LAPORAN
DATA ANALYTICS COMPETITION FIND IT! 2024

Tim AI Geniuses

Disusun Oleh :

Jemmy Febryan
German Mindo Simarmata
Meirida Karisma Putri

MALANG

2024

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam beberapa dekade terakhir, perekonomian global telah mengalami transformasi yang signifikan akibat pesatnya perkembangan teknologi digital. Ekonomi global modern berada di bawah pengaruh signifikan teknologi digital, yang dengan cepat mengubah praktik industri, meningkatkan kinerja industri, dan mendorong inovasi di semua bidang industri (Fajrillah et al., 2020). Peran teknologi digital dalam mengubah praktik industri tidak dapat dipandang sebelah mata. Teknologi digital dianggap mampu memberikan kontribusi signifikan terhadap peningkatan efisiensi operasional, peningkatan integrasi sistem, dan kemudahan akses untuk menjangkau pasar global (Bagale, 2021). Teknologi digital memungkinkan bentuk-bentuk baru perilaku pasar, interaksi, atau pengalaman (Lamberton & Stephen, 2016). Teknologi digital juga menjadi katalis inovasi di semua sektor industri, termasuk ritel. Kemajuan teknologi telah mempercepat perubahan besar dan menyebabkan gangguan signifikan pada lanskap ritel (Sanskar et al., 2021).

Industri ritel telah mengalami transformasi yang signifikan dalam beberapa tahun terakhir, terutama seiring dengan perkembangan teknologi dan pergeseran perilaku konsumen (Boari et al., 2024). Salah satu tantangan utama yang dihadapi oleh industri ritel adalah meningkatnya persaingan, yang menuntut mereka untuk tidak hanya fokus menawarkan produk yang menarik tetapi juga memperhatikan bagaimana mereka dapat bersaing secara efektif dengan mampu mengidentifikasi dan memahami kebutuhan, preferensi, dan perilaku pembelian pelanggan. Menurut penelitian Balaka (2023), memahami pelanggan tidak hanya memungkinkan industri untuk meningkatkan strategi pemasaran dan promosi, namun juga meningkatkan layanan pelanggan, retensi, dan pada akhirnya keuntungan. Selain itu, pergeseran paradigma juga terlihat pada strategi pemasaran. Sejarah pemasaran telah menyaksikan evolusi ini, dimana metode yang berhasil di masa lalu tidak lagi relevan atau efektif di era baru ini (Budiarti, 2023). Konsumen modern memiliki akses lebih baik terhadap informasi dan preferensi yang semakin dinamis. Industri ritel harus mampu mengikuti tren dan mengubah pendekatan pemasaran mereka untuk memenuhi kebutuhan dan ekspektasi pelanggan.

Salah satu cara yang efektif untuk mencapai kepuasan pelanggan dalam suatu industri ritel adalah melalui penggunaan data. Penggunaan data dalam menganalisis perilaku pelanggan telah menjadi fokus utama dalam strategi pemasaran dan promosi (Kwon, 2017). Industri ritel dapat mengumpulkan, menganalisis, dan mengambil keputusan berdasarkan informasi yang relevan dan akurat dengan memanfaatkan perkembangan teknologi digital. Perkembangan teknologi digital, mulai dari komputasi awan hingga *Artificial Intelligence* (AI), telah merevolusi berbagai praktik industri, termasuk lanskap ritel yang saat ini sedang berkembang pesat (Muttaqin et al., 2023) Teknologi AI, khususnya *machine learnin*g, telah menjadi katalisator utama dalam transformasi industri ritel. *Machine learning* memiliki kemampuan untuk mengumpulkan, menganalisis, dan memahami data dengan cepat dan akurat sehingga

memungkinkan industri ritel untuk mengambil keputusan yang lebih cerdas dan tepat waktu.

Penggunaan data dan teknologi *machine learning* juga memungkinkan industri ritel untuk meningkatkan personalisasi layanan pelanggan. *Machine learning* dapat digunakan untuk menganalisis pola pembelian pelanggan, mengidentifikasi preferensi produk, dan memprediksi respons pelanggan terhadap berbagai jenis promosi (Apriana & Yuliansyah, 2024). Hal ini dapat membantu industri ritel dalam mengirimkan tawaran promosi yang disesuaikan dengan preferensi belanja dan perilaku pembelian individu pelanggan. Teknologi *machine learning* meningkatkan kemungkinan respon positif dari pelanggan industri ritel. Selain itu, *machine learning* juga membantu industri ritel dalam mengoptimalkan alokasi sumber daya. Dengan menganalisis data secara cermat, mereka dapat menentukan strategi promosi yang paling efektif, menargetkan segmentasi pelanggan yang tepat, dan mengalokasikan anggaran promosi dengan lebih bijak.

Penulis melakukan penelitian lebih lanjut dengan memanfaatkan data yang kompleks dan beragam untuk mengembangkan model yang akurat dan andal. Model tersebut secara akurat dapat menyediakan informasi yang berharga tentang pelanggan, termasuk data demografi seperti usia, pendidikan, dan status pernikahan, data ekonomi seperti pendapatan, serta perilaku belanja seperti pola pembelian dan partisipasi dalam promosi. Selain itu, penulis juga melakukan inovasi lebih lanjut dalam bentuk *website* dengan mengintegrasikan teknologi *machine learning* untuk menciptakan sistem prediksi yang mampu menentukan pada promosi ke berapa pelanggan akan menerima program promosi tertentu. Dengan demikian, penulis berharap inovasi ini dapat memberikan nilai tambah yang signifikan bagi industri ritel dalam merencanakan dan melaksanakan strategi pemasaran dan promosi, memungkinkan mereka untuk mengoptimalkan penggunaan anggaran promosi, dan meningkatkan efektivitas kampanye promosi secara keseluruhan.

1.2 Tujuan

Adapun tujuan yang diperoleh dari proses analisis data ini dapat dipaparkan sebagai berikut:

1.2.1 Mengoptimalkan sumber daya melalui model sehingga dapat membantu industri ritel menentukan di mana dan kapan harus mengalokasikan sumber daya promosi mereka untuk mencapai hasil terbaik.

1.2.2 Membantu industri ritel untuk menargetkan promosi kepada pelanggan yang memiliki potensi tinggi untuk merespon dan melakukan pembelian.

1.2.3 Mengembangkan strategi pemasaran dalam industri ritel yang lebih efektif dan tepat sasaran.

1.2.4 Meningkatkan efektivitas dan efisiensi program promosi industri ritel di masa depan melalui pemanfaatan teknik *machine learning* untuk memprediksi respon pelanggan terhadap program promosi tertentu.

1.3 Manfaat

Adapun manfaat yang diperoleh dari proses analisis data ini dapat dipaparkan sebagai berikut:

1.3.1 Bagi Peneliti:

Memberikan pengalaman dan keterampilan dalam melakukan pengolahan data kompleks, pengembangan model, dan analisis prediktif yang efektif melalui program promosi dan penggunaan *machine learning*.

1.3.2 Bagi Pihak Penyelenggara:

Meningkatkan reputasi sebagai penyelenggara yang mendukung inovasi dan pengembangan di industri ritel sehingga menarik minat peserta dan sponsor untuk lomba di masa mendatang.

1.3.3 Bagi Pelaku Industri:

Mendapatkan model prediktif yang dapat membantu industri ritel mengoptimalkan alokasi sumber daya promosi, meningkatkan target pasar yang tepat, dan memperkuat posisi persaingan dengan memanfaatkan data dan teknologi untuk mengambil keputusan yang lebih cerdas dan tepat waktu.

1.3.4 Bagi Pemerintah:

Mendorong pertumbuhan ekonomi melalui peningkatan kinerja industri ritel sehingga memberikan dampak positif terhadap lapangan kerja dan pendapatan nasional.

