\$grade)

test_pred	Average	Excellent	Good	Great	LessThanAvg	MoreThanAvg	NotAccept
Average	206	0	10	0	31	103	1
Excellent	0	0	0	0	0	0	0
Good	3	1	68	17	0	52	0
Great	1	0	2	3	0	1	0
LessThanAvg	5	0	0	1	1	2	0
MoreThanAvg	228	0	182	28	21	499	4
NotAccept	0	0	0	0	0	0	0

> table(test_pred_gini, test_data\$grade)

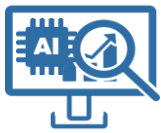
test_pred_gini	Average	Excellent	Good	Great	LessThanAvg	MoreThanAvg	NotAccept
Average	246	0	14	1	30	138	1
Excellent	0	0	0	0	0	0	0
Good	7	0	93	20	1	81	0
Great	0	0	0	1	0	0	0
LessThanAvg	8	0	0	0	2	3	0
MoreThanAvg	182	1	155	27	20	435	4
NotAccept	0	0	0	0	0	0	0

>

> # Measure model performance with AUC

> library(ROCR)

Loading required package: gplots



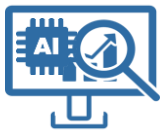
Attaching package: 'gplots'

The following object is masked from 'package:stats':

lowess

Warning messages:

```
1: package 'ROCR' was built under R version 3.4.4
2: package 'gplots' was built under R version 3.4.4
> # Calculate the probability of new observations belonging to each class
> # Source_pred_ROC_AUC will be a matrix with dimensions (data_set_size x
number_of_classes)
> grade_pred_ROC_AUC_dtree <- predict(wine.dTree, test_data_new1, type="prob")
> # Use colors for roc curves for the three classes (one for each Source):
> colors <- c("green", "orange", "purple", "yellow", "blue", "black", "red")
> # Specify the different classes
> ROC_AUC_classes <- levels(test_data1$grade)
> # For each class
> for (i in 1:7)
+ {
+ # Define observations belonging to each class i.e., class[i]
+ Actualgrade_values <- ifelse(test_data1$grade==ROC_AUC_classes[i],1,0)
+ # Assess the performance of classifier for class[i]
+ pred <- prediction(grade_pred_ROC_AUC_dtree[,i], Actualgrade_values)
+ perf <- performance(pred, "tpr", "fpr")
+ pdf(paste("Decision_Tree_v7_ROC_Curve_for_", ROC_AUC_classes[i], ".pdf"))
+ plot(perf, main=paste("Decision Tree ROC Curve for ",
ROC_AUC_classes[i]), col=colors[i])
+ dev.off()
+ # Calculate the AUC and print it to screen
+ auc.perf <- performance(pred, measure = "auc")
+ print(paste("Decision Tree AUC for ", ROC_AUC_classes[i], ":", auc.perf@y.values))
+ }
[1] "Decision Tree AUC for Average : 0.770696828958966"
[1] "Decision Tree AUC for Excellent : 0.466643975493533"
[1] "Decision Tree AUC for Good : 0.754293893129771"
[1] "Decision Tree AUC for Great : 0.763747863677491"
[1] "Decision Tree AUC for LessThanAvg : 0.72961079080172"
[1] "Decision Tree AUC for MoreThanAvg : 0.598024678876926"
[1] "Decision Tree AUC for NotAccept : 0.437133105802048"
> results_dTree_gini <- data.frame(Actual = test_data1$grade, Predicted = predict(wine.
dTree_gini, test_data_new1, type = "prob"))
> head(results_dTree_gini)
```



```

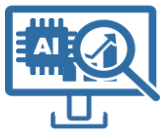
      Actual Predicted.Average Predicted.Excellent Predicted.Good
5 MoreThanAvg          0.2214386              0  0.142454161
7 MoreThanAvg          0.6749049              0  0.007604563
8 MoreThanAvg          0.2666667              0  0.133333333
9 MoreThanAvg          0.6749049              0  0.007604563
12 Average            0.2214386              0  0.142454161
14 Good               0.1805825              0  0.213592233
      Predicted.Great Predicted.LessThanAvg Predicted.MoreThanAvg
5      0.01410437          0.01410437          0.6064880
7      0.00000000          0.05513308          0.2566540
8      0.00000000          0.00000000          0.6000000
9      0.00000000          0.05513308          0.2566540
12     0.01410437          0.01410437          0.6064880
14     0.05825243          0.01747573          0.5242718
      Predicted.NotAccept
5      0.001410437
7      0.005703422
8      0.000000000
9      0.005703422
12     0.001410437
14     0.005825243
> results_dTree_gini$predicted_grade <- ifelse(results_dTree_gini$Predicted.Average
> 0.5, "Average",
+       ifelse(results_dTree_gini$Predicted.Excellent > 0.5, "Excellent",
+       ifelse(results_dTree_gini$Predicted.Good > 0.5, "Good",
+       ifelse(results_dTree_gini$Predicted.Great > 0.5, "Great",
+       ifelse(results_dTree_gini$Predicted.LessThanAvg > 0.5, "LessThanAvg",
+       ifelse(results_dTree_gini$Predicted.MoreThanAvg > 0.5, "MoreThanAvg",
+       ifelse(results_dTree_gini$Predicted.NotAccept > 0.5, "NotAccept",
+       '))))))
> results_dTree_gini$predicted_grade <- factor(results_dTree_gini$predicted_grade,
levels=levels(results_dTree_gini$Actual))
> cm_original_dtree_gini <- confusionMatrix(results_dTree_gini$predicted_grade,
test_data$grade)
> cm_original_dtree_gini

```

Confusion Matrix and Statistics

Prediction	Reference						
	Average	Excellent	Good	Great	LessThanAvg	MoreThanAvg	NotAccept
Average	245	0	14	1	30	135	1
Excellent	0	0	0	0	0	0	0
Good	7	0	93	20	1	81	0
Great	0	0	0	1	0	0	0
LessThanAvg	8	0	0	0	2	3	0
MoreThanAvg	182	1	155	27	20	435	4
NotAccept	0	0	0	0	0	0	0

Overall Statistics



```
Accuracy : 0.5293
 95% CI : (0.5034, 0.5551)
No Information Rate : 0.4461
P-Value [Acc > NIR] : 1.003e-10
```

```
Kappa : 0.2607
McNemar's Test P-Value : NA
```

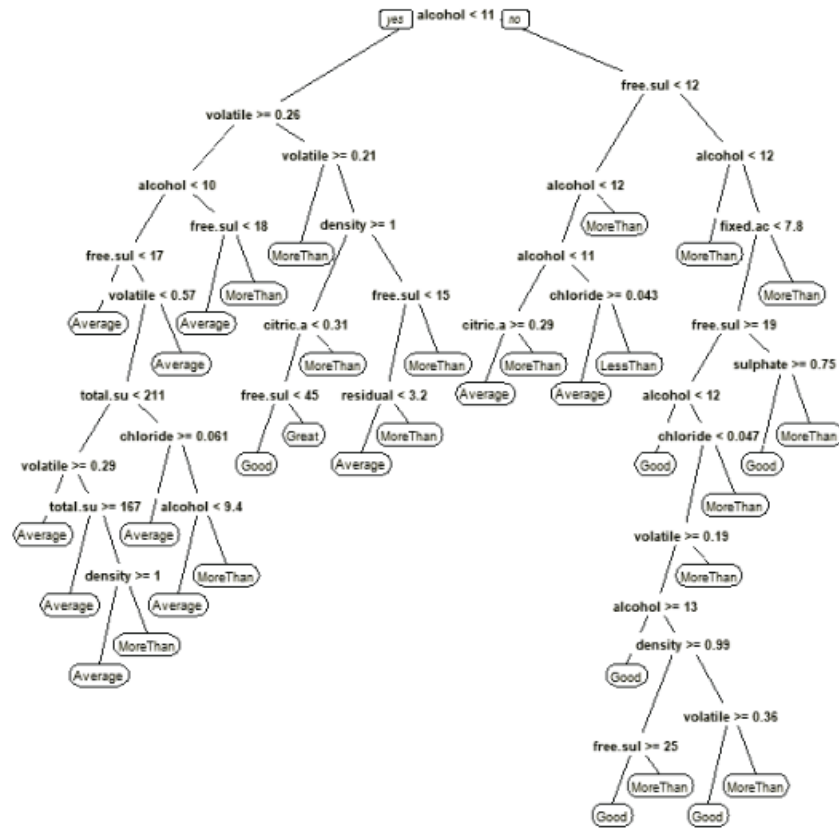
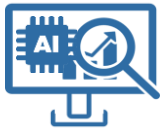
Statistics by Class:

```
Class: Average Class: Excellent Class: Good Class: Great
Sensitivity      0.5543      0.0000000      0.35496      0.0204082
Specificity      0.8232      1.0000000      0.90947      1.0000000
Pos Pred Value   0.5751      NaN           0.46040      1.0000000
Neg Pred Value   0.8106      0.9993179     0.86630      0.9672355
Prevalence       0.3015      0.0006821     0.17872      0.0334243
Detection Rate   0.1671      0.0000000     0.06344      0.0006821
Detection Prevalence 0.2906      0.0000000     0.13779      0.0006821
Balanced Accuracy 0.6888      0.5000000     0.63222      0.5102041
```

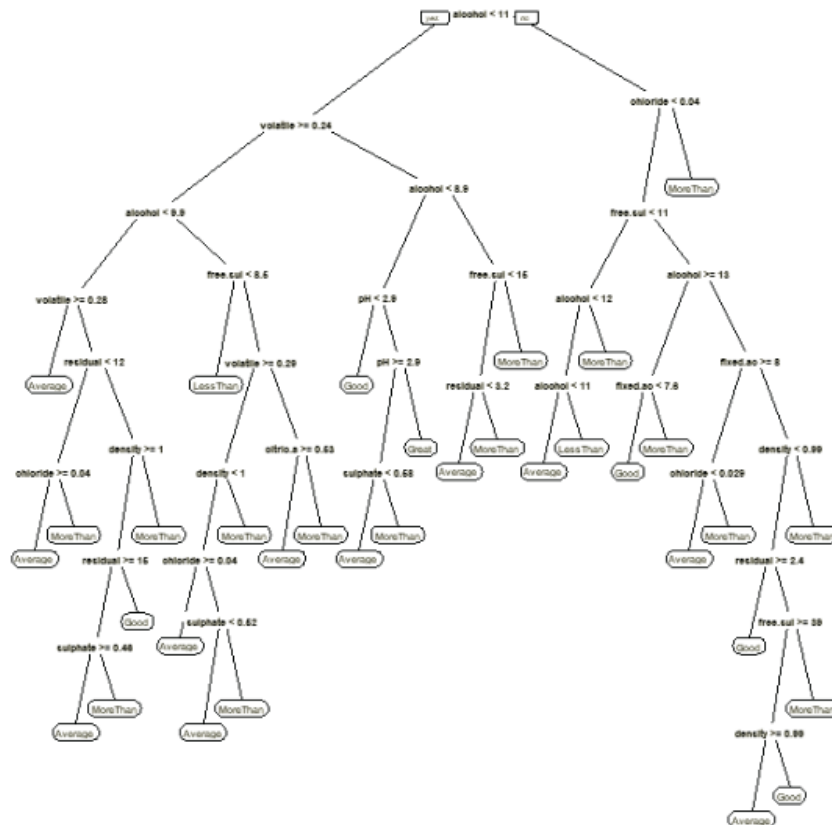
```
Class: LessThanAvg Class: MoreThanAvg Class: NotAccept
Sensitivity      0.037736      0.6651      0.000000
Specificity      0.992215      0.5209      1.000000
Pos Pred Value   0.153846      0.5279      NaN
Neg Pred Value   0.964900      0.6589      0.996589
Prevalence       0.036153      0.4461      0.003411
Detection Rate   0.001364      0.2967      0.000000
Detection Prevalence 0.008868      0.5621      0.000000
Balanced Accuracy 0.514975      0.5930      0.500000
```

```
> pairs(table(test_pred_gini, test_data1$grade), main="Wine Quality Classification
using Decision Tree")
>
```

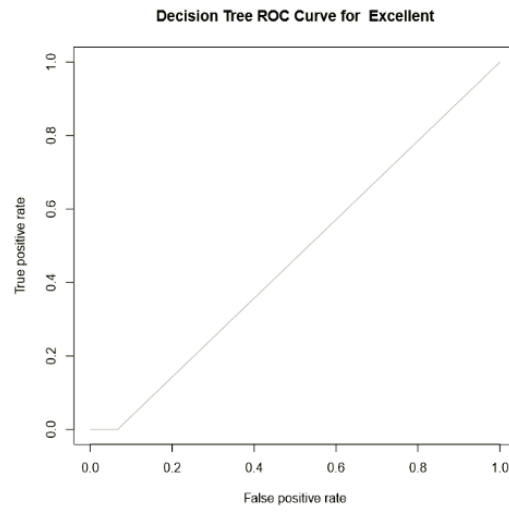
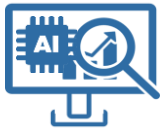
Struktur utama pohon keputusan yang digunakan untuk klasifikasi kualitas anggur dengan kriteria "perolehan informasi" dan "gini" masing-masing ditunjukkan pada Gambar 4.10 dan 4.11. Kurva ROC untuk model Pohon Keputusan untuk masing-masing dari tujuh kelas ditunjukkan pada Gambar 4.12 hingga 4.18.



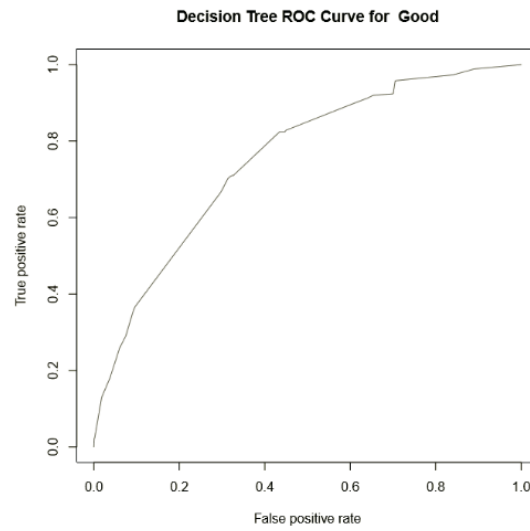
Gambar 4.10 Struktur utama Pohon Keputusan (dengan Informasi sebagai kriteria).



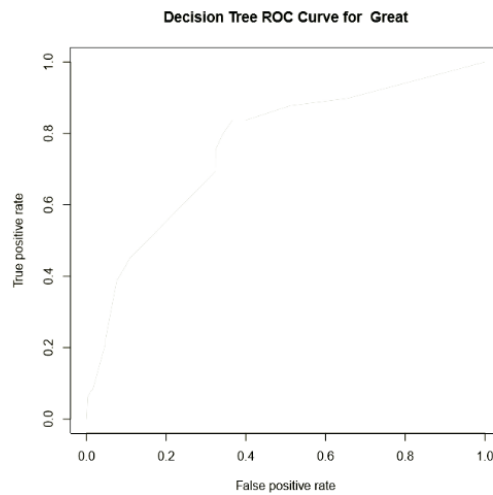
Gambar 4.11 Struktur utama Pohon Keputusan (dengan Gini sebagai kriteria).



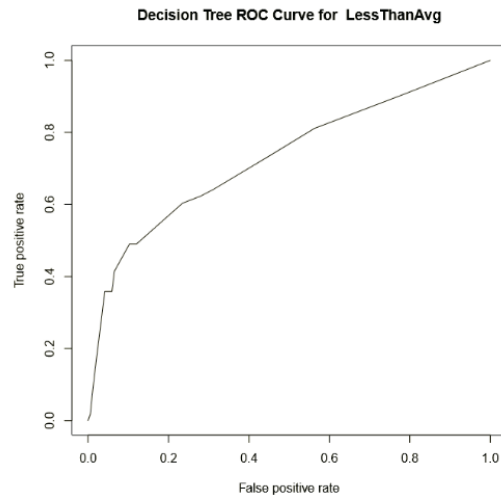
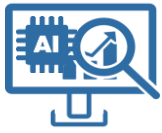
Gambar 4.12 Kurva ROC Pohon Keputusan untuk Sangat Baik.



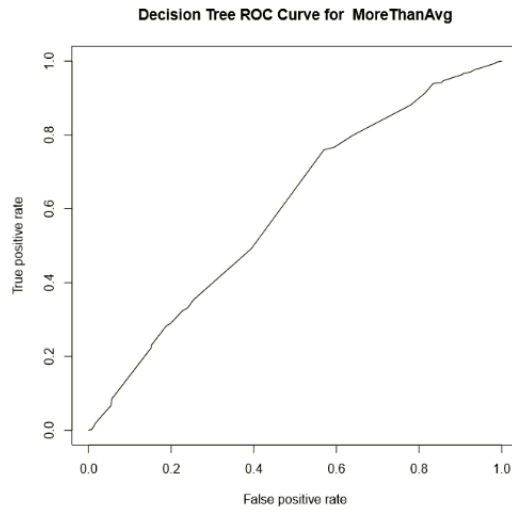
Gambar 4.13 Kurva ROC Pohon Keputusan untuk Baik.



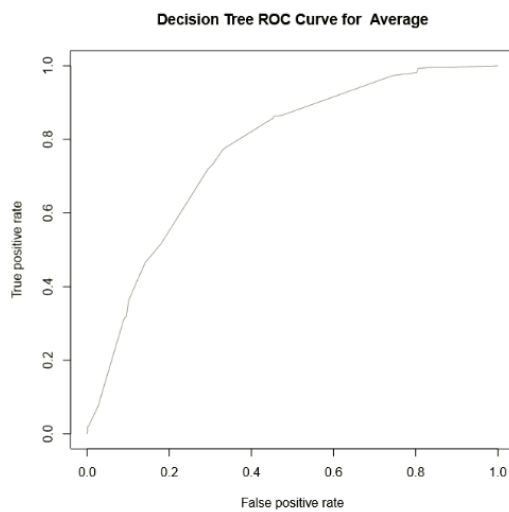
Gambar 4.14 Kurva ROC Pohon Keputusan untuk Hebat.



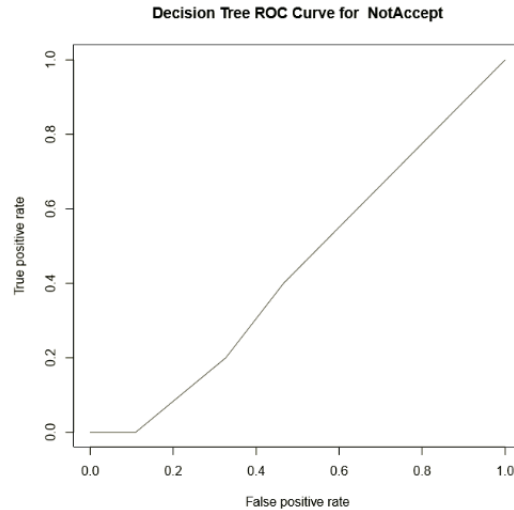
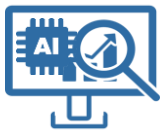
Gambar 4.15 Kurva ROC Pohon Keputusan untuk LessThanAvg.



Gambar 4.16 Kurva ROC Pohon Keputusan untuk MoreThanAvg.



Gambar 4.17 Kurva ROC Pohon Keputusan untuk Rata-rata.

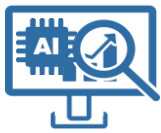


Gambar 4.18 Kurva ROC Pohon Keputusan untuk NotAccept

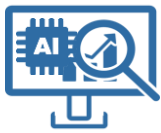
Dari plot varImp hutan acak dan plot struktur primer pohon keputusan (dengan kriteria informasi dan Gini), kita melihat bahwa "alkohol" adalah fitur yang paling penting. Namun, nilai "Kappa" (berbeda dengan metrik "akurasi") bervariasi dari 49,64% hingga 60% dalam kasus model hutan acak, sementara itu bervariasi dari sedikit di atas 25% hingga 26% dalam kasus model pohon keputusan. Selain itu, overfitting lebih rendah, seperti yang diamati dari kurva ROC untuk kelas yang ada dalam data uji dibandingkan dengan model pohon keputusan. Hal ini menunjukkan bahwa model hutan acak berkinerja lebih baik daripada pohon keputusan dalam skenario ini. Namun, ketidakseimbangan dalam kelas-kelas dataset dan pembangunan model tanpa pra-pemrosesan data turut berperan dalam proses pemilihan fitur dan model serta overfitting. Performa algoritma hutan acak dapat ditingkatkan lebih lanjut dengan menggunakan berbagai metode resampling untuk menyeimbangkan distribusi data di setiap kelas, seperti "up" sampling pada kelas yang kurang terdistribusi, yang mana duplikasi acak sampel dari kelas dengan lebih sedikit instans atau pembangkitan data tambahan untuk meratakan distribusi data di setiap kelas dilakukan, "down" sampling pada kelas yang lebih terdistribusi, yang mana pemilihan sampel secara acak dari kelas yang lebih terdistribusi untuk meratakan data di setiap kelas dilakukan, menggunakan kombinasi "up" dan "down" sampling yang disebut SMOTE, dll. Teknik-teknik ini diterapkan pada tahap validasi silang sebelum membangun model untuk meningkatkan performanya.

Menggunakan "Up" Sampling pada Model Hutan Acak untuk Meningkatkan Performa

Sebagaimana dijelaskan pada sisipan di atas, di subbagian ini kami menunjukkan bagaimana "up" sampling pada langkah validasi silang meningkatkan performa model hutan acak. Daftar 4.5 dan 4.6 menyediakan kode untuk hal ini, diikuti dengan keluaran eksekusi kode.

**Daftar 4.5** Menggunakan "Up" Sampling dengan Model Hutan Acak

```
setwd("<directory-name>")
library(caret)
library(e1071)
library(DMwR)
# The file winequality-white.csv is in the working directory
winedata2 <- read.csv("winequality-white.csv", header=TRUE, sep=';')
head(winedata2)
winedata2$grade <- ifelse(winedata2$quality <= 3, 'NotAccept',
                          ifelse(winedata2$quality == 4, 'LessThanAvg',
                                  ifelse(winedata2$quality == 5, 'Average',
                                          ifelse(winedata2$quality == 6, 'MoreThanAvg',
                                                  ifelse(winedata2$quality == 7, 'Good',
                                                          ifelse(winedata2$quality == 8, 'Great',
                                                                  ifelse(winedata2$quality >= 9, 'Excellent',
                                                                          'CannotGrade'))))))))
class(winedata2$grade)
winedata2$grade <- as.factor(winedata2$grade)
head(winedata2$grade)
anyNA(winedata2)
set.seed(12345)
sample_size2 <- 0.70 * nrow(winedata2)
sampledata2 <- sample(seq_len(nrow(winedata2)), sample_size2)
training_data2 <- winedata2[sampledata2, ]
test_data2 <- winedata2[-sampledata2, ]
nrow(training_data2) #
nrow(test_data2) #
head(training_data2$grade)
training_data2$grade <- as.factor(training_data2$grade)
summary(training_data2$grade) # to get distribution of each grade
training_data2 <- training_data2[, -12]
head(training_data2)
levels(training_data2$grade)
training_data2$grade <- factor(as.character(training_data2$grade))
levels(training_data2$grade)
# Using "up" to perform over-sampling for imbalanced training_data
# (Creating a balanced dataset for original TRAINING_SET for imbalanced classes)
trctrl5 <- trainControl(method = "repeatedcv", number = 10, repeats = 10,
                        savePredictions = TRUE, sampling = "up")
# Use same seed as used for splitting data to get better control
```



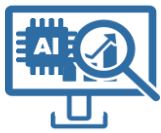
```
# model randomness i.e., for reusable results
set.seed(12345)
rf_up5 <- train(grade ~ ., data = training_data2, method = "rf", preProcess =
c("scale", "center"), trControl = trctrl5)
print(rf_up5)
rf_up5$finalModel$confusion
# Feature importance
imp5 <- rf_up5$finalModel$importance
dim(imp5)
class(imp5)
imp5
imp5[order(imp5, decreasing = TRUE), ]
# estimate variable importance
importance <- varImp(rf_up5, scale = TRUE)
plot(importance)
# predicting on test data
head(test_data2)
test_data_new2 <- test_data2[,-12]
test_data_new2 <- test_data_new2[,-12]
head(test_data_new2)
head(training_data2)
rf_up5_pred <- predict(rf_up5, test_data_new2)
cm_up5 <- confusionMatrix(rf_up5_pred, test_data2$grade)
cm_up5

results_rf_over <- data.frame(Actual = test_data2$grade, Predicted = predict(rf_
up5, test_data_new2, type = "prob"))
head(results_rf_over)
results_rf_over$predicted_grade <- ifelse(results_rf_over$Predicted.Average > 0.5,
"Average",
      ifelse(results_rf_over$Predicted.Excellent > 0.5, "Excellent",
            ifelse(results_rf_over$Predicted.Good > 0.5, "Good",
                  ifelse(results_rf_over$Predicted.Great > 0.5, "Great",
                        ifelse(results_rf_over$Predicted.LessThanAvg > 0.5, "LessThanAvg",
                              ifelse(results_rf_over$Predicted.MoreThanAvg > 0.5, "MoreThanAvg",
                                    ifelse(results_rf_over$Predicted.NotAccept > 0.5, "NotAccept",
                                          ')))))))))

results_rf_over$predicted_grade <- factor(results_rf_over$predicted_grade,
levels=levels(results_rf_over$Actual))

cm_over <- confusionMatrix(results_rf_over$predicted_grade, test_data2$grade)
cm_over

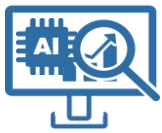
# Measure model performance with ROCR and AUC
library(ROCR)
# Calculate the probability of new observations belonging to each class
```



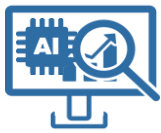
```
# grade_pred_up_ROC_AUC will be a matrix with dimensions (data_set_size x
number_of_classes)
grade_pred_up_ROC_AUC <- predict(rf_up5, test_data_new2, type="prob")
# Use colors for roc curves for the classes (one for each grade measure):
colors <- c("green", "orange", "purple", "yellow", "blue", "black", "red")
# Specify the different classes
ROC_AUC_classes <- levels(test_data2$grade)
# For each class
for (i in 1:7)
{
  # Define observations belonging to each class i.e., class[i]
  Actualgrade_values <- ifelse(test_data2$grade==ROC_AUC_classes[i], 1, 0)
  # Assess the performance of classifier for class[i]
  pred <- prediction(grade_pred_up_ROC_AUC[, i], Actualgrade_values)
  perf <- performance(pred, "tpr", "fpr")
  pdf(paste("Random_Forest_up_ROC_Curve_for_", ROC_AUC_classes[i], ".pdf"))
  plot(perf, main=paste("Random Forest up ROC Curve for ",
ROC_AUC_classes[i]), col=colors[i])
  dev.off()
  # Calculate the AUC and print it to screen
  auc.perf <- performance(pred, measure = "auc")
  print(paste("Random_Forest_up_AUC for ", ROC_AUC_classes[i], ":", auc.perf@y.values))
}
pairs(table(rf_up5_pred, test_data2$grade), main="Wine Quality Classification")
models2 <- list(original = model_rf, over = rf_up5)
resampling <- resamples(models2)
bwplot(resampling)
```

Daftar 4.6 Keluaran Kode pada Daftar 4.5

```
> setwd("<directory-name>")
> library(caret)
> library(e1071)
> library(DMwR)
> # The file winequality-white.csv is in the working directory
> winedata2 <- read.csv("winequality-white.csv", header=TRUE, sep=';')
> head(winedata2)
  fixed.acidity volatile.acidity citric.acid residual.sugar chlorides free.sulfur.dioxide
1          7.0           0.27         0.36          20.7      0.045                45
2          6.3           0.30         0.34           1.6      0.049                14
3          8.1           0.28         0.40           6.9      0.050                30
4          7.2           0.23         0.32           8.5      0.058                47
5          7.2           0.23         0.32           8.5      0.058                47
6          8.1           0.28         0.40           6.9      0.050                30
```



```
total.sulfur.dioxide density    pH sulphates alcohol quality
1          170  1.0010 3.00      0.45    8.8      6
2          132  0.9940 3.30      0.49    9.5      6
3           97  0.9951 3.26      0.44   10.1      6
4          186  0.9956 3.19      0.40    9.9      6
5          186  0.9956 3.19      0.40    9.9      6
6           97  0.9951 3.26      0.44   10.1      6
> winedata2$grade <- ifelse(winedata2$quality <= 3, 'NotAccept',
+ ifelse(winedata2$quality == 4, 'LessThanAvg',
+ ifelse(winedata2$quality == 5, 'Average',
+ ifelse(winedata2$quality == 6, 'MoreThanAvg',
+ ifelse(winedata2$quality == 7, 'Good',
+ ifelse(winedata2$quality == 8, 'Great',
+ ifelse(winedata2$quality >= 9, 'Excellent',
+ 'CannotGrade')))))))
> class(winedata2$grade)
[1] "character"
> winedata2$grade <- as.factor(winedata2$grade)
> head(winedata2$grade)
[1] MoreThanAvg MoreThanAvg MoreThanAvg MoreThanAvg MoreThanAvg MoreThanAvg
Levels: Average Excellent Good Great LessThanAvg MoreThanAvg NotAccept
> anyNA(winedata2)
[1] FALSE
> set.seed(12345)
> sample_size2 <- 0.70 * nrow(winedata2)
> sampledata2 <- sample(seq_len(nrow(winedata2)), sample_size2)
> training_data2 <- winedata2[sampledata2, ]
> test_data2 <- winedata2[-sampledata2, ]
> nrow(training_data2) #
[1] 3428
> nrow(test_data2) #
[1] 1470
> head(training_data2$grade)
```



```
[1] Average      MoreThanAvg MoreThanAvg Great      Average      MoreThanAvg
Levels: Average Excellent Good Great LessThanAvg MoreThanAvg NotAccept
> training_data2$grade <- as.factor(training_data2$grade)
> summary(training_data2$grade) # to get distribution of each grade
      Average      Excellent      Good      Great LessThanAvg MoreThanAvg      NotAccept
      1014           4           618           126           110           1541           15
> training_data2 <- training_data2[,-12]
> head(training_data2)
      fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
3531          7.2           0.31           0.35           7.2           0.046
4289          7.0           0.20           0.33           4.7           0.030
3726          6.6           0.26           0.28           9.4           0.028
4338          7.3           0.19           0.27           13.9          0.057
2235          7.2           0.23           0.38           6.1           0.067
815           7.9           0.35           0.24           15.6          0.072

      free.sulfur.dioxide total.sulfur.dioxide density  pH sulphates alcohol      grade
3531           45           178 0.99550 3.14      0.53      9.7      Average
4289           25           76 0.99202 2.88      0.54     10.5 MoreThanAvg
3726           13           121 0.99254 3.17      0.34     12.1 MoreThanAvg
4338           45           155 0.99807 2.94      0.41      8.8      Great
2235           20           90 0.99496 3.17      0.79      9.7      Average
815            44           229 0.99785 3.03      0.59     10.5 MoreThanAvg

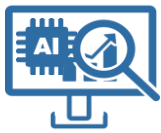
> # Using "up" to perform over-sampling for imbalanced training_data
> # (Creating a balanced dataset for original TRAINING_SET for imbalanced classes)
> trctrl5 <- trainControl(method = "repeatedcv", number = 10, repeats = 10,
savePredictions = TRUE, sampling = "up")
> set.seed(12345)
> rf_up5 <- train(grade ~ ., data = training_data2, method = "rf", preProcess =
c("scale", "center"), trControl = trctrl5)
> print(rf_up5)
Random Forest

3428 samples
 11 predictor
 7 classes: 'Average', 'Excellent', 'Good', 'Great', 'LessThanAvg', 'MoreThanAvg',
'NotAccept'

Pre-processing: scaled (11), centered (11)
Resampling: Cross-Validated (10 fold, repeated 10 times)
Summary of sample sizes: 3084, 3085, 3086, 3088, 3082, 3086, ...
Additional sampling using up-sampling prior to pre-processing

Resampling results across tuning parameters:
```

mtry	Accuracy	Kappa
2	0.6604751	0.4813420
6	0.6487449	0.4692309
11	0.6290210	0.4455761

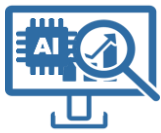


Accuracy was used to select the optimal model using the largest value.
The final value used for the model was `mtry = 2`.

```
> rf_up5$finalModel$confusion
      Average Excellent Good Great LessThanAvg MoreThanAvg NotAccept class.error
Average    1354         0   15    0         11        161         0 0.12134977
Excellent     0      1541    0    0         0         0         0 0.00000000
Good          3         0 1482    4         0         52         0 0.03828683
Great         0         0  0 1541    0         0         0         0 0.00000000
LessThanAvg  0         0  0  0    1541        0         0         0 0.00000000
MoreThanAvg 253         0 151    2         4        1131        0 0.26606100
NotAccept    0         0  0  0         0         0        1541 0.00000000

> # Feature importance
> imp5 <- rf_up5$finalModel$importance
> imp5
      MeanDecreaseGini
fixed.acidity      872.4905
volatile.acidity   800.6993
citric.acid        733.8792
residual.sugar     731.9511
chlorides          911.0409
free.sulfur.dioxide 995.7841
total.sulfur.dioxide 814.4053
density            979.8786
pH                830.4776
sulphates         591.4882
alcohol           980.1300
> imp5[order(imp5, decreasing = TRUE), ]
      free.sulfur.dioxide      alcohol      density      chlorides
      995.7841      980.1300      979.8786      911.0409
      fixed.acidity      pH total.sulfur.dioxide      volatile.acidity
      872.4905      830.4776      814.4053      800.6993
      citric.acid      residual.sugar      sulphates
      733.8792      731.9511      591.4882

> # estimate variable importance
> importance <- varImp(rf_up5, scale = TRUE)
> plot(importance)
> head(test_data2)
      fixed.acidity volatile.acidity citric.acid residual.sugar chlorides free.sulfur.dioxide
5          7.2          0.23          0.32          8.5          0.058          47
7          6.2          0.32          0.16          7.0          0.045          30
8          7.0          0.27          0.36          20.7         0.045          45
9          6.3          0.30          0.34          1.6          0.049          14
12         8.6          0.23          0.40          4.2          0.035          17
14         6.6          0.16          0.40          1.5          0.044          48
```