BAB II

DASAR TEORI

2.1 *Preprocessing* Data

Preprocessing data telah menjadi aspek terpenting dalam penerapan teknik *machine learning* pada data yang tersedia (Meneses & Rodriguez, 2018). *Preprocessing* data merupakan teknik yang digunakan peneliti sebelum menerapkan pendekatan pembelajaran mesin. Teknik ini termasuk mengubah data mentah menjadi kumpulan data yang bersih. Data dikumpulkan dari berbagai sumber yang awalnya tidak cocok untuk dianalisis (Farquad & Bose, 2012). Pada awalnya, data yang dikumpulkan diproses terlebih dahulu menggunakan teknik yang berbeda seperti mengisi nilai kosong, normalisasi, dan membagi data menjadi bagian *training* dan *testing* sesuai dengan rasio pemisahan (Sekhar & Kumar, 2022). Dengan demikian, melalui langkah-langkah *preprocessing* data yang tepat dan pembagian data yang baik, proses *machine learning* dapat menghasilkan model yang lebih akurat dan dapat diandalkan. (Dablain et al, 2022).

2.1.1 *Feature Selection*

Feature selection (*variable elimination*) adalah proses pemilihan subset dari fitur atau variabel yang ada dalam dataset untuk digunakan dalam membangun model *machine learning*. Menurut Girish Chandrashekar & Ferat Sahin (2014), *feature selection* dapat membantu dalam memahami data, meningkatkan kecepatan komputasi, mengurangi dimensi data, meningkatkan kinerja prediktor, dan menghindari *overfitting*. *Feature selection* berfokus pada pemilihan subset variabel dari masukan yang dapat mendeskripsikan data masukan secara efisien sekaligus mengurangi efek dari *noise* atau variabel yang tidak relevan sehingga tetap memberikan hasil prediksi yang baik (Guyon, I. & Elisseeff, A., 2003).

Dalam *feature selection*, terdapat beberapa metode yang umum digunakan, diantaranya adalah *Univariate Feature Selection*, *Recursive Feature Elimination* (RFE), *Feature Importance*, *Principal Component Analysis* (PCA), *Lasso Regression*, dan *Forward* dan *Backward Feature Selection* (Chandrashekar & Sahin, 2014). Setiap teknik memiliki pendekatan yang berbeda dalam memilih subset fitur yang optimal tergantung pada karakteristik data dan tujuan analisis. Dengan menerapkan teknik *feature selection*, akan dihasilkan model yang lebih efisien, lebih mudah diinterpretasikan, dan memiliki kinerja yang lebih baik secara keseluruhan. Kombinasi dari beberapa teknik Feature Selection sering digunakan untuk memperoleh subset fitur yang optimal sesuai dengan kebutuhan dan karakteristik dataset (Putra et al, 2023).

2.1.2 *Feature Engineering*

Feature engineering merupakan proses penting dalam pra-pemrosesan data yang bertujuan untuk mengubah data mentah menjadi format yang dapat digunakan oleh algoritma *machine learning*. Salah satu metode yang umum digunakan dalam *feature engineering* adalah LabelEncoder. LabelEncoder

berfungsi untuk mengonversi label kategorikal menjadi nilai numerik sehingga algoritma *machine learning* dapat memprosesnya dengan lebih efisien (Zheng & Casari, 2018). Dalam konteks ini, setiap kategori unik dalam fitur kategorikal diberikan nilai integer yang berbeda. Penggunaan LabelEncoder sangat bermanfaat ketika menangani dataset dengan banyak fitur kategorikal, karena metode ini menyederhanakan proses pengkodean tanpa mengubah makna asli dari data (Muller & Guido, 2016). Dengan demikian, LabelEncoder membantu dalam meningkatkan kinerja model *machine learning* dengan memastikan bahwa data input berada dalam format yang optimal untuk analisis lebih lanjut.

Selain LabelEncoder, salah satu metode alternatif dalam *feature engineering* adalah menggunakan teknik *One-Hot Encoding*, di mana setiap kategori dalam fitur kategorikal diubah menjadi sebuah vektor biner. *One-Hot Encoding* hanya memiliki satu elemen yang bernilai 1 untuk menunjukkan keanggotaan kategori tersebut, sementara elemen lainnya bernilai 0. Pendekatan ini membantu mencegah model untuk menafsirkan hubungan ordinal atau hierarkis antar-kategori yang tidak ada, sehingga cocok digunakan untuk data yang tidak memiliki urutan atau tingkat prioritas tertentu di antara kategorinya (Raschka & Mirjalili, 2019).

2.1.3 Penanganan *Outlier*.

Penanganan *outlier* merupakan langkah yang krusial dalam *preprocessing* data karena *outlier* memiliki potensi besar untuk mengganggu kualitas dan interpretasi dari model *machine learning* (Geron, 2022). Identifikasi *outlier* dapat dilakukan dengan berbagai metode, seperti menggunakan metrik statistik atau teknik visualisasi. Setelah *outlier* teridentifikasi, langkah selanjutnya adalah memutuskan bagaimana menanganinya secara efektif. Penghapusan *outlier* adalah pendekatan yang umum, tetapi perlu diperhatikan bahwa penghapusan *outlier* dapat mengurangi jumlah data yang tersedia untuk melatih model, sehingga dapat mengurangi performa model jika data yang sudah langka menjadi semakin terbatas.

Selain itu, transformasi data dan penggunaan metode *robust* seperti penggunaan median dapat menjadi alternatif yang lebih aman dalam menangani *outlier* tanpa mengorbankan jumlah data. Imputasi *outlier* juga dapat menjadi solusi jika outlier tersebut penting untuk tetap dipertahankan dalam analisis (Raschka & Mirjalili, 2019). Dengan memilih teknik penanganan outlier yang sesuai dengan karakteristik data dan tujuan analisis, kita dapat memastikan bahwa model *machine learning* yang dihasilkan mampu memberikan prediksi yang akurat dan dapat digeneralisasi dengan baik terhadap data baru.

2.1.4 Pengisian Nilai Kosong (*Missing Value Imputation*)

Dalam bukunya "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" Aurelien Geron (2022), menyebutkan bahwa *missing value imputation* adalah proses mengisi nilai-nilai yang hilang (*missing values*) dengan nilai yang dihitung berdasarkan nilai-nilai yang ada dalam dataset. Tujuannya adalah untuk menjaga integritas data dan memastikan bahwa

dataset yang digunakan dalam analisis atau model tidak memiliki kekosongan yang dapat mengganggu hasilnya. Selanjutnya, Geron membahas berbagai teknik yang dapat digunakan dalam *missing value imputation*, seperti mengisi nilai kosong dengan nilai rata-rata, median, atau modus dari kolom yang bersangkutan, atau menggunakan teknik berbasis model prediksi untuk memprediksi nilai yang hilang. Dengan melakukan langkah ini, dataset menjadi lebih lengkap dan siap digunakan dalam analisis atau pembangunan model *machine learning*, sehingga meningkatkan kualitas dan reliabilitas hasil yang dihasilkan.

Missing value imputation adalah salah satu tahap penting dalam proses *preprocessing* data dalam *machine learning*. Aurelien Geron menjelaskan bahwa dengan mengisi nilai-nilai yang hilang dengan nilai yang dihitung berdasarkan informasi yang ada dalam dataset, kita dapat mempertahankan integritas data dan memastikan dataset yang digunakan dalam analisis atau model tidak memiliki kekosongan yang dapat mengganggu hasilnya. Dengan demikian, *missing value imputation* menjadi langkah penting dalam *preprocessing* data yang mengatasi kekurangan data dan memastikan kelengkapan data untuk digunakan dalam analisis atau pembangunan model *machine learning*.

2.1.5 Normalisasi Data

Normalisasi data merupakan proses standarisasi atau pengukuran ulang skala nilai-nilai dalam dataset sehingga memiliki skala yang seragam atau relatif terhadap satu sama lain (Geron, A., 2022). Tujuan dari normalisasi data adalah untuk mengurangi kesenjangan antara nilai-nilai yang berbeda dalam dataset, sehingga memungkinkan algoritma *machine learning* untuk bekerja lebih efektif dan menghasilkan model yang lebih baik. Andreas C. Muller & Sarah Guido (2016), menjelaskan beberapa teknik dalam normalisasi data, seperti:

- *Min-Max Scaling*

  Metode ini mengubah nilai-nilai pada fitur menjadi rentang tertentu, biasanya antara 0 dan 1. Metode *Min-Max Scaling* dilakukan dengan menggunakan rumus:

  $$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

  Dimana X adalah nilai asli, X_{min} adalah nilai minimum dalam dataset, dan X_{max} adalah nilai maksimum dalam dataset.

- *Standardization* (*Z-Score Normalization*)

  Metode ini mengubah distribusi data sehingga memiliki rata-rata 0 dan standar deviasi 1. Metode *Standardization* (*Z-Score Normalization*) dilakukan dengan menggunakan rumus:

  $$X_{scaled} = \frac{X-\mu}{\sigma}$$

Dimana X adalah nilai asli, μ adalah rata-rata dari dataset, dan σ adalah standar deviasi dari dataset.

- *Robust Scaling*

  Metode ini mirip dengan Min-Max Scaling, namun lebih tahan terhadap outlier dalam dataset. Metode *Robust Scaling* dilakukan dengan menggunakan rumus:

$$X_{scaled} = \frac{X - Q_1}{Q_3 - Q_1}$$

  Dimana X adalah nilai asli, adalah kuartil pertama (*25th percentile*) dari dataset, dan Q_3 adalah kuartil ketiga (*75th percentile*) dari dataset.

- *Normalization*

  Metode ini mengubah nilai-nilai pada fitur sehingga setiap baris data memiliki panjang 1 dalam ruang vektor. Metode *Normalization* dilakukan dengan menggunakan rumus:

$$X_{scaled} = \frac{X}{||X||}$$

  Dimana X adalah nilai asli, dan $||X||$ adalah norma L_2 dari vektor X.

Teknik Normalisasi data seperti *Min-Max Scaling* dan *Standardization* sangat penting dalam pra-pemrosesan data karena membantu menghindari masalah seperti dominasi fitur atau variabel yang memiliki rentang nilai yang sangat besar dibandingkan dengan fitur lainnya (Geron, 2022). Dengan normalisasi, dipastikan setiap fitur memiliki pengaruh yang seimbang terhadap pembentukan model. Dengan demikian, model yang dihasilkan menjadi lebih stabil dan konsisten dalam memprediksi data baru.

2.1.6 *Train-Validation Split*

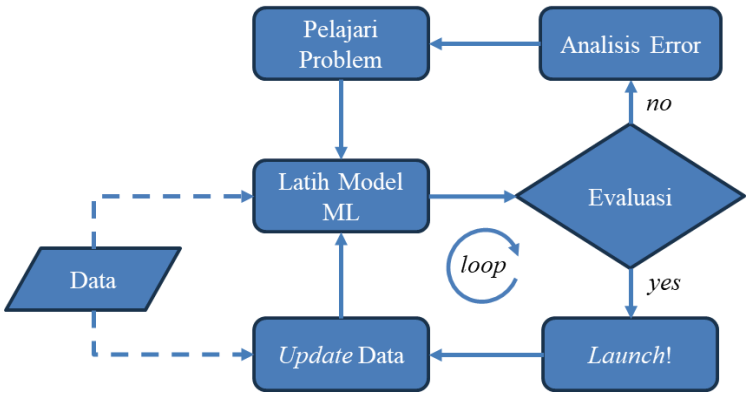
Train-validation split atau yang umumnya dikenal sebagai *train-test split* adalah salah satu konsep penting dalam *machine learning* yang digunakan untuk membagi *dataset* menjadi dua *subse*t yang berbeda, yaitu *subset* pelatihan (*train set*) dan *subset* validasi (*validation se*t) (Geron, A., 2022). Dua subset yang berbeda tersebut dapat menguji model pada data yang independen, sehingga memperoleh perkiraan yang lebih realistis tentang kinerja model di dunia nyata. Tujuan dari *train-validation split* adalah untuk menguji kinerja model yang telah dilatih pada data yang tidak pernah "dilihat" oleh model sebelumnya, sehingga menghindari *overfitting* dan menghasilkan model yang lebih umum dan dapat digeneralisasi dengan baik (Müller, A.C. & Guido, S., 2016).

Train-validation split seringkali digunakan untuk melatih model menggunakan *subset* pelatihan, menguji kinerja model menggunakan *subset* validasi, dan mengevaluasi performa model berdasarkan metrik evaluasi yang relevan. *Train-validation split* adalah langkah penting dalam proses pengembangan model *machine learning* karena membantu memastikan bahwa

model yang dihasilkan mampu menggeneralisasi dengan baik terhadap data baru yang belum pernah dilihat sebelumnya (Geron, A., 2022). Dengan menggunakan *train-validation split* secara bijaksana, dapat mengoptimalkan model *machine learning* untuk mencapai kinerja yang optimal dan mampu mengatasi berbagai tantangan dalam pemodelan data.

2.2 *Machine Learning*

Menurut Arthur Samuel (1950), *machine learning* merupakan bidang studi yang memberikan komputer kemampuan untuk belajar tanpa diprogram secara eksplisit. Geron, dkk (2021) mendefinisikan *machine learning* juga sering disebut sebagai ilmu dan seni tentang pemrograman komputer yang dipelajari melalui data. *Machine learning* berhubungan dengan penemuan pola data dan ketidakteraturan terkait data (Furnkranz, J., 2012). Tom Mitchell (1997) juga memberikan sebuah definisi dimana *machine learning* adalah suatu program komputer yang dikatakan telah melakukan pembelajaran dari pengalaman E (*experience*) terhadap tugas T (*task*) dan mengukur peningkatan kinerja P (*performance measure*), jika kinerja Tugas T diukur oleh kinerja P, maka meningkatkan pengalaman E. Alur kerja *machine learning* dalam menyelesaikan suatu permasalahan dapat dilihat pada **Gambar 2.1**.



Gambar 2.1 Alur Kerja *Machine Learning*

Machine learning memiliki beberapa kelebihan yang penting bagi dunia teknologi saat ini. Salah satunya adalah kemampuan mengolah data dengan cepat dan akurat sehingga memungkinkan pengambilan keputusan lebih cepat (Jadhav, T., dkk. 2023). Selain itu, menurut Katrin Sophie Bohnsack (2023), model *machine learning* dapat belajar dari data yang diberikan dan secara otomatis meningkatkan performanya seiring waktu. Hal ini memungkinkan sistem untuk mengidentifikasi pola-pola yang kompleks dan menarik, bahkan pada data yang sangat besar dan kompleks sekalipun. *Machine learning* dapat digunakan untuk mengotomatisasi tugas-tugas yang repetitif dan memakan waktu, seperti klasifikasi email, deteksi fraud, atau pengelompokan data (Kuswardani, D., dkk. 2024).