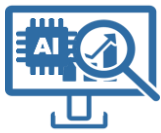
```

total.sulfur.dioxide density    pH sulphates alcohol quality    grade
5          186 0.9956 3.19      0.40    9.9      6 MoreThanAvg
7          136 0.9949 3.18      0.47    9.6      6 MoreThanAvg
8          170 1.0010 3.00      0.45    8.8      6 MoreThanAvg
9          132 0.9940 3.30      0.49    9.5      6 MoreThanAvg
12         109 0.9947 3.14      0.53    9.7      5   Average
14         143 0.9912 3.54      0.52   12.4      7     Good
> test_data_new2 <- test_data2[,-12]
> test_data_new2 <- test_data_new2[,-12]
> head(test_data_new2)
fixed.acidity volatile.acidity citric.acid residual.sugar chlorides free.sulfur.dioxide
5          7.2          0.23      0.32          8.5      0.058          47
7          6.2          0.32      0.16          7.0      0.045          30
8          7.0          0.27      0.36          20.7     0.045          45
9          6.3          0.30      0.34          1.6      0.049          14
12         8.6          0.23      0.40          4.2      0.035          17
14         6.6          0.16      0.40          1.5      0.044          48

total.sulfur.dioxide density    pH sulphates alcohol
5          186 0.9956 3.19      0.40    9.9
7          136 0.9949 3.18      0.47    9.6
8          170 1.0010 3.00      0.45    8.8
9          132 0.9940 3.30      0.49    9.5
12         109 0.9947 3.14      0.53    9.7
14         143 0.9912 3.54      0.52   12.4
> head(training_data2)
fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
3531         7.2          0.31      0.35          7.2      0.046
4289         7.0          0.20      0.33          4.7      0.030
3726         6.6          0.26      0.28          9.4      0.028
4338         7.3          0.19      0.27          13.9     0.057
2235         7.2          0.23      0.38          6.1      0.067
815          7.9          0.35      0.24          15.6     0.072
free.sulfur.dioxide total.sulfur.dioxide density    pH sulphates alcohol    grade
3531         45          178 0.99550 3.14      0.53    9.7   Average
4289         25          76 0.99202 2.88      0.54   10.5 MoreThanAvg
3726         13          121 0.99254 3.17      0.34   12.1 MoreThanAvg
4338         45          155 0.99807 2.94      0.41    8.8     Great
2235         20          90 0.99496 3.17      0.79    9.7   Average
815          44          229 0.99785 3.03      0.59   10.5 MoreThanAvg

> rf_up5_pred <- predict(rf_up5, test_data_new2)
> cm_up5 <- confusionMatrix(rf_up5_pred, test_data2$grade)
> cm_up5
Confusion Matrix and Statistics

```



Prediction	Reference						
	Average	Excellent	Good	Great	LessThanAvg	MoreThanAvg	NotAccept
Average	314	0	8	0	34	84	0
Excellent	0	0	0	0	0	0	0
Good	5	0	164	15	1	65	0
Great	0	0	1	19	0	1	0
LessThanAvg	3	0	0	1	3	2	0
MoreThanAvg	121	1	89	14	13	505	5
NotAccept	0	0	0	0	2	0	0

Overall Statistics

Accuracy : 0.6837
 95% CI : (0.6592, 0.7074)
 No Information Rate : 0.4469
 P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.5144

Mcnemar's Test P-Value : NA

Statistics by Class:

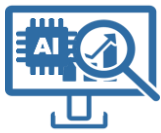
	Class: Average	Class: Excellent	Class: Good	Class: Great
Sensitivity	0.7088	0.0000000	0.6260	0.38776
Specificity	0.8773	1.0000000	0.9288	0.99859
Pos Pred Value	0.7136	NaN	0.6560	0.90476
Neg Pred Value	0.8748	0.9993197	0.9197	0.97930
Prevalence	0.3014	0.0006803	0.1782	0.03333
Detection Rate	0.2136	0.0000000	0.1116	0.01293
Detection Prevalence	0.2993	0.0000000	0.1701	0.01429
Balanced Accuracy	0.7931	0.5000000	0.7774	0.69317

	Class: LessThanAvg	Class: MoreThanAvg	Class: NotAccept
Sensitivity	0.056604	0.7686	0.000000
Specificity	0.995766	0.7011	0.998635
Pos Pred Value	0.333333	0.6751	0.000000
Neg Pred Value	0.965777	0.7895	0.996594
Prevalence	0.036054	0.4469	0.003401
Detection Rate	0.002041	0.3435	0.000000
Detection Prevalence	0.006122	0.5088	0.001361
Balanced Accuracy	0.526185	0.7349	0.499317

```
> results_rf_over <- data.frame(Actual = test_data2$grade, Predicted = predict(rf_
up5, test_data_new2, type = "prob"))
```

```
> head(results_rf_over)
```

	Actual	Predicted.Average	Predicted.Excellent	Predicted.Good	Predicted.Great
5	MoreThanAvg	0.164	0.000	0.034	0.004
7	MoreThanAvg	0.820	0.000	0.018	0.000
8	MoreThanAvg	0.220	0.000	0.040	0.000
9	MoreThanAvg	0.214	0.000	0.014	0.002
12	Average	0.250	0.002	0.052	0.004
14	Good	0.084	0.000	0.306	0.282

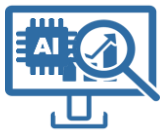


```
Predicted.LessThanAvg Predicted.MoreThanAvg Predicted.NotAccept
5                0.002                0.796                0.000
7                0.006                0.156                0.000
8                0.002                0.738                0.000
9                0.024                0.746                0.000
12               0.082                0.584                0.026
14               0.002                0.314                0.012
> results_rf_over$predicted_grade <- ifelse(results_rf_over$Predicted.Average > 0.5,
"Average",
+     ifelse(results_rf_over$Predicted.Excellent > 0.5, "Excellent",
+     ifelse(results_rf_over$Predicted.Good > 0.5, "Good",
+     ifelse(results_rf_over$Predicted.Great > 0.5, "Great",
+     ifelse(results_rf_over$Predicted.LessThanAvg > 0.5, "LessThanAvg",
+     ifelse(results_rf_over$Predicted.MoreThanAvg > 0.5, "MoreThanAvg",
+     ifelse(results_rf_over$Predicted.NotAccept > 0.5, "NotAccept",
+     '))))))
> results_rf_over$predicted_grade <- factor(results_rf_over$predicted_grade,
levels=levels(results_rf_over$Actual))
>
> cm_over <- confusionMatrix(results_rf_over$predicted_grade, test_data2$grade)
> cm_over
Confusion Matrix and Statistics

              Reference
Prediction   Average Excellent Good Great LessThanAvg MoreThanAvg NotAccept
Average      268          0    3    0         20         54          0
Excellent    0           0    0    0          0          0          0
Good         1           0  122    3          0         26          0
Great        0           0    0   18          0          1          0
LessThanAvg  0           0    0    0          2          0          0
MoreThanAvg  56           0   39    4          5        317          1
NotAccept    0           0    0    0          1          0          0

Overall Statistics

              Accuracy : 0.7726
              95% CI : (0.7444, 0.799)
              No Information Rate : 0.423
              P-Value [Acc > NIR] : < 2.2e-16
```



Kappa : 0.6528

McNemar's Test P-Value : NA

Statistics by Class:

	Class: Average	Class: Excellent	Class: Good	Class: Great
Sensitivity	0.8246	NA	0.7439	0.72000
Specificity	0.8750	1	0.9614	0.99891
Pos Pred Value	0.7768	NA	0.8026	0.94737
Neg Pred Value	0.9044	NA	0.9468	0.99241
Prevalence	0.3454	0	0.1743	0.02657
Detection Rate	0.2848	0	0.1296	0.01913
Detection Prevalence	0.3666	0	0.1615	0.02019
Balanced Accuracy	0.8498	NA	0.8526	0.85945

	Class: LessThanAvg	Class: MoreThanAvg	Class: NotAccept
Sensitivity	0.071429	0.7965	0.000000
Specificity	1.000000	0.8066	0.998936
Pos Pred Value	1.000000	0.7512	0.000000
Neg Pred Value	0.972311	0.8439	0.998936
Prevalence	0.029756	0.4230	0.001063
Detection Rate	0.002125	0.3369	0.000000
Detection Prevalence	0.002125	0.4485	0.001063
Balanced Accuracy	0.535714	0.8016	0.499468

```
> # Measure model performance with ROCR and AUC
> library(ROCR)
Loading required package: gplots
```

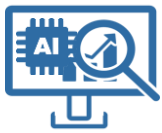
Attaching package: 'gplots'

The following object is masked from 'package:stats':

lowess

Warning messages:

```
1: package 'ROCR' was built under R version 3.4.4
2: package 'gplots' was built under R version 3.4.4
> # Calculate the probability of new observations belonging to each class
> # grade_pred_up_ROC_AUC will be a matrix with dimensions (data_set_size x
number_of_classes)
> grade_pred_up_ROC_AUC <- predict(rf_up5, test_data_new2, type="prob")
> # Use colors for roc curves for the classes (one for each grade measure):
> colors <- c("green", "orange", "purple", "yellow", "blue", "black", "red")
> # Specify the different classes
> ROC_AUC_classes <- levels(test_data2$grade)
> # For each class
> for (i in 1:7)
+ {
```

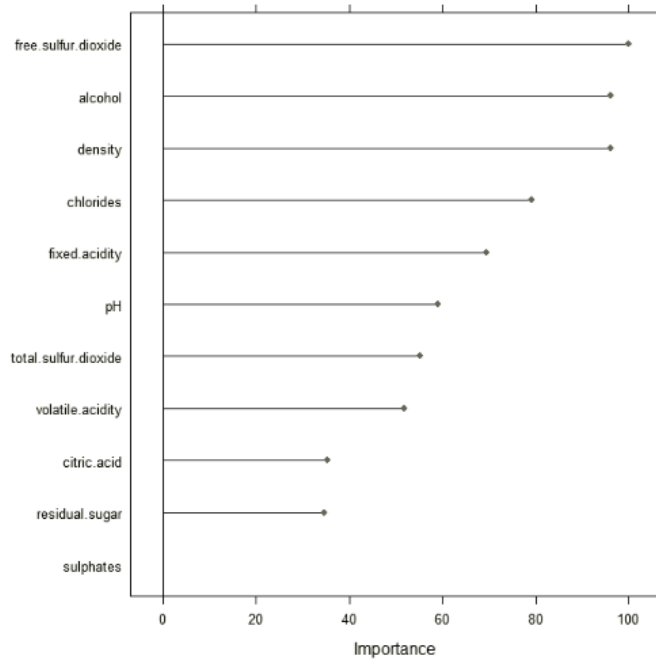
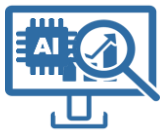


```
+ # Define observations belonging to each class i.e., class[i]
+ Actualgrade_values <- ifelse(test_data2$grade==ROC_AUC_classes[i],1,0)
+ # Assess the performance of classifier for class[i]
+ pred <- prediction(grade_pred_up_ROC_AUC[,i],Actualgrade_values)
+ perf <- performance(pred, "tpr", "fpr")
+   pdf(paste("Random_Forest_up_ROC_Curve_for_",ROC_AUC_classes[i],".pdf"))
+   plot(perf,main=paste("Random Forest up ROC Curve for ",
+     ROC_AUC_classes[i]),col=colors[i])
+   dev.off()
+ # Calculate the AUC and print it to screen
+ auc.perf <- performance(pred, measure = "auc")
+ print(paste("Random_Forest_up_AUC for ",ROC_AUC_classes[i],":",auc.perf@y.
+ values))
+ }
[1] "Random_Forest_up_AUC for Average : 0.895005286167386"
[1] "Random_Forest_up_AUC for Excellent : 0.993192648059905"
[1] "Random_Forest_up_AUC for Good : 0.897973434103432"
[1] "Random_Forest_up_AUC for Great : 0.887855634864783"
[1] "Random_Forest_up_AUC for LessThanAvg : 0.876040265775422"
[1] "Random_Forest_up_AUC for MoreThanAvg : 0.818323251725668"
[1] "Random_Forest_up_AUC for NotAccept : 0.678907849829351"
> pairs(table(rf_up5_pred, test_data2$grade), main="Wine Quality Classification")
> models2 <- list(original = model_rf, over = rf_up5)
> resampling <- resamples(models2)
> bwplot(resampling)
```

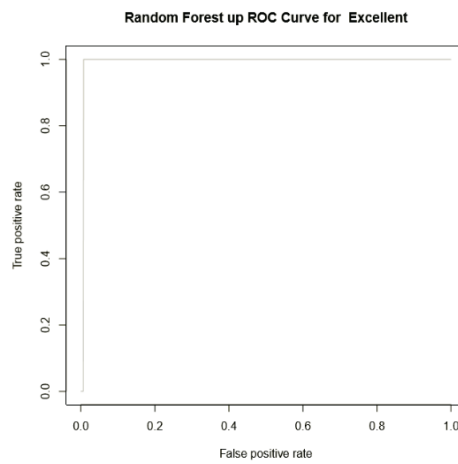
Poin-poin berikut patut diperhatikan:

- *Fitur dengan kepentingan tertinggi berubah dari "alkohol" menjadi "bebas sulfur dioksida".*
 - *Matriks kebingungan menunjukkan lebih banyak contoh data dalam kelas yang salah diklasifikasikan serta memiliki nilai Kappa yang lebih baik, yaitu > 65, dibandingkan dengan model hutan acak asli, yang sedikit di atas 60.*
 - *Kurva ROC menunjukkan lebih sedikit overfitting dibandingkan dengan model asli.*
 - *Performa AUC ROC meningkat dibandingkan dengan model asli.*
-

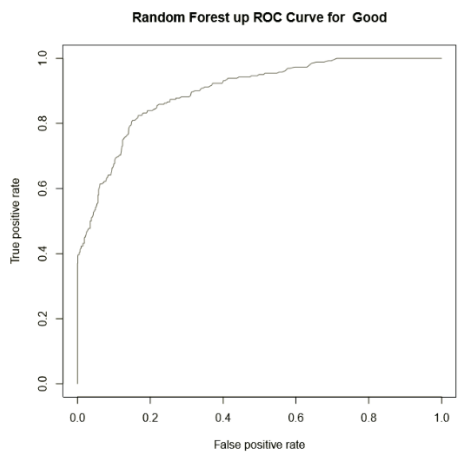
Gambar 4.19–4.28 menunjukkan plot kepentingan variabel dari model yang ditingkatkan beserta kurva ROC untuk setiap kelas dan bwplot yang membandingkan model asli dengan model yang ditingkatkan, dengan mempertimbangkan nilai akurasi dan Kappa, serta plot pasangan dari klasifikasi kualitas anggur. Seperti yang dapat dilihat, model pengambilan sampel "over" atau "up" memiliki nilai Kappa yang lebih tinggi daripada model asli.



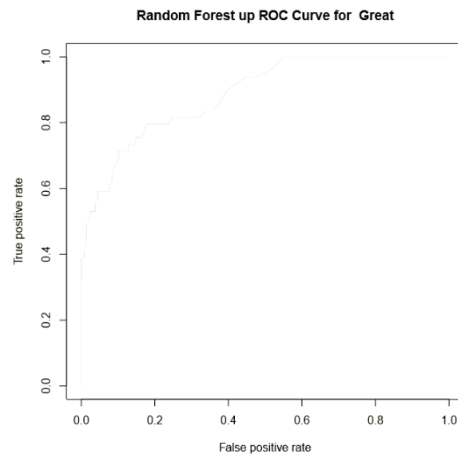
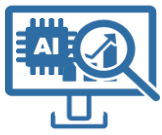
Gambar 4.19 Plot kepentingan variabel dari model hutan acak yang ditingkatkan.



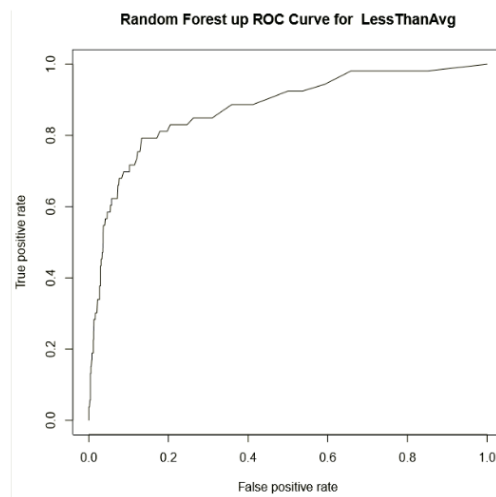
Gambar 4.20 Kurva ROC Hutan Acak menggunakan pengambilan sampel “naik” untuk Sangat Baik.



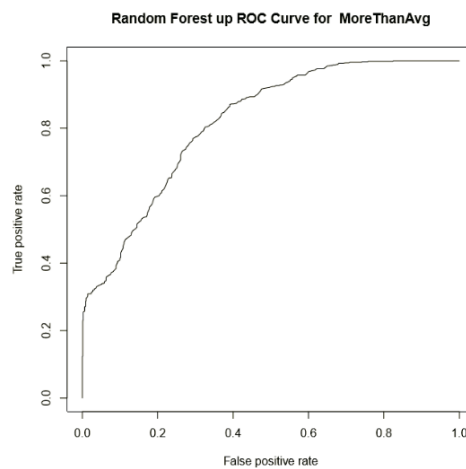
Gambar 4.21 Kurva ROC Hutan Acak menggunakan pengambilan sampel “naik” untuk Baik.



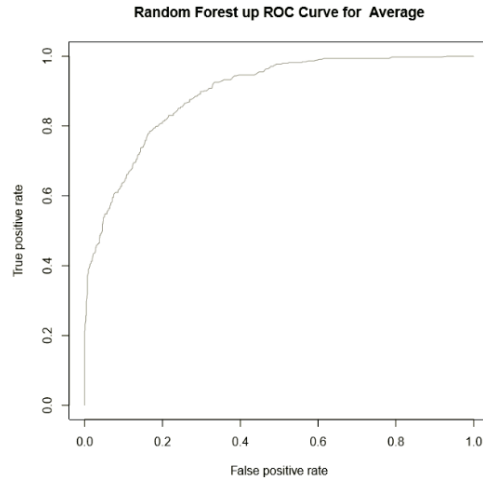
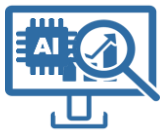
Gambar 4.22 Kurva ROC Hutan Acak menggunakan pengambilan sampel “naik” untuk Great.



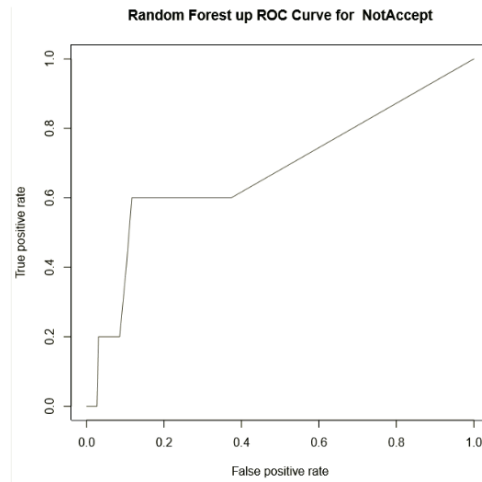
Gambar 4.23 Kurva ROC Hutan Acak menggunakan pengambilan sampel “naik” untuk LessThanAvg.



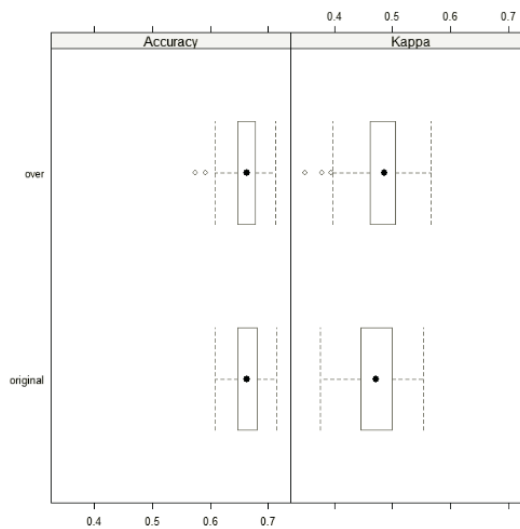
Gambar 4.24 Kurva ROC Hutan Acak menggunakan pengambilan sampel “naik” untuk MoreThanAvg.



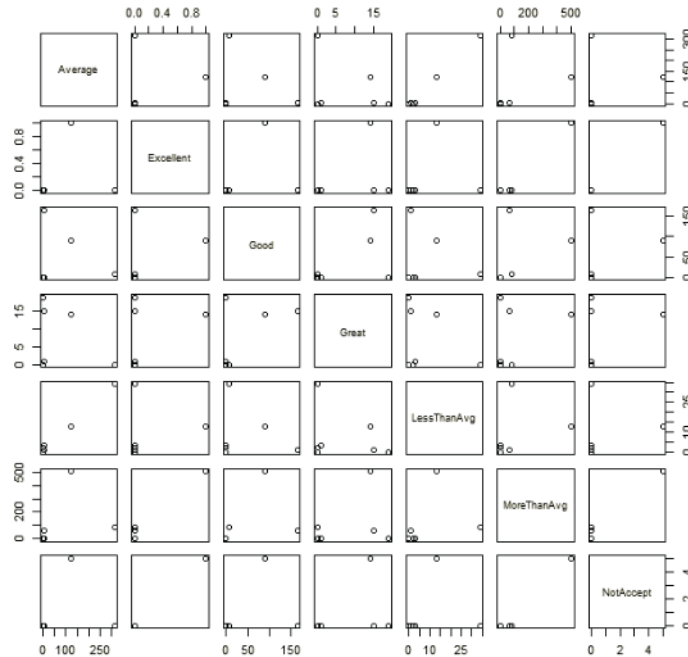
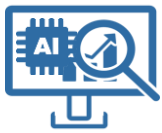
Gambar 4.25 Kurva ROC Hutan Acak menggunakan pengambilan sampel “naik” untuk Rata-rata.



Gambar 4.26 Kurva ROC Hutan Acak menggunakan pengambilan sampel “naik” untuk NotAccept.



Gambar 4.27 Plot perbandingan antara model hutan acak asli dan yang telah disempurnakan, yang menunjukkan nilai Akurasi dan Kappa.



Gambar 4.28 Plot berpasangan model hutan acak yang ditingkatkan untuk Klasifikasi Tingkat Anggur.

Pemberdayaan BI dan Dampaknya terhadap Perusahaan

Seperti yang terlihat dari gambar untuk Hutan Acak dan Pohon Keputusan, variabel terpenting adalah alkohol untuk klasifikasi tingkat anggur. Hal ini perlu diingat ketika melakukan analisis bisnis untuk pengadaan dan penjualan anggur di pasar yang kompetitif. Selain itu, akurasi nilai Kappa dari klasifikasi ke dalam tujuh kelas lebih baik dalam kasus implementasi hutan acak dibandingkan dengan implementasi pohon keputusan. Namun, penggunaan teknik pengambilan sampel "naik" untuk meningkatkan kinerja klasifikasi karena data yang tidak seimbang di setiap kelas dalam set pelatihan menunjukkan bahwa cukup dengan memilih sebanyak mungkin data yang tersedia di setiap kelas untuk menyeimbangkan set data.

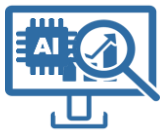
4.3 MEMPREDIKSI HARGA KOMODITAS DI MUKA

Pada bagian ini, kami menjelaskan studi kasus di bidang metalurgi dengan memprediksi harga logam mulia berdasarkan data bulanan yang tersedia. Kami mulai dengan menjelaskan tujuan kasus penggunaan dari sudut pandang teknis, kemudian merinci spesifikasi implementasinya.

Metodologi AI Analisis Prediktif Menggunakan Jaringan Saraf Tiruan

Tujuan Kasus Penggunaan

- Sasaran
Untuk memprediksi harga logam mulia. Ini adalah salah satu aplikasi utama dalam industri logam. Hal ini memungkinkan harga emas batangan untuk ditetapkan sesuai



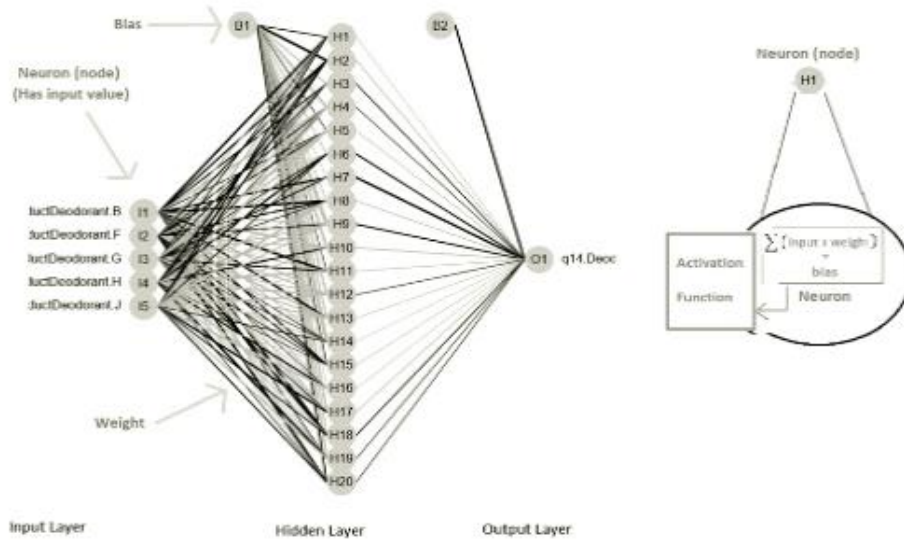
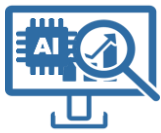
dengan berbagai mata uang dan berbagai belahan dunia. Ini juga dapat digunakan untuk mempelajari dampak harga emas batangan di pasar.

- Teknologi
AI dan ML menggunakan jaringan saraf tiruan dalam (DNN).
- Implementasi
Menggunakan pustaka keras dengan kerangka kerja Tensorflow® di R.
- Set Data Input
Nilai emas batangan di awal setiap bulan dari tahun 1950 hingga 2019 dalam file .csv dengan kolom Tanggal dan Harga.

Pengantar Singkat Jaringan Syaraf Tiruan dan Jaringan Syaraf Tiruan Dalam Berbasis Pembelajaran Mesin

Jaringan saraf tiruan, juga disebut jaringan saraf tiruan (JST), adalah algoritma/model/sistem berbasis pembelajaran mesin (ML) yang meniru pola otak manusia dan memproses serta belajar dari data untuk memecahkan masalah kompleks. Dengan kata lain, jaringan saraf tiruan dilatih dengan memasukkan data yang menghasilkan pengetahuan yang disimpan dan digunakan sistem saat memecahkan masalah dalam konteks. Jaringan saraf tiruan terstruktur sebagai tingkatan simpul yang mirip dengan neuron di otak manusia. Semakin banyak data yang dimasukkan, semakin banyak ia mempelajari dan menyimpan keluaran dari satu lapisan ke lapisan lainnya, menghasilkan keluaran akhir yang merupakan implementasi dari solusi. Gambar 4.29 menggambarkan struktur JST. Model jaringan saraf tiruan sama baiknya dengan data yang dipelajarinya. JST terdiri dari komponen-komponen berikut:

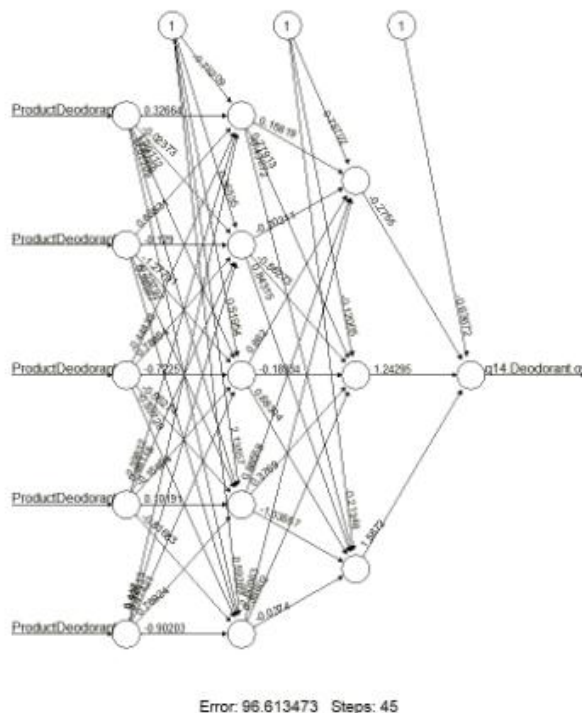
1. Simpul lapisan masukan (yang menyimpan nilai masukan).
2. Bobot (yang diinisialisasi secara acak) pada waktu pembelajaran model. Dalam AI, algoritma berbasis model jaringan saraf tiruan menginisialisasi dan menetapkan bobot secara acak tanpa perlu melakukannya secara terprogram.
3. Lapisan antara, juga disebut lapisan tersembunyi.
4. Istilah bias untuk setiap simpul di lapisan antara dan lapisan keluaran. Bias ini digunakan untuk menggeser nilai keluaran ke nilai keluaran optimal yang diinginkan.
5. Fungsi aktivasi yang memicu setiap neuron antara untuk melakukan komputasi. Fungsi aktivasi menerima jumlah bobot (yaitu, masukan*bobot) ditambah istilah bias untuk setiap neuron tersembunyi.
6. Pengoptimal yang berfungsi untuk meningkatkan akurasi model (selama fase propagasi maju dan komputasi).



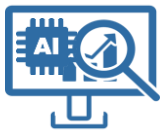
Gambar 4.29 Struktur tipikal jaringan saraf tiruan (JST).

7. Fungsi kerugian yang digunakan untuk meminimalkan tingkat kesalahan dalam fase pembelajaran (yaitu, membangun dan melatih, selama fase propagasi balik).
8. Lapisan keluaran (dengan satu atau lebih neuron keluaran). Untuk masalah regresi, lapisan keluaran memiliki satu simpul yang mewakili nilai biner, dan untuk masalah klasifikasi, lapisan keluaran memiliki jumlah simpul yang sama dengan jumlah kelas.

Jaringan saraf dalam (JSD) adalah jaringan saraf tiruan yang terdiri dari lapisan tersembunyi (>1) di antara lapisan masukan dan keluaran. Gambar 4.30 menunjukkan JSD tipikal.



Gambar 4.30 Jaringan saraf dalam (DNN) yang umum.



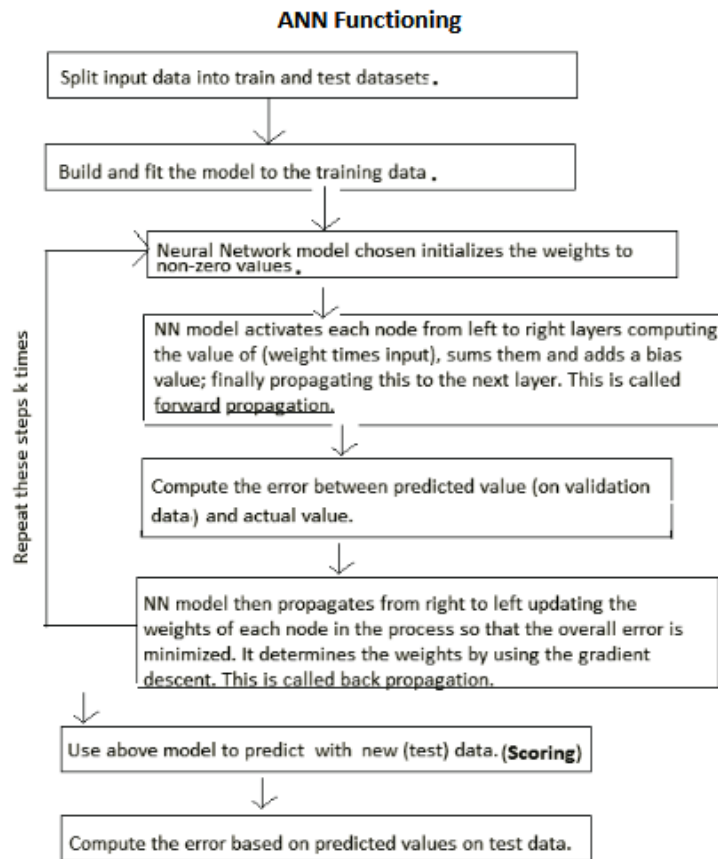
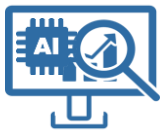
Gambar 4.30 menunjukkan berbagai fase kerja ANN. Fase-fase utamanya adalah sebagai berikut:

1. Membangun dan melatih model menggunakan data pelatihan. Dalam fase ini, ANN mempelajari berbagai fitur dan korelasi antar pola data, seperti yang diamati pada masukan. Yang terpenting, ANN juga mengekstraksi nonlinieritas dalam data. Ini melibatkan pembangunan, kompilasi modus (menentukan fungsi pengoptimal dan fungsi kerugian), dan pemasangan model menggunakan data pelatihan dan/atau validasi secara batch. Pemasangan dilakukan dalam iterasi yang disebut *epoch*.
2. Validasi model yang dibangun menggunakan data validasi. Ini digunakan untuk memverifikasi model yang telah dilatih pada sampel data secara bertahap, untuk memastikan apakah model tersebut mampu memberikan keluaran atau solusi yang diinginkan.
3. Skoring model pada data baru (atau data uji). Ini melibatkan prediksi keluaran (untuk masalah regresi) atau kelas (untuk masalah klasifikasi).
4. Validasi K-fold pada set pelatihan dan uji dalam k-fold (yaitu, iterasi) untuk mencapai akurasi optimal dan meminimalkan kerugian. Hal ini dilakukan untuk mengukur kinerja prediktif model.

Hal ini meliputi:

- Membagi data menjadi set pelatihan dan uji
- Membangun dan menyesuaikan model dengan set pelatihan
- Memberi skor (memprediksi) model pada set uji
- Menghitung kesalahan prediksi (MSE atau RMSE)
- Mengulangi proses di atas sebanyak K kali
- Menghitung rata-rata kesalahan kinerja.