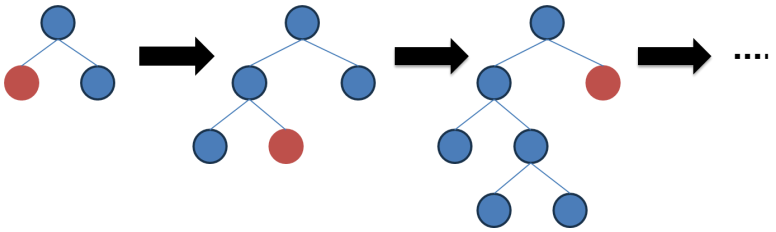
2.3 *Machine Learning* Tradisional

Machine learning tradisional adalah pendekatan dalam bidang kecerdasan buatan yang berfokus pada pengembangan model matematika dan algoritma untuk mempelajari pola-pola dari data (Gong, X., dkk. 2019). Dalam *machine learning* tradisional, data yang digunakan untuk melatih model biasanya harus memiliki struktur yang terstruktur dan jelas, serta memerlukan fitur-fitur yang dihasilkan secara manual oleh ahli domain (Wang, P., dkk. 2021). Proses pelatihan model melibatkan

optimasi parameter-parameter model untuk mencapai kinerja yang baik dalam memprediksi atau mengklasifikasikan data baru. *Machine learning* tradisional memiliki kelebihan dalam interpretabilitas yang tinggi, dimana kita dapat memahami secara intuitif bagaimana model membuat keputusan berdasarkan fitur-fitur yang ada (Wang, P., dkk. 2021). Christopher M. Bishop menyoroti keandalan *machine learning* tradisional pada data terbatas. Menurutnya, model seperti regresi linier dan pohon keputusan dapat memberikan hasil yang memuaskan meski dengan dataset yang tidak terlalu besar (Khan. A., dkk. 2022).

2.3.1 *Light Gradient Boosting Machine* (LightGBM)

Light Gradient Boosting Machine (LightGBM) adalah salah satu algoritma *machine learning* yang sangat efisien dalam menangani data besar dan memiliki performa yang baik dalam tugas-tugas seperti klasifikasi dan regresi (Ke, G., Meng, Q., dkk, 2017). Menurut Haijian Shi (2007), LightGBM menggunakan pendekatan *leaf-wise* untuk membangun pohon keputusan (*decision tree*). Sebagian besar algoritma pembelajaran pohon keputusan menumbuhkan pohon berdasarkan tingkat (kedalaman), seperti yang tertera pada **Gambar 2.2.**

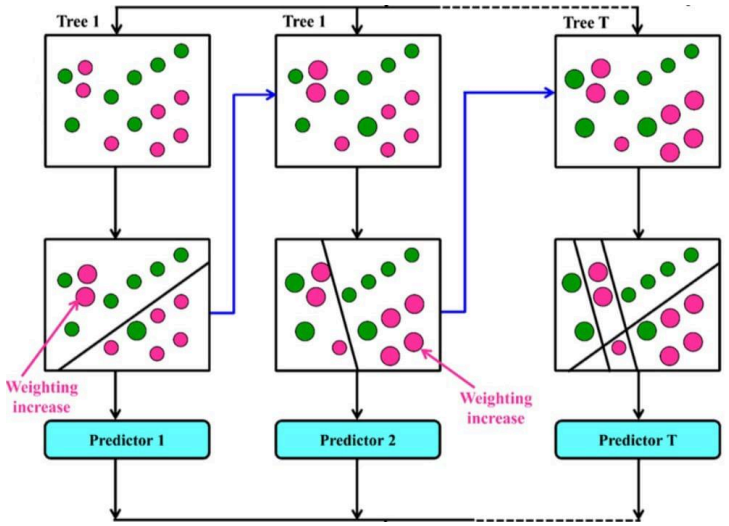


Gambar 2.2 Ilustrasi *Leaf-Wise* Pohon Keputusan

LightGBM memanfaatkan histogram untuk menghitung gradien dan mempercepat proses pelatihan dengan mengelompokkan nilai-nilai fitur ke dalam bin-bin yang berbeda (Chen, T., & Guestrin, C., 2016). Selain itu, LightGBM memiliki kemampuan untuk secara langsung menangani fitur-fitur kategori (*categorical features*) tanpa perlu mengonversi mereka menjadi angka. Zhang, T., & Zhou, Y. (2020) menyatakan bahwa keberadaan LightGBM mendukung pelatihan model secara paralel, yang memungkinkan penggunaan sumber daya komputasi yang lebih besar dan mempercepat proses pelatihan, terutama pada dataset besar .

2.3.2 CatBoost

CatBoost merupakan algoritma *machine learning* yang dikembangkan oleh Yandex, yang dirancang khusus untuk menangani data kategorikal dan memiliki performa yang baik dalam tugas-tugas klasifikasi dan regresi (Prokhorenkova, L., dkk., 2018). Dalam konteks *machine learning* tradisional, CatBoost memiliki kelebihan dengan beberapa fitur kunci yang membuatnya efektif dan mudah digunakan. CatBoost secara otomatis menangani data kategorik tanpa memerlukan *encoding* sebelumnya (Maria, C., & Loureiro, J., 2019). CatBoost menggunakan pendekatan pertumbuhan pohon berbasis penyesuaian gradien (*gradient-based boosting*), yang memungkinkan model untuk belajar secara efisien dari kesalahan sebelumnya dan meningkatkan akurasi prediksi (Anand, S., & Kumar, A., 2019), seperti pada **Gambar 2.3.**

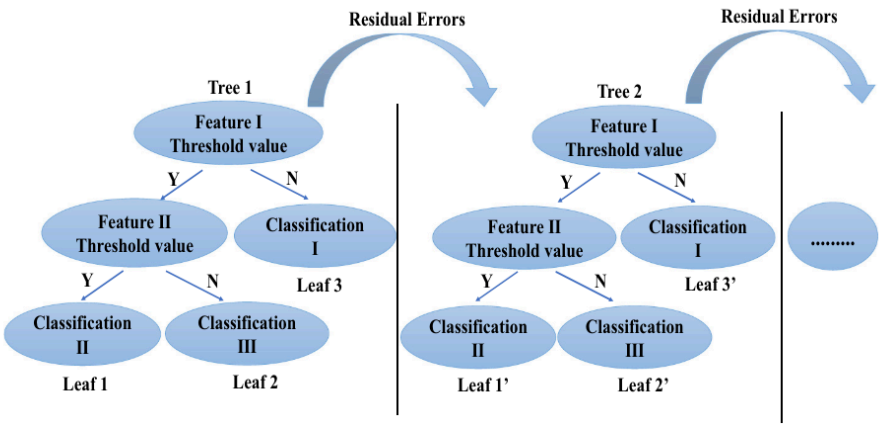


Gambar 2.3 Ilustrasi CatBoost

Dalam mencegah *overfitting*, CatBoost memiliki mekanisme bawaan, seperti penggunaan regularisasi, penurunan kecepatan pembelajaran (*learning rate*), dan penempatan batasan kedalaman pohon (Sadiq, M., dkk, 2020). Menurut Mustafa Sarı & İbrahim Onur Akgül (2021), CatBoost dirancang untuk menangani dataset dengan volume besar dan fitur-fitur yang kompleks. Selain itu, CatBoost menyediakan fitur interpretabilitas yang memudahkan analisis dan pemahaman terhadap bagaimana model membuat prediksi.

2.3.3 *Extreme Gradient Boosting* (XGBoost)

*Extreme Gradient Boostin*g (XGBoost) merupakan salah satu algoritma *ensemble learning* yang sangat populer dalam *machine learning* tradisional. Algoritma ini dikembangkan oleh Tianqi Chen pada tahun 2014 dan sejak itu telah menjadi salah satu pilihan utama dalam berbagai kompetisi *science data* dan aplikasi *machine learning* yang kompleks. XGBoost memadukan kekuatan dari *gradient boosting* dengan optimasi yang sangat efisien, sehingga mampu menghasilkan model prediktif yang sangat kuat dan akurat. Konsep dasar XGBoost adalah membangun serangkaian model prediktif yang sederhana, seperti *decision trees*, dan menggabungkannya untuk membentuk model kompleks yang dapat memprediksi target dengan tingkat akurasi yang tinggi, seperti yang tertera pada **Gambar 2.4.**

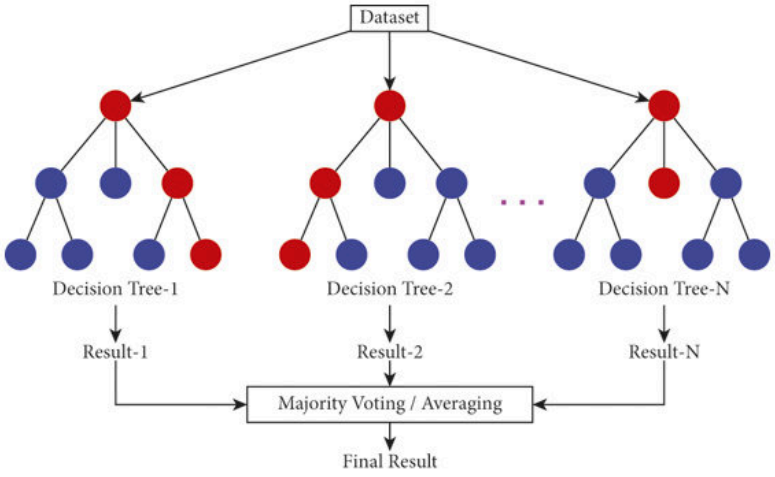


Gambar 2.4 Ilustrasi XGBoost

Ilustrasi di atas menggambarkan *decision trees* yang disusun secara berurutan, dimulai dari *decision tree* pertama hingga *decision tree* terakhir dalam proses pembentukan model XGBoost. Setiap *decision tree* menghasilkan prediksi awal yang mungkin mengandung kesalahan. *Decision tree* berikutnya kemudian memperbaiki kesalahan tersebut dengan fokus pada *instance* yang sebelumnya salah diprediksi. Ilustrasi tersebut juga dapat menggambarkan bagaimana XGBoost mampu menangani kompleksitas data, seperti interaksi variabel, pola yang rumit, dan noise dalam data.

2.3.4 Random Forest

Random forest adalah salah satu algoritma yang populer dalam *machine learning* untuk masalah klasifikasi dan regresi (Y Ao dkk, 2019). Algoritma ini menggunakan konsep *ensemble learning* yang membangun beberapa pohon keputusan (*decision trees*) pada dataset latihan dan menggabungkan hasil prediksi dari setiap pohon untuk menentukan prediksi akhir (Nitze dkk, 2012). Dasar teori di balik *random forest* adalah konsep "pohon keputusan" atau *decision tree*. Pohon keputusan merupakan struktur berhirarki yang digunakan untuk mengambil keputusan berdasarkan serangkaian aturan logis, seperti pada **Gambar 2.5**

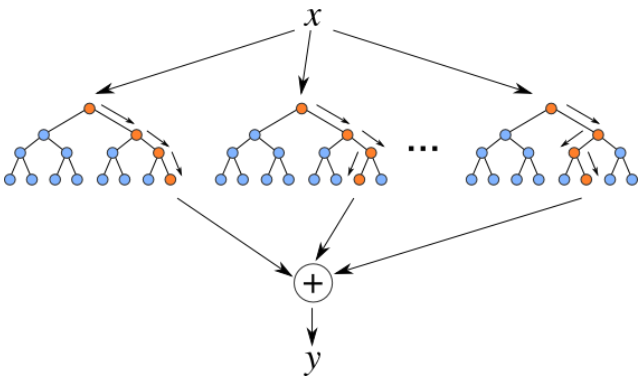


Gambar 2.5 Ilustrasi *Random Forest*

Setiap node dalam pohon keputusan mewakili suatu atribut, dan setiap cabang dari *node* tersebut mewakili nilai yang mungkin dari atribut tersebut. Proses pembelajaran pada pohon keputusan adalah membagi dataset menjadi subset yang lebih kecil berdasarkan nilai-nilai atribut, dengan tujuan untuk menghasilkan subset yang homogen dalam hal label kelas atau nilai target (Breiman, 2001). *Random forest* memperkenalkan konsep *ensemble learning* dengan cara membangun sejumlah besar pohon keputusan secara acak (Ahmad dkk, 2018). Setiap pohon keputusan dibangun menggunakan subset acak dari data pelatihan dan subset acak dari fitur (atribut). Proses ini mengurangi varians model dan meningkatkan performa prediksi, serta mengurangi risiko *overfitting*.

2.3.5 ExtraTrees

ExtraTrees, singkatan dari *Extremely Randomized Trees*, adalah varian dari algoritma *random forest* dalam *machine learnin*g. Seperti *random forest*, ExtraTrees juga menggunakan konsep *ensemble learning* dengan membangun beberapa pohon keputusan pada dataset latihan (Pagliaro, 2023). Namun, yang membedakan ExtraTrees adalah pendekatan yang lebih ekstrim dalam pembentukan pohon keputusan. Pada setiap node pembelajaran, ExtraTrees memilih nilai pemisah secara acak dari subset acak dari fitur yang dipilih secara acak pula, tidak seperti *random forest* yang melakukan pemilihan nilai pemisah terbaik, seperti pada **Gambar 2.6.**



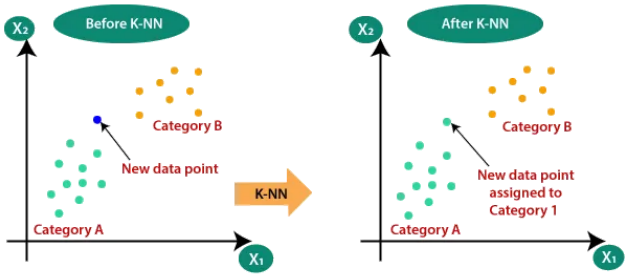
Gambar 2.6 Ilustrasi ExtraTress

Pendekatan ini membuat ExtraTrees lebih cepat dalam pembangunan pohon keputusan, karena tidak perlu mempertimbangkan semua kemungkinan nilai pemisah untuk setiap fitur (Pagliaro, 2023). Meskipun proses pembelajaran lebih cepat, ExtraTrees mampu menghasilkan model yang seringkali memiliki performa yang setara atau bahkan lebih baik daripada *random forest*, terutama untuk dataset yang besar atau dengan fitur-fitur yang banyak. Ketika melakukan prediksi, hasil prediksi dari setiap pohon keputusan diambil, dan prediksi akhir diambil berdasarkan mayoritas hasil dari semua pohon keputusan (dalam kasus klasifikasi) atau rerata hasil (dalam kasus regresi). Dengan cara ini, ExtraTrees menghasilkan model *ensemble* yang kuat dan seringkali lebih tahan terhadap *overfitting* daripada pohon keputusan tunggal (Geurts dkk, 2006).

2.3.6 *K-Nearest Neighbor* (KNN)

K-Nearest Neighbor (KNN) adalah salah satu algoritma klasifikasi dan regresi yang sederhana tetapi efektif dalam *machine learning* (Peterson, 2009). Prinsip dasar dari KNN adalah melakukan prediksi berdasarkan mayoritas kelas (dalam klasifikasi) atau rerata nilai (dalam regresi) dari K tetangga terdekat dari titik data yang akan diprediksi (Hastie dkk, 2009). "K" dalam KNN menunjukkan jumlah tetangga terdekat yang akan dipertimbangkan dalam proses prediksi. Untuk melakukan prediksi, KNN mengukur jarak antara titik data yang akan diprediksi dengan setiap titik data dalam dataset latihan, menggunakan metrik jarak seperti jarak Euclidean atau jarak Manhattan (Peterson, 2009). Kemudian, KNN akan memilih K tetangga terdekat berdasarkan jarak tersebut, dan menentukan kelas mayoritas (dalam klasifikasi)

atau nilai rerata (dalam regresi) dari tetangga-tetangga tersebut sebagai prediksi untuk titik data yang akan diprediksi.