Gambar 4.31 menunjukkan fungsi-fungsi utama ANN.



Gambar 4.31 Fungsi utama jaringan saraf tiruan.

Setelah fase penilaian, data uji digunakan sebagai set data dan dibagi secara acak menjadi set pelatihan dan pengujian sebanyak k kali, serta dihitung performa model untuk setiap lipatan dan kemudian dihitung rata-rata performanya. Validasi k -fold memberikan gambaran seberapa baik peningkatan performa tersebut.

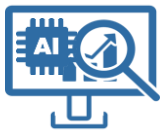
Berikut ini adalah beberapa aplikasi bisnis jaringan saraf tiruan dalam:

- Analisis preskriptif merekomendasikan produk berdasarkan "peringkat pengguna"
- Pembuatan dan kueri bahasa alami
- Pengenalan gambar seperti pada penandaan gambar Facebook™
- Prediksi cuaca
- Pemrosesan ucapan-ke-teks kontekstual
- Aplikasi genetik

BI dapat memiliki solusi DNN yang tertanam dalam tumpukan implementasinya untuk mewujudkan hasil yang kompetitif. Misalnya, untuk memprediksi cuaca, keluaran sistem DNN dapat dimanfaatkan untuk memperkirakan pola cuaca yang akan datang seperti badai atau hujan salju.

Pragmatik Implementasi

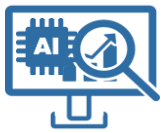
Kami akan menggunakan jaringan saraf tiruan dalam berbasis keras di R untuk mengimplementasikan model ini.



- Keras—Pustaka pembelajaran mesin sumber terbuka yang awalnya tersedia dalam Python® dan selanjutnya ditulis dalam R.
- Keras adalah API tingkat tinggi yang hanya membutuhkan beberapa baris kode dan bekerja di atas kerangka kerja yang disebut TensorFlow®. Keras juga dapat menggunakan kerangka kerja lain seperti Theano.
- Dengan menggunakan keras, model jaringan saraf tiruan dibangun dengan memulai secara berurutan dan menambahkan lapisan tambahan di atasnya. Model dapat dibangun menggunakan set data pelatihan, dikompilasi (menggunakan fungsi kerugian = $M(ean)S(kuadrat)E(rror)$ dan pengoptimal (= "*adam*"—untuk laju pembelajaran)), dicocokkan dan dievaluasi, lalu diberi skor berdasarkan data baru.
- Keluaran rekomendasi dapat diintegrasikan secara visual ke dalam dasbor BI untuk mendapatkan wawasan yang dapat ditindaklanjuti (misalnya, Tableau® Dashboard, Oracle® BI, Microsoft® BI, dll.)

Berikut langkah-langkah implementasi model:

- ❖ Model ini menggunakan model pembelajaran mesin (ML) jaringan saraf tiruan dalam yang dibangun dengan melatih dirinya sendiri berdasarkan harga yang berkorelasi dengan tahun dan tanggal pertama setiap bulan.
- ❖ Kasus penggunaan ini menggunakan data dari berkas `monthly_csv.csv` untuk memprediksi "Harga". Berkas ini memiliki tanggal dalam format Tanggal, Harga, dengan Tanggal sebagai nilai string "YYYY-MM".
- ❖ Siapkan data masukan dan pastikan masukan dinormalisasi (untuk memastikan akurasi yang lebih baik). Kami mengonversi string Tanggal menjadi nilai tanggal dalam format YYYY-MM-DD dengan DD, seperti biasa sebagai "01"—tanggal pertama setiap bulan menggunakan fungsi `ymd()` di pustaka R `lubridate`.
- ❖ Mulailah dengan memanggil fungsi inisialisasi `keras_model_sequential()`.
- ❖ Tambahkan lapisan padat untuk membuat koneksi ke lapisan perantara. Lapisan ini mengambil bentuk input dan aktivasi '*relu*'.
- ❖ Tambahkan dua lapisan padat tambahan menggunakan aktivasi '*relu*'. Lapisan-lapisan ini berfungsi sebagai lapisan tersembunyi (masing-masing dengan 12 dan 8 neuron).
- ❖ Tambahkan lapisan keluaran (dengan satu simpul) dan aktivasi 'linier' yang menjadi variabel respons "Harga". Karena kita hanya memprediksi satu variabel respons, yaitu "Harga", jumlah neuron di lapisan keluaran adalah 1 dan fungsi aktivasinya adalah 'softmax', untuk nilai float "Harga" (berbeda dengan prediksi nilai biner di mana fungsi aktivasi yang digunakan adalah 'sigmoid').
- ❖ Sesuaikan data dengan menerapkan keluaran model yang diprediksi terhadap data uji. Variabel prediktor yang digunakan adalah Tahun, Bulan, dan tanggal 1 setiap bulan, dan variabel respons yang digunakan adalah "Harga".
- ❖ Beri skor model pada dataset (uji) baru untuk mendapatkan "Harga" yang diprediksi.



Struktur DNN Berbasis Keras yang Dibangun dan Dilatih

```
summary(dnnFinal)
```

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 12)	48
dense_2 (Dense)	(None, 8)	104
dense_3 (Dense)	(None, 1)	9

Total params: 161
Trainable params: 161
Non-trainable params: 0

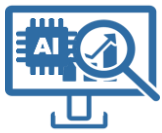
Program R yang Mengimplementasikan DNN

Daftar 4.7 menunjukkan kode R untuk mengimplementasikan model dan memberikan skor pada data baru.

Daftar 4.7 Program R untuk Mengimplementasikan DNN dalam Memprediksi Harga Logam Mulia

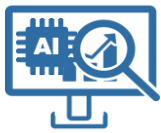
```
setwd("<directory-name>")
dataset <- read.csv("monthly_csv.csv")
library(tidyverse)
library(keras)
library(lubridate)
date1 <- paste0(dataset[, 1], "-01")
date1.date <- ymd(date1)
date1.date.df <- as.data.frame(date1.date)
head(date1.date.df)
df_with_DateVal <- cbind(date1.date.df, dataset[, 2])
colnames(df_with_DateVal) <- c("Date", "Price")

df_with_DateVal$Year <- as.numeric(substr(as.character(df_with_DateVal[, 1]),0,4))
df_with_DateVal$Month <- as.numeric(substr(as.character(df_with_DateVal[, 1]),6,7))
df_with_DateVal$Day <- as.numeric(substr(as.character(df_with_DateVal[, 1]),9, 10))
head(cbind(df_with_DateVal$Date, df_with_DateVal$Year, df_with_DateVal$Month, df_
with_DateVal$Day, df_with_DateVal$Price))
head(df_with_DateVal)
dfFinal <- df_with_DateVal[, c(1, 3:5,2)]
head(dfFinal)
```



```
maxminnorm <- function(x) {
  return ((x-min(x)) / (max(x) - min(x))) }

scaleddfFinal <- as.data.frame(lapply(dfFinal[, c(2,3,5)], maxminnorm))
head(scaleddfFinal)
attach(scaleddfFinal)
train_index <- sample(1:nrow(scaleddfFinal), 0.8 * nrow(scaleddfFinal))
test_index <- setdiff(1:nrow(scaleddfFinal), train_index)
X_train <- as.matrix(scaleddfFinal[train_index, -15])
y_train <- as.matrix(scaleddfFinal[train_index, "Price"])
X_test <- as.matrix(scaleddfFinal[test_index, -15])
y_test <- as.matrix(scaleddfFinal[test_index, "Price"])
dnnFinal <- keras_model_sequential()
dnnFinal %>%
layer_dense(units = 12, activation = 'relu',
kernel_initializer='RandomNormal',
input_shape = c(3)) %>%
layer_dense(units = 8, activation = 'relu') %>%
layer_dense(units = 1, activation = 'linear')
summary(dnnFinal)
dnnFinal %>% compile(
loss = 'mean_squared_error',
optimizer = 'adam',
metrics = c('mae')
)
dnnFinal.fit <- dnnFinal %>% fit(
X_train, y_train,
epochs = 150, batch_size = 50,
validation_split = 0.2
)
```

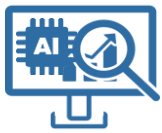


```
dnnFinal %>% evaluate(X_test, y_test)
$loss
$mean_absolute_error
dnnFinal.pred <- data.frame(y = predict(dnnFinal, as.matrix(X_test)))
dfFinal2 <- data.frame(dnnFinal.pred, X_test)
attach(dfFinal2)
dfFinal2$y[1]
X_test.df <- as.data.frame(X_test)
y_test.df <- as.data.frame(y_test)
dfFinal2$y[1]
dfFinal2$Price[1]
dfFinal2$y[1] - dfFinal2$Price[1]
dfFinal2$Year[1]
dfFinal2$Month[1]
head(dfFinal2)
plot(dfFinal2$Price, dfFinal2$y, col='green',

main='Actual vs. Predicted Price using DNN (keras)',
xlab='dfFinal2$Price (Actual)', ylab='dfFinal2$y (Predicted)',
pch=20, cex=1)
abline(0,1, lwd=0.5)
legend('bottomright', legend='Actual vs. Predicted Price - DNN (keras)', pch=20,
col='green', bty='n')

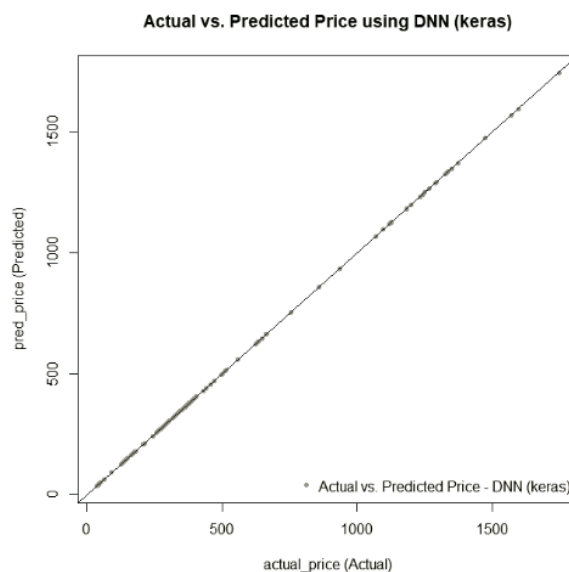
pred_price <- dfFinal2$y*(max(dfFinal$Price) - min(dfFinal$Price))+min(dfFinal$Price)
actual_price <- dfFinal2$Price*(max(dfFinal$Price) - min(dfFinal$Price))+min(dfFinal$Price)
MSE.nn <- sum((actual_price-pred_price)^2)/nrow(X_test)
MSE.nn
[1] 1.571772
plot(actual_price, pred_price, col='green',
main='Actual vs. Predicted Price
using DNN (keras)',
xlab='actual_price (Actual)', ylab='pred_price (Predicted)',
pch=20, cex=1)
abline(0,1, lwd=0.5)
legend('bottomright', legend='Actual vs. Predicted Price - DNN (keras)'
, pch=20, col='green', bty='n')

# Comparing pred with y_test or the Price response variable
dfFinal4 <- data.frame(dnnFinal.pred, y_test)
dfFinal4$y[1]
head(dfFinal4)
pred_price <- dfFinal4$y*(max(dfFinal$Price) - min(dfFinal$Price))+min(dfFinal$Price)
y_test[1]
# [1] 0.0001374446
```



```
actual_price <- y_test*(max(dfFinal$Price) - min(dfFinal$Price))+min(dfFinal$Price)
plot(actual_price, pred_price, col='green',
main='Actual vs. Predicted Price
using DNN (keras)',
xlab='actual_price (Actual)',
ylab='pred_price (Predicted)',
pch=20,cex=1)
abline(0,1,lwd=0.5)
legend('bottomright',legend='Actual vs. Predicted Price - DNN (keras)',pch=20,
col='green',bty='n')
MSE1 <- sum((actual_price-pred_price)^2)/nrow(y_test)
MSE1
# [1] 0.16877
MSE2 <- sum((actual_price-pred_price)^2)/nrow(X_test)
MSE2
# [1] 0.16877
```

Gambar 4.32 menyediakan plot R dari nilai harga aktual vs. nilai harga yang diprediksi.



Gambar 4.32 Plot DNN yang Dipasang Harga Aktual vs. Harga Prediksi.

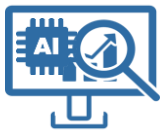
Pemberdayaan BI dan Dampaknya terhadap Perusahaan

Analisis Rekomendasi dari Sudut Pandang AI dan BI

Gambar 4.32 menunjukkan akurasi DNN yang dipasang, menunjukkan harga produk aktual versus harga prediksi dengan MSE 0,16. Berdasarkan plot DNN yang dipasang dan terintegrasi ke dalam platform BI, perusahaan dapat melakukan analisis untuk menentukan harga optimal untuk daya jual dan analisis proses bisnis terkait.

Deduksi untuk Analisis BI

Gambar 4.32 menunjukkan bahwa harga yang baru diprediksi selaras dengan garis kecocokan terbaik dan, oleh karena itu, hampir mendekati harga sebenarnya (aktual).



Meskipun terdapat beberapa overfitting, hal ini berarti model dapat digeneralisasi pada data baru, terutama dataset yang lebih besar, sehingga meningkatkan akurasi prediksi harga. Hal ini memungkinkan harga emas batangan ditetapkan dalam berbagai mata uang di berbagai belahan dunia. Hal ini menunjukkan bahwa model ini cukup akurat dan dapat digunakan untuk mempelajari dampak harga emas batangan di pasar.

4.4 SISTEM REKOMENDASI PILIHAN OPTIMAL BERDASARKAN SKOR

Pada bagian ini, kami menyajikan kasus analitik preskriptif dengan merekomendasikan sumber produk yang layak berdasarkan prediksi skor, diikuti dengan memprediksi "bagaimana kinerja produk di pasar" berdasarkan pilihan yang direkomendasikan sebuah kasus analitik prediktif, keduanya menggunakan DNN.

Metodologi AI—Analisis Preskriptif dengan Merekomendasikan Sumber yang Layak Berdasarkan Prediksi Skor (menggunakan DNN) dan Analisis Prediktif dengan Memprediksi "Bagaimana Kinerja Produk di Pasar" Berdasarkan Pilihan yang Direkomendasikan

Tujuan Kasus Penggunaan

- Sasaran

Bagian 1—Merekomendasikan produk yang layak berdasarkan peringkat setiap produk pada skala 1 hingga 10—sebuah solusi analitik preskriptif.

Bagian 2—Memprediksi bagaimana kinerja produk yang direkomendasikan di pasar berdasarkan "suka"-nya—sebuah solusi analitik prediktif.

Analisis prediktif dan preskriptif merupakan salah satu aplikasi utama AI dalam kehidupan nyata dan digunakan di berbagai industri, termasuk produk seperti ritel, keuangan (instrumen keuangan, dll.), hiburan (film, serial televisi, dll.), dan banyak lagi yang menjadi subjek bisnis atau minat BI.

- Teknologi

AI dan ML menggunakan jaringan saraf dalam (DNN).

- Implementasi

Menggunakan pustaka *neuralnet()* dan pustaka visualisasi *ggplot2()* dalam R.

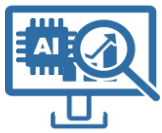
- Set Data Masukan

Set data masukan adalah set data Kaggle yang terdiri dari lima deodoran beserta berbagai atributnya dan peringkat berdasarkan penggunaan pelanggan. Data sumber untuk pelatihan terdapat dalam berkas `Data_train_reduced.csv`, yang tersedia daring. Data uji untuk penilaian dan prediksi skala terdapat dalam berkas `test_data.csv`, yang juga tersedia daring (<https://www.kaggle.com/ramkumar/deodorant-instant-liking-data/>).

Pragmatik Implementasi

Berikut adalah langkah-langkah utama yang terlibat untuk Bagian 1 dan 2 sebagaimana dinyatakan dalam tujuan kasus penggunaan di atas:

- Merekomendasikan produk yang layak berdasarkan peringkatnya pada skala 1 hingga 10, dengan terlebih dahulu memprediksi skala yang menganalisis data berdasarkan Produk sebagai prediktor utama.



- Produk dengan skala tertinggi akan dikeluarkan sebagai produk yang direkomendasikan.
- Skala diprediksi menggunakan jaringan saraf tiruan dalam (DNN) dan kemudian menggunakan nilai tersebut untuk menentukan hasil dan merekomendasikan produk tersebut.
- Untuk analitik preskriptif, hal ini memungkinkan untuk merekomendasikan produk yang layak guna memaksimalkan penjualan.
- Lebih lanjut, kami memprediksi kinerja produk di pasar berdasarkan produk yang direkomendasikan (analitik prediktif).

Pragmatik lebih lanjut dapat didefinisikan melalui kognisi dan merekomendasikan produk "optimal" untuk memaksimalkan penjualan dan dengan demikian meningkatkan persaingan. Hal ini memperkenalkan "kognisi" ke dalam model dan melatihnya menggunakan model rekomendasi berbasis komputasi kognitif untuk pengambilan keputusan dalam hal menghasilkan "produk paling optimal" yang akan digunakan—serta "memahami" mengapa sumber tertentu optimal untuk digunakan (yaitu, tindakan terbaik berikutnya)—dari dukungan keputusan hingga pengambilan keputusan.

Dua bagian berikut menjelaskan detail implementasi solusi analitik preskriptif dan prediktif.

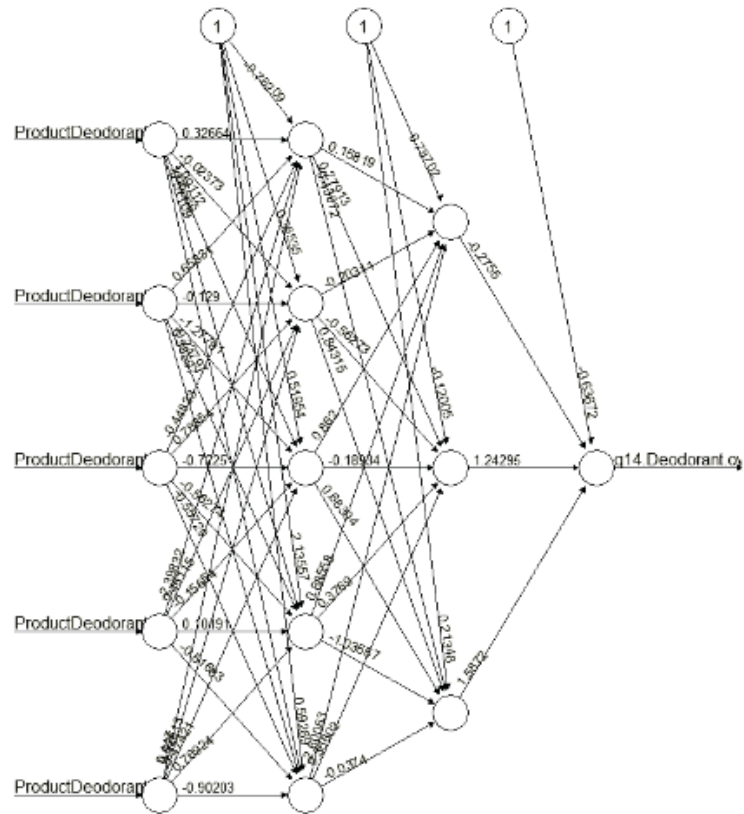
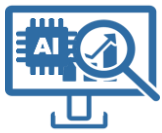
Solusi Analisis Preskriptif

Sebagaimana dijelaskan sebelumnya, solusi ini terdiri dari prediksi skor Produk pada skala 1 hingga 10 berdasarkan Produk membangun model jaringan saraf tiruan dalam berbasis AI menggunakan `neuralnet()` berbasis R. Berikut langkah-langkahnya:

- Solusi ini menggunakan data dari berkas `Data_train_reduced.csv` untuk memprediksi `"q14.Deodorant.overall.on.a.scale.from.1.to.10"` dan berdasarkan nilainya untuk merekomendasikan Produk.
- Solusi ini menggunakan model ML jaringan saraf tiruan dalam yang dibangun dengan melatih dirinya sendiri berdasarkan data yang mengkorelasikan lima produk dengan masing-masing skalanya.
- Solusi ini kemudian memprediksi skala pada data uji dan, berdasarkan nilainya, merekomendasikan Produk tersebut.
- Sebuah DNN dengan satu lapisan masukan dengan lima simpul masukan (neuron), dua lapisan tengah (tersembunyi) (masing-masing dengan lima dan tiga simpul), dan satu lapisan keluaran (dengan satu simpul) dibangun dan dilatih.
- Kemudian, data tersebut dicocokkan dengan menerapkan keluaran model yang diprediksi terhadap masing-masing dari lima Produk. Variabel prediktor yang digunakan adalah 'Produk', dan variabel respons yang digunakan adalah skala.

Solusi analitik preskriptif diimplementasikan menggunakan model R `neuralnet()` untuk membangun dan melatih DNN, kemudian menggunakan fungsi `compute()` untuk memprediksi keluaran dan menggunakannya untuk rekomendasi. Plot DNN yang dicocokkan untuk analitik preskriptif ditunjukkan pada Gambar 4.33.

Daftar 4.8 dan 4.9 menunjukkan kode R dan keluarannya untuk solusi ini.

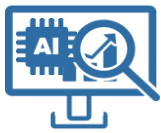


Error: 96.613473 Steps: 45

Gambar 4.33 Plot DNN untuk Analisis Preskriptif

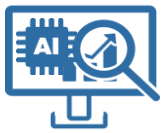
Daftar 4.8 Solusi Analisis Preskriptif Kode R

```
setwd("<directory-name>")
product_data <- read.csv("Data_train_reduced.csv")
names(product_data)
test_data <- read.csv("<directory-name>test_data.csv")
names(test_data)
nrow(product_data)
nrow(test_data)
head(product_data)
product_dataF <- product_data[, c(1:3,5:37,39,42:44,47:64)]
Product.matrix <- model.matrix(~Product - 1, data = product_dataF)
product_dataF <- data.frame(product_dataF, Product.matrix)
```



```
head(product_dataF)
ncol(product_dataF)
maxs <- apply(product_dataF[, c(1:2,4:58)], 2, max)
mins <- apply(product_dataF[, c(1:2,4:58)], 2, min)
scaled <- as.data.frame(scale(product_dataF[, c(1:2,4:58)], center = mins, scale =
maxs - mins))
scaledF <- as.data.frame(cbind(scaled, product_dataF[, c(59:63)]))
set.seed(500)
library(neuralnet)
library(ggplot2)
head(scaledF) # had some NA and NaN
scaledFF <- scaledF[, c(1:32,34:35,37,39:40,42:48,50:62)]
head(scaledFF)
ncol(scaledFF) # 57 # 43rd column is scale1.to.10
scaledFF <- scaledFF[, c(1:42,44:57,43)]
ncol(scaledFF) #57
head(scaledFF) # last column (57th) is scale1.to.10
n <- names(scaledFF)
f <- as.formula(q14.Deodorant.overall.on.a.scale.from.1.to.10 ~ ProductDeodorant.B +
+ ProductDeodorant.F + ProductDeodorant.G + ProductDeodorant.H +
+ ProductDeodorant.J)
f
DNNf3 <- neuralnet(f, data = scaledFF, hidden=c(5,3), linear.output=TRUE)
plot(DNNf3)
ncol(scaledFF) # 57
head(test_data)
nrow(test_data)
test.Product.matrix <- model.matrix( ~Product - 1, data = test_data)
test_dataF <- data.frame(test_data, test.Product.matrix)
ncol(test_dataF) # 63
head(test_dataF)

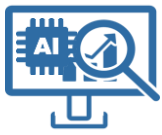
maxs <- apply(test_dataF[, c(1:2,4:58)], 2, max)
mins <- apply(test_dataF[, c(1:2,4:58)], 2, min)
scaledTest <- as.data.frame(scale(test_dataF[,c(1:2, 4:58)], center = mins, scale =
maxs - mins))
set.seed(500)
n1 <- names(scaledTest)
head(scaledTest)
ncol(scaledTest) # 57
scaledTestF <- as.data.frame(cbind(scaledTest, test_dataF[, c(59:63)]))
ncol(scaledTestF)
scaledTestFF <- scaledTestF[, c(1:32, 34:35, 37,39:40,42:48,50:57,58:62)]
ncol(scaledTestFF)
n2 <- names(scaledTestFF)
head(test_data)
head(scaledTestFF)
pr.DNNf3 <- compute(DNNf3, scaledTestFF[, c(53:57)])
```



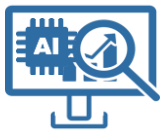
```
pr.DNNf3_r <- pr.DNNf3$net.result*(max(test_data$q14.Deodorant.overall.
on.a.scale.from.1.to.10)-min(test_data$q14.Deodorant.overall.on.a.scale.
from.1.to.10))+min(test_data$q14.Deodorant.overall.on.a.scale.from.1.to.10)
test_data_r <- (scaledTestFF$q14.Deodorant.overall.on.a.scale.
from.1.to.10)*(max(test_data$q14.Deodorant.overall.on.a.scale.from.1.to.10)-
min(test_data$q14.Deodorant.overall.on.a.scale.from.1.to.10))+min(test_data$q14.
Deodorant.overall.on.a.scale.from.1.to.10)
MSE.DNNf3 <- sum((test_data_r - pr.DNNf3_r)^2)/nrow(scaledTestFF)
MSE.DNNf3
# [1] 6.400654791
plot(test_data$q14.Deodorant.overall.on.a.scale.from.1.to.10, pr.DNNf3_r,col='red',
main='Actual vs predicted DNN',pch=18,cex=0.7)
abline(0,1,lwd=2)
legend('bottomright',legend='DNN',pch=18,col='red',bty='n')
pr.DNNf3_r2 <- pr.DNNf3$net.result
pdf(" Plots ProductDeodorants vs Predicted.pdf")
plot(test_dataF$ProductDeodorant.B, pr.DNNf3_r2, col='green',main='Product
Deodorant B vs Predicted', xlab='Product Deodorant B',pch=18,cex=2)
abline(0,1,lwd=2)
legend('bottom',legend='Product Deodorant B vs Predicted',pch=2,col='green',bty='n')
plot(test_dataF$ProductDeodorant.F, pr.DNNf3_r2, col='green',main='Product
Deodorant F vs Predicted', xlab='Product Deodorant F',pch=18,cex=2)
abline(0,1,lwd=2)
legend('bottom',legend='Product Deodorant F vs Predicted',pch=2,col='green',bty='n')
plot(test_dataF$ProductDeodorant.G, pr.DNNf3_r2, col='green',main='Product
Deodorant G vs Predicted', xlab='Product Deodorant G',pch=18,cex=2)
abline(0,1,lwd=2)
legend('bottom',legend='Product Deodorant G vs Predicted',pch=2,col='green',bty='n')
plot(test_dataF$ProductDeodorant.H, pr.DNNf3_r2, col='green',main='Product Deodorant
H vs Predicted', xlab='Product Deodorant H',pch=18,cex=2)
abline(0,1,lwd=2)
legend('bottom',legend='Product Deodorant H vs Predicted',pch=2,col='green',bty='n')
plot(test_dataF$ProductDeodorant.J, pr.DNNf3_r2, col='green',main='Product Deodorant
J vs Predicted', xlab='Product Deodorant J',pch=18,cex=2)
abline(0,1,lwd=2)
legend('bottom',legend='Product Deodorant J vs Predicted',pch=2,col='green',bty='n')
dev.off()
```

Daftar 4.9 Output Kode R pada Daftar 4.8

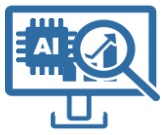
```
> setwd("<directory-name>")
> product_data <- read.csv("Data_train_reduced.csv")
> names(product_data)
[1] "Respondent.ID"
[2] "Product.ID"
[3] "Product"
[4] "Instant.Liking"
[5] "q1_1.personal.opinion.of.this.Deodorant"
```



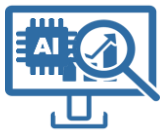
- [6] "q2_all.words"
- [7] "q3_1.strength.of.the.Deodorant"
- [8] "q4_1.artificial.chemical"
- [9] "q4_2.attractive"
- [10] "q4_3.bold"
- [11] "q4_4.boring"
- [12] "q4_5.casual"
- [13] "q4_6.cheap"
- [14] "q4_7.clean"
- [15] "q4_8.easy.to.wear"
- [16] "q4_9.elegant"
- [17] "q4_10.feminine"
- [18] "q4_11.for.someone.like.me"
- [19] "q4_12.heavy"
- [20] "q4_13.high.quality"
- [21] "q4_14.long.lasting"
- [22] "q4_15.masculine"
- [23] "q4_16.memorable"
- [24] "q4_17.natural"
- [25] "q4_18.old.fashioned"
- [26] "q4_19.ordinary"
- [27] "q4_20.overpowering"
- [28] "q4_21.sharp"
- [29] "q4_22.sophisticated"
- [30] "q4_23.upscale"



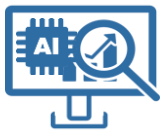
```
[31] "q4_24.well.rounded"
[32] "q5_1.Deodorant.is.addictive"
[33] "q7"
[34] "q8.1"
[35] "q8.2"
[36] "q8.5"
[37] "q8.6"
[38] "q8.7"
[39] "q8.8"
[40] "q8.9"
[41] "q8.10"
[42] "q8.11"
[43] "q8.12"
[44] "q8.13"
[45] "q8.17"
[46] "q8.18"
[47] "q8.19"
[48] "q8.20"
[49] "q9.how.likely.would.you.be.to.purchase.this.Deodorant"
[50] "q10.prefer.this.Deodorant.or.your.usual.Deodorant"
[51] "q11.time.of.day.would.this.Deodorant.be.appropriate"
[52] "q12.which.occasions.would.this.Deodorant.be.appropriate"
[53] "Q13_Liking.after.30.minutes"
[54] "q14.Deodorant.overall.on.a.scale.from.1.to.10"
[55] "ValSegb"
[56] "s7.involved.in.the.selection.of.the.cosmetic.products"
[57] "s8.ethnic.background"
[58] "s9.education"
[59] "s10.income"
[60] "s11.marital.status"
[61] "s12.working.status"
[62] "s13.2"
[63] "s13a.b.most.often"
[64] "s13b.bottles.of.Deodorant.do.you.currently.own"
> test_data <- read.csv("<directory-name>test_data.csv")
> names(test_data)
[1] "Respondent.ID"
[2] "Product.ID"
[3] "Product"
```



- [4] "q1_1.personal.opinion.of.this.Deodorant"
- [5] "q2_all.words"
- [6] "q3_1.strength.of.the.Deodorant"
- [7] "q4_1.artificial.chemical"
- [8] "q4_2.attractive"
- [9] "q4_3.bold"
- [10] "q4_4.boring"
- [11] "q4_5.casual"
- [12] "q4_6.cheap"
- [13] "q4_7.clean"
- [14] "q4_8.easy.to.wear"
- [15] "q4_9.elegant"
- [16] "q4_10.feminine"
- [17] "q4_11.for.someone.like.me"
- [18] "q4_12.heavy"
- [19] "q4_13.high.quality"
- [20] "q4_14.long.lasting"
- [21] "q4_15.masculine"
- [22] "q4_16.memorable"
- [23] "q4_17.natural"
- [24] "q4_18.old.fashioned"
- [25] "q4_19.ordinary"
- [26] "q4_20.overpowering"
- [27] "q4_21.sharp"
- [28] "q4_22.sophisticated"
- [29] "q4_23.upscale"
- [30] "q4_24.well.rounded"
- [31] "q5_1.Deodorant.is.addictive"
- [32] "q7"
- [33] "q8.1"
- [34] "q8.2"
- [35] "q8.5"
- [36] "q8.6"
- [37] "q8.8"
- [38] "q8.11"
- [39] "q8.12"
- [40] "q8.13"
- [41] "q8.19"
- [42] "q8.20"



```
[43] "q9.how.likely.would.you.be.to.purchase.this.Deodorant"
[44] "q10.prefer.this.Deodorant.or.your.usual.Deodorant"
[45] "q11.time.of.day.would.this.Deodorant.be.appropriate"
[46] "q12.which.occasions.would.this.Deodorant.be.appropriate"
[47] "Q13_Liking.after.30.minutes"
[48] "q14.Deodorant.overall.on.a.scale.from.1.to.10"
[49] "ValSegb"
[50] "s7.involved.in.the.selection.of.the.cosmetic.products"
[51] "s8.ethnic.background"
[52] "s9.education"
[53] "s10.income"
[54] "s11.marital.status"
[55] "s12.working.status"
[56] "s13.2"
[57] "s13a.b.most.often"
[58] "s13b.bottles.of.Deodorant.do.you.currently.own"
> nrow(product_data)
[1] 2500
> nrow(test_data)
[1] 5105
> head(product_data)
  Respondent.ID Product.ID      Product Instant.Liking
1           3800         121 Deodorant B              1
2           3801         121 Deodorant B              0
3           3802         121 Deodorant B              0
4           3803         121 Deodorant B              1
5           3804         121 Deodorant B              1
6           3805         121 Deodorant B              0
  q1_1.personal.opinion.of.this.Deodorant q2_all.words
1                      4                      1
2                      5                      1
3                      6                      1
4                      4                      0
5                      4                      1
6                      5                      1
```

4				4	3	0	0	0	0	NA	0	NA	NA	0
5				3	2	0	0	0	0	NA	0	NA	NA	0
6				1	3	0	0	0	0	NA	0	NA	NA	0

	q8.12	q8.13	q8.17	q8.18	q8.19	q8.20
1	0	0	NA	NA	0	0
2	0	0	NA	NA	0	0
3	0	0	NA	NA	0	0
4	0	0	NA	NA	0	0
5	0	0	NA	NA	0	0
6	0	0	NA	NA	0	0

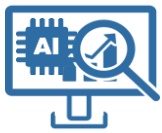
q9.how.likely.would.you.be.to.purchase.this.Deodorant	
1	2
2	3
3	5
4	5
5	5
6	5

q10.prefer.this.Deodorant.or.your.usual.Deodorant	
1	1
2	5
3	1
4	4
5	3
6	4

q11.time.of.day.would.this.Deodorant.be.appropriate	
1	1
2	3
3	3
4	1
5	3
6	2

q12.which.occasions.would.this.Deodorant.be.appropriate	
1	2
2	3
3	3
4	3
5	2
6	2

Q13_Liking.after.30.minutes q14.Deodorant.overall.on.a.scale.from.1.to.10		
1	1	7
2	3	8
3	2	5
4	6	8
5	5	4
6	6	7



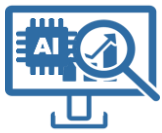
```
ValSegb s7.involved.in.the.selection.of.the.cosmetic.products
1      7      4
2      6      4
3      7      4
4      1      4
5      4      4
6      7      4

s8.ethnic.background s9.education s10.income s11.marital.status
1      1      4      3      1
2      1      4      3      1
3      1      3      5      1
4      1      4      9      1
5      1      3      5      1
6      1      4      5      2

s12.working.status s13.2 s13a.b.most.often
1      1      0      0
2      1      0      0
3      1      0      0
4      3      0      0
5      2      0      0
6      2      0      0

s13b.bottles.of.Deodorant.do.you.currently.own
1      3
2      4
3      2
4      3
5      3
6      2

> product_dataF <- product_data[, c(1:3,5:37,39,42:44,47:64)]
> Product.matrix <-model.matrix( ~Product - 1, data = product_dataF)
> product_dataF <- data.frame(product_dataF, Product.matrix)
> head(product_dataF)
  Respondent.ID Product.ID      Product q1_1.personal.opinion.of.this.Deodorant
1          3800         121 Deodorant B                                     4
2          3801         121 Deodorant B                                     5
3          3802         121 Deodorant B                                     6
4          3803         121 Deodorant B                                     4
5          3804         121 Deodorant B                                     4
6          3805         121 Deodorant B                                     5
```



	q2_all.words	q3_1.strength.of.the.Deodorant	q4_1.artificial.chemical	
1	1		4	2
2	1		4	4
3	1		3	2
4	0		4	5
5	1		2	1
6	1		5	5

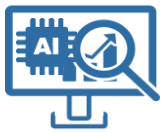
	q4_2.attractive	q4_3.bold	q4_4.boring	q4_5.casual	q4_6.cheap	q4_7.clean
1	5	4	2	3	5	5
2	2	2	1	3	2	4
3	5	2	4	2	4	3
4	5	4	3	5	2	5
5	3	1	1	3	3	5
6	5	2	1	4	5	5

	q4_8.easy.to.wear	q4_9.elegant	q4_10.feminine	q4_11.for.someone.like.me	
1	5	4	5		3
2	4	4	3		1
3	5	4	4		4
4	3	5	5		5
5	3	5	5		5
6	2	4	4		3

	q4_12.heavy	q4_13.high.quality	q4_14.long.lasting	q4_15.masculine	
1	1	5	1	2	
2	1	3	4	4	
3	3	1	2	1	
4	1	4	3	3	
5	1	4	4	2	
6	2	4	5	2	

	q4_16.memorable	q4_17.natural	q4_18.old.fashioned	q4_19.ordinary	
1	4	5	4	5	
2	5	3	3	4	
3	4	2	4	3	
4	5	5	4	2	
5	3	5	1	2	
6	3	4	2	2	

	q4_20.overpowering	q4_21.sharp	q4_22.sophisticated	q4_23.upscale	
1	1	1	4	1	
2	2	2	5	4	
3	2	5	4	4	
4	5	3	3	5	
5	4	2	3	1	
6	3	1	5	5	



	q4_24.well.rounded	q5_1.Deodorant.is.addictive	q7	q8.1	q8.2	q8.5	q8.6	q8.8
1	4		1	1	0	0	0	1
2	4		4	4	0	0	0	1
3	3		4	3	0	0	0	0
4	4		4	3	0	0	0	0
5	5		3	2	0	0	0	0
6	4		1	3	0	0	0	0

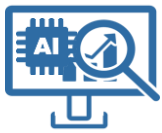
	q8.11	q8.12	q8.13	q8.19	q8.20
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0

4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0

	q9.how.likely.would.you.be.to.purchase.this.Deodorant
1	2
2	3
3	5
4	5
5	5
6	5

	q10.prefer.this.Deodorant.or.your.usual.Deodorant
1	1
2	5
3	1
4	4
5	3
6	4

	q11.time.of.day.would.this.Deodorant.be.appropriate
1	1
2	3
3	3
4	1
5	3
6	2



```

q12.which.occasions.would.this.Deodorant.be.appropriate
1 2
2 3
3 3
4 3
5 2
6 2

Q13_Liking.after.30.minutes q14.Deodorant.overall.on.a.scale.from.1.to.10
1 1 7
2 3 8
3 2 5
4 6 8
5 5 4
6 6 7

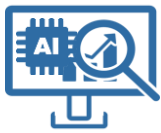
ValSegb s7.involved.in.the.selection.of.the.cosmetic.products
1 7 4
2 6 4
3 7 4
4 1 4
5 4 4
6 7 4

s8.ethnic.background s9.education s10.income s11.marital.status
1 1 4 3 1
2 1 4 3 1
3 1 3 5 1
4 1 4 9 1
5 1 3 5 1
6 1 4 5 2

s12.working.status s13.2 s13a.b.most.often
1 1 0 0
2 1 0 0
3 1 0 0
4 3 0 0
5 2 0 0
6 2 0 0

s13b.bottles.of.Deodorant.do.you.currently.own ProductDeodorant.B
1 3 1
2 4 1
3 2 1
4 3 1
5 3 1
6 2 1

```

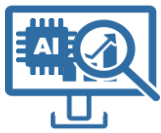


```
ProductDeodorant.F ProductDeodorant.G ProductDeodorant.H ProductDeodorant.J
1 0 0 0 0
2 0 0 0 0
3 0 0 0 0
4 0 0 0 0
5 0 0 0 0
6 0 0 0 0
> ncol(product_dataF)
[1] 63
> maxs <- apply(product_dataF[, c(1:2,4:58)], 2, max)
> mins <- apply(product_dataF[, c(1:2,4:58)], 2, min)
> scaled <- as.data.frame(scale(product_dataF[, c(1:2,4:58)], center = mins, scale
= maxs - mins))
> scaledF <- as.data.frame(cbind(scaled, product_dataF[, c(59:63)]))
> set.seed(500)
> library(neuralnet)
Warning message:
package 'neuralnet' was built under R version 3.5.0
> library(ggplot2)
Warning message:
package 'ggplot2' was built under R version 3.4.4
> head(scaledF) # had some NA and NaN
  Respondent.ID Product.ID q1_1.personal.opinion.of.this.Deodorant q2_all.words
1 0.000000e+00 0 0.5000000 0.2
2 9.709681e-05 0 0.6666667 0.2
3 1.941936e-04 0 0.8333333 0.2

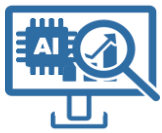
4 2.912904e-04 0 0.5000000 0.0
5 3.883872e-04 0 0.5000000 0.2
6 4.854840e-04 0 0.6666667 0.2

  q3_1.strength.of.the.Deodorant q4_1.artificial.chemical q4_2.attractive
1 0.75 0.25 1.00
2 0.75 0.75 0.25
3 0.50 0.25 1.00
4 0.75 1.00 1.00
5 0.25 0.00 0.50
6 1.00 1.00 1.00

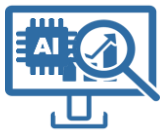
  q4_3.bold q4_4.boring q4_5.casual q4_6.cheap q4_7.clean q4_8.easy.to.wear
1 0.75 0.25 0.50 1.00 1.00 1.00
2 0.25 0.00 0.50 0.25 0.75 0.75
3 0.25 0.75 0.25 0.75 0.50 1.00
4 0.75 0.50 1.00 0.25 1.00 0.50
5 0.00 0.00 0.50 0.50 1.00 0.50
6 0.25 0.00 0.75 1.00 1.00 0.25
```



	q4_9.elegant	q4_10.feminine	q4_11.for.someone.like.me	q4_12.heavy					
1	0.75	1.00	0.50	0.00					
2	0.75	0.50	0.00	0.00					
3	0.75	0.75	0.75	0.50					
4	1.00	1.00	1.00	0.00					
5	1.00	1.00	1.00	0.00					
6	0.75	0.75	0.50	0.25					
	q4_13.high.quality	q4_14.long.lasting	q4_15.masculine	q4_16.memorable					
1	1.00	0.00	0.25	0.75					
2	0.50	0.75	0.75	1.00					
3	0.00	0.25	0.00	0.75					
4	0.75	0.50	0.50	1.00					
5	0.75	0.75	0.25	0.50					
6	0.75	1.00	0.25	0.50					
	q4_17.natural	q4_18.old.fashioned	q4_19.ordinary	q4_20.overpowering					
1	1.00	0.75	1.00	0.00					
2	0.50	0.50	0.75	0.25					
3	0.25	0.75	0.50	0.25					
4	1.00	0.75	0.25	1.00					
5	1.00	0.00	0.25	0.75					
6	0.75	0.25	0.25	0.50					
	q4_21.sharp	q4_22.sophisticated	q4_23.upscale	q4_24.well.rounded					
1	0.00	0.75	0.00	0.75					
2	0.25	1.00	0.75	0.75					
3	1.00	0.75	0.75	0.50					
4	0.50	0.50	1.00	0.75					
5	0.25	0.50	0.00	1.00					
6	0.00	1.00	1.00	0.75					
	q5_1.Deodorant.is.addictive	q7	q8.1	q8.2	q8.5	q8.6	q8.8	q8.11	q8.12
1	0.00	0.4545455	0	NA	0	0	NA	0	NA
2	0.75	0.7272727	0	NA	0	1	NA	0	NA
3	0.75	0.6363636	0	NA	0	0	NA	0	NA
4	0.75	0.6363636	0	NA	0	0	NA	0	NA
5	0.50	0.5454545	0	NA	0	0	NA	0	NA
6	0.00	0.6363636	0	NA	0	0	NA	0	NA
	q8.13	q8.19	q8.20	q9.how.likely.would.you.be.to.purchase.this.Deodorant					
1	0	0	NA	0.25					
2	0	0	NA	0.50					
3	0	0	NA	1.00					
4	0	0	NA	1.00					
5	0	0	NA	1.00					
6	0	0	NA	1.00					



q10.prefer.this.Deodorant.or.your.usual.Deodorant				
1	0.00			
2	1.00			
3	0.00			
4	0.75			
5	0.50			
6	0.75			
q11.time.of.day.would.this.Deodorant.be.appropriate				
1	0.0			
2	1.0			
3	1.0			
4	0.0			
5	1.0			
6	0.5			
q12.which.occasions.would.this.Deodorant.be.appropriate				
1	0.5			
2	1.0			
3	1.0			
4	1.0			
5	0.5			
6	0.5			
Q13_Liking.after.30.minutes q14.Deodorant.overall.on.a.scale.from.1.to.10				
1	0.0000000	0.6666667		
2	0.3333333	0.7777778		
3	0.1666667	0.4444444		
4	0.8333333	0.7777778		
5	0.6666667	0.3333333		
6	0.8333333	0.6666667		
ValSegb s7.involved.in.the.selection.of.the.cosmetic.products				
1	1.0000000	NaN		
2	0.8333333	NaN		
3	1.0000000	NaN		
4	0.0000000	NaN		
5	0.5000000	NaN		
6	1.0000000	NaN		
s8.ethnic.background s9.education s10.income s11.marital.status				
1	0	0.4	0.125	0.00
2	0	0.4	0.125	0.00
3	0	0.2	0.375	0.00
4	0	0.4	0.875	0.00
5	0	0.2	0.375	0.00
6	0	0.4	0.375	0.25



```

s12.working.status s13.2 s13a.b.most.often
1      0.0000000    0      0
2      0.0000000    0      0
3      0.0000000    0      0
4      0.3333333    0      0
5      0.1666667    0      0
6      0.1666667    0      0

s13b.bottles.of.Deodorant.do.you.currently.own ProductDeodorant.B
1      0.4      1
2      0.6      1
3      0.2      1
4      0.4      1
5      0.4      1
6      0.2      1

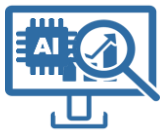
ProductDeodorant.F ProductDeodorant.G ProductDeodorant.H ProductDeodorant.J
1      0      0      0      0
2      0      0      0      0
3      0      0      0      0
4      0      0      0      0
5      0      0      0      0
6      0      0      0      0

> scaledFF <- scaledF[, c(1:32,34:35,37,39:40,42:48,50:62)]
> head(scaledFF)
Respondent.ID Product.ID q1_1.personal.opinion.of.this.Deodorant q2_all.words
1 0.000000e+00      0      0.5000000      0.2
2 9.709681e-05      0      0.6666667      0.2
3 1.941936e-04      0      0.8333333      0.2
4 2.912904e-04      0      0.5000000      0.0
5 3.883872e-04      0      0.5000000      0.2
6 4.854840e-04      0      0.6666667      0.2

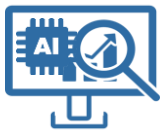
q3_1.strength.of.the.Deodorant q4_1.artificial.chemical q4_2.attractive
1      0.75      0.25      1.00
2      0.75      0.75      0.25
3      0.50      0.25      1.00
4      0.75      1.00      1.00
5      0.25      0.00      0.50
6      1.00      1.00      1.00

q4_3.bold q4_4.boring q4_5.casual q4_6.cheap q4_7.clean q4_8.easy.to.wear
1      0.75      0.25      0.50      1.00      1.00      1.00
2      0.25      0.00      0.50      0.25      0.75      0.75
3      0.25      0.75      0.25      0.75      0.50      1.00
4      0.75      0.50      1.00      0.25      1.00      0.50
5      0.00      0.00      0.50      0.50      1.00      0.50
6      0.25      0.00      0.75      1.00      1.00      0.25

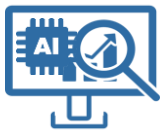
```



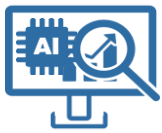
	q4_9.elegant	q4_10.feminine	q4_11.for.someone.like.me	q4_12.heavy				
1	0.75	1.00		0.50				
2	0.75	0.50		0.00				
3	0.75	0.75		0.75				
4	1.00	1.00		1.00				
5	1.00	1.00		1.00				
6	0.75	0.75		0.50				
	q4_13.high.quality	q4_14.long.lasting	q4_15.masculine	q4_16.memorable				
1	1.00	0.00	0.25	0.75				
2	0.50	0.75	0.75	1.00				
3	0.00	0.25	0.00	0.75				
4	0.75	0.50	0.50	1.00				
5	0.75	0.75	0.25	0.50				
6	0.75	1.00	0.25	0.50				
	q4_17.natural	q4_18.old.fashioned	q4_19.ordinary	q4_20.overpowering				
1	1.00	0.75	1.00	0.00				
2	0.50	0.50	0.75	0.25				
3	0.25	0.75	0.50	0.25				
4	1.00	0.75	0.25	1.00				
5	1.00	0.00	0.25	0.75				
6	0.75	0.25	0.25	0.50				
	q4_21.sharp	q4_22.sophisticated	q4_23.upscale	q4_24.well.rounded				
1	0.00	0.75	0.00	0.75				
2	0.25	1.00	0.75	0.75				
3	1.00	0.75	0.75	0.50				
4	0.50	0.50	1.00	0.75				
5	0.25	0.50	0.00	1.00				
6	0.00	1.00	1.00	0.75				
	q5_1.Deodorant.is.addictive	q7	q8.1	q8.5	q8.6	q8.11	q8.13	q8.19
1	0.00	0.4545455	0	0	0	0	0	0
2	0.75	0.7272727	0	0	1	0	0	0
3	0.75	0.6363636	0	0	0	0	0	0
4	0.75	0.6363636	0	0	0	0	0	0
5	0.50	0.5454545	0	0	0	0	0	0
6	0.00	0.6363636	0	0	0	0	0	0
	q9.how.likely.would.you.be.to.purchase.this.Deodorant							
1	0.25							
2	0.50							
3	1.00							
4	1.00							
5	1.00							
6	1.00							



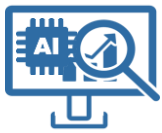
q10.prefer.this.Deodorant.or.your.usual.Deodorant					
1	0.00				
2	1.00				
3	0.00				
4	0.75				
5	0.50				
6	0.75				
q11.time.of.day.would.this.Deodorant.be.appropriate					
1	0.0				
2	1.0				
3	1.0				
4	0.0				
5	1.0				
6	0.5				
q12.which.occasions.would.this.Deodorant.be.appropriate					
1	0.5				
2	1.0				
3	1.0				
4	1.0				
5	0.5				
6	0.5				
Q13_Liking.after.30.minutes					
1	0.0000000				
2	0.3333333				
3	0.1666667				
4	0.8333333				
5	0.6666667				
6	0.8333333				
q14.Deodorant.overall.on.a.scale.from.1.to.10					
1	0.6666667				
2	0.7777778				
3	0.4444444				
4	0.7777778				
5	0.3333333				
6	0.6666667				
ValSegb s8.ethnic.background s9.education s10.income s11.marital.status					
1	1.0000000	0	0.4	0.125	0.00
2	0.8333333	0	0.4	0.125	0.00
3	1.0000000	0	0.2	0.375	0.00
4	0.0000000	0	0.4	0.875	0.00
5	0.5000000	0	0.2	0.375	0.00
6	1.0000000	0	0.4	0.375	0.25
s12.working.status s13.2 s13a.b.most.often					
1	0.0000000	0	0		
2	0.0000000	0	0		
3	0.0000000	0	0		
4	0.3333333	0	0		
5	0.1666667	0	0		
6	0.1666667	0	0		



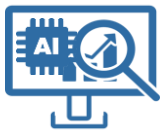
```
s13b.bottles.of.Deodorant.do.you.currently.own ProductDeodorant.B
1          0.4          1
2          0.6          1
3          0.2          1
4          0.4          1
5          0.4          1
6          0.2          1
ProductDeodorant.F ProductDeodorant.G ProductDeodorant.H ProductDeodorant.J
1          0          0          0          0
2          0          0          0          0
3          0          0          0          0
4          0          0          0          0
5          0          0          0          0
6          0          0          0          0
> ncol(scaledFF) # 57 # 43rd column is scaled.to.10
[1] 57
> scaledFF <- scaledFF[, c(1:42,44:57,43)]
> ncol(scaledFF) #57
[1] 57
> head(scaledFF) # last column (57th) is scaled.to.10
Respondent.ID Product.ID q1_1.personal.opinion.of.this.Deodorant q2_all.words
1 0.000000e+00      0          0.5000000          0.2
2 9.709681e-05      0          0.6666667          0.2
3 1.941936e-04      0          0.8333333          0.2
4 2.912904e-04      0          0.5000000          0.0
5 3.883872e-04      0          0.5000000          0.2
6 4.854840e-04      0          0.6666667          0.2
q3_1.strength.of.the.Deodorant q4_1.artificial.chemical q4_2.attractive
1          0.75          0.25          1.00
2          0.75          0.75          0.25
3          0.50          0.25          1.00
4          0.75          1.00          1.00
5          0.25          0.00          0.50
6          1.00          1.00          1.00
q4_3.bold q4_4.boring q4_5.casual q4_6.cheap q4_7.clean q4_8.easy.to.wear
1 0.75 0.25 0.50 1.00 1.00 1.00
2 0.25 0.00 0.50 0.25 0.75 0.75
3 0.25 0.75 0.25 0.75 0.50 1.00
4 0.75 0.50 1.00 0.25 1.00 0.50
5 0.00 0.00 0.50 0.50 1.00 0.50
6 0.25 0.00 0.75 1.00 1.00 0.25
q4_9.elegant q4_10.feminine q4_11.for.someone.like.me q4_12.heavy
1 0.75 1.00          0.50 0.00
2 0.75 0.50          0.00 0.00
3 0.75 0.75          0.75 0.50
```



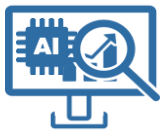
4	1.00	1.00	1.00	0.00				
5	1.00	1.00	1.00	0.00				
6	0.75	0.75	0.50	0.25				
	q4_13.high.quality q4_14.long.lasting q4_15.masculine q4_16.memorable							
1	1.00	0.00	0.25	0.75				
2	0.50	0.75	0.75	1.00				
3	0.00	0.25	0.00	0.75				
4	0.75	0.50	0.50	1.00				
5	0.75	0.75	0.25	0.50				
6	0.75	1.00	0.25	0.50				
	q4_17.natural q4_18.old.fashioned q4_19.ordinary q4_20.overpowering							
1	1.00	0.75	1.00	0.00				
2	0.50	0.50	0.75	0.25				
3	0.25	0.75	0.50	0.25				
4	1.00	0.75	0.25	1.00				
5	1.00	0.00	0.25	0.75				
6	0.75	0.25	0.25	0.50				
	q4_21.sharp q4_22.sophisticated q4_23.upscale q4_24.well.rounded							
1	0.00	0.75	0.00	0.75				
2	0.25	1.00	0.75	0.75				
3	1.00	0.75	0.75	0.50				
4	0.50	0.50	1.00	0.75				
5	0.25	0.50	0.00	1.00				
6	0.00	1.00	1.00	0.75				
	q5_1.Deodorant.is.addictive q7 q8.1 q8.5 q8.6 q8.11 q8.13 q8.19							
1	0.00	0.4545455	0	0	0	0	0	0
2	0.75	0.7272727	0	0	1	0	0	0
3	0.75	0.6363636	0	0	0	0	0	0
4	0.75	0.6363636	0	0	0	0	0	0
5	0.50	0.5454545	0	0	0	0	0	0
6	0.00	0.6363636	0	0	0	0	0	0
	q9.how.likely.would.you.be.to.purchase.this.Deodorant							
1				0.25				
2				0.50				
3				1.00				
4				1.00				
5				1.00				
6				1.00				



q10.prefer.this.Deodorant.or.your.usual.Deodorant					
1					0.00
2					1.00
3					0.00
4					0.75
5					0.50
6					0.75
q11.time.of.day.would.this.Deodorant.be.appropriate					
1					0.0
2					1.0
3					1.0
4					0.0
5					1.0
6					0.5
q12.which.occasions.would.this.Deodorant.be.appropriate					
1					0.5
2					1.0
3					1.0
4					1.0
5					0.5
6					0.5
Q13_Liking.after.30.minutes	ValSegb	s8.ethnic.background	s9.education		
1	0.0000000	1.0000000	0		0.4
2	0.3333333	0.8333333	0		0.4
3	0.1666667	1.0000000	0		0.2
4	0.8333333	0.0000000	0		0.4
5	0.6666667	0.5000000	0		0.2
6	0.8333333	1.0000000	0		0.4
s10.income	s11.marital.status	s12.working.status	s13.2	s13a.b.most.often	
1	0.125	0.00	0.0000000	0	0
2	0.125	0.00	0.0000000	0	0
3	0.375	0.00	0.0000000	0	0
4	0.875	0.00	0.3333333	0	0
5	0.375	0.00	0.1666667	0	0
6	0.375	0.25	0.1666667	0	0
s13b.bottles.of.Deodorant.do.you.currently.own ProductDeodorant.B					
1				0.4	1
2				0.6	1
3				0.2	1
4				0.4	1
5				0.4	1
6				0.2	1



```
ProductDeodorant.F ProductDeodorant.G ProductDeodorant.H ProductDeodorant.J
1          0          0          0          0
2          0          0          0          0
3          0          0          0          0
4          0          0          0          0
5          0          0          0          0
6          0          0          0          0
q14.Deodorant.overall.on.a.scale.from.1.to.10
1          0.6666667
2          0.7777778
3          0.4444444
4          0.7777778
5          0.3333333
6          0.6666667
> n <- names(scaledFF)
> f <- as.formula(q14.Deodorant.overall.on.a.scale.from.1.to.10 ~ ProductDeodorant.B +
+ + ProductDeodorant.F + ProductDeodorant.G + ProductDeodorant.H +
+ + + ProductDeodorant.J)
> f
q14.Deodorant.overall.on.a.scale.from.1.to.10 ~ ProductDeodorant.B +
++ProductDeodorant.F + ProductDeodorant.G + ProductDeodorant.H +
++ProductDeodorant.J
> DNNf3 <- neuralnet(f, data = scaledFF, hidden=c(5,3), linear.output=TRUE)
> plot(DNNf3)
> ncol(scaledFF) # 57
[1] 57
> head(test_data)
  Respondent.ID Product.ID      Product q1_1.personal.opinion.of.this.Deodorant
1          4300      121 Deodorant B              7
2          4301      121 Deodorant B              6
3          4302      121 Deodorant B              6
4          4303      121 Deodorant B              5
5          4304      121 Deodorant B              6
6          4305      121 Deodorant B              7
q2_all.words q3_1.strength.of.the.Deodorant q4_1.artificial.chemical
1          2          3          3
2          0          3          2
3          1          3          3
4          1          5          3
5          2          3          2
6          4          3          2
```



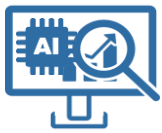
	q4_2.attractive	q4_3.bold	q4_4.boring	q4_5.casual	q4_6.cheap	q4_7.clean
1	1	4	2	3	3	5
2	4	2	1	4	4	5
3	5	3	3	5	4	5
4	4	5	2	5	2	4
5	4	3	1	2	1	3
6	4	4	2	5	1	4

	q4_8.easy.to.wear	q4_9.elegant	q4_10.feminine	q4_11.for.someone.like.me
1	2	1	5	3
2	5	3	2	1
3	3	5	5	5
4	5	3	4	4
5	4	5	5	3
6	1	4	4	5

	q4_12.heavy	q4_13.high.quality	q4_14.long.lasting	q4_15.masculine
1	2	3	5	1
2	3	5	4	1
3	3	5	5	1
4	5	5	5	1
5	3	4	3	1
6	5	3	4	1

	q4_16.memorable	q4_17.natural	q4_18.old.fashioned	q4_19.ordinary
1	4	4	4	2
2	3	1	1	5
3	5	4	1	1
4	5	4	4	4
5	5	5	5	2
6	4	3	2	1

	q4_20.overpowering	q4_21.sharp	q4_22.sophisticated	q4_23.upscale
1	3	2	3	4
2	1	4	3	2
3	4	5	3	5
4	2	3	5	3
5	4	4	4	4
6	4	5	5	5



	q4_24.well.rounded	q5_1.Deodorant.is.addictive	q7	q8.1	q8.2	q8.5	q8.6	q8.8
1	4	5	1	0	0	0	0	0
2	3	4	2	0	0	0	0	0
3	3	1	4	0	0	0	1	0
4	4	4	1	0	0	0	0	0
5	4	4	3	0	0	0	0	0
6	2	5	2	0	0	0	0	0

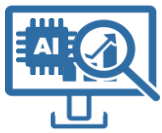
	q8.11	q8.12	q8.13	q8.19	q8.20
1	0	0	0	0	0
2	0	1	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0

	q9.how.likely.would.you.be.to.purchase.this.Deodorant
1	4
2	4
3	5
4	5
5	2
6	4

	q10.prefer.this.Deodorant.or.your.usual.Deodorant
1	1
2	4
3	3
4	3
5	5
6	3

	q11.time.of.day.would.this.Deodorant.be.appropriate
1	3
2	3
3	2
4	3
5	3
6	3

	q12.which.occasions.would.this.Deodorant.be.appropriate
1	2
2	3
3	3
4	3
5	3
6	1



```
Q13_Liking.after.30.minutes q14.Deodorant.overall.on.a.scale.from.1.to.10
1 1 5
2 3 10
3 6 1
4 6 8
5 6 4
6 4 7

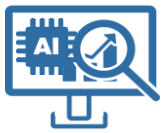
ValSegb s7.involved.in.the.selection.of.the.cosmetic.products
1 7 4
2 1 4
3 6 4
4 2 4
5 5 4
6 2 4

s8.ethnic.background s9.education s10.income s11.marital.status
1 2 3 2 2
2 1 4 4 1
3 1 3 3 1
4 2 3 10 1
5 1 2 3 3
6 2 3 2 2

s12.working.status s13.2 s13a.b.most.often
1 1 0 0
2 1 0 0
3 1 0 0
4 1 0 0
5 1 0 0
6 1 0 0

s13b.bottles.of.Deodorant.do.you.currently.own
1 4
2 1
3 6
4 4
5 4
6 1

> nrow(test_data)
[1] 5105
> test.Product.matrix <-model.matrix( ~Product - 1, data = test_data)
> test_dataF <- data.frame(test_data, test.Product.matrix)
> ncol(test_dataF) # 63
[1] 63
> head(test_dataF)
```



Respondent.ID	Product.ID	Product	q1_1.personal.opinion.of.this.Deodorant
1	4300	121 Deodorant B	7
2	4301	121 Deodorant B	6
3	4302	121 Deodorant B	6
4	4303	121 Deodorant B	5
5	4304	121 Deodorant B	6
6	4305	121 Deodorant B	7

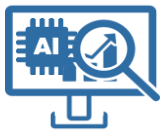
	q2_all.words	q3_1.strength.of.the.Deodorant	q4_1.artificial.chemical
1	2	3	3
2	0	3	2
3	1	3	3
4	1	5	3
5	2	3	2
6	4	3	2

	q4_2.attractive	q4_3.bold	q4_4.boring	q4_5.casual	q4_6.cheap	q4_7.clean
1	1	4	2	3	3	5
2	4	2	1	4	4	5
3	5	3	3	5	4	5
4	4	5	2	5	2	4
5	4	3	1	2	1	3
6	4	4	2	5	1	4

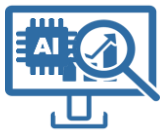
	q4_8.easy.to.wear	q4_9.elegant	q4_10.feminine	q4_11.for.someone.like.me
1	2	1	5	3
2	5	3	2	1
3	3	5	5	5
4	5	3	4	4
5	4	5	5	3
6	1	4	4	5

	q4_12.heavy	q4_13.high.quality	q4_14.long.lasting	q4_15.masculine
1	2	3	5	1
2	3	5	4	1
3	3	5	5	1
4	5	5	5	1
5	3	4	3	1
6	5	3	4	1

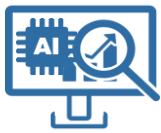
	q4_16.memorable	q4_17.natural	q4_18.old.fashioned	q4_19.ordinary
1	4	4	4	2
2	3	1	1	5
3	5	4	1	1
4	5	4	4	4
5	5	5	5	2
6	4	3	2	1



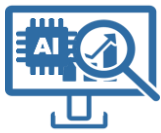
	q4_20.overpowering	q4_21.sharp	q4_22.sophisticated	q4_23.upscale						
1	3	2	3	4						
2	1	4	3	2						
3	4	5	3	5						
4	2	3	5	3						
5	4	4	4	4						
6	4	5	5	5						
	q4_24.well.rounded	q5_1.Deodorant.is.addictive	q7	q8.1	q8.2	q8.5	q8.6	q8.8		
1	4		5	1	0	0	0	0	0	0
2	3		4	2	0	0	0	0	0	0
3	3		1	4	0	0	0	1	0	0
4	4		4	1	0	0	0	0	0	0
5	4		4	3	0	0	0	0	0	0
6	2		5	2	0	0	0	0	0	0
	q8.11	q8.12	q8.13	q8.19	q8.20					
1	0	0	0	0	0					
2	0	1	0	0	0					
3	0	0	0	0	0					
4	0	0	0	0	0					
5	0	0	0	0	0					
6	0	0	0	0	0					
	q9.how.likely.would.you.be.to.purchase.this.Deodorant									
1										4
2										4
3										5
4										5
5										2
6										4
	q10.prefer.this.Deodorant.or.your.usual.Deodorant									
1										1
2										4
3										3
4										3
5										5
6										3
	q11.time.of.day.would.this.Deodorant.be.appropriate									
1										3
2										3
3										2
4										3
5										3
6										3



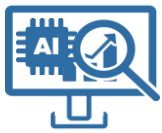
q12.which.occasions.would.this.Deodorant.be.appropriate					
1					2
2					3
3					3
4					3
5					3
6					1
Q13_Liking.after.30.minutes q14.Deodorant.overall.on.a.scale.from.1.to.10					
1		1			5
2		3			10
3		6			1
4		6			8
5		6			4
6		4			7
ValSegb s7.involved.in.the.selection.of.the.cosmetic.products					
1	7				4
2	1				4
3	6				4
4	2				4
5	5				4
6	2				4
s8.ethnic.background s9.education s10.income s11.marital.status					
1		2	3	2	2
2		1	4	4	1
3		1	3	3	1
4		2	3	10	1
5		1	2	3	3
6		2	3	2	2
s12.working.status s13.2 s13a.b.most.often					
1		1	0		0
2		1	0		0
3		1	0		0
4		1	0		0
5		1	0		0
6		1	0		0
s13b.bottles.of.Deodorant.do.you.currently.own ProductDeodorant.B					
1				4	1
2				1	1
3				6	1
4				4	1
5				4	1
6				1	1