Gambar 2.7 Ilustrasi *K-Nearest Neighbor* (KNN)

Meskipun sederhana, KNN dapat memberikan hasil yang baik untuk masalah klasifikasi dan regresi pada dataset yang relatif kecil atau memiliki struktur yang jelas. Namun, kelemahan utama dari KNN adalah kinerjanya yang lambat ketika diterapkan pada dataset besar, karena memerlukan perhitungan jarak antara titik data. Selain itu, KNN juga sensitif terhadap skala fitur dan memerlukan penyesuaian yang tepat terhadap jumlah tetangga (nilai K) untuk mengoptimalkan kinerjanya (Dhanabal & Chandramathi, 2011).

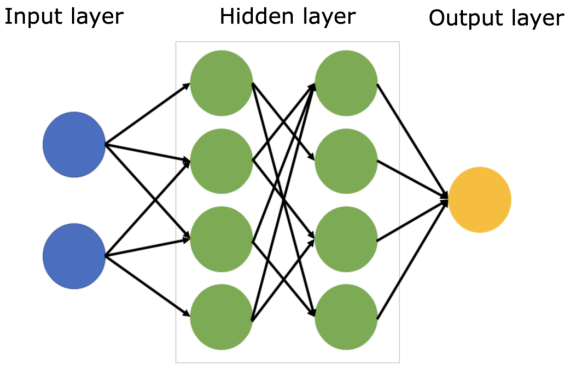
2.4 *Deep Learning*

Deep learning merupakan bagian dari *machine learning* yang menggunakan tingkat representasi untuk mendapatkan fungsi-fungsi kompleks dalam data yang berdimensi tinggi (LeCun et al, 2015). Representasi tersebut didapatkan melalui jaringan-jaringan yang disertai dengan operasi matematika tingkat lanjut, seperti contohnya adalah konvolusi (Janiesch, 2021). Keunggulan *deep learning* berada pada penyelesaian masalah terhadap dimensi data yang tinggi. Algoritma tersebut dapat menganalisis dan belajar dari data yang sangat besar dan tidak terawasi, menjadikannya alat yang berharga untuk analisis *Big Data* (Najafabadi et al, 2015). Hal tersebut karena *deep learning* terdiri dari beberapa lapisan dengan algoritma dan arsitektur yang sangat optimal, yang memungkinkan peningkatan akurasi dan efisiensi waktu pelatihan (Shrestha & Mahmood, 2019).

2.4.1 Jaringan Syaraf Tiruan

Jaringan syaraf tiruan adalah sistem pemrosesan informasi yang terinspirasi oleh sistem saraf dan otak manusia, yang terdiri dari neuron-neuron yang bekerja secara paralel (Kruse et al, 2016). Jaringan saraf bekerja dengan menghubungkan elemen pemrosesan (neuron) yang disusun agar bekerja secara serempak untuk memecahkan masalah tertentu, susunan elemen pemrosesan tersebut disebut arsitektur (Awodele & Jegede, 2009). Arsitektur jaringan umumnya terdiri dari lapisan input, lapisan tersembunyi, dan lapisan output seperti pada **Gambar 2.8**. Jaringan syaraf tiruan merupakan paradigma pemrograman yang berbeda dari pemrograman tradisional. Paradigma ini memungkinkan program belajar untuk observasi data yang nantinya akan mendapatkan solusi tertentu. Jaringan saraf menawarkan keuntungan seperti membutuhkan pelatihan statistik yang lebih sedikit, mendeteksi hubungan non

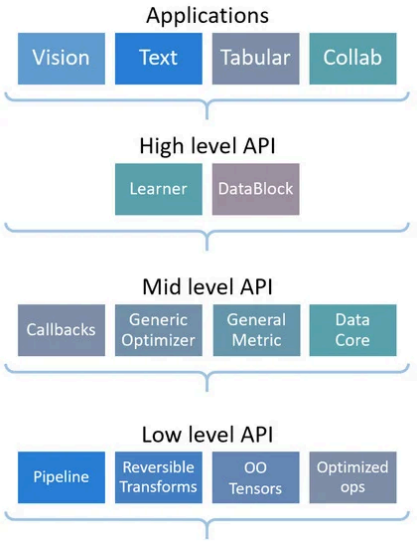
linear yang kompleks, dan mendeteksi semua interaksi yang mungkin terjadi antara variabel prediktor (Tu, 1996).



Gambar 2.8 Ilustrasi Jaringan Syaraf Tiruan

2.4.2 FastAI

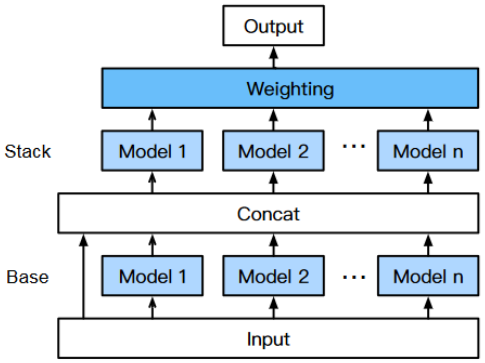
FastAI adalah pustaka *deep learning* yang unik karena menyediakan dua tingkatan komponen yang berbeda. Komponen tingkat tinggi memungkinkan pengguna untuk dengan mudah mengakses teknologi terbaru dalam *deep learning* untuk berbagai domain standar seperti *computer vision* dan *natural language processing*. Sedangkan, komponen tingkat rendah memberikan fleksibilitas kepada pengguna untuk mengembangkan pendekatan baru dalam *deep learning*. Dengan menggunakan arsitektur berlapis, FastAI menyediakan keseimbangan yang baik antara kemudahan penggunaan, fleksibilitas, dan kinerja (Howard & Gugger, 2020). *Application Programming Interface* (API) merupakan sekumpulan protokol yang dapat yang memungkinkan antar perangkat lunak saling berkomunikasi (Lomborg et al, 2014). Dalam FastAI, API disediakan dengan abstraksi tingkat tinggi sehingga mudah dipahami oleh pengguna.



Gambar 2.9 *Layered API* pada FastAI

2.5 *Weighted Ensemble* Model

Weighted Ensemble memungkinkan pembobotan individual dari masing-masing model, yang dapat meningkatkan kinerja pada basis data statistik standar (Granitto et al, 2005) . Ide utama dari model *Weighted Ensemble* adalah mengurangi *loss* dengan melakukan agregasi pada hasil prediksi dari beberapa model dasar yang berbeda, **Gambar 2.10** (Li et al, 2016). Hasil prediksi atau *loss* yang didapat oleh masing-masing model kemudian dilakukan pembobotan dengan harapan masing-masing model dapat saling menanggung *loss* antara model satu dengan yang lain sehingga dapat menghasilkan model yang mereduksi *loss* tersebut (Zhang & Zhou, 2011).



Gambar 2.10. Strategi *Weighted Ensemble*

Dengan memberikan bobot pada setiap model klasifikasi, algoritma ini dapat mengatasi ketidakseimbangan dalam kualitas atau keandalan dari masing-masing model. Hal ini memungkinkan *ensemble* untuk lebih baik menyesuaikan diri dengan data yang kompleks atau tidak seimbang, yang sering kali sulit untuk ditangani oleh model tunggal. Dengan demikian, *Weighted Ensemble* dapat memberikan kinerja yang lebih baik daripada model tunggal atau *ensemble* tanpa pembobotan model secara individu, terutama pada dataset yang besar atau kompleks (Mao et al, 2021).

2.6 *Large Language Model* (LLM)

Large Language Model (LLM) adalah *Language Model* yang besar dan canggih. LLM memiliki skala parameter yang sangat besar, sering kali mencapai jutaan hingga miliaran, yang memungkinkan model untuk menyimpan dan memahami informasi yang kompleks (Dehghani, 2023). Model ini dilatih pada kumpulan data teks yang sangat besar dalam proses yang disebut pre-training, sehingga mereka dapat memahami berbagai pola dan struktur dalam bahasa (Zhao, 2023). LLM dapat disesuaikan lebih lanjut untuk tugas-tugas spesifik melalui proses *fine-tuning,* yang melibatkan pelatihan tambahan pada dataset yang lebih kecil dan khusus (Kuang, 2023). Kemampuan generatif yang kuat memungkinkannya untuk menghasilkan teks yang koheren dan kontekstual berdasarkan input yang diberikan, sehingga sangat berguna dalam berbagai aplikasi seperti penerjemahan bahasa, pembuatan konten otomatis, chatbots, dan analisis sentimen.

Model ini adalah alat yang sangat bermanfaat dalam pemrosesan bahasa alami yang mempelajari representasi universal dan dapat meningkatkan sistem rekomendasi

dengan menggunakan teknik *fine-tuning* dan *prompt tuning* (Wu et al, 2023). Hal ini dapat dimanfaatkan dalam mendapatkan rekomendasi terkait hasil dari data input. Dalam konteks sistem rekomendasi, LLM dapat digunakan untuk memproses input dari berbagai sumber data, seperti ulasan pengguna, riwayat pencarian, dan interaksi sebelumnya dengan sistem. Teknik *fine-tuning* memungkinkan LLM untuk disesuaikan secara spesifik dengan dataset tertentu, memperbaiki akurasi dan relevansi rekomendasi yang dihasilkan. Misalnya, LLM yang dilatih dengan data ulasan produk dapat mengidentifikasi preferensi dan kebutuhan individu berdasarkan teks yang mereka tulis, dan kemudian menghasilkan rekomendasi produk yang lebih sesuai dengan keinginan mereka. *Prompt tuning*, di sisi lain, memungkinkan model untuk memberikan rekomendasi berdasarkan prompt atau perintah spesifik yang diberikan oleh pengguna (Oymak et al, 2023). Misalnya, jika seorang pengguna meminta rekomendasi buku tentang topik tertentu, LLM dapat menghasilkan daftar rekomendasi yang relevan dengan topik tersebut berdasarkan analisis teks dan pola yang telah dipelajarinya

2.7 *Deployment*

Deployment merupakan proses menerapkan model *machine learning* dari tahap pengembangan menuju lingkungan produksi, yang mana model dapat diinputkan data baru (Paleyes et alk, 2020). Proses ini memastikan bahwa model dapat diakses dan dimanfaatkan dalam aplikasi nyata untuk memberikan nilai operasional. *Deployment* model *machine learning* melibatkan mengemas model yang telah dilatih dalam format yang sesuai, menyiapkan infrastruktur seperti *server* atau *cloud*, membuat API menggunakan *framework* tertentu, mengintegrasikan API dengan aplikasi, memantau kinerja model, melakukan maintenance jika diperlukan, dan memastikan keamanan dan kepatuhan terhadap regulasi privasi data (Klaise et al, 2020). Deployment bertujuan agar model dapat digunakan secara efektif dalam aplikasi nyata, memberikan prediksi yang akurat, dan memenuhi kebutuhan operasional serta keamanan yang diinginkan (Ackermann, 2018).

BAB III

METODE PELAKSANAAN

3.1 *Software*

Dalam laporan penelitian ini, bahasa pemrograman Python menjadi alat utama untuk menganalisis dan memproses kumpulan data yang dikumpulkan di industri ritel. Berbagai pustaka dan alat analisis data yang tersedia di ekosistem Python memungkinkan peneliti melakukan berbagai tugas analisis seperti pemrosesan data, eksplorasi statistik, visualisasi, dan pembuatan model prediktif. Python juga memungkinkan integrasi tanpa batas dengan platform penyimpanan data dan infrastruktur *cloud computing*, memungkinkan peneliti mengelola dan menganalisis kumpulan data dengan skalabilitas yang mereka perlukan. Dengan memanfaatkan manfaat analisis data dari Python, penelitian ini memberikan wawasan mendalam tentang perilaku pelanggan, memungkinkan ritel untuk lebih efektif mengoptimalkan strategi pemasaran, periklanan, dan layanan pelanggan.

Dalam penelitian ini, penggunaan Python menjadi penting karena keunggulan yang ditawarkannya di bidang *artificial intelligence* dan *machine learning*. Python sangat populer di kalangan analisis data karena berbagai alasan, termasuk berbagai *package* canggih seperti PyTorch, Scikit-learn, NumPy, dan Pandas yang menyediakan algoritma canggih untuk analisis data dan pembelajaran mesin. Selain itu, Python menyediakan sintaksis yang mudah dipahami dan fleksibilitas yang memungkinkan peneliti mengembangkan dan menguji model prediktif yang kompleks dengan cepat. Dengan memanfaatkan manfaat kecerdasan buatan dan pembelajaran mesin Python, penelitian ini dapat mengeksplorasi pola tersembunyi dalam catatan pelanggan dan memberikan wawasan berharga bagi ritel.

3.2 *Dataset*

Dataset yang digunakan dalam penelitian ini mencakup berbagai karakteristik yang terkait dengan demografi pelanggan ritel dan perilaku pembelian. Karakteristik tersebut meliputi tahun lahir pelanggan, tingkat pendidikan, status pernikahan, pendapatan, jumlah anak balita dan remaja, serta pembelian, termasuk pengeluaran untuk berbagai jenis produk seperti buah, daging, ikan, dan kue sejarah. Selain itu, dataset ini juga mengumpulkan informasi tentang perilaku pembelian, seperti jumlah pembelian diskon, jumlah pembelian melalui web dan pembelian di toko, dan apakah pelanggan pernah menyampaikan keluhan. Terdapat juga informasi mengenai tanggal pertama kali pelanggan menjadi anggota toko dan jumlah promosi yang diterima dari toko dari total 6 promosi yang dijalankan.

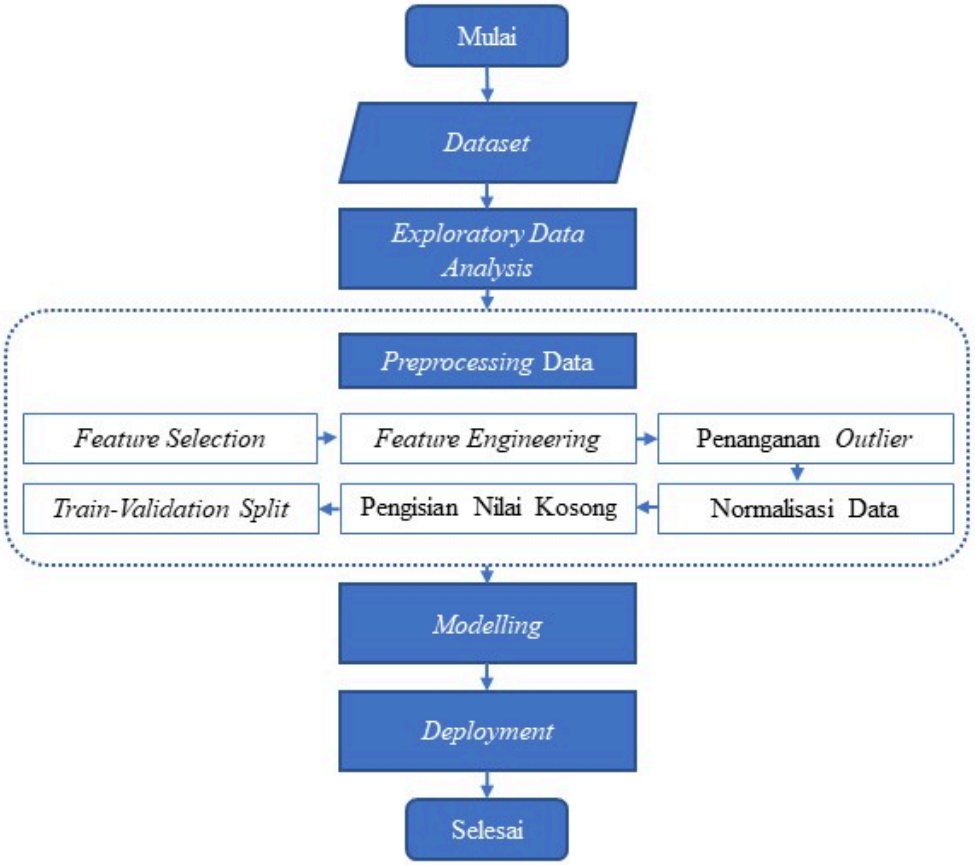
Dengan fitur-fitur tersebut, *dataset* memberikan gambaran komprehensif tentang perilaku dan preferensi pelanggan, yang dapat digunakan untuk menganalisis tren belanja, mengembangkan strategi pemasaran yang lebih efektif, dan meningkatkan loyalitas pelanggan. Selain itu, *dataset* telah dibagi menjadi dua bagian yaitu data latih sebanyak 3817 data dan data tes sebanyak 3818 data. Dalam hal ini, data latih akan digunakan untuk melatih model, sedangkan data tes digunakan untuk submisi jawaban pada babak penyisihan lalu. **Tabel 3.1.** memuat fitur-fitur dari *dataset* beserta jenis data dan penjelasannya.