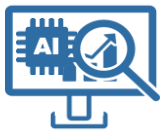
```
ProductDeodorant.F ProductDeodorant.G ProductDeodorant.H ProductDeodorant.J
1 0 0 0 0
2 0 0 0 0
3 0 0 0 0
4 0 0 0 0
5 0 0 0 0
6 0 0 0 0
> maxs <- apply(test_dataF[, c(1:2,4:58)], 2, max)
> mins <- apply(test_dataF[, c(1:2,4:58)], 2, min)
> scaledTest <- as.data.frame(scale(test_dataF[,c(1:2, 4:58)], center = mins, scale
= maxs - mins))
> set.seed(500)
> n1 <- names(scaledTest)
> head(scaledTest)
  Respondent.ID Product.ID q1_1.personal.opinion.of.this.Deodorant
1 0.0000000000000000 0 1.000000000000
2 0.00008392077878 0 0.833333333333
3 0.00016784155757 0 0.833333333333
4 0.00025176233635 0 0.666666666667
5 0.00033568311514 0 0.833333333333
6 0.00041960389392 0 1.000000000000
  q2_all.words q3_1.strength.of.the.Deodorant q4_1.artificial.chemical
1 0.4 0.5 0.50
2 0.0 0.5 0.25
3 0.2 0.5 0.50
4 0.2 1.0 0.50
5 0.4 0.5 0.25
6 0.8 0.5 0.25
  q4_2.attractive q4_3.bold q4_4.boring q4_5.casual q4_6.cheap q4_7.clean
1 0.00 0.75 0.25 0.50 0.50 1.00
2 0.75 0.25 0.00 0.75 0.75 1.00
3 1.00 0.50 0.50 1.00 0.75 1.00
4 0.75 1.00 0.25 1.00 0.25 0.75
5 0.75 0.50 0.00 0.25 0.00 0.50
6 0.75 0.75 0.25 1.00 0.00 0.75
  q4_8.easy.to.wear q4_9.elegant q4_10.feminine q4_11.for.someone.like.me
1 0.25 0.00 1.00 0.50
2 1.00 0.50 0.25 0.00
3 0.50 1.00 1.00 1.00
4 1.00 0.50 0.75 0.75
5 0.75 1.00 1.00 0.50
6 0.00 0.75 0.75 1.00
```



	q4_12.heavy	q4_13.high.quality	q4_14.long.lasting	q4_15.masculine																
1	0.25		0.50		1.00															
2	0.50		1.00		0.75															
3	0.50		1.00		1.00															
4	1.00		1.00		1.00															
5	0.50		0.75		0.50															
6	1.00		0.50		0.75															
	q4_16.memorable				q4_17.natural				q4_18.old.fashioned				q4_19.ordinary							
1		0.75		0.75			0.75				0.75								0.25	
2		0.50		0.00			0.00				0.00								1.00	
3		1.00		0.75			0.00				0.00								0.00	
4		1.00		0.75			0.75				0.75								0.75	
5		1.00		1.00			1.00				1.00								0.25	
6		0.75		0.50			0.25				0.25								0.00	
	q4_20.overpowering				q4_21.sharp				q4_22.sophisticated				q4_23.upscale							
1		0.50		0.25			0.50				0.75									
2		0.00		0.75			0.50				0.25									
3		0.75		1.00			0.50				1.00									
4		0.25		0.50			1.00				0.50									
5		0.75		0.75			0.75				0.75									
6		0.75		1.00			1.00				1.00									
	q4_24.well.rounded				q5_1.Deodorant.is.addictive				q7				q8.1 q8.2 q8.5							
1		0.75					1.00	0.4166666667			0	0	0							
2		0.50					0.75	0.5000000000			0	0	0							
3		0.50					0.00	0.6666666667			0	0	0							
4		0.75					0.75	0.4166666667			0	0	0							
5		0.75					0.75	0.5833333333			0	0	0							
6		0.25					1.00	0.5000000000			0	0	0							
	q8.6 q8.8 q8.11 q8.12 q8.13 q8.19 q8.20																			
1	0	0	0	0	0	0	0	0												
2	0	0	0	1	0	0	0	0												
3	1	0	0	0	0	0	0	0												
4	0	0	0	0	0	0	0	0												
5	0	0	0	0	0	0	0	0												
6	0	0	0	0	0	0	0	0												
	q9.how.likely.would.you.be.to.purchase.this.Deodorant																			
1																			0.75	
2																			0.75	
3																			1.00	
4																			1.00	
5																			0.25	
6																			0.75	



q10.prefer.this.Deodorant.or.your.usual.Deodorant		
1		0.00
2		0.75
3		0.50
4		0.50
5		1.00
6		0.50
q11.time.of.day.would.this.Deodorant.be.appropriate		
1		1.0
2		1.0
3		0.5
4		1.0
5		1.0
6		1.0
q12.which.occasions.would.this.Deodorant.be.appropriate		
1		0.5
2		1.0
3		1.0
4		1.0
5		1.0
6		0.0
Q13_Liking.after.30.minutes q14.Deodorant.overall.on.a.scale.from.1.to.10		
1	0.0000000000	0.4444444444
2	0.3333333333	1.0000000000
3	0.8333333333	0.0000000000
4	0.8333333333	0.7777777778
5	0.8333333333	0.3333333333
6	0.5000000000	0.6666666667
ValSegb s7.involved.in.the.selection.of.the.cosmetic.products		
1	1.0000000000	1
2	0.0000000000	1
3	0.8333333333	1
4	0.1666666667	1
5	0.6666666667	1
6	0.1666666667	1

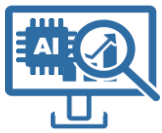


```
s8.ethnic.background s9.education s10.income s11.marital.status
1                0.25          0.2        0.000          0.25
2                0.00          0.4        0.250          0.00
3                0.00          0.2        0.125          0.00
4                0.25          0.2        1.000          0.00
5                0.00          0.0        0.125          0.50
6                0.25          0.2        0.000          0.25

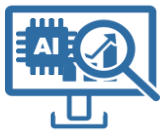
s12.working.status s13.2 s13a.b.most.often
1                0      0                0
2                0      0                0
3                0      0                0
4                0      0                0
5                0      0                0
6                0      0                0

s13b.bottles.of.Deodorant.do.you.currently.own
1                0.6
2                0.0
3                1.0
4                0.6
5                0.6
6                0.0

> ncol(scaledTest) # 57
[1] 57
> scaledTestF <- as.data.frame(cbind(scaledTest, test_dataF[, c(59:63)]))
> ncol(scaledTestF)
[1] 62
> scaledTestFF <- scaledTestF[, c(1:32, 34:35, 37, 39:40, 42:48, 50:57, 58:62)]
> ncol(scaledTestFF)
[1] 57
> n2 <- names(scaledTestFF)
> head(test_data)
  Respondent.ID Product.ID      Product q1_1.personal.opinion.of.this.Deodorant
1           4300       121 Deodorant B                               7
2           4301       121 Deodorant B                               6
3           4302       121 Deodorant B                               6
4           4303       121 Deodorant B                               5
5           4304       121 Deodorant B                               6
6           4305       121 Deodorant B                               7
```



	q2_all.words	q3_1.strength.of.the.Deodorant	q4_1.artificial.chemical			
1	2		3		3	
2	0		3		2	
3	1		3		3	
4	1		5		3	
5	2		3		2	
6	4		3		2	
	q4_2.attractive	q4_3.bold	q4_4.boring	q4_5.casual	q4_6.cheap	q4_7.clean
1	1	4	2	3	3	5
2	4	2	1	4	4	5
3	5	3	3	5	4	5
4	4	5	2	5	2	4
5	4	3	1	2	1	3
6	4	4	2	5	1	4
	q4_8.easy.to.wear	q4_9.elegant	q4_10.feminine	q4_11.for.someone.like.me		
1	2	1	5			3
2	5	3	2			1
3	3	5	5			5
4	5	3	4			4
5	4	5	5			3
6	1	4	4			5
	q4_12.heavy	q4_13.high.quality	q4_14.long.lasting	q4_15.masculine		
1	2		3	5		1
2	3		5	4		1
3	3		5	5		1
4	5		5	5		1
5	3		4	3		1
6	5		3	4		1
	q4_16.memorable	q4_17.natural	q4_18.old.fashioned	q4_19.ordinary		
1	4	4		4		2
2	3	1		1		5
3	5	4		1		1
4	5	4		4		4
5	5	5		5		2
6	4	3		2		1
	q4_20.overpowering	q4_21.sharp	q4_22.sophisticated	q4_23.upscale		
1	3	2		3		4
2	1	4		3		2
3	4	5		3		5
4	2	3		5		3
5	4	4		4		4
6	4	5		5		5



	q4_24.well.rounded	q5_1.Deodorant.is.addictive	q7	q8.1	q8.2	q8.5	q8.6	q8.8
1	4		5	1	0	0	0	0
2	3		4	2	0	0	0	0
3	3		1	4	0	0	0	1
4	4		4	1	0	0	0	0
5	4		4	3	0	0	0	0
6	2		5	2	0	0	0	0

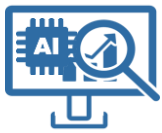
	q8.11	q8.12	q8.13	q8.19	q8.20
1	0	0	0	0	0
2	0	1	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0

	q9.how.likely.would.you.be.to.purchase.this.Deodorant
1	4
2	4
3	5
4	5
5	2
6	4

	q10.prefer.this.Deodorant.or.your.usual.Deodorant
1	1
2	4
3	3
4	3
5	5
6	3

	q11.time.of.day.would.this.Deodorant.be.appropriate
1	3
2	3
3	2
4	3
5	3
6	3

	q12.which.occasions.would.this.Deodorant.be.appropriate
1	2
2	3
3	3
4	3
5	3
6	1



```
Q13_Liking.after.30.minutes q14.Deodorant.overall.on.a.scale.from.1.to.10
1          1          5
2          3         10
3          6          1
4          6          8
5          6          4
6          4          7

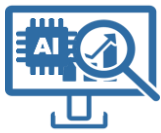
ValSegb s7.involved.in.the.selection.of.the.cosmetic.products
1      7          4
2      1          4
3      6          4
4      2          4
5      5          4
6      2          4

s8.ethnic.background s9.education s10.income s11.marital.status
1          2          3          2          2
2          1          4          4          1
3          1          3          3          1
4          2          3          10         1
5          1          2          3          3
6          2          3          2          2

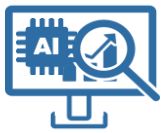
s12.working.status s13.2 s13a.b.most.often
1          1          0          0
2          1          0          0
3          1          0          0
4          1          0          0
5          1          0          0
6          1          0          0

s13b.bottles.of.Deodorant.do.you.currently.own
1          4
2          1
3          6
4          4
5          4
6          1

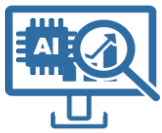
> head(scaledTestFF)
```



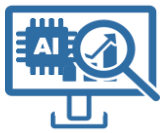
	Respondent.ID	Product.ID	q1_1.personal.opinion.of.this.Deodorant			
1	0.0000000000000000	0	1.0000000000			
2	0.00008392077878	0	0.8333333333			
3	0.00016784155757	0	0.8333333333			
4	0.00025176233635	0	0.6666666667			
5	0.00033568311514	0	0.8333333333			
6	0.00041960389392	0	1.0000000000			
	q2_all.words	q3_1.strength.of.the.Deodorant	q4_1.artificial.chemical			
1	0.4	0.5	0.50			
2	0.0	0.5	0.25			
3	0.2	0.5	0.50			
4	0.2	1.0	0.50			
5	0.4	0.5	0.25			
6	0.8	0.5	0.25			
	q4_2.attractive	q4_3.bold	q4_4.boring	q4_5.casual	q4_6.cheap	q4_7.clean
1	0.00	0.75	0.25	0.50	0.50	1.00
2	0.75	0.25	0.00	0.75	0.75	1.00
3	1.00	0.50	0.50	1.00	0.75	1.00
4	0.75	1.00	0.25	1.00	0.25	0.75
5	0.75	0.50	0.00	0.25	0.00	0.50
6	0.75	0.75	0.25	1.00	0.00	0.75
	q4_8.easy.to.wear	q4_9.elegant	q4_10.feminine	q4_11.for.someone.like.me		
1	0.25	0.00	1.00	0.50		
2	1.00	0.50	0.25	0.00		
3	0.50	1.00	1.00	1.00		
4	1.00	0.50	0.75	0.75		
5	0.75	1.00	1.00	0.50		
6	0.00	0.75	0.75	1.00		
	q4_12.heavy	q4_13.high.quality	q4_14.long.lasting	q4_15.masculine		
1	0.25	0.50	1.00	0		
2	0.50	1.00	0.75	0		
3	0.50	1.00	1.00	0		
4	1.00	1.00	1.00	0		
5	0.50	0.75	0.50	0		
6	1.00	0.50	0.75	0		
	q4_16.memorable	q4_17.natural	q4_18.old.fashioned	q4_19.ordinary		
1	0.75	0.75	0.75	0.25		
2	0.50	0.00	0.00	1.00		
3	1.00	0.75	0.00	0.00		
4	1.00	0.75	0.75	0.75		
5	1.00	1.00	1.00	0.25		
6	0.75	0.50	0.25	0.00		



	q4_20.overpowering	q4_21.sharp	q4_22.sophisticated	q4_23.upscale				
1	0.50	0.25	0.50	0.75				
2	0.00	0.75	0.50	0.25				
3	0.75	1.00	0.50	1.00				
4	0.25	0.50	1.00	0.50				
5	0.75	0.75	0.75	0.75				
6	0.75	1.00	1.00	1.00				
	q4_24.well.rounded	q5_1.Deodorant.is.addictive	q7	q8.1	q8.5	q8.6		
1	0.75	1.00	0.4166666667	0	0	0		
2	0.50	0.75	0.5000000000	0	0	0		
3	0.50	0.00	0.6666666667	0	0	1		
4	0.75	0.75	0.4166666667	0	0	0		
5	0.75	0.75	0.5833333333	0	0	0		
6	0.25	1.00	0.5000000000	0	0	0		
	q8.11	q8.13	q8.19	q9.how.likely.would.you.be.to.purchase.this.Deodorant				
1	0	0	0				0.75	
2	0	0	0				0.75	
3	0	0	0				1.00	
4	0	0	0				1.00	
5	0	0	0				0.25	
6	0	0	0				0.75	
	q10.prefer.this.Deodorant.or.your.usual.Deodorant							
1				0.00				
2				0.75				
3				0.50				
4				0.50				
5				1.00				
6				0.50				
	q11.time.of.day.would.this.Deodorant.be.appropriate							
1				1.0				
2				1.0				
3				0.5				
4				1.0				
5				1.0				
6				1.0				
	q12.which.occasions.would.this.Deodorant.be.appropriate							
1				0.5				
2				1.0				
3				1.0				
4				1.0				
5				1.0				
6				0.0				



Q13_Liking.after.30.minutes q14.Deodorant.overall.on.a.scale.from.1.to.10					
1	0.0000000000				0.4444444444
2	0.3333333333				1.0000000000
3	0.8333333333				0.0000000000
4	0.8333333333				0.7777777778
5	0.8333333333				0.3333333333
6	0.5000000000				0.6666666667
ValSegb s8.ethnic.background s9.education s10.income s11.marital.status					
1	1.0000000000	0.25	0.2	0.000	0.25
2	0.0000000000	0.00	0.4	0.250	0.00
3	0.8333333333	0.00	0.2	0.125	0.00
4	0.1666666667	0.25	0.2	1.000	0.00
5	0.6666666667	0.00	0.0	0.125	0.50
6	0.1666666667	0.25	0.2	0.000	0.25
s12.working.status s13.2 s13a.b.most.often					
1	0	0			0
2	0	0			0
3	0	0			0
4	0	0			0
5	0	0			0
6	0	0			0
s12.working.status s13.2 s13a.b.most.often					
1	0	0			0
2	0	0			0
3	0	0			0
4	0	0			0
5	0	0			0
6	0	0			0
s13b.bottles.of.Deodorant.do.you.currently.own ProductDeodorant.B					
1			0.6		1
2			0.0		1
3			1.0		1
4			0.6		1
5			0.6		1
6			0.0		1
ProductDeodorant.F ProductDeodorant.G ProductDeodorant.H ProductDeodorant.J					
1	0	0	0		0
2	0	0	0		0
3	0	0	0		0
4	0	0	0		0
5	0	0	0		0
6	0	0	0		0



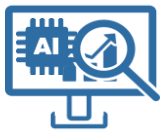
```
> pr.DNNf3 <- compute(DNNf3, scaledTestFF[, c(53:57)])
> pr.DNNf3_r <- pr.DNNf3$net.result*(max(test_data$q14.Deodorant.overall.
on.a.scale.from.1.to.10)-min(test_data$q14.Deodorant.overall.on.a.scale.
from.1.to.10))+min(test_data$q14.Deodorant.overall.on.a.scale.from.1.to.10)
> test_data_r <- (scaledTestFF$q14.Deodorant.overall.on.a.scale.
from.1.to.10)*(max(test_data$q14.Deodorant.overall.on.a.scale.from.1.to.10)-
min(test_data$q14.Deodorant.overall.on.a.scale.from.1.to.10))+min(test_data$q14.
Deodorant.overall.on.a.scale.from.1.to.10)
> MSE.DNNf3 <- sum((test_data_r - pr.DNNf3_r)^2)/nrow(scaledTestFF)
> MSE.DNNf3
[1] 6.400654791
> # [1] 6.400654791
> plot(test_data$q14.Deodorant.overall.on.a.scale.from.1.to.10, pr.DNNf3_r,col='red',
',main='Actual vs predicted DNN',pch=18,cex=0.7)

> abline(0,1,lwd=2)
> legend('bottomright',legend='DNN',pch=18,col='red',bty='n')
> pr.DNNf3_r2 <- pr.DNNf3$net.result
> pdf(" Plots ProductDeodorants vs Predicted.pdf")
> plot(test_dataF$ProductDeodorant.B, pr.DNNf3_r2, col='green',main='Product
Deodorant B vs Predicted', xlab='Product Deodorant B',pch=18,cex=2)
> abline(0,1,lwd=2)
> legend('bottom',legend='Product Deodorant B vs Predicted',pch=2,col='green',bty='n')
> plot(test_dataF$ProductDeodorant.F, pr.DNNf3_r2, col='green',main='Product
Deodorant F vs Predicted', xlab='Product Deodorant F',pch=18,cex=2)
> abline(0,1,lwd=2)
> legend('bottom',legend='Product Deodorant F vs Predicted',pch=2,col='green',bty='n')
> plot(test_dataF$ProductDeodorant.G, pr.DNNf3_r2, col='green',main='Product
Deodorant G vs Predicted', xlab='Product Deodorant G',pch=18,cex=2)
> abline(0,1,lwd=2)
> legend('bottom',legend='Product Deodorant G vs Predicted',pch=2,col='green',bty='n')
> plot(test_dataF$ProductDeodorant.H, pr.DNNf3_r2, col='green',main='Product
Deodorant H vs Predicted', xlab='Product Deodorant H',pch=18,cex=2)
> abline(0,1,lwd=2)
> legend('bottom',legend='Product Deodorant H vs Predicted',pch=2,col='green',bty='n')
> plot(test_dataF$ProductDeodorant.J, pr.DNNf3_r2, col='green',main='Product
Deodorant J vs Predicted', xlab='Product Deodorant J',pch=18,cex=2)
> abline(0,1,lwd=2)
> legend('bottom',legend='Product Deodorant J vs Predicted',pch=2,col='green',bty='n')
> dev.off()
windows
      2
>
```

Solusi Analisis Prediktif

Ini melibatkan prediksi rasa suka instan berdasarkan Produk dan Skala sebagai variabel prediktor membangun model jaringan saraf tiruan dalam berbasis AI yang serupa menggunakan *neuralnet()* berbasis R. Langkah-langkahnya adalah sebagai berikut:

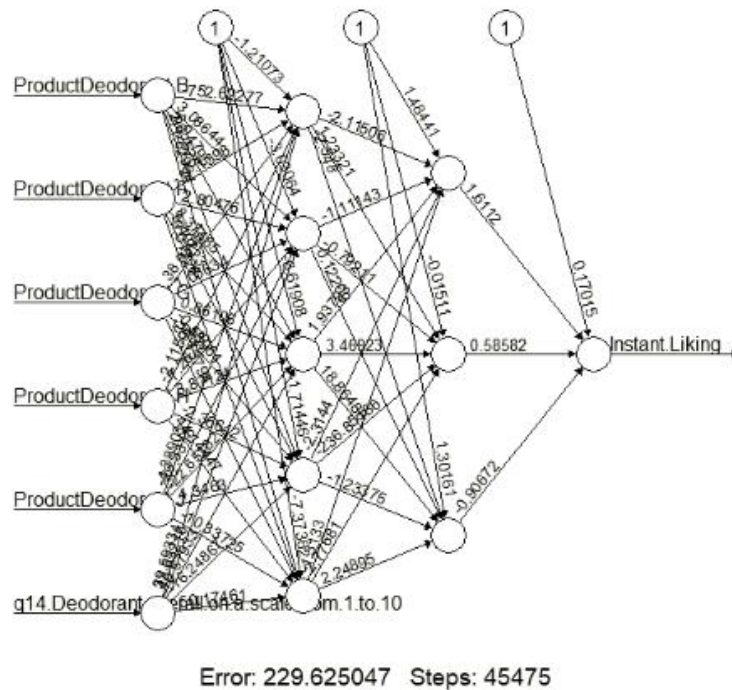
- Menggunakan data dari berkas *Data_train_reduced.csv* untuk memprediksi " Instant Liking" berdasarkan " Product" dan " q14.Deodorant.overall.on.a.scale.from.1.to.10".



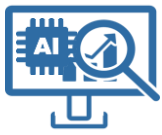
- Menggunakan model pembelajaran mesin jaringan saraf tiruan dalam yang dibangun dengan melatih dirinya sendiri berdasarkan data yang mengkorelasikan lima produk dengan masing-masing skalanya.
- Kemudian, memprediksi skala pada data uji dan berdasarkan nilai tersebut, merekomendasikan Produk tersebut.
- Sebuah DNN dengan satu lapisan masukan berisi enam simpul masukan (neuron), dua lapisan tengah (tersembunyi) (masing-masing berisi lima dan tiga simpul), dan satu lapisan keluaran (dengan satu simpul) dibangun dan dilatih.
- Kemudian, data tersebut dicocokkan dengan menerapkan keluaran model yang diprediksi terhadap masing-masing dari lima Produk dan skalanya. Variabel respons yang digunakan adalah Instant.Liking.
- Data tersebut diimplementasikan menggunakan model *neuralnet()* berbasis R untuk membangun dan melatih DNN, lalu menggunakan fungsi *compute()* untuk memprediksi keluaran dan menggunakannya untuk rekomendasi.

Plot keluaran berbasis DNN beserta plot keluaran yang telah dicocokkan ditunjukkan pada Gambar 4.34.

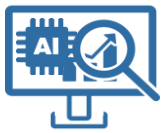
Lampiran 4.10 menampilkan kode R dan keluarannya untuk solusi analitik prediktif.



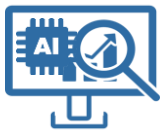
Gambar 4.34 DNN untuk Analisis Prediktif.

**Daftar 4.10** Kode R untuk Solusi Analisis Prediktif

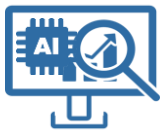
```
> setwd("<directory-name>")
> product_data <- read.csv("Data_train_reduced.csv")
> names(product_data)
 [1] "Respondent.ID"
 [2] "Product.ID"
 [3] "Product"
 [4] "Instant.Liking"
 [5] "q1_1.personal.opinion.of.this.Deodorant"
 [6] "q2_all.words"
 [7] "q3_1.strength.of.the.Deodorant"
 [8] "q4_1.artificial.chemical"
 [9] "q4_2.attractive"
[10] "q4_3.bold"
[11] "q4_4.boring"
[12] "q4_5.casual"
[13] "q4_6.cheap"
[14] "q4_7.clean"
[15] "q4_8.easy.to.wear"
[16] "q4_9.elegant"
[17] "q4_10.feminine"
[18] "q4_11.for.someone.like.me"
[19] "q4_12.heavy"
[20] "q4_13.high.quality"
[21] "q4_14.long.lasting"
[22] "q4_15.masculine"
[23] "q4_16.memorable"
[24] "q4_17.natural"
[25] "q4_18.old.fashioned"
[26] "q4_19.ordinary"
[27] "q4_20.overpowering"
[28] "q4_21.sharp"
[29] "q4_22.sophisticated"
[30] "q4_23.upscale"
[31] "q4_24.well.rounded"
[32] "q5_1.Deodorant.is.addictive"
[33] "q7"
[34] "q8.1"
```



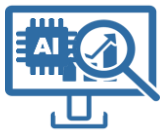
- [35] "q8.2"
- [36] "q8.5"
- [37] "q8.6"
- [38] "q8.7"
- [39] "q8.8"
- [40] "q8.9"
- [41] "q8.10"
- [42] "q8.11"
- [43] "q8.12"
- [44] "q8.13"
- [45] "q8.17"
- [46] "q8.18"
- [47] "q8.19"
- [48] "q8.20"
- [49] "q9.how.likely.would.you.be.to.purchase.this.Deodorant"
- [50] "q10.prefer.this.Deodorant.or.your.usual.Deodorant"
- [51] "q11.time.of.day.would.this.Deodorant.be.appropriate"
- [52] "q12.which.occasions.would.this.Deodorant.be.appropriate"
- [53] "Q13_Liking.after.30.minutes"
- [54] "q14.Deodorant.overall.on.a.scale.from.1.to.10"
- [55] "ValSegb"
- [56] "s7.involved.in.the.selection.of.the.cosmetic.products"
- [57] "s8.ethnic.background"
- [58] "s9.education"
- [59] "s10.income"
- [60] "s11.marital.status"
- [61] "s12.working.status"
- [62] "s13.2"
- [63] "s13a.b.most.often"
- [64] "s13b.bottles.of.Deodorant.do.you.currently.own"



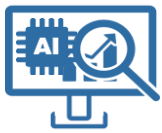
```
> test_data <- read.csv("<directory-name>test_data.csv")
> names(test_data)
 [1] "Respondent.ID"
 [2] "Product.ID"
 [3] "Product"
 [4] "q1_1.personal.opinion.of.this.Deodorant"
 [5] "q2_all.words"
 [6] "q3_1.strength.of.the.Deodorant"
 [7] "q4_1.artificial.chemical"
 [8] "q4_2.attractive"
 [9] "q4_3.bold"
[10] "q4_4.boring"
[11] "q4_5.casual"
[12] "q4_6.cheap"
[13] "q4_7.clean"
[14] "q4_8.easy.to.wear"
[15] "q4_9.elegant"
[16] "q4_10.feminine"
[17] "q4_11.for.someone.like.me"
[18] "q4_12.heavy"
[19] "q4_13.high.quality"
[20] "q4_14.long.lasting"
[21] "q4_15.masculine"
[22] "q4_16.memorable"
[23] "q4_17.natural"
[24] "q4_18.old.fashioned"
[25] "q4_19.ordinary"
[27] "q4_21.sharp"
[28] "q4_22.sophisticated"
[29] "q4_23.upscale"
[30] "q4_24.well.rounded"
[31] "q5_1.Deodorant.is.addictive"
[32] "q7"
[33] "q8.1"
[34] "q8.2"
[35] "q8.5"
[36] "q8.6"
[37] "q8.8"
[38] "q8.11"
[39] "q8.12"
[40] "q8.13"
[41] "q8.19"
[42] "q8.20"
```



```
[43] "q9.how.likely.would.you.be.to.purchase.this.Deodorant"
[44] "q10.prefer.this.Deodorant.or.your.usual.Deodorant"
[45] "q11.time.of.day.would.this.Deodorant.be.appropriate"
[46] "q12.which.occasions.would.this.Deodorant.be.appropriate"
[47] "Q13_Liking.after.30.minutes"
[48] "q14.Deodorant.overall.on.a.scale.from.1.to.10"
[49] "ValSegb"
[50] "s7.involved.in.the.selection.of.the.cosmetic.products"
[51] "s8.ethnic.background"
[52] "s9.education"
[53] "s10.income"
[54] "s11.marital.status"
[55] "s12.working.status"
[56] "s13.2"
[57] "s13a.b.most.often"
[58] "s13b.bottles.of.Deodorant.do.you.currently.own"
> nrow(product_data)
[1] 2500
> nrow(test_data)
[1] 5105
> head(product_data)
  Respondent.ID Product.ID      Product Instant.Liking
1           3800         121 Deodorant B                1
2           3801         121 Deodorant B                0
3           3802         121 Deodorant B                0
4           3803         121 Deodorant B                1
5           3804         121 Deodorant B                1
6           3805         121 Deodorant B                0
  q1_1.personal.opinion.of.this.Deodorant q2_all.words
1                                     4                1
2                                     5                1
3                                     6                1
4                                     4                0
5                                     4                1
6                                     5                1
  q3_1.strength.of.the.Deodorant q4_1.artificial.chemical q4_2.attractive
1                                 4                        2                5
2                                 4                        4                2
3                                 3                        2                5
4                                 4                        5                5
5                                 2                        1                3
6                                 5                        5                5
```



	q4_3.bold	q4_4.boring	q4_5.casual	q4_6.cheap	q4_7.clean	q4_8.easy.to.wear							
1	4	2	3	5	5	5							
2	2	1	3	2	4	4							
3	2	4	2	4	3	5							
4	4	3	5	2	5		3						
5	1	1	3	3	5		3						
6	2	1	4	5	5		2						
	q4_9.elegant	q4_10.feminine	q4_11.for.someone.like.me	q4_12.heavy									
1	4		5		3		1						
2	4		3		1		1						
3	4		4		4		3						
4	5		5		5		1						
5	5		5		5		1						
6	4		4		3		2						
	q4_13.high.quality	q4_14.long.lasting	q4_15.masculine	q4_16.memorable									
1		5	1	2			4						
2		3	4	4			5						
3		1	2	1			4						
4		4	3	3			5						
5		4	4	2			3						
6		4	5	2			3						
	q4_17.natural	q4_18.old.fashioned	q4_19.ordinary	q4_20.overpowering									
1	5		4	5			1						
2	3		3	4			2						
3	2		4	3			2						
4	5		4	2			5						
5	5		1	2			4						
6	4		2	2			3						
	q4_21.sharp	q4_22.sophisticated	q4_23.upscale	q4_24.well.rounded									
1	1		4	1			4						
2	2		5	4			4						
3	5		4	4			3						
4	3		3	5			4						
5	2		3	1			5						
6	1		5	5			4						
	q5_1.Deodorant.is.addictive	q7	q8.1	q8.2	q8.5	q8.6	q8.7	q8.8	q8.9	q8.10	q8.11		
1			1	1	0	0	0	NA	1	NA	NA	0	
2			4	4	0	0	0	1	NA	0	NA	NA	0
3			4	3	0	0	0	0	NA	0	NA	NA	0
4			4	3	0	0	0	0	NA	0	NA	NA	0
5			3	2	0	0	0	0	NA	0	NA	NA	0
6			1	3	0	0	0	0	NA	0	NA	NA	0



	q8.12	q8.13	q8.17	q8.18	q8.19	q8.20
1	0	0	NA	NA	0	0
2	0	0	NA	NA	0	0
3	0	0	NA	NA	0	0
4	0	0	NA	NA	0	0
5	0	0	NA	NA	0	0
6	0	0	NA	NA	0	0

q9.how.likely.would.you.be.to.purchase.this.Deodorant

1	2
2	3
3	5
4	5
5	5
6	5

q10.prefer.this.Deodorant.or.your.usual.Deodorant

1	1
2	5
3	1
4	4
5	3
6	4

q11.time.of.day.would.this.Deodorant.be.appropriate

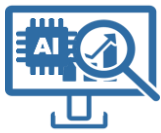
1	1
2	3
3	3
4	1
5	3
6	2

q12.which.occasions.would.this.Deodorant.be.appropriate

1	2
2	3
3	3
4	3
5	2
6	2

Q13_Liking.after.30.minutes q14.Deodorant.overall.on.a.scale.from.1.to.10

1	1	7
2	3	8
3	2	5
4	6	8
5	5	4
6	6	7



```
ValSegb s7.involved.in.the.selection.of.the.cosmetic.products
1      7      4
2      6      4
3      7      4
4      1      4
5      4      4
6      7      4

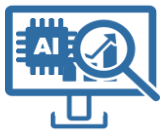
s8.ethnic.background s9.education s10.income s11.marital.status
1      1      4      3      1
2      1      4      3      1
3      1      3      5      1

4      1      4      9      1
5      1      3      5      1
6      1      4      5      2

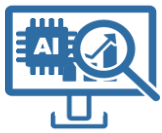
s12.working.status s13.2 s13a.b.most.often
1      1      0      0
2      1      0      0
3      1      0      0
4      3      0      0
5      2      0      0
6      2      0      0

s13b.bottles.of.Deodorant.do.you.currently.own
1      3
2      4
3      2
4      3
5      3
6      2

> product_dataF <- product_data[, c(1:3,5:37,39,42:44,47:64)]
> Product.matrix <- model.matrix( ~Product - 1, data = product_dataF)
> product_dataF <- data.frame(product_dataF, Product.matrix)
> head(product_dataF)
  Respondent.ID Product.ID      Product q1_1.personal.opinion.of.this.Deodorant
1           3800         121 Deodorant B                                     4
2           3801         121 Deodorant B                                     5
3           3802         121 Deodorant B                                     6
4           3803         121 Deodorant B                                     4
5           3804         121 Deodorant B                                     4
6           3805         121 Deodorant B                                     5
```



	q2_all.words	q3_1.strength.of.the.Deodorant	q4_1.artificial.chemical			
1	1		4		2	
2	1		4		4	
3	1		3		2	
4	0		4		5	
5	1		2		1	
6	1		5		5	
	q4_2.attractive	q4_3.bold	q4_4.boring	q4_5.casual	q4_6.cheap	q4_7.clean
1		5	4	2	3	5
2		2	2	1	3	2
3		5	2	4	2	4
4		5	4	3	5	2
5		3	1	1	3	3
6		5	2	1	4	5
	q4_8.easy.to.wear	q4_9.elegant	q4_10.feminine	q4_11.for.someone.like.me		
1		5	4	5		3
2		4	4	3		1
3		5	4	4		4
4		3	5	5		5
5		3	5	5		5
6		2	4	4		3
	q4_12.heavy	q4_13.high.quality	q4_14.long.lasting	q4_15.masculine		
1		1	5	1		2
2		1	3	4		4
3		3	1	2		1
4		1	4	3		3
5		1	4	4		2
6		2	4	5		2
	q4_16.memorable	q4_17.natural	q4_18.old.fashioned	q4_19.ordinary		
1		4	5	4		5
2		5	3	3		4
3		4	2	4		3
4		5	5	4		2
5		3	5	1		2
6		3	4	2		2
	q4_20.overpowering	q4_21.sharp	q4_22.sophisticated	q4_23.upscale		
1		1	1	4		1
2		2	2	5		4
3		2	5	4		4
4		5	3	3		5
5		4	2	3		1
6		3	1	5		5



	q4_24.well.rounded	q5_1.Deodorant.is.addictive	q7	q8.1	q8.2	q8.5	q8.6	q8.8	
1	4		1	1	0	0	0	1	
2	4		4	4	0	0	0	1	0
3	3		4	3	0	0	0	0	0
4	4		4	3	0	0	0	0	0
5	5		3	2	0	0	0	0	0
6	4		1	3	0	0	0	0	0

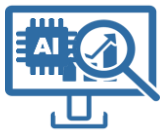
	q8.11	q8.12	q8.13	q8.19	q8.20
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0

	q9.how.likely.would.you.be.to.purchase.this.Deodorant
1	2
2	3
3	5
4	5
5	5
6	5

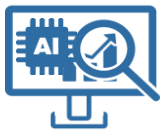
	q10.prefer.this.Deodorant.or.your.usual.Deodorant
1	1
2	5
3	1
4	4
5	3
6	4

	q11.time.of.day.would.this.Deodorant.be.appropriate
1	1
2	3
3	3
4	1
5	3
6	2

	q12.which.occasions.would.this.Deodorant.be.appropriate
1	2
2	3
3	3
4	3
5	2
6	2



Q13_Liking.after.30.minutes q14.Deodorant.overall.on.a.scale.from.1.to.10				
1		1		7
2		3		8
3		2		5
4		6		8
5		5		4
6		6		7
ValSegb s7.involved.in.the.selection.of.the.cosmetic.products				
1	7			4
2	6			4
3	7			4
4	1			4
5	4			4
6	7			4
s8.ethnic.background s9.education s10.income s11.marital.status				
1	1	4	3	1
2	1	4	3	1
3	1	3	5	1
4	1	4	9	1
5	1	3	5	1
6	1	4	5	2
s12.working.status s13.2 s13a.b.most.often				
1	1	0		0
2	1	0		0
3	1	0		0
4	3	0		0
5	2	0		0
6	2	0		0
s13b.bottles.of.Deodorant.do.you.currently.own ProductDeodorant.B				
1			3	1
2			4	1
3			2	1
4			3	1
5			3	1
6			2	1
ProductDeodorant.F ProductDeodorant.G ProductDeodorant.H ProductDeodorant.J				
1	0		0	0
2	0		0	0
3	0		0	0
4	0		0	0
5	0		0	0
6	0		0	0

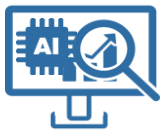


```
> ncol(product_dataF)
[1] 63
> maxs <- apply(product_dataF[, c(1:2,4:58)], 2, max)
> mins <- apply(product_dataF[, c(1:2,4:58)], 2, min)
> scaled <- as.data.frame(scale(product_dataF[, c(1:2,4:58)], center = mins, scale
= maxs - mins))
> scaledF <- as.data.frame(cbind(scaled, product_dataF[, c(59:63)]))
> set.seed(500)
> library(neuralnet)
Warning message:
package 'neuralnet' was built under R version 3.5.0
> library(ggplot2)
Warning message:
package 'ggplot2' was built under R version 3.4.4
> head(scaledF) # had some NA and NaN
  Respondent.ID Product.ID q1_1.personal.opinion.of.this.Deodorant q2_all.words
1  0.000000e+00         0                0.5000000                0.2
2  9.709681e-05         0                0.6666667                0.2
3  1.941936e-04         0                0.8333333                0.2
4  2.912904e-04         0                0.5000000                0.0
5  3.883872e-04         0                0.5000000                0.2
6  4.854840e-04         0                0.6666667                0.2

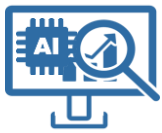
  q3_1.strength.of.the.Deodorant q4_1.artificial.chemical q4_2.attractive
1                0.75                0.25                1.00
2                0.75                0.75                0.25
3                0.50                0.25                1.00
4                0.75                1.00                1.00
5                0.25                0.00                0.50
6                1.00                1.00                1.00

  q4_3.bold q4_4.boring q4_5.casual q4_6.cheap q4_7.clean q4_8.easy.to.wear
1    0.75    0.25    0.50    1.00    1.00    1.00
2    0.25    0.00    0.50    0.25    0.75    0.75
3    0.25    0.75    0.25    0.75    0.50    1.00
4    0.75    0.50    1.00    0.25    1.00    0.50
5    0.00    0.00    0.50    0.50    1.00    0.50
6    0.25    0.00    0.75    1.00    1.00    0.25

  q4_9.elegant q4_10.feminine q4_11.for.someone.like.me q4_12.heavy
1    0.75    1.00                0.50    0.00
2    0.75    0.50                0.00    0.00
3    0.75    0.75                0.75    0.50
4    1.00    1.00                1.00    0.00
5    1.00    1.00                1.00    0.00
6    0.75    0.75                0.50    0.25
```



	q4_13.high.quality	q4_14.long.lasting	q4_15.masculine	q4_16.memorable					
1	1.00	0.00	0.25	0.75					
2	0.50	0.75	0.75	1.00					
3	0.00	0.25	0.00	0.75					
4	0.75	0.50	0.50	1.00					
5	0.75	0.75	0.25	0.50					
6	0.75	1.00	0.25	0.50					
	q4_17.natural	q4_18.old.fashioned	q4_19.ordinary	q4_20.overpowering					
1	1.00	0.75	1.00	0.00					
2	0.50	0.50	0.75	0.25					
3	0.25	0.75	0.50	0.25					
4	1.00	0.75	0.25	1.00					
5	1.00	0.00	0.25	0.75					
6	0.75	0.25	0.25	0.50					
	q4_21.sharp	q4_22.sophisticated	q4_23.upscale	q4_24.well.rounded					
1	0.00	0.75	0.00	0.75					
2	0.25	1.00	0.75	0.75					
3	1.00	0.75	0.75	0.50					
4	0.50	0.50	1.00	0.75					
5	0.25	0.50	0.00	1.00					
6	0.00	1.00	1.00	0.75					
	q5_1.Deodorant.is.addictive	q7	q8.1	q8.2	q8.5	q8.6	q8.8	q8.11	q8.12
1	0.00	0.4545455	0	NA	0	0	NA	0	NA
2	0.75	0.7272727	0	NA	0	1	NA	0	NA
3	0.75	0.6363636	0	NA	0	0	NA	0	NA
4	0.75	0.6363636	0	NA	0	0	NA	0	NA
5	0.50	0.5454545	0	NA	0	0	NA	0	NA
6	0.00	0.6363636	0	NA	0	0	NA	0	NA
	q8.13	q8.19	q8.20	q9.how.likely.would.you.be.to.purchase.this.Deodorant					
1	0	0	NA						0.25
2	0	0	NA						0.50
3	0	0	NA						1.00
4	0	0	NA						1.00
5	0	0	NA						1.00
6	0	0	NA						1.00
	q10.prefer.this.Deodorant.or.your.usual.Deodorant								
1					0.00				
2					1.00				
3					0.00				
4					0.75				
5					0.50				
6					0.75				



```
q11.time.of.day.would.this.Deodorant.be.appropriate
1 0.0
2 1.0
3 1.0
4 0.0
5 1.0
6 0.5

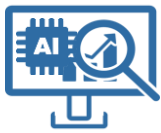
q12.which.occasions.would.this.Deodorant.be.appropriate
1 0.5
2 1.0
3 1.0
4 1.0
5 0.5
6 0.5

Q13_Liking.after.30.minutes q14.Deodorant.overall.on.a.scale.from.1.to.10
1 0.0000000 0.6666667
2 0.3333333 0.7777778
3 0.1666667 0.4444444
4 0.8333333 0.7777778
5 0.6666667 0.3333333
6 0.8333333 0.6666667

ValSegb s7.involved.in.the.selection.of.the.cosmetic.products
1 1.0000000 NaN
2 0.8333333 NaN
3 1.0000000 NaN
4 0.0000000 NaN
5 0.5000000 NaN
6 1.0000000 NaN

s8.ethnic.background s9.education s10.income s11.marital.status
1 0 0.4 0.125 0.00
2 0 0.4 0.125 0.00
3 0 0.2 0.375 0.00
4 0 0.4 0.875 0.00
5 0 0.2 0.375 0.00
6 0 0.4 0.375 0.25

s12.working.status s13.2 s13a.b.most.often
1 0.0000000 0 0
2 0.0000000 0 0
3 0.0000000 0 0
4 0.3333333 0 0
5 0.1666667 0 0
6 0.1666667 0 0
```

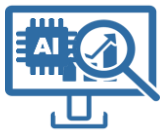


```

s13b.bottles.of.Deodorant.do.you.currently.own ProductDeodorant.B
1          0.4          1
2          0.6          1
3          0.2          1
4          0.4          1
5          0.4          1
6          0.2          1

ProductDeodorant.F ProductDeodorant.G ProductDeodorant.H ProductDeodorant.J
1          0          0          0          0
2          0          0          0          0
3          0          0          0          0
4          0          0          0          0
5          0          0          0          0
6          0          0          0          0
> scaledFF <- scaledF[, c(1:32,34:35,37,39:40,42:48,50:62)]
> head(scaledFF)
Respondent.ID Product.ID q1_1.personal.opinion.of.this.Deodorant q2_all.words
1 0.000000e+00          0          0.5000000          0.2
2 9.709681e-05          0          0.6666667          0.2
3 1.941936e-04          0          0.8333333          0.2
4 2.912904e-04          0          0.5000000          0.0
5 3.883872e-04          0          0.5000000          0.2
6 4.854840e-04          0          0.6666667          0.2
q3_1.strength.of.the.Deodorant q4_1.artificial.chemical q4_2.attractive
1          0.75          0.25          1.00
2          0.75          0.75          0.25
3          0.50          0.25          1.00
4          0.75          1.00          1.00
5          0.25          0.00          0.50
6          1.00          1.00          1.00
q4_3.bold q4_4.boring q4_5.casual q4_6.cheap q4_7.clean q4_8.easy.to.wear
1          0.75          0.25          0.50          1.00          1.00          1.00
2          0.25          0.00          0.50          0.25          0.75          0.75
3          0.25          0.75          0.25          0.75          0.50          1.00
4          0.75          0.50          1.00          0.25          1.00          0.50
5          0.00          0.00          0.50          0.50          1.00          0.50
6          0.25          0.00          0.75          1.00          1.00          0.25
q4_9.elegant q4_10.feminine q4_11.for.someone.like.me q4_12.heavy
1          0.75          1.00          0.50          0.00
2          0.75          0.50          0.00          0.00
3          0.75          0.75          0.75          0.50
4          1.00          1.00          1.00          0.00
5          1.00          1.00          1.00          0.00
6          0.75          0.75          0.50          0.25

```



	q4_13.high.quality	q4_14.long.lasting	q4_15.masculine	q4_16.memorable
1	1.00	0.00	0.25	0.75
2	0.50	0.75	0.75	1.00
3	0.00	0.25	0.00	0.75
4	0.75	0.50	0.50	1.00
5	0.75	0.75	0.25	0.50
6	0.75	1.00	0.25	0.50

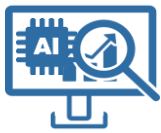
	q4_17.natural	q4_18.old.fashioned	q4_19.ordinary	q4_20.overpowering
1	1.00	0.75	1.00	0.00
2	0.50	0.50	0.75	0.25
3	0.25	0.75	0.50	0.25
4	1.00	0.75	0.25	1.00
5	1.00	0.00	0.25	0.75
6	0.75	0.25	0.25	0.50

	q4_21.sharp	q4_22.sophisticated	q4_23.upscale	q4_24.well.rounded
1	0.00	0.75	0.00	0.75
2	0.25	1.00	0.75	0.75
3	1.00	0.75	0.75	0.50
4	0.50	0.50	1.00	0.75
5	0.25	0.50	0.00	1.00
6	0.00	1.00	1.00	0.75

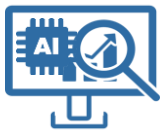
	q5_1.Deodorant.is.addictive	q7	q8.1	q8.5	q8.6	q8.11	q8.13	q8.19
1	0.00	0.4545455	0	0	0	0	0	0
2	0.75	0.7272727	0	0	1	0	0	0
3	0.75	0.6363636	0	0	0	0	0	0
4	0.75	0.6363636	0	0	0	0	0	0
5	0.50	0.5454545	0	0	0	0	0	0
6	0.00	0.6363636	0	0	0	0	0	0

	q9.how.likely.would.you.be.to.purchase.this.Deodorant
1	0.25
2	0.50
3	1.00
4	1.00
5	1.00
6	1.00

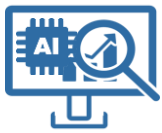
	q10.prefer.this.Deodorant.or.your.usual.Deodorant
1	0.00
2	1.00
3	0.00
4	0.75
5	0.50
6	0.75



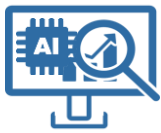
q11.time.of.day.would.this.Deodorant.be.appropriate					
1	0.0				
2	1.0				
3	1.0				
4	0.0				
5	1.0				
6	0.5				
q12.which.occasions.would.this.Deodorant.be.appropriate					
1	0.5				
2	1.0				
3	1.0				
4	1.0				
5	0.5				
6	0.5				
Q13_Liking.after.30.minutes q14.Deodorant.overall.on.a.scale.from.1.to.10					
1	0.0000000	0.6666667			
2	0.3333333	0.7777778			
3	0.1666667	0.4444444			
4	0.8333333	0.7777778			
5	0.6666667	0.3333333			
6	0.8333333	0.6666667			
ValSegb s8.ethnic.background s9.education s10.income s11.marital.status					
1	1.0000000	0	0.4	0.125	0.00
2	0.8333333	0	0.4	0.125	0.00
3	1.0000000	0	0.2	0.375	0.00
4	0.0000000	0	0.4	0.875	0.00
5	0.5000000	0	0.2	0.375	0.00
6	1.0000000	0	0.4	0.375	0.25
s12.working.status s13.2 s13a.b.most.often					
1	0.0000000	0	0		
2	0.0000000	0	0		
3	0.0000000	0	0		
4	0.3333333	0	0		
5	0.1666667	0	0		
6	0.1666667	0	0		
s13b.bottles.of.Deodorant.do.you.currently.own ProductDeodorant.B					
1	0.4	1			
2	0.6	1			
3	0.2	1			
4	0.4	1			
5	0.4	1			
6	0.2	1			



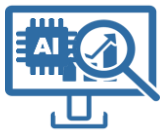
```
ProductDeodorant.F ProductDeodorant.G ProductDeodorant.H ProductDeodorant.J
1 0 0 0 0
2 0 0 0 0
3 0 0 0 0
4 0 0 0 0
5 0 0 0 0
6 0 0 0 0
> ncol(scaledFF) # 57 # 43rd column is scale1.to.10
[1] 57
> scaledFF <- scaledFF[, c(1:42,44:57,43)]
> ncol(scaledFF) #57
[1] 57
> head(scaledFF) # last column (57th) is scale1.to.10
Respondent.ID Product.ID q1_1.personal.opinion.of.this.Deodorant q2_all.words
1 0.000000e+00 0 0.5000000 0.2
2 9.709681e-05 0 0.6666667 0.2
3 1.941936e-04 0 0.8333333 0.2
4 2.912904e-04 0 0.5000000 0.0
5 3.883872e-04 0 0.5000000 0.2
6 4.854840e-04 0 0.6666667 0.2
q3_1.strength.of.the.Deodorant q4_1.artificial.chemical q4_2.attractive
1 0.75 0.25 1.00
2 0.75 0.75 0.25
3 0.50 0.25 1.00
4 0.75 1.00 1.00
5 0.25 0.00 0.50
6 1.00 1.00 1.00
q4_3.bold q4_4.boring q4_5.casual q4_6.cheap q4_7.clean q4_8.easy.to.wear
1 0.75 0.25 0.50 1.00 1.00 1.00
2 0.25 0.00 0.50 0.25 0.75 0.75
3 0.25 0.75 0.25 0.75 0.50 1.00
4 0.75 0.50 1.00 0.25 1.00 0.50
5 0.00 0.00 0.50 0.50 1.00 0.50
6 0.25 0.00 0.75 1.00 1.00 0.25
q4_9.elegant q4_10.feminine q4_11.for.someone.like.me q4_12.heavy
1 0.75 1.00 0.50 0.00
2 0.75 0.50 0.00 0.00
3 0.75 0.75 0.75 0.50
4 1.00 1.00 1.00 0.00
5 1.00 1.00 1.00 0.00
6 0.75 0.75 0.50 0.25
q4_13.high.quality q4_14.long.lasting q4_15.masculine q4_16.memorable
1 1.00 0.00 0.25 0.75
2 0.50 0.75 0.75 1.00
3 0.00 0.25 0.00 0.75
```



4	0.75	0.50	0.50	1.00				
5	0.75	0.75	0.25	0.50				
6	0.75	1.00	0.25	0.50				
	q4_17.natural	q4_18.old.fashioned	q4_19.ordinary	q4_20.overpowering				
1	1.00	0.75	1.00	0.00				
2	0.50	0.50	0.75	0.25				
3	0.25	0.75	0.50	0.25				
4	1.00	0.75	0.25	1.00				
5	1.00	0.00	0.25	0.75				
6	0.75	0.25	0.25	0.50				
	q4_21.sharp	q4_22.sophisticated	q4_23.upscale	q4_24.well.rounded				
1	0.00	0.75	0.00	0.75				
2	0.25	1.00	0.75	0.75				
3	1.00	0.75	0.75	0.50				
4	0.50	0.50	1.00	0.75				
5	0.25	0.50	0.00	1.00				
6	0.00	1.00	1.00	0.75				
	q5_1.Deodorant.is.addictive	q7	q8.1	q8.5	q8.6	q8.11	q8.13	q8.19
1	0.00	0.4545455	0	0	0	0	0	0
2	0.75	0.7272727	0	0	1	0	0	0
3	0.75	0.6363636	0	0	0	0	0	0
4	0.75	0.6363636	0	0	0	0	0	0
5	0.50	0.5454545	0	0	0	0	0	0
6	0.00	0.6363636	0	0	0	0	0	0
	q9.how.likely.would.you.be.to.purchase.this.Deodorant							
1					0.25			
2					0.50			
3					1.00			
4					1.00			
5					1.00			
6					1.00			
	q10.prefer.this.Deodorant.or.your.usual.Deodorant							
1					0.00			
2					1.00			
3					0.00			
4					0.75			
5					0.50			
6					0.75			



q11.time.of.day.would.this.Deodorant.be.appropriate					
1				0.0	
2				1.0	
3				1.0	
4				0.0	
5				1.0	
6				0.5	
q12.which.occasions.would.this.Deodorant.be.appropriate					
1				0.5	
2				1.0	
3				1.0	
4				1.0	
5				0.5	
6				0.5	
Q13_Liking.after.30.minutes ValSegb s8.ethnic.background s9.education					
1	0.0000000	1.0000000	0	0.4	
2	0.3333333	0.8333333	0	0.4	
3	0.1666667	1.0000000	0	0.2	
4	0.8333333	0.0000000	0	0.4	
5	0.6666667	0.5000000	0	0.2	
6	0.8333333	1.0000000	0	0.4	
s10.income s11.marital.status s12.working.status s13.2 s13a.b.most.often					
1	0.125	0.00	0.0000000	0	0
2	0.125	0.00	0.0000000	0	0
3	0.375	0.00	0.0000000	0	0
4	0.875	0.00	0.3333333	0	0
5	0.375	0.00	0.1666667	0	0
6	0.375	0.25	0.1666667	0	0
s13b.bottles.of.Deodorant.do.you.currently.own ProductDeodorant.B					
1			0.4		1
2			0.6		1
3			0.2		1
4			0.4		1
5			0.4		1
6			0.2		1
ProductDeodorant.F ProductDeodorant.G ProductDeodorant.H ProductDeodorant.J					
1	0		0		0
2	0		0		0
3	0		0		0
4	0		0		0
5	0		0		0
6	0		0		0

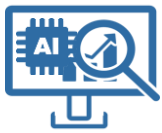


```

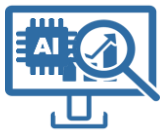
q14.Deodorant.overall.on.a.scale.from.1.to.10
1          0.6666667
2          0.7777778
3          0.4444444
4          0.7777778
5          0.3333333
6          0.6666667
> scaledFF_for_PA <- cbind(scaledFF, product_data[, c(4)])
> ncol(scaledFF_for_PA)
[1] 58
> names(scaledFF_for_PA)[58] <- "Instant.Liking"
> head(scaledFF_for_PA)

  Respondent.ID Product.ID q1_1.personal.opinion.of.this.Deodorant
1 0.0000000000000000      0          0.5000000000
2 0.00009709680552      0          0.6666666667
3 0.00019419361103      0          0.8333333333
4 0.00029129041655      0          0.5000000000
5 0.00038838722206      0          0.5000000000
6 0.00048548402758      0          0.6666666667
  q2_all.words q3_1.strength.of.the.Deodorant q4_1.artificial.chemical
1          0.2          0.75          0.25
2          0.2          0.75          0.75
3          0.2          0.50          0.25
4          0.0          0.75          1.00
5          0.2          0.25          0.00
6          0.2          1.00          1.00
  q4_2.attractive q4_3.bold q4_4.boring q4_5.casual q4_6.cheap q4_7.clean
1          1.00          0.75          0.25          0.50          1.00          1.00
2          0.25          0.25          0.00          0.50          0.25          0.75
3          1.00          0.25          0.75          0.25          0.75          0.50
4          1.00          0.75          0.50          1.00          0.25          1.00
5          0.50          0.00          0.00          0.50          0.50          1.00
6          1.00          0.25          0.00          0.75          1.00          1.00
  q4_8.easy.to.wear q4_9.elegant q4_10.feminine q4_11.for.someone.like.me
1          1.00          0.75          1.00          0.50
2          0.75          0.75          0.50          0.00
3          1.00          0.75          0.75          0.75
4          0.50          1.00          1.00          1.00
5          0.50          1.00          1.00          1.00
6          0.25          0.75          0.75          0.50

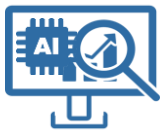
```



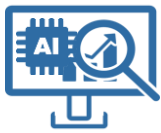
	q4_12.heavy	q4_13.high.quality	q4_14.long.lasting	q4_15.masculine				
1	0.00	1.00	0.00	0.25				
2	0.00	0.50	0.75	0.75				
3	0.50	0.00	0.25	0.00				
4	0.00	0.75	0.50	0.50				
5	0.00	0.75	0.75	0.25				
6	0.25	0.75	1.00	0.25				
	q4_16.memorable	q4_17.natural	q4_18.old.fashioned	q4_19.ordinary				
1	0.75	1.00	0.75	1.00				
2	1.00	0.50	0.50	0.75				
3	0.75	0.25	0.75	0.50				
4	1.00	1.00	0.75	0.25				
5	0.50	1.00	0.00	0.25				
6	0.50	0.75	0.25	0.25				
	q4_20.overpowering	q4_21.sharp	q4_22.sophisticated	q4_23.upscale				
1	0.00	0.00	0.75	0.00				
2	0.25	0.25	1.00	0.75				
3	0.25	1.00	0.75	0.75				
4	1.00	0.50	0.50	1.00				
5	0.75	0.25	0.50	0.00				
6	0.50	0.00	1.00	1.00				
	q4_24.well.rounded	q5_1.Deodorant.is.addictive	q7	q8.1	q8.5	q8.6		
1	0.75	0.00	0.4545454545	0	0	0		
2	0.75	0.75	0.7272727273	0	0	1		
3	0.50	0.75	0.6363636364	0	0	0		
4	0.75	0.75	0.6363636364	0	0	0		
5	1.00	0.50	0.5454545455	0	0	0		
6	0.75	0.00	0.6363636364	0	0	0		
	q8.11	q8.13	q8.19	q9.how.likely.would.you.be.to.purchase.this.Deodorant				
1	0	0	0	0.25				
2	0	0	0	0.50				
3	0	0	0	1.00				
4	0	0	0	1.00				
5	0	0	0	1.00				
6	0	0	0	1.00				
	q10.prefer.this.Deodorant.or.your.usual.Deodorant							
1	0.00							
2	1.00							
3	0.00							
4	0.75							
5	0.50							
6	0.75							



q11.time.of.day.would.this.Deodorant.be.appropriate					
1					0.0
2					1.0
3					1.0
4					0.0
5					1.0
6					0.5
q12.which.occasions.would.this.Deodorant.be.appropriate					
1					0.5
2					1.0
3					1.0
4					1.0
5					0.5
6					0.5
Q13_Liking.after.30.minutes	ValSegb	s8.ethnic.background	s9.education		
1	0.0000000000	1.0000000000	0		0.4
2	0.3333333333	0.8333333333	0		0.4
3	0.1666666667	1.0000000000	0		0.2
4	0.8333333333	0.0000000000	0		0.4
5	0.6666666667	0.5000000000	0		0.2
6	0.8333333333	1.0000000000	0		0.4
s10.income	s11.marital.status	s12.working.status	s13.2	s13a.b.most.often	
1	0.125	0.00	0.0000000000	0	0
2	0.125	0.00	0.0000000000	0	0
3	0.375	0.00	0.0000000000	0	0
4	0.875	0.00	0.3333333333	0	0
5	0.375	0.00	0.1666666667	0	0
6	0.375	0.25	0.1666666667	0	0
s13b.bottles.of.Deodorant.do.you.currently.own	ProductDeodorant.B				
1	0.4				1
2	0.6				1
3	0.2				1
4	0.4				1
5	0.4				1
6	0.2				1
ProductDeodorant.F	ProductDeodorant.G	ProductDeodorant.H	ProductDeodorant.J		
1	0	0	0		0
2	0	0	0		0
3	0	0	0		0
4	0	0	0		0
5	0	0	0		0
6	0	0	0		0

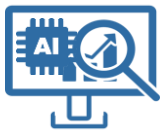


```
q14.Deodorant.overall.on.a.scale.from.1.to.10 Instant.Liking
1 0.66666666667 1
2 0.77777777778 0
3 0.44444444444 0
4 0.77777777778 1
5 0.33333333333 1
6 0.66666666667 0
> n_for_PA <- names(scaledFF_for_PA)
> set.seed(456)
> f_for_PA <- as.formula(Instant.Liking ~ ProductDeodorant.B + ProductDeodorant.F
+ ProductDeodorant.G + ProductDeodorant.H + ProductDeodorant.J + q14.Deodorant.
overall.on.a.scale.from.1.to.10)
> f_for_PA
Instant.Liking ~ ProductDeodorant.B + ProductDeodorant.F + ProductDeodorant.G +
  ProductDeodorant.H + ProductDeodorant.J + q14.Deodorant.overall.on.a.scale.from.1.to.10
> DNNf3_for_PA <- neuralnet(f_for_PA, data = scaledFF_for_PA, hidden=c(5,3),
linear.output=TRUE)
> plot(DNNf3_for_PA)
> colnames(scaledFF_for_PA)
> predicted_scale_value <- pr.DNNf3_r2
> scaledTestFF <- scaledTestFF[, c(1:42, 44:52, 53:57, 43)]
> colnames(scaledTestFF)
> scaledTestFF_for_PA <- cbind(scaledTestFF, predicted_scale_value)
> colnames(scaledTestFF_for_PA)
> scaledTestFF_for_PA <- scaledTestFF_for_PA[, c(1:56, 58, 57)]
> pr.DNNf3_r2_for_PA <- compute(DNNf3_for_PA, scaledTestFF_for_PA[, c(52:57)])
> pr.DNNf3_r2_for_PA_r2 <- pr.DNNf3_r2_for_PA$net.result
> class(pr.DNNf3_r2_for_PA_r2)
[1] "matrix"
> dim(pr.DNNf3_r2_for_PA_r2)
[1] 5105 1
> head(pr.DNNf3_r2_for_PA_r2)
      [,1]
1 0.2462013271
2 0.2462013271
3 0.2462013271
4 0.2462013271
5 0.2462013271
6 0.2462013271
>
> plot(scaledTestFF_for_PA$ProductDeodorant.G,pr.DNNf3_r2_for_PA_
r2,col='green',main='Product Deodorant G vs Predicted',
+ xlab='Product Deodorant G',pch=18,cex=2)
> abline(0,1,lwd=2)
> legend('bottom',legend='Product Deodorant G vs Predicted',pch=2,col='green',bty='n')
```



```
> plot(scaledTestFF_for_PA$ProductDeodorant.H, pr.DNNf3_r2_for_PA_
r2, col='green', main='Product Deodorant H vs Predicted',
+ xlab='Product Deodorant H', pch=18, cex=2)
> abline(0,1, lwd=2)
> legend('bottom', legend='Product Deodorant H vs Predicted', pch=2, col='green', bty='n')
> colnames(scaledTestFF_for_PA)

> pr.DNNf3_r2_for_PA2 <- compute(DNNf3_for_PA, scaledTestFF_for_PA[, c(52:56, 58)])
> pr.DNNf3_r2_for_PA_r2_2 <- pr.DNNf3_r2_for_PA2$net.result
> dim(pr.DNNf3_r2_for_PA_r2_2)
[1] 5105    1
> head(pr.DNNf3_r2_for_PA_r2_2)
      [,1]
1 0.2327684731
2 0.2827996446
3 0.1303827389
4 0.2617500624
5 0.3360772057
6 0.2532206749
>
> plot(scaledTestFF_for_PA$ProductDeodorant.G, pr.DNNf3_r2_for_PA_r2_2,
+ col='green', main='Product Deodorant G vs Predicted',
+ xlab='Product Deodorant G', pch=18, cex=2)
> abline(0,1, lwd=2)
> legend('bottom', legend='Product Deodorant G vs Predicted', pch=2, col='green', bty='n')
> > plot(scaledTestFF_for_PA$ProductDeodorant.H, pr.DNNf3_r2_for_PA_r2_2,
+ col='green', main='Product Deodorant H vs Predicted',
+ xlab='Product Deodorant G', pch=18, cex=2)
> abline(0,1, lwd=2)
> legend('bottom', legend='Product Deodorant H vs Predicted', pch=2, col='green', bty='n')
> plot(scaledTestFF_for_PA$ProductDeodorant.B, pr.DNNf3_r2_for_PA_r2_2,
+ col='green', main='Product Deodorant B vs Predicted',
+ xlab='Product Deodorant B', pch=18, cex=2)
> abline(0,1, lwd=2)
> legend('bottom', legend='Product Deodorant B vs Predicted', pch=2, col='green', bty='n')
> plot(scaledTestFF_for_PA$ProductDeodorant.F, pr.DNNf3_r2_for_PA_r2_2,
+ col='green', main='Product Deodorant F vs Predicted',
+ xlab='Product Deodorant F', pch=18, cex=2)
> abline(0,1, lwd=2)
> legend('bottom', legend='Product Deodorant F vs Predicted', pch=2, col='green', bty='n')
> plot(scaledTestFF_for_PA$ProductDeodorant.J, pr.DNNf3_r2_for_PA_r2_2,
+ col='green', main='Product Deodorant J vs Predicted',
+ xlab='Product Deodorant J', pch=18, cex=2)
> abline(0,1, lwd=2)
> legend('bottom', legend='Product Deodorant J vs Predicted', pch=2, col='green', bty='n')
>
```



Pemberdayaan BI dan Dampaknya terhadap Perusahaan

Keluaran rekomendasi dapat diintegrasikan secara visual ke dalam dasbor BI untuk mendapatkan wawasan yang dapat ditindaklanjuti (misalnya, Dasbor Tableau, Oracle BI, Microsoft BI, dll.)

Pemberdayaan BI dan Dampaknya terhadap Perusahaan (Analisis Preskriptif)

Keluaran rekomendasi dapat diintegrasikan secara visual ke dalam dasbor BI untuk mendapatkan wawasan yang dapat ditindaklanjuti (misalnya, Dasbor Tableau, Oracle BI, Microsoft BI, dll.)

Gambar 4.35 menunjukkan plot DNN yang telah disesuaikan skala aktual versus prediksi. Lima produk digambarkan di sini.

Analisis Preskriptif dari Sudut Pandang AI dan BI

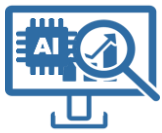
Gambar 4.35 menunjukkan akurasi DNN yang telah disesuaikan yang menunjukkan skala produk aktual versus prediksi dengan MSE 6,4.

Garis tunggal yang hampir vertikal dan sejajar dengan sumbu y adalah garis yang paling sesuai untuk DNN kami. Titik-titik di sebelah kanan garis vertikal ini menunjukkan bahwa titik-titik tersebut mewakili nilai aktual yang lebih tinggi dibandingkan dengan prediksi. Dan garis-garis di sebelah kiri garis ini menunjukkan bahwa garis-garis tersebut mewakili nilai aktual yang lebih rendah dibandingkan dengan prediksi. Selain itu, kita menyadari bahwa rata-rata nilai aktual adalah 6,7, yang jelas menandai nilai aktual yang lebih tinggi dan lebih rendah di sekitarnya untuk kelima produk.

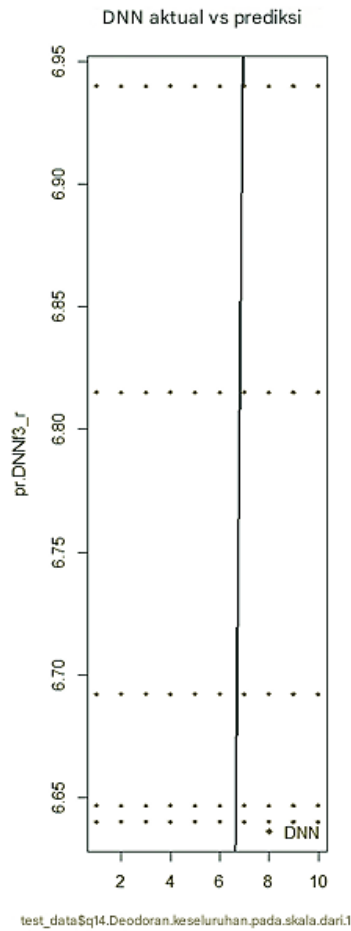
Garis putus-putus horizontal menunjukkan hal berikut: Garis putus-putus yang lebih dekat ke sumbu x menunjukkan nilai prediksi yang rendah dan karenanya deviasi yang lebih tinggi dibandingkan dengan nilai aktual, dan garis putus-putus yang lebih jauh dari sumbu x menunjukkan nilai prediksi yang lebih tinggi dan karenanya deviasi yang lebih rendah dibandingkan dengan nilai aktual.