Tabel 3.1. Deskripsi Dataset

Fitur	Jenis Data	Penjelasan
tahun_kelahiran	Numerik	Tahun kelahiran pelanggan
pendidikan	Kategorikal	Tingkat pendidikan pelanggan (SMP, SMA, Sarjana, Magister, dan Doktor)
status_pernikahan	Kategorikal	Status pernikahan pelanggan (Sendiri, Rencana Menikah, Menikah, Cerai, dan Cerai Mati)
pendapatan	Numerik	Pendapatan pelanggan dalam rupiah
jumlah_anak_balita	Numerik	Banyaknya anak pelanggan yang masih balita
jumlah_anak_remaja	Numerik	Banyaknya anak pelanggan yang sudah remaja
terakhir_belanja	Numerik	Jumlah hari berlalu setelah terakhir belanja
belanja_buah	Numerik	Biaya yang dikeluarkan untuk belanja buah
belanja_daging	Numerik	Biaya yang dikeluarkan untuk belanja daging
belanja_ikan	Numerik	Biaya yang dikeluarkan untuk belanja ikan
belanja_kue	Numerik	Biaya yang dikeluarkan untuk belanja kue
pembelian_diskon	Numerik	Banyaknya pembelian yang dilakukan saat diskon
pembelian_web	Numerik	Banyaknya pembelian yang dilakukan secara daring
pembelian_toko	Numerik	Banyaknya pembelian yang dilakukan secara luring
keluhan	Numerik	1 - pernah memberikan keluhan, 0 - tidak pernah
tanggal_menjadi_anggota	Tanggal	Tanggal pertama kali terdaftar sebagai anggota
Jumlah_promosi	Numerik	Kolom target, yaitu pada promosi ke berapa pelanggan menerima program dari toko, dari total 6 kali promosi yang dilakukan. Nilai 0 berarti tidak menerima promosi sama sekali

3.3 Alur Penelitian

Secara keseluruhan, penelitian akan dilakukan dengan mempersiapkan *dataset*, melakukan *Exploratory Data Analysis*, *preprocessing* data, *modelling*, dan *deployment*. Alur penelitian secara lengkap dapat dilihat pada **Gambar 3.1.**



Gambar 3.1 Alur penelitian

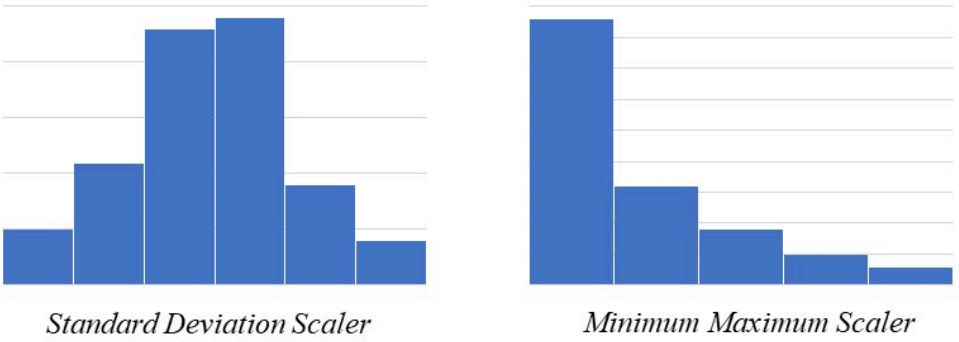
3.4 *Exploratory Data Analysis* (EDA)

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

3.5 *Preprocessing* Data

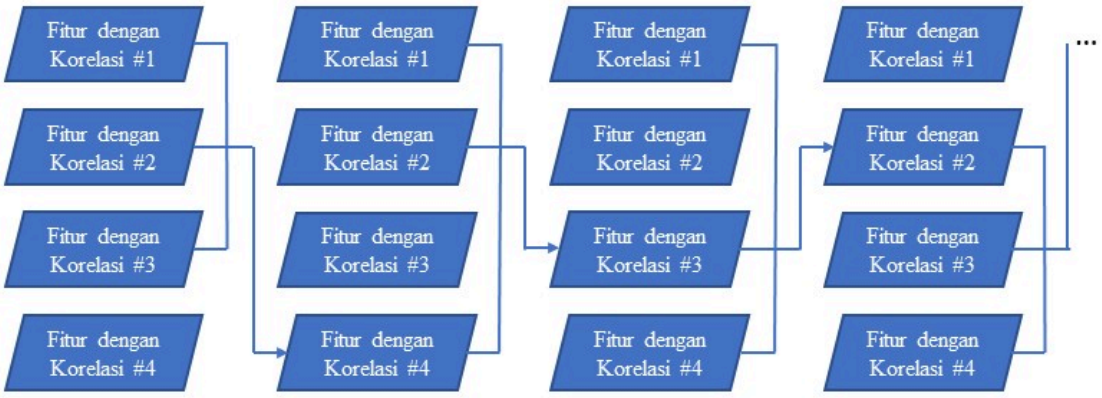
Alur *preprocessing* data yang dilakukan adalah *feature selection*, *feature engineering*, normalisasi data, pengisian nilai kosong, dan *Train-Validation Split*. *Feature selection* dilakukan dengan menghapus kolom yang paling tidak berkontribusi pada model, dapat ditunjukkan dengan fitur berkorelasi rendah, maupun fitur dengan banyak nilai kosong. *Feature engineering* juga dilakukan dengan mengganti nilai pada fitur agar semakin mudah dimengerti model, serta mengubah fitur dengan jenis data kategorik menjadi numerik. Penanganan *outlier* dan normalisasi data akan dilakukan sesuai dengan bentuk distribusi. Fitur yang memiliki distribusi menyerupai distribusi normal akan dinormalisasi dengan metode *Standardization Scaler*, sedangkan fitur

dengan distribusi yang tidak menyerupai distribusi normal akan dinormalisasi dengan metode *Min-Max Scaler* yang dapat dilihat pada **Gambar 3.2.**



Gambar 3.2 Normalisasi berdasarkan distribusi

Preprocessing data yang dilakukan berikutnya adalah pengisian nilai kosong. Pengisian nilai kosong menggunakan prediksi model yang dilatih dengan fitur selain fitur yang akan diisi, dimana nilai kosong pada fitur-fitur latih tersebut akan diisi dengan rata-rata fitur untuk sementara. Fitur yang diisi adalah fitur dengan korelasi paling rendah, hingga korelasi paling tinggi, sehingga