Grafik untuk masing-masing produk versus prediksi pada Gambar 4.35 hingga 4.40 dapat diinterpretasikan sebagai berikut:

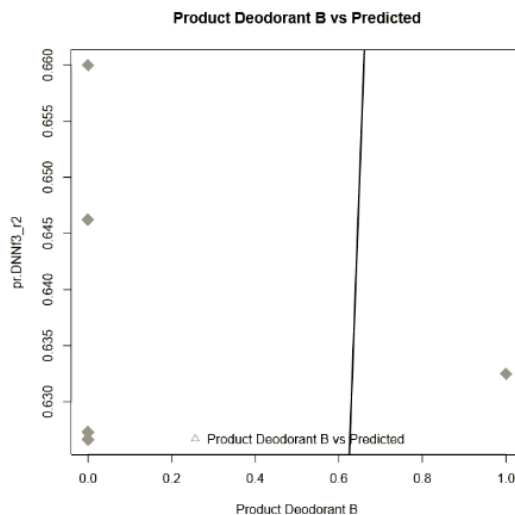
- Pada grafik untuk produk G dan H (Gambar 4.38 dan 4.39), titik-titik di sebelah kanan garis vertikal merepresentasikan aktual yang tinggi deviasi yang jauh lebih kecil dari aktual, karena letaknya lebih jauh dari sumbu y dan juga sangat terprediksi (lebih jauh dari sumbu x). Oleh karena itu, titik-titik tersebut memiliki fungsionalitas dan kualitas yang tinggi karena prediktabilitasnya yang lebih tinggi.
- Produk lain memiliki prediktabilitas yang jauh lebih rendah (lebih dekat ke sumbu x) dan, oleh karena itu, kualitas dan fungsionalitasnya lebih rendah.
- Secara umum, empat titik di sebelah kiri garis vertikal memiliki beberapa linearitas dalam grafik, kecuali untuk produk G dan H, tetapi merepresentasikan aktual yang lebih sedikit meskipun sejajar dengan 0,0 pada sumbu x. Namun, titik-titik tersebut dapat diprediksi menggunakan model matematika, statistik, atau sistem informasi.
- Pada prinsipnya, untuk tujuan rekomendasi terbaik, kami hanya mengambil nilai prediktabilitas yang lebih tinggi. Dalam kasus penggunaan ini, tiga kategori tersebut dipilih dari atas.



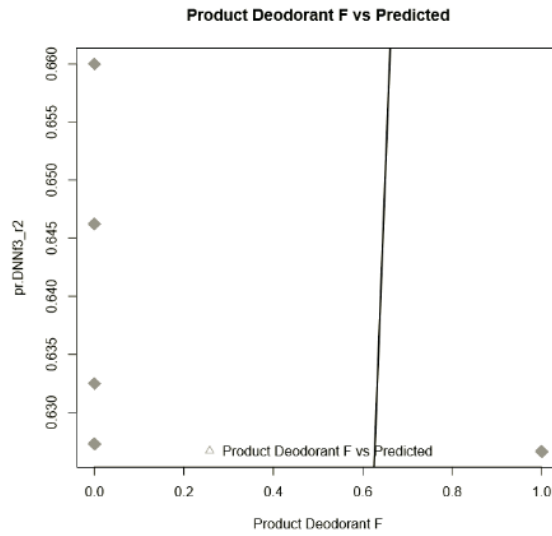
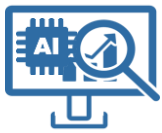
- Plot pada Gambar 4.35 dan plot dengan produk individual versus produk yang diprediksi (Gambar 4.36 hingga 4.45) dapat diintegrasikan dengan platform BI seperti dasbor Tableau, Oracle BI, Microsoft BI, dll., menggunakan antarmuka R atau antarmuka SQL untuk analisis lebih lanjut/tambahan untuk keperluan proses bisnis.



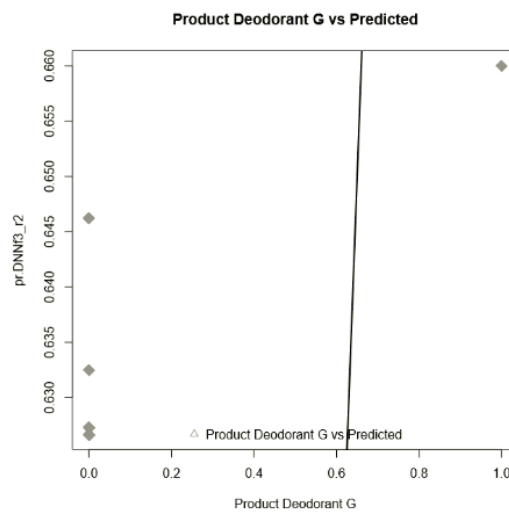
Gambar 4.35 Akurasi DNN yang dipasang menunjukkan skala produk aktual vs. skala produk prediksi.



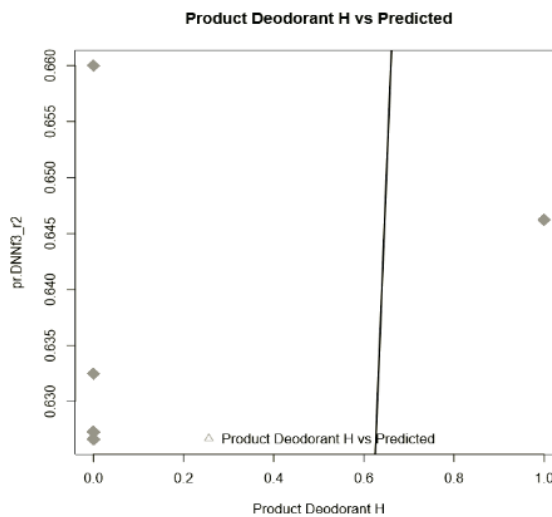
Gambar 4.36 Produk Deodorant B vs. skala produk prediksi.



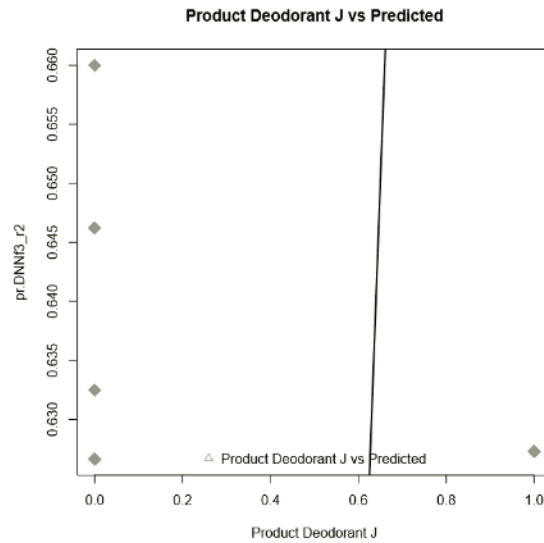
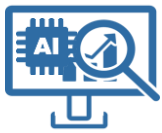
Gambar 4.37 Produk Deodoran F vs. Prediksi.



Gambar 4.38 Produk Deodoran G vs. Prediksi.



Gambar 4.39 Produk Deodoran H vs. Prediksi.



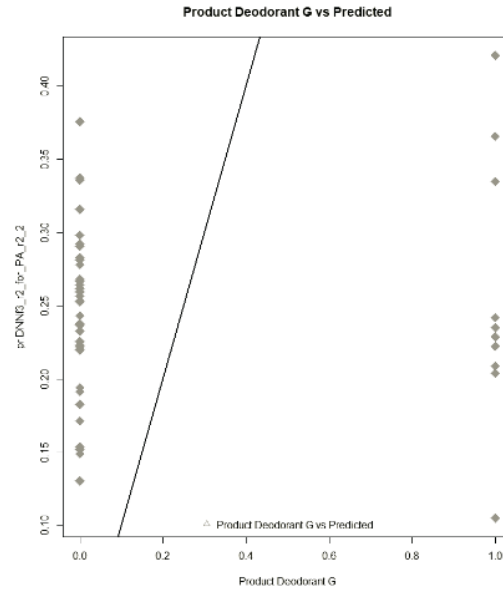
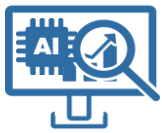
Gambar 4.40 Produk Deodoran J vs. Produk yang Diprediksi.

Pemberdayaan BI dan Dampaknya terhadap Perusahaan (Analisis Prediktif)

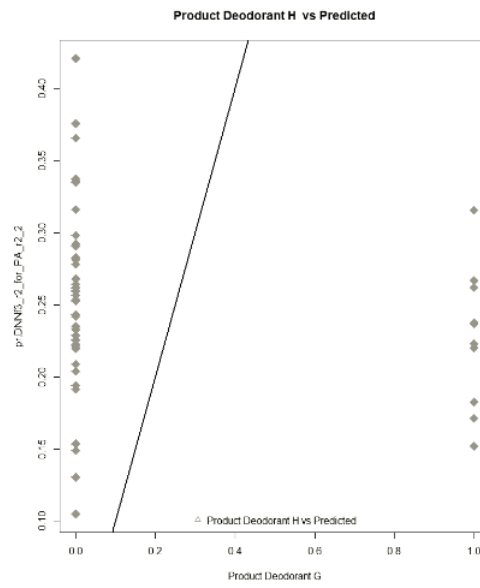
Hasil rekomendasi dapat diintegrasikan secara visual ke dalam dasbor BI untuk mendapatkan wawasan yang dapat ditindaklanjuti (misalnya, Tableau Dashboard, Oracle BI, Microsoft BI, dll.).

Grafik untuk masing-masing produk versus produk yang diprediksi ditunjukkan pada Gambar 4.41 hingga 4.45.

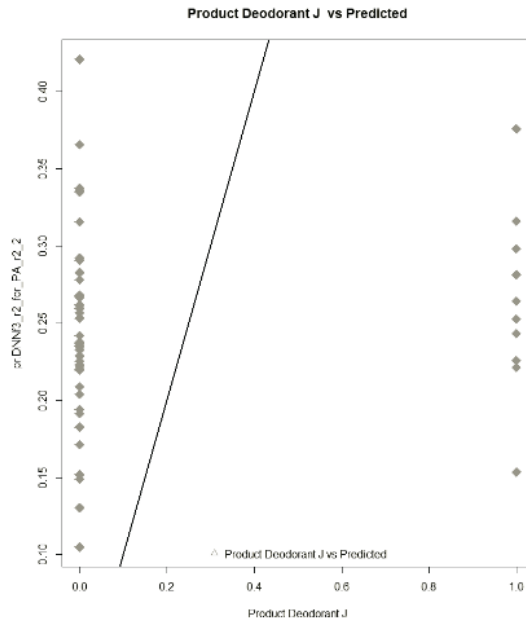
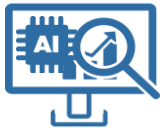
- Di antara semua grafik, hanya G yang memiliki nilai probabilitas tinggi pada $x = 1,0$.
- Pada semua grafik, pada $x = 0,0$, G memiliki probabilitas lebih rendah dibandingkan dengan grafik lainnya.
 - Probabilitas kemiripan instan sangat tinggi pada $x = 1,0$ hanya pada G (tidak pada grafik lainnya).
 - Probabilitas kemiripan instan berada pada $x = 0,0$ di semua grafik kecuali G.
- Titik $x = 0$ dan $x = 1$ signifikan karena skala produk dari 1 hingga 10 telah diperhitungkan dalam prediksi.
- Oleh karena itu, di antara produk G dan H yang direkomendasikan, Produk G memiliki kesukaan instan paling tinggi. Dengan kata lain, produk G adalah yang paling disukai.



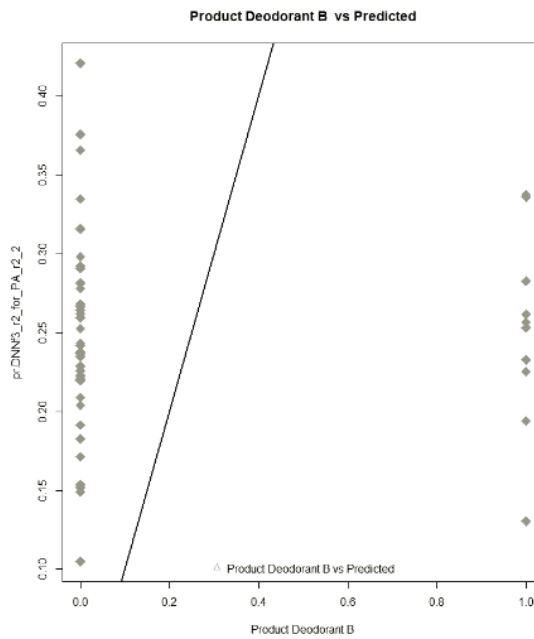
Gambar 4.41 Produk Deodoran G vs. Prediksi (untuk Analisis Prediktif).



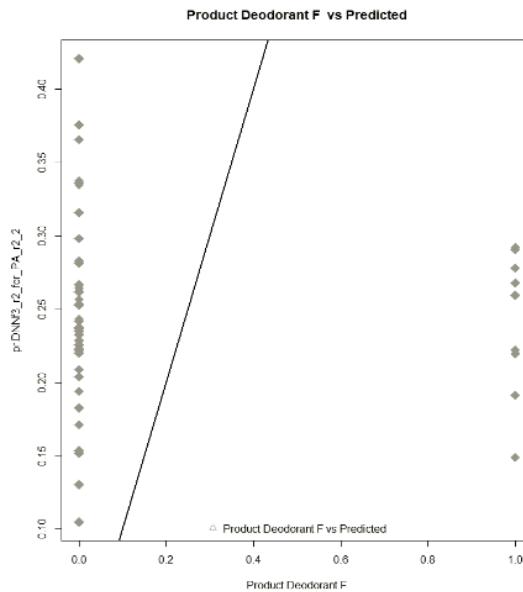
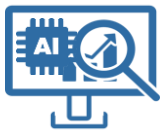
Gambar 4.42 Produk Deodoran H vs. Prediksi (untuk Analisis Prediktif).



Gambar 4.43 Produk Deodoran J vs. Prediksi (untuk Analisis Prediktif).



Gambar 4.44 Produk Deodoran B vs. Prediksi (untuk Analisis Prediktif).



Gambar 4.45 Produk Deodorant F vs. Predicted (untuk Analisis Prediktif).

4.5 PENGENALAN GAMBAR OTOMATIS

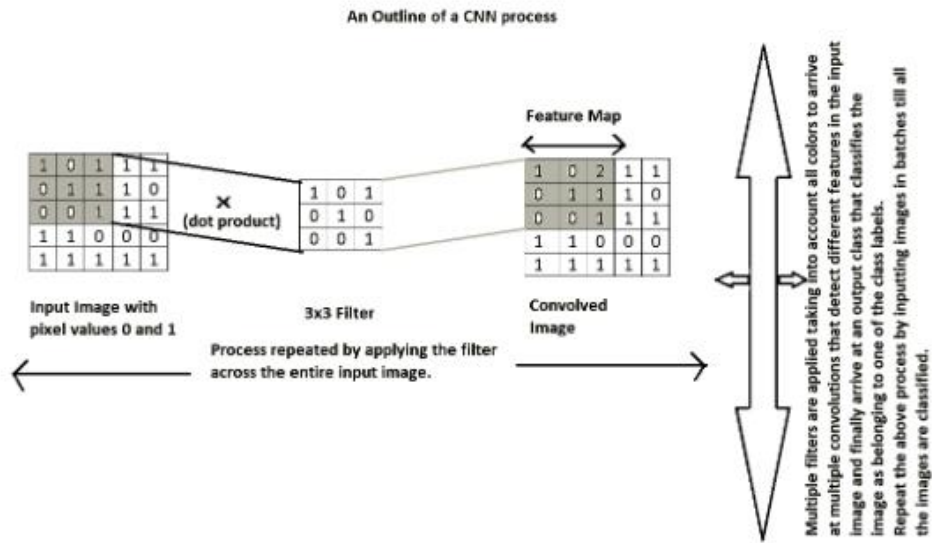
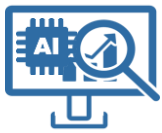
Metodologi AI Analisis Preskriptif dan Tertambah Menggunakan Jaringan Syaraf Tiruan Konvolusional Berbasis Pembelajaran Mendalam untuk Klasifikasi

Tujuan Kasus Penggunaan

- Sasaran
Pengenalan gambar menggunakan pembelajaran mendalam. Ini adalah salah satu aplikasi utama AI dalam kehidupan nyata dan digunakan untuk penandaan gambar Facebook dan pembuatan teks otomatis untuk gambar, dll.
- Teknologi
AI dan ML menggunakan jaringan saraf tiruan dalam (DNN) khususnya jaringan saraf tiruan konvolusional (CNN).
- Implementasi
Menggunakan pustaka keras dengan kerangka kerja TensorFlow di R.
- Set Data Masukan
Set data gambar Fashion MNIST. Fashion MNIST adalah kumpulan data 60.000 gambar skala abu-abu, masing-masing berukuran 28 x 28 piksel, beserta kumpulan uji 10.000 gambar. Label kelas dikodekan sebagai 10 bilangan bulat, berkisar antara 0 hingga 9, yang mewakili kaus/atasan, celana panjang, pullover, gaun, mantel, sandal, kemeja, sepatu kets, sag, dan sepatu semata kaki.
Struktur kumpulan data ini menghasilkan daftar data pelatihan dan pengujian: x-part adalah larik gambar berdimensi (60000, 28, 28), dan y-part adalah larik label yang bersesuaian, berkisar antara 0 hingga 9, berdimensi (10000).

Garis Besar Jaringan Saraf Tiruan Konvolusional (CNN)

Gambar 4.46 dan 4.47 menunjukkan garis besar Jaringan Saraf Tiruan Konvolusional. Gambar 4.48 menunjukkan struktur CNN yang diperoleh melalui proses "Build". Gambar 4.49 menunjukkan representasi CNN dan berbagai lapisannya.



Gambar 4.46 Garis besar proses jaringan saraf konvolusional (CNN).

Bagaimana CNN melihat gambar?—Sebagai matriks [jumlah baris piksel x jumlah kolom piksel x jumlah saluran warna]

Jika jumlah gambar masukan difaktorkan, perkalian di atas menjadi:

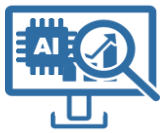
[Jumlah gambar masukan x jumlah baris piksel x jumlah kolom piksel x jumlah saluran warna]

Apa itu filter?—Filter seperti ekstraktor fitur yang ditentukan oleh ukuran $m \times n$, misalnya, 3×3 . Filter ditempatkan di atas suatu wilayah gambar, dan perkalian matriks dilakukan di wilayah ini dan filter.

Mengapa filter?—Karena filter ditempatkan pada bagian gambar yang terlokalisasi dan diproses, filter membantu dalam mengekstrak detail spesifik di wilayah gambar tersebut.

Filter berukuran lebih besar lebih mahal daripada filter berukuran lebih kecil dalam hal jumlah perkalian yang harus dilakukan.

Gambar 4.47 CNN dijelaskan lebih detail.



Structure of CNN model for Automatic Image Recognition

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 28, 28, 32)	320
activation_1 (Activation)	(None, 28, 28, 32)	0
conv2d_2 (Conv2D)	(None, 26, 26, 32)	9248
activation_2 (Activation)	(None, 26, 26, 32)	0
max_pooling2d_1 (MaxPooling2D)	(None, 13, 13, 32)	0
dropout_1 (Dropout)	(None, 13, 13, 32)	0
conv2d_3 (Conv2D)	(None, 13, 13, 32)	9248
activation_3 (Activation)	(None, 13, 13, 32)	0
conv2d_4 (Conv2D)	(None, 11, 11, 32)	9248
activation_4 (Activation)	(None, 11, 11, 32)	0
max_pooling2d_2 (MaxPooling2D)	(None, 5, 5, 32)	0
dropout_2 (Dropout)	(None, 5, 5, 32)	0
flatten_1 (Flatten)	(None, 800)	0
dense_1 (Dense)	(None, 512)	410112
activation_5 (Activation)	(None, 512)	0
dropout_3 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 10)	5130
activation_6 (Activation)	(None, 10)	0

Total params: 443,306
Trainable params: 443,306
Non-trainable params: 0

Repeat above layers as needed

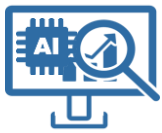
Repeat as needed

← Output Layer (10 classes)

Gambar 4.48 Struktur model CNN yang digunakan untuk pengenalan gambar otomatis.

Menggunakan Dataset Fashion MNIST dan Model CNN Berbasis Keras di R

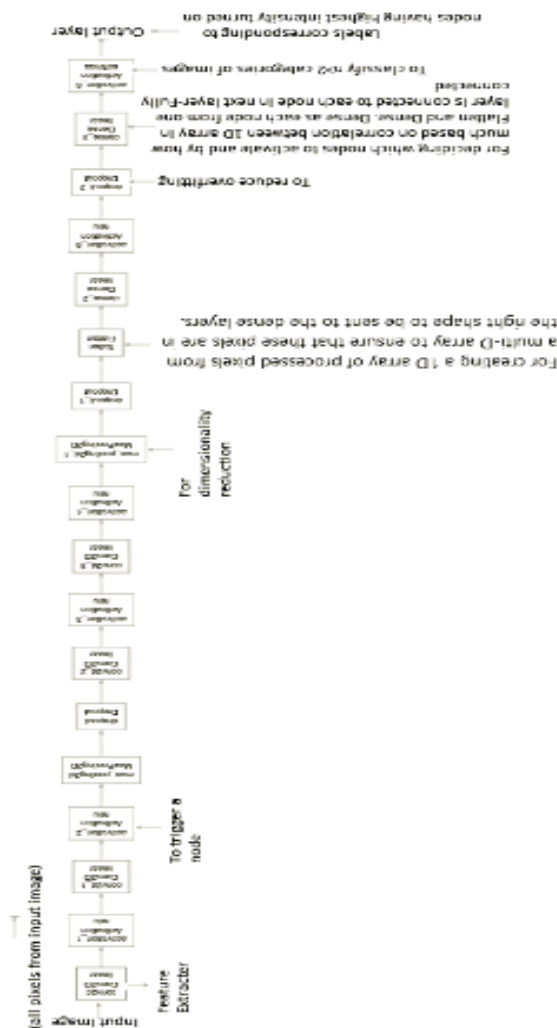
- Keras adalah API tingkat tinggi yang awalnya ditulis dengan Python dan kini juga tersedia di R. Kedua API ini menggunakan TensorFlow sebagai platform backend untuk menjalankannya.
- Fashion MNIST adalah dataset berisi 60.000 gambar skala abu-abu, masing-masing berukuran 28 x 28 piksel, beserta set uji yang berisi 10.000 gambar. Label kelas dikodekan sebagai 10 bilangan bulat, dengan rentang 0 hingga 9, yang mewakili kaus/atasan, celana panjang, pullover, gaun, mantel, sandal, kemeja, sepatu kets, tas, dan sepatu semata kaki.
- Struktur dataset sedemikian rupa sehingga menghasilkan daftar data latih dan uji: x-part adalah larik gambar berdimensi (60000, 28, 28), dan y-part adalah larik label yang bersesuaian, berkisar dari 0 hingga 9, berdimensi (10000).
- Model CNN pertama-tama dilatih dengan data latih, kemudian mengklasifikasikan gambar dalam data uji. Dengan cara ini, model CNN mengenali gambar dalam data uji dengan menerapkan model berbasis AI. Keluaran dari klasifikasi (dan pengenalan) adalah menentukan label gambar dengan tepat untuk setiap gambar yang diberikan dan memeriksa tampilan raster dari gambar yang sama dalam set uji. Salah satu contoh teknik ini adalah yang digunakan untuk penandaan gambar otomatis, seperti di Facebook, dll.



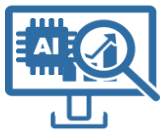
- Langkah pertama untuk melanjutkan lebih lanjut adalah mempersiapkan data gambar agar berada dalam format yang dapat diterima oleh model CNN. Ini terdiri dari:
 - Membentuk ulang x-array menjadi dimensi peringkat 4, yaitu (jumlah_sampel, lebar dalam piksel, tinggi dalam piksel, saluran). Dengan demikian, kita membentuk ulang x-part sebagai berikut:

```
train_images <- array_reshape(train x-part, c(60000,28,28,1))  
test_images <- array_reshape(test x-part, c(10000,28,28,1))
```
 - Menormalkan x-data sehingga setiap piksel gambar berada dalam interval [0,1], bukan interval awal [0,255]. Ini dilakukan dengan membagi train_images dan test_images dengan 255.
 - Pengodean one-hot untuk label train dan test (y-part). Pengodean one-hot berarti nilai label diubah menjadi array yang hanya memiliki satu nilai '1' dan sisanya memiliki nilai '0'. Ini diperlukan untuk variabel target jaringan saraf.

```
train_labels <- to_categorical(train y-part)  
test_labels <- to_categorical(test y-part)
```



Gambar 4.49 Representasi model CNN yang dibangun untuk Pengenalan Gambar Otomatis dan berbagai lapisannya.

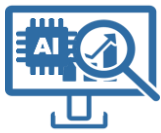


Membangun dan Mensko Model Menggunakan Algoritma CNN dalam R untuk Pengenalan Gambar Berdasarkan Dataset Mode MNIST Menggunakan Pustaka Keras dengan Backend TensorFlow

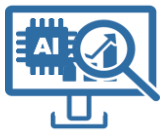
Berikut adalah langkah-langkah yang diperlukan untuk membangun model CNN (kode ditampilkan dalam Demo):

- Membangun arsitektur CNN (menambahkan tumpukan lapisan linear).
 - Menambahkan lapisan tersembunyi.
- Menambahkan konvolusi (-al layer) yang menentukan jumlah filter, ukuran setiap filter, dan bentuk (dimensi) format input lebar, tinggi, saluran. Ini menjalankan filter di seluruh gambar, sehingga membantu dalam ekstraksi fitur.
- Menambahkan aktivasi (lapisan). Lapisan ini digunakan untuk mengurangi waktu pelatihan dan menggunakan aktivasi "*relu*" yang terkenal, yang mengganti semua nilai negatif dalam matriks menjadi 0 dan mempertahankan semua nilai lainnya tetap sama.
- Menambahkan penggabungan (lapisan) (untuk mengurangi dimensionalitas peta fitur dan membuat model lebih mudah dihitung).
- Tambahkan dropout (lapisan) (untuk menghindari overfitting).
 - Tambahkan lapisan untuk meratakan masukan. (Ini diperlukan untuk memasukkan lapisan yang sudah ditambahkan ke lapisan padat.)
 - Tambahkan lapisan padat. (Dengan ini, CNN mengklasifikasikan masukan.)
 - Tambahkan lapisan keluaran (menggunakan aktivasi softmax untuk menghitung entropi silang kategoris, karena kita memiliki 10 kelas yang berkorespondensi dengan 10 citra).
- Kompilasi CNN.
- Latih CNN dengan cara menyesuaikannya menggunakan fitur (train x-part), target (train y-part), jumlah epoch, ukuran batch, dan pemisahan validasi.
- Plot keluaran yang telah disesuaikan untuk kurva kerugian dan akurasi (ditunjukkan pada Gambar 4.50).
- Evaluasi CNN pada dataset uji (test x-part, test y-part).
- Prediksi kelas menggunakan model dan citra uji (test x-part).
- Uji validitas model menggunakan tabel nilai prediksi dan aktual (keluaran prediksi di atas dan test y-part). Keluarannya berupa tabel yang terdiri dari 10 baris dan 10 kolom.
- Periksa nilai label (prediksi vs. aktual) untuk setiap label yang diberikan, yaitu `pred_test[1]` dan `test y-part [1]`. Selain itu, plot citra raster test x-part asli [1, ,] dan pastikan labelnya cocok dengan nomor label yang diprediksi (kelas) (citra raster ditunjukkan pada Gambar 4.51).

Daftar 4.11 memberikan kode R dari implementasi ini beserta keluarannya.

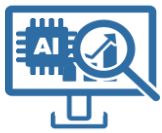
**Daftar 4.11** Kode R Implementasi CNN untuk Klasifikasi Citra Otomatis

```
> library(keras)
Warning message:
package 'keras' was built under R version 3.5.0
> fashion_mnist <- dataset_fashion_mnist()
Using TensorFlow backend.
> train_images <- fashion_mnist$train$x
> train_labels <- fashion_mnist$train$y
> test_images <- fashion_mnist$test$x
> test_labels <- fashion_mnist$test$y
> str(fashion_mnist)
List of 2
 $ train:List of 2
  ..$ x: int [1:60000, 1:28, 1:28] 0 0 0 0 0 0 0 0 0 0 0 0 ...
  ..$ y: int [1:60000(1d)] 9 0 0 3 0 2 7 2 5 5 ...
 $ test :List of 2
  ..$ x: int [1:10000, 1:28, 1:28] 0 0 0 0 0 0 0 0 0 0 0 0 ...
  ..$ y: int [1:10000(1d)] 9 2 1 1 6 1 4 6 5 7 ...
> train_images[1, , ]
  [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15]
[1,]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
[2,]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
[3,]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
[4,]  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0
[5,]  0  0  0  0  0  0  0  0  0  0  0  0  0  3  0
[6,]  0  0  0  0  0  0  0  0  0  0  0  0  0  6  0
[7,]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
[8,]  0  0  0  0  0  0  0  0  0  0  0  0  1  0  69
[9,]  0  0  0  0  0  0  0  0  0  0  1  1  1  0  200
[10,] 0  0  0  0  0  0  0  0  0  0  0  0  0  0  183
```

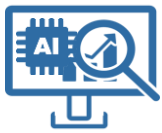


[11,]	0	0	0	0	0	0	0	0	0	0	0	0	0	193	228
[12,]	0	0	0	0	0	0	0	0	0	1	3	0	12	219	220
[13,]	0	0	0	0	0	0	0	0	0	0	6	0	99	244	222
[14,]	0	0	0	0	0	0	0	0	0	4	0	0	55	236	228
[15,]	0	0	1	4	6	7	2	0	0	0	0	0	237	226	217
[16,]	0	3	0	0	0	0	0	0	0	62	145	204	228	207	213
[17,]	0	0	0	0	18	44	82	107	189	228	220	222	217	226	200
[18,]	0	57	187	208	224	221	224	208	204	214	208	209	200	159	245
[19,]	3	202	228	224	221	211	211	214	205	205	205	220	240	80	150
[20,]	98	233	198	210	222	229	229	234	249	220	194	215	217	241	65
[21,]	75	204	212	204	193	205	211	225	216	185	197	206	198	213	240
[22,]	48	203	183	194	213	197	185	190	194	192	202	214	219	221	220
[23,]	0	122	219	193	179	171	183	196	204	210	213	207	211	210	200
[24,]	0	0	74	189	212	191	175	172	175	181	185	188	189	188	193
[25,]	2	0	0	0	66	200	222	237	239	242	246	243	244	221	220
[26,]	0	0	0	0	0	0	0	40	61	44	72	41	35	0	0
[27,]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[28,]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	[,16]	[,17]	[,18]	[,19]	[,20]	[,21]	[,22]	[,23]	[,24]	[,25]	[,26]	[,27]	[,28]
[1,]	0	0	0	0	0	0	0	0	0	0	0	0	0
[2,]	0	0	0	0	0	0	0	0	0	0	0	0	0
[3,]	0	0	0	0	0	0	0	0	0	0	0	0	0
[4,]	13	73	0	0	1	4	0	0	0	0	1	1	0
[5,]	136	127	62	54	0	0	0	1	3	4	0	0	3
[6,]	204	176	134	144	123	23	0	0	0	0	12	10	0
[7,]	236	207	178	107	156	161	109	64	23	77	130	72	15
[8,]	223	218	216	216	163	127	121	122	146	141	88	172	66
[9,]	232	233	229	223	223	215	213	164	127	123	196	229	0
[10,]	216	223	228	235	227	224	222	224	221	223	245	173	0
[11,]	218	213	198	180	212	210	211	213	223	220	243	202	0
[12,]	212	218	192	169	227	208	218	224	212	226	197	209	52
[13,]	220	218	203	198	221	215	213	222	220	245	119	167	56
[14,]	230	228	240	232	213	218	223	234	217	217	209	92	0
[15,]	223	222	219	222	221	216	223	229	215	218	255	77	0
[16,]	221	218	208	211	218	224	223	219	215	224	244	159	0
[17,]	205	211	230	224	234	176	188	250	248	233	238	215	0
[18,]	193	206	223	255	255	221	234	221	211	220	232	246	0
[19,]	255	229	221	188	154	191	210	204	209	222	228	225	0
[20,]	73	106	117	168	219	221	215	217	223	223	224	229	29
[21,]	195	227	245	239	223	218	212	209	222	220	221	230	67
[22,]	236	225	216	199	206	186	181	177	172	181	205	206	115
[23,]	196	194	191	195	191	198	192	176	156	167	177	210	92
[24,]	198	204	209	210	210	211	188	188	194	192	216	170	0
[25,]	193	191	179	182	182	181	176	166	168	99	58	0	0
[26,]	0	0	0	0	0	0	0	0	0	0	0	0	0
[27,]	0	0	0	0	0	0	0	0	0	0	0	0	0
[28,]	0	0	0	0	0	0	0	0	0	0	0	0	0



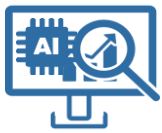
```
> mage <- train_images[1, , ]
> image <- train_images[1, , ]
> plot(as.raster(image, max = 255))
> train_labels[1]
[1] 9
> train_images <- array_reshape(train_images, c(60000, 28, 28, 1))
> train_images <- train_images / 255
> test_images <- array_reshape(test_images, c(10000, 28, 28, 1))
> test_images <- test_images / 255
> train_labels <- to_categorical(train_labels)
> test_labels <- to_categorical(test_labels)
> train_y <- fashion_mnist$train$y
> test_y <- fashion_mnist$test$y
> model <- keras_model_sequential()
> model %>%
+ layer_conv_2d(
+ filter = 32, kernel_size = c(3,3), padding = "same",
+ input_shape = c(28, 28, 1)
+ ) %>%
+ layer_activation("relu") %>%
+ layer_conv_2d(filter = 32, kernel_size = c(3,3)) %>%
+ layer_activation("relu") %>%
+ layer_max_pooling_2d(pool_size = c(2,2)) %>%
+ layer_dropout(0.25) %>%
+ layer_conv_2d(filter = 32, kernel_size = c(3,3), padding = "same") %>%
+ layer_activation("relu") %>%
+ layer_conv_2d(filter = 32, kernel_size = c(3,3)) %>%
+ layer_activation("relu") %>%
+ layer_max_pooling_2d(pool_size = c(2,2)) %>%
+ layer_dropout(0.25) %>%
+ layer_flatten() %>%
+ layer_dense(512) %>%
+ layer_activation("relu") %>%
+ layer_dropout(0.5) %>%
+ layer_dense(10) %>%
+ layer_activation("softmax")
> str(model)
```



Model

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 28, 28, 32)	320
activation_1 (Activation)	(None, 28, 28, 32)	0
conv2d_2 (Conv2D)	(None, 26, 26, 32)	9248
activation_2 (Activation)	(None, 26, 26, 32)	0
max_pooling2d_1 (MaxPooling2D)	(None, 13, 13, 32)	0
dropout_1 (Dropout)	(None, 13, 13, 32)	0
conv2d_3 (Conv2D)	(None, 13, 13, 32)	9248
activation_3 (Activation)	(None, 13, 13, 32)	0
conv2d_4 (Conv2D)	(None, 11, 11, 32)	9248
activation_4 (Activation)	(None, 11, 11, 32)	0
max_pooling2d_2 (MaxPooling2D)	(None, 5, 5, 32)	0
dropout_2 (Dropout)	(None, 5, 5, 32)	0
flatten_1 (Flatten)	(None, 800)	0
dense_1 (Dense)	(None, 512)	410112
activation_5 (Activation)	(None, 512)	0
dropout_3 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 10)	5130
activation_6 (Activation)	(None, 10)	0
=====		
Total params: 443,306		
Trainable params: 443,306		
Non-trainable params: 0		

```
> set.seed(1234)
> model %>% compile(
+ optimizer = "rmsprop",
+ loss = "categorical_crossentropy",
+ metrics = c("accuracy")
+ )
> model %>%
+ fit(train_images, train_labels, epochs=2, batch_size=128)
> model %>%
+ fit(train_images, train_labels, epochs=2, batch_size=128)
Epoch 1/2
60000/60000 [=====] - 158s 3ms/step - loss: 0.5863 - acc: 0.7849
Epoch 2/2
60000/60000 [=====] - 161s 3ms/step - loss: 0.3646 - acc: 0.8677
```



```
> model %>% evaluate(test_images, test_labels)
10000/10000 [=====] - 8s 822us/step
$loss
[1] 0.3220585

$acc
[1] 0.8831

> pred_test <- model %>% predict_classes(test_images)

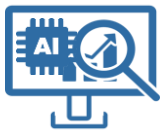
> str(pred_test)
 num [1:10000(1d)] 9 2 1 1 6 1 4 6 5 7 ...
> pred_test[1]
[1] 9
> test_y[1]
[1] 9
> table(Predicted = pred_test, Actual = test_y)
      Actual
Predicted  0  1  2  3  4  5  6  7  8  9
  0 911  1  14  32  1  0 226  0  4  0
  1  0 969  0  2  0  0  0  0  0  0
  2  23  1 916  22 133  0 137  0  9  0
  3  14  20  6 895  29  0  22  0  6  0
  4  6  5  38  30 796  0  98  0  1  0
  5  2  0  0  0  0 980  0  15  3  20
  6  38  2  26  17  40  0 502  0  7  0
  7  0  0  0  0  0  18  0 977  4  60
  8  6  2  0  2  1  0  15  0 965  0
  9  0  0  0  0  0  2  0  8  1 920

> test_x <- fashion_mnist$test$x
> plot(as.raster(test_x[1, , ], max = 255))
>
```

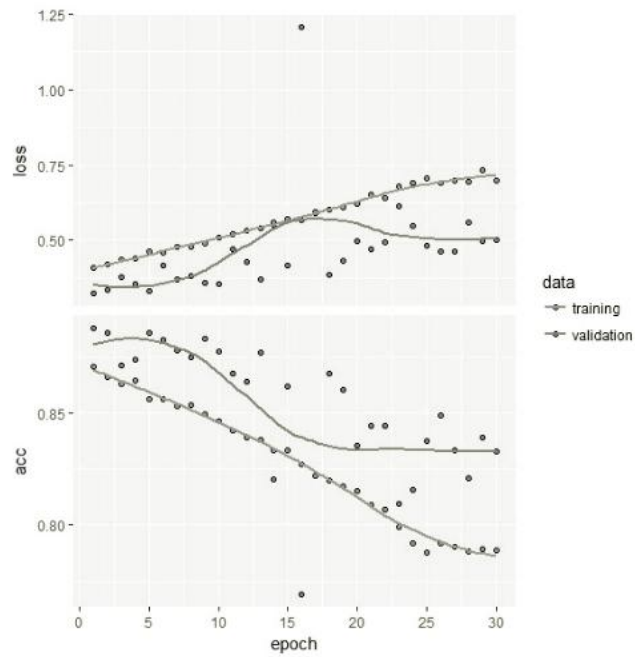
Gambar 4.50 dan 4.51 memperlihatkan kurva kerugian dan akurasi model CNN pada gambar pelatihan dan gambar uji beserta plot raster gambar yang dikenali dari CNN—yang termasuk kelas 9—sepatu kaki.

```
pred_test[1]
9
```

cocok dengan yang



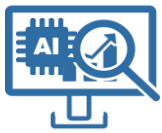
`test_y[1]`
9



Gambar 4.50 Plot CNN yang dipasang menggunakan Keras dan TensorFlow.



Gambar 4.51 Citra prediksi menggunakan CNN (sepatu ankle foot).



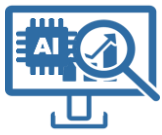
Ringkasan Model CNN yang Dibangun

```
s
Model
```

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 28, 28, 32)	320
activation_1 (Activation)	(None, 28, 28, 32)	0
conv2d_2 (Conv2D)	(None, 26, 26, 32)	9248
activation_2 (Activation)	(None, 26, 26, 32)	0
max_pooling2d_1 (MaxPooling2D)	(None, 13, 13, 32)	0
dropout_1 (Dropout)	(None, 13, 13, 32)	0
conv2d_3 (Conv2D)	(None, 13, 13, 32)	9248
activation_3 (Activation)	(None, 13, 13, 32)	0
conv2d_4 (Conv2D)	(None, 11, 11, 32)	9248
activation_4 (Activation)	(None, 11, 11, 32)	0
max_pooling2d_2 (MaxPooling2D)	(None, 5, 5, 32)	0
dropout_2 (Dropout)	(None, 5, 5, 32)	0
flatten_1 (Flatten)	(None, 800)	0
dense_1 (Dense)	(None, 512)	410112
activation_5 (Activation)	(None, 512)	0
dropout_3 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 10)	5130
activation_6 (Activation)	(None, 10)	0
Total params: 443,306		
Trainable params: 443,306		
Non-trainable params: 0		

4.6 RINGKASAN

Bab ini berfokus pada penggunaan AI dan interseksinya dengan BI generasi ke-3 untuk berbagai kasus penggunaan di dunia nyata berdasarkan klasifikasi menggunakan hutan acak, pohon keputusan, dan jaringan saraf tiruan. Bab ini menunjukkan bagaimana solusi AI dapat diimplementasikan menggunakan model pembelajaran mesin dan kode R, serta bagaimana pengambilan keputusan dapat dicapai untuk tujuan BI. Bab ini juga menjelaskan kemampuan prediktif AI dengan kasus penggunaan yang memprediksi harga logam mulia.



BAB 5

APA SELANJUTNYA DALAM AI BERTEMU BI?

5.1 PENDAHULUAN

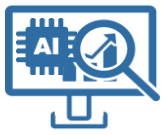
Bab ini menguraikan batas baru dalam BI bertenaga AI. Bab ini dimulai dengan deskripsi komputasi kognitif bertenaga AI. Kemudian, bab ini menyoroti aspek keamanan dan tata kelola BI bertenaga AI, diikuti oleh aspek etika penerapannya di perusahaan.

5.2 KOMPUTASI KOGNITIF BERTENAGA AI

Gelombang komputasi bertenaga AI berikutnya adalah komputasi kognitif, yang melibatkan komputasi konteks menggunakan teknik AI.

-
- *[AI] telah menyediakan area di mana AI dan komputasi kognitif dapat bekerja sama, menghasilkan platform kecerdasan otonom, augmented, dan adaptif yang optimal.*
 - *Melalui AI, komputasi kognitif dapat terdiri dari teknologi-teknologi berikut: pemrosesan bahasa alami (NLP), pembelajaran mesin dengan jaringan saraf tiruan dan jaringan saraf tiruan dalam, algoritma yang belajar dan beradaptasi dengan AI, pembelajaran mendalam, pengenalan gambar, otomatisasi penalaran dan keputusan, serta kecerdasan emosional.*
 - *AI telah menjadi konvergensi dari tiga teknologi canggih, yaitu pembelajaran mesin, pemrosesan bahasa alami, dan komputasi kognitif. Ke sinilah masa depan AI sedang menuju.*
 - *Lebih lanjut, pragmatik dapat didefinisikan melalui kognisi dan merekomendasikan pilihan "optimal", sehingga meningkatkan persaingan. Hal ini mengintegrasikan "kognisi" ke dalam model dan melatihnya menggunakan model rekomendasi berbasis komputasi kognitif untuk pengambilan keputusan dalam hal menghasilkan pilihan "paling optimal" yang akan digunakan, serta "memahami" mengapa pilihan tertentu optimal untuk digunakan (yaitu, tindakan terbaik berikutnya) mulai dari dukungan keputusan hingga pengambilan keputusan yang mendalam.*
 - *Tingkatan analitik data yang umum meliputi analitik deskriptif, analitik diagnostik, analitik prediktif, analitik preskriptif, dan komputasi kognitif. Jenis analitik data kelima komputasi kognitif menjawab pertanyaan seperti, Apa tindakan terbaik berikutnya yang dapat diambil? Ini melibatkan komputasi konteks sehubungan dengan kasus penggunaan bisnis.*
-

Kemampuan sistem bertenaga AI yang memanfaatkan big data, BI, AI, dan pembelajaran mendalam yang menghasilkan kecerdasan kognitif dan kompetitif yang bekerja secara paralel sangat membantu dalam komputasi konteks dan penalaran yang dihasilkannya. Dengan menggunakan komputasi kognitif, sistem BI perusahaan dapat memperoleh manfaat dari pengambilan keputusan yang dapat ditindaklanjuti yang memanfaatkan analitik tambahan. Perusahaan yang mengadopsi komputasi kognitif bertenaga AI dapat menggunakannya untuk berbagai kasus penggunaan seperti pengalaman pelanggan 360 derajat bertenaga AI, chatbot, solusi layanan kesehatan cerdas seperti terapi perilaku kognitif, dan analitik kapan pun, di mana pun, dan oleh siapa pun. Sistem AI memungkinkan mesin untuk meningkatkan



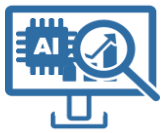
kemampuan berpikir manusia dengan menggunakan algoritma dan teknik serta memberikan hasil dengan tingkat akurasi yang wajar, sementara komputasi kognitif bertenaga AI memungkinkan mesin untuk bernalar dengan cara yang mirip dengan manusia dan mengatasi ambiguitas serta kekacauan yang terlibat dalam proses penyediaan solusi. Contoh yang baik dari sistem bertenaga AI adalah pemrosesan gambar, sedangkan contoh yang baik dari komputasi kognitif bertenaga AI adalah pengenalan wajah. Hal ini semakin diperkuat oleh fakta bahwa pemrosesan gambar dapat digunakan untuk pengenalan suara, sedangkan pengenalan wajah dapat digunakan untuk analisis sentimen.

AI berfokus pada hasil yang tepat, sementara komputasi kognitif berfokus pada hasil terbaik menggunakan penalaran berbasis bukti. Keduanya melibatkan analisis konten, tetapi komputasi kognitif membawa komputasi ke tingkat yang lebih tinggi.

Fungsi utama sistem kognitif berbasis AI adalah:

1. Adaptif—Belajar seiring perubahan data dan perkembangan kebutuhan, serta dari informasi berdasarkan masukan pengguna.
2. Interaktif—Memungkinkan interaksi manusia-komputer sehingga pengguna bisnis dapat menentukan tujuan dan sasaran mereka sendiri, sambil berinteraksi dengan sistem lain. Dengan cara ini, mereka dapat belajar dari umpan balik pengguna yang, pada gilirannya, dapat kembali ke sistem kognitif. Hal ini, pada gilirannya, memungkinkan sistem kognitif untuk belajar dari umpan balik interaktif dan berkembang dalam fungsinya.
3. Iteratif dan stateful—Memiliki kemampuan untuk "menghafal" umpan balik pengguna yang dipelajarinya dari interaksi tersebut sehingga dapat menghasilkan informasi yang peka terhadap konteks dan spesifik pengguna kapan saja.
4. Kontekstual—Sadar konteks karena kemampuannya untuk bernalar dan mengekstrak informasi spesifik konteks—domain data, domain pengguna, dan domain bisnis—sebagaimana diterapkan pada pemanfaatan data berbasis masukan dan berbasis pengetahuan (yang mencakup semua data relevan). Kuncinya di sini adalah relevansi dari semua sudut pandang yang berkaitan dengan pemrosesan tersebut dan dapat berupa informasi visual, berbasis suara, atau lainnya misalnya, memberikan makna pada data IoT melalui penalaran yang cermat.
5. Terintegrasi—Memungkinkan integrasi dengan sistem bertenaga AI lainnya untuk menghasilkan solusi yang melibatkan rekomendasi kasus terbaik.
6. Bertenaga AI—Berbasis pembelajaran mendalam, jaringan saraf tiruan, pemrosesan bahasa alami (NLP), dan analisis sentimen, selain menggunakan teknik pembelajaran mandiri, seperti pengenalan pola dan penambangan data cerdas.

Contoh nyata sistem komputasi kognitif meliputi IBM Watson® untuk analisis konten mendalam dan optimalisasi hasil; sistem layanan kesehatan concierge; dan sistem lain yang melibatkan berbagai domain, seperti yang digunakan untuk terapi, untuk mempelajari pola



penyakit, SparkCognition™ untuk dunia industri, dan sebagainya. Kerangka kerja kognisi berbasis AI dapat dibangun berdasarkan prototipe berikut:

1. Penalaran berbasis bukti dan pembuatan hipotesis yang berkembang menggunakan informasi, hasil, dan tindakan yang dipelajari
2. Keterlibatan melalui pemberian saran ahli, sebagaimana manusia
3. Penemuan pengetahuan melalui sintesis semantik informasi yang beragam dan luas secara bersamaan
4. Kepatuhan keamanan dan tata kelola yang mengutamakan manfaat bisnis perusahaan

5.3 KEAMANAN DAN TATA KELOLA DALAM BI BERBASIS AI

Sebagaimana diungkapkan banyak pakar,

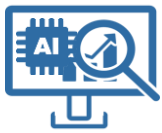
$$AI = Data + Governance + Trust$$

Sistem BI bertenaga AI generasi berikutnya dapat ditekankan sebagai

$$AI - Powered BI = Machine + Human + BI$$

Sistem BI berbasis AI sama amannya dengan keputusan yang dibuatnya dan data yang mendorong keputusan tersebut. Hal ini mencakup keamanan di seluruh alur AI end-to-end dan alur proses bisnis. Berikut ini menggambarkan beberapa poin yang perlu dipertimbangkan dari aspek keamanan sistem BI berbasis AI:

1. Pastikan aturan data tersedia untuk mendukung perlindungan data penting, mulai dari penerapan di lokasi hingga cloud. Selain itu, pastikan validasi cerdas berdasarkan data dan aturan ini tersedia di setiap langkah solusi BI berbasis AI.
2. Penting untuk memastikan penyimpanan data dan keputusan aman di lokasi atau di cloud, dan tidak ada penyimpanan sisa.
3. Sistem ini dapat mendukung penggunaan jarak jauh yang aman selain dikelola di lokasi tanpa kehilangan data, dan dapat mempertahankan keamanan yang sama ketika data dan keputusan mengalir dua arah dari lokasi ke cloud.
4. Sistem dapat menangani redundansi data dan keputusan terkait.
5. Sistem dapat memanfaatkan semua data relevan dalam bentuk input tanpa mengorbankan validitasnya.
6. Konteks eksekusi sistem BI berbasis AI selaras dengan skenario bisnis kontekstual secara terintegrasi.
7. Sistem lulus uji dampak proses BI yang dikelola oleh bisnis.
8. Sistem memberikan wawasan dari output positif dan negatif.
9. Sistem memperhitungkan konteks kueri dan konteks kumpulan hasil, serta memungkinkan parameter berbasis input pengguna.
10. Sistem dapat memperhitungkan ketidakpastian dan ambiguitas data dan output serta metode untuk mengeksposnya melalui kueri khusus berbasis AI atau SQL.



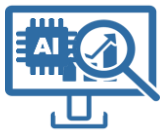
11. Sistem diimplementasikan untuk menangani pemisahan tugas dalam hal otorisasi dan autentisitas penggunaan.

Mengenai aspek tata kelola sistem BI berbasis AI, daftar berikut mencakup beberapa kriteria implementasi:

1. Persetujuan:
 - a. Pastikan terdapat mekanisme yang memungkinkan persetujuan atas informasi dan keluaran analitik oleh sistem, serta yang ditambah oleh pengguna yang berwenang dari perspektif strategis.
 - b. Pastikan bahwa aturan bisnis yang diterapkan oleh sistem dapat divalidasi dan disetujui berdasarkan hasilnya.
 - c. Sistem memungkinkan validasi untuk hasil positif dan negatif, beserta detail hasil negatif dan pengecualian yang berlaku.
 - d. Proses persetujuan dapat digunakan kembali melalui alur kerja yang terintegrasi dalam sistem.
 - e. Persetujuan tersebut dapat dikomunikasikan ke ruang kerja pengguna bisnis melalui email atau laporan instan.
2. Kualitas data yang transparan dan melibatkan pengguna dalam menentukan validasi serta persetujuannya.
3. Sistem memungkinkan audit data-ke-keputusan secara menyeluruh, secara transparan, dan membuat keluaran audit tersebut dapat dikomunikasikan melalui pesan.
4. Sistem dapat menyerahkan kepemilikan proses bisnis kepada lini bisnis kontekstual tanpa pengkodean tambahan menggunakan kecerdasan tertanam. Pengguna harus dapat meminta perubahan pada sistem yang paling sesuai dengan fungsionalitasnya.
5. Sistem memberikan konsistensi fungsionalitas di seluruh spektrum pengguna, dengan kemampuan untuk berdampingan dengan sistem BI yang ada.
6. Setiap perubahan pada sistem BI bertenaga AI didorong oleh pengguna. Misalnya, perubahan aturan dapat diinisiasi oleh pengguna, dan setiap perubahan teknis untuk meningkatkan kinerja dapat didorong oleh pengguna, berdasarkan penggunaan sistem mereka. Selain itu, perubahan ini dapat dilacak secara waktu nyata (*real-time*).
7. Tim tata kelola membuat rekomendasi tentang proses yang diterapkan, sebagaimana tercantum di atas. Tim juga memastikan bahwa rekomendasi tersebut diimplementasikan agar sistem sesuai dengan kriteria tata kelola. Kriteria ini dapat berkaitan dengan segmentasi penggunaan, umpan balik, dan pelaporan kepatuhan terhadap standar pada sistem bertenaga AI yang ada, termasuk proses yang dicakup dari permintaan hingga respons.

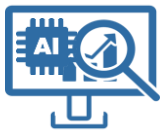
5.4 FAKTOR KEPERCAYAAN DALAM BI BERBASIS AI

Faktor kepercayaan dalam BI berbasis AI didasarkan pada data (dan informasi yang telah diproses) dan apakah penyertaan AI mengubah atau menambah proses bisnis yang ada



serta cara kerja pengguna sistem AI tersebut. Kerangka kerja untuk AI yang tepercaya dapat mencakup hal-hal berikut:

1. Sistem BI berbasis AI dapat dijelaskan karena tidak menggantikan sistem yang ada, melainkan menambahnya dengan meningkatkan keputusan dan presisi keputusan tersebut. Hal ini menjembatani kesenjangan antara pengguna bisnis AI dan ilmuwan data yang paham teknologi tentang AI. Dengan demikian, pengguna bisnis dapat memperoleh penjelasan tentang bagaimana BI berbasis AI etis di perusahaan dan tidak merusak citra bisnis.
2. Wawasan dan pengambilan keputusan berbasis data berbasis AI harus spesifik konteks. Ini berarti melibatkan pengguna lob untuk mengetahui bahwa solusi BI berbasis AI relevan dengan apa yang ingin dihasilkannya sejauh menyangkut kasus penggunaan bisnis yang dihadapi pengguna ini.
3. Sistem BI bertenaga AI memungkinkan umpan balik berulang yang berkelanjutan dari pengguna ketika mencoba menganalisis atau menggunakan hasil sistem. Analisis dan wawasan berbasis mesin terkadang bisa ambigu atau tidak akurat, dan semakin banyak informasi dalam bentuk umpan balik atau ulasan yang dimasukkan ke dalam sistem AI dan BI, semakin berkembang pula sistem tersebut dalam menghasilkan keputusan yang cerdas. Misalnya, untuk melayani konsumen yang lebih suka melihat foto orang-orang yang membeli produk tertentu, seperti jas, sistem pemrosesan gambar yang dapat mengidentifikasi wajah orang-orang tersebut, serta warna dan merek jas yang mereka kenakan, dan terintegrasi ke dalam BI, akan lebih tepercaya jika dapat menjelaskan bagaimana sistem tersebut sampai pada keputusan tertentu dan mengapa keputusan tersebut optimal. Jika sistem memprediksi bahwa peningkatan produksi merek tertentu dapat mengoptimalkan permintaan, memberi tahu pengguna bahwa kampanye tertentu yang diterapkan di seluruh segmen bisnis adalah alasan prediksi tersebut memungkinkan pengguna untuk memercayai prediksi tersebut. Selain itu, bagaimana korelasi dengan kampanye serupa yang telah diterapkan telah meningkatkan permintaan tersebut menunjukkan bahwa keputusan yang dibuat sudah tepat, baik dari sudut pandang bisnis maupun pengguna. Dengan demikian, menggabungkan analitik prediktif dan preskriptif yang menjawab aspek bagaimana dan mengapa suatu keputusan dapat menanamkan kepercayaan pada pengguna bisnis untuk sistem semacam itu.
4. Eksplorasi umpan balik yang dianggap palsu oleh pengguna BI dapat berdampak negatif pada kepercayaan. Hal ini mengecilkan faktor kepercayaan dalam sistem yang diberdayakan AI. Dalam kasus seperti itu, sistem BI yang diberdayakan AI dapat menghasilkan konten berdasarkan umpan balik negatif dan positif yang dapat diinterpretasikan oleh pengguna BI sebagai lebih relevan dengan keputusan dalam konteksnya. Interaksi dengan pengguna, umpan balik dari mereka, dan kemampuan untuk mengirim notifikasi otomatis berdasarkan umpan balik tersebut yang berisi keluaran berbasis AI berbasis data, serta keberlakuan umpan balik tersebut, sangat

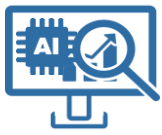


membantu dalam melindungi kepercayaan terhadap sistem yang diberdayakan AI— dari masukan data hingga keluaran keputusan.

5. Sistem ini sadar konteks bisnis dan aman di seluruh alur AI, dari awal hingga akhir.
6. Dan yang tak kalah pentingnya, kumpulan data dapat ditingkatkan untuk menyertakan hasil dari sistem BI yang diberdayakan AI sehingga sistem tersebut dapat menggunakannya untuk meningkatkan pengambilan keputusan. Kumpulan data yang disempurnakan ini dapat disediakan bagi pengguna sebagai data uji yang disempurnakan agar mereka dapat menguji sendiri dan melihat peningkatan hasilnya. Hal ini terjadi karena kumpulan data telah dibuat dengan umpan balik berbasis pengguna.

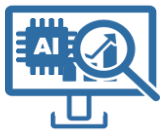
5.5 RINGKASAN

Bab ini mengeksplorasi aspek-aspek gelombang AI dan BI bertenaga AI berikutnya dalam hal komputasi kognitif, keamanan dan tata kelola, serta faktor kepercayaan dari solusi AI yang bertemu dengan BI. Bab ini mencoba menguraikan pertimbangan utama di setiap area ini dan menunjukkan bagaimana sistem BI bertenaga AI yang mencakup komputasi kognitif dan, pada saat yang sama, aman, didukung tata kelola, dan tepercaya merupakan solusi yang terbukti dan menonjol sebagai sistem masa depan yang tangguh.

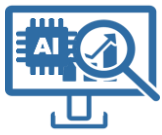


DAFTAR PUSTAKA

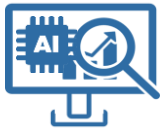
- Ahmad, H., Hanandeh, R., Alazzawi, F. R. Y., Al-Daradkah, A., ElDmrar, A. A. T., Ghaith, Y. M., & Darawsheh, S. R. (2023). The effects of big data, artificial intelligence, and business intelligence on e-learning and business performance: Evidence from Jordanian telecommunication firms. *International Journal of Data & Network Science*, 7(1).
- Alghamdi, N. A., & Al-Baity, H. H. (2022). Augmented analytics driven by AI: A digital transformation beyond business intelligence. *Sensors*, 22(20), 8071.
- Al-Okaily, A., Teoh, A. P., & Al-Okaily, M. (2023). Evaluation of data analytics-oriented business intelligence technology effectiveness: an enterprise-level analysis. *Business Process Management Journal*, 29(3), 777-800.
- Amirineni, S. (2024). Enhancing Predictive Analytics in Business Intelligence through Explainable AI: A Case Study in Financial Products. *Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023*, 6(1), 258-288.
- 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.
- Bauskar, S. (2024). Business analytics in enterprise system based on application of artificial intelligence. *International Research Journal of Modernization in Engineering Technology and Science*.
- Bharadiya, J. (2023). The impact of artificial intelligence on business processes. *European journal of technology*, 7(2), 15-25.
- Bharadiya, J. P. (2023). The role of machine learning in transforming business intelligence. *International Journal of Computing and Artificial Intelligence*, 4(1), 16-24.
- Bharadiya, J. P. (2023). The role of machine learning in transforming business intelligence. *International Journal of Computing and Artificial Intelligence*, 4(1), 16-24.
- Cahyati, I., Fauzi, A., Hasanuddin, H., Zuhri, I., Hibatullah, H., Dwi, N., ... & Felisyana, R. (2024). Penerapan Business Intelligence Dengan Artificial Intelligence Pada E-Commerce. *SENTRI: Jurnal Riset Ilmiah*, 3(6), 2741-2756.
- Caruso, S., Bruccoleri, M., Pietrosi, A., & Scaccianoce, A. (2023). Artificial intelligence to counteract "KPI overload" in business process monitoring: the case of anti-corruption in public organizations. *Business process management journal*, 29(4), 1227-1248.
- Çelebi, H. İ. (2021). Artificial intelligence applications in management information systems: a comprehensive systematic review with business analytics perspective. *Artificial Intelligence Theory and Applications*, 1(1), 25-56.
- Charles, V., Garg, P., Gupta, N., & Agarwal, M. (2023). Data Analytics and Business Intelligence. *Data Analytics and Business Intelligence*.
- Chen, Y., & Lin, Z. (2021). Business intelligence capabilities and firm performance: A study in China. *International journal of information management*, 57, 102232.



- Chen, Y., Li, C., & Wang, H. (2022). Big data and predictive analytics for business intelligence: A bibliographic study (2000–2021). *Forecasting*, 4(4), 767-786.
- Chinta, S. (2022). Integrating artificial intelligence with cloud business intelligence: Enhancing predictive analytics and data visualization. *Iconic Research And Engineering Journals*, 5(9).
- Chintala, S. (2024). Next-Gen BI: Leveraging AI for competitive advantage. *International Journal of Science and Research (IJSR)*, 13(7), 972-977.
- 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.
- Chintala, S., & Thiyagarajan, V. (2023). Harnessing AI for Transformative Business Intelligence Strategies. *ESP International Journal of Advancements in Computational Technology (ESPIJACT) Volume*, 1, 81-96.
- Citak, J., Owoc, M. L., & Weichbroth, P. (2021). A note on the applications of artificial intelligence in the hospitality industry: preliminary results of a survey. *Procedia Computer Science*, 192, 4552-4559.
- Collins, A., Hamza, O., Eweje, A., & Babatunde, G. O. (2024). Integrating 5G core networks with business intelligence platforms: Advancing data-driven decision-making. *International Journal of Multidisciplinary Research and Growth Evaluation*, 5(1), 1082-1099.
- Doshi, R., Hiran, K. K., Mijwil, M. M., & Anand, D. (2023). To that of artificial intelligence, passing through business intelligence. In *Handbook of Research on AI and Knowledge Engineering for Real-Time Business Intelligence* (pp. 1-16). IGI Global.
- Eboigbe, E. O., Farayola, O. A., Olatoye, F. O., Nnabugwu, O. C., & Daraojimba, C. (2023). Business intelligence transformation through AI and data analytics. *Engineering Science & Technology Journal*, 4(5), 285-307.
- Emon, M. M. H., Khan, T., Rahman, M. A., & Siam, S. A. J. (2024, September). Factors influencing the usage of artificial intelligence among Bangladeshi professionals: Mediating role of attitude towards the technology. In *2024 IEEE International Conference on Computing, Applications and Systems (COMPAS)* (pp. 1-7). IEEE.
- Farayola, O. A. (2024). Revolutionizing banking security: integrating artificial intelligence, blockchain, and business intelligence for enhanced cybersecurity. *Finance & Accounting Research Journal*, 6(4), 501-514.
- Figalist, I., Elsner, C., Bosch, J., & Olsson, H. H. (2022). Breaking the vicious circle: A case study on why AI for software analytics and business intelligence does not take off in practice. *Journal of Systems and Software*, 184, 111135.
- Gad-Elrab, A. A. (2021). Modern business intelligence: Big data analytics and artificial intelligence for creating the data-driven value. In *E-Business-Higher Education and Intelligence Applications*. IntechOpen.
- Hariguna, T., & Ruangkanjanases, A. (2024). Assessing the impact of artificial intelligence on customer performance: A quantitative study using partial least squares methodology. *Data Science and Management*, 7(3), 155-163.



- Hmoud, B. (2021). The adoption of artificial intelligence in human resource management and the role of human resources. In *Forum Scientiae Oeconomia* (Vol. 9, No. 1, pp. 105-118). Wydawnictwo Naukowe Akademii WSB.
- Jiménez-Partearroyo, M., & Medina-López, A. (2024). Leveraging business intelligence systems for enhanced corporate competitiveness: Strategy and evolution. *Systems*, 12(3), 94.
- Khaddam, A. A., Alzghoul, A., Abusweilem, M. A., & Abousweilem, F. (2023). Business intelligence and firm performance: a moderated-mediated model. *The Service Industries Journal*, 43(13-14), 923-939.
- Kumar, M., Raut, R. D., Mangla, S. K., Ferraris, A., & Choubey, V. K. (2024). The adoption of artificial intelligence powered workforce management for effective revenue growth of micro, small, and medium scale enterprises (MSMEs). *Production Planning & Control*, 35(13), 1639-1655.
- Kundurur, A. R. (2023). Effective usage of artificial intelligence in enterprise resource planning applications. *International Journal of Computer Trends and Technology*, 71(4), 73-80.
- Mehrotra, P. (2021). Applications of Artificial Intelligence in the Realm of Business Intelligence. In *Research Anthology on Artificial Intelligence Applications in Security* (pp. 358-386). IGI Global Scientific Publishing.
- Michael, C. I., Ipede, O. J., Adejumo, A. D., Adenekan, I. O., Adebayo, D., Ojo, A. S., & Ayodele, P. A. (2024). Data-driven decision making in IT: Leveraging AI and data science for business intelligence. *World Journal of Advanced Research and Reviews*, 23(01), 432-439.
- 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).
- Ramírez, J. G. C. (2023). Incorporating Information Architecture (ia), Enterprise Engineering (ee) and Artificial Intelligence (ai) to Improve Business Plans for Small Businesses in the United States. *Journal of Knowledge Learning and Science Technology* ISSN: 2959-6386 (online), 2(1), 115-127.
- 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.
- Selvarajan, G. P. (2024). The Role of Machine Learning Algorithms in Business Intelligence: Transforming Data into Strategic Insights. *International Journal of Advanced Research and Interdisciplinary Scientific Endeavours*, 1(7), 391-400.
- Taherdoost, H., & Madanchian, M. (2023). Artificial intelligence and sentiment analysis: A review in competitive research. *Computers*, 12(2), 37.
- Tong-On, P., Siripipatthanakul, S., & Phayaphrom, B. (2021). The implementation of business intelligence using data analytics and its effects towards on performance in the hotel industry in Thailand. *International Journal of Behavioral Analytics*, 1(2).
- Wang, J., Omar, A. H., Alotaibi, F. M., Daradkeh, Y. I., & Althubiti, S. A. (2022). Business intelligence ability to enhance organizational performance and performance evaluation capabilities by



improving data mining systems for competitive advantage. *Information Processing & Management*, 59(6), 103075.

Wang, Z., Li, M., Lu, J., & Cheng, X. (2022). Business Innovation based on artificial intelligence and Blockchain technology. *Information Processing & Management*, 59(1), 102759.

Weber, F. (2023). *Artificial intelligence for business analytics*. Springer Fachmedien Wiesbaden, Wiesbaden.

Zohuri, B., Moghaddam, M., & Mossavar-Rahmani, F. (2022). Business resilience system integrated artificial intelligence system. *International Journal of Theoretical & Computational Physics*, 3(1), 1-7.

Hubungan Antara Kecerdasan Buatan (AI) dan Kecerdasan Bisnis (BI)

Dr. 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 978-634-7227-33-1 (PDF)



9

786347

227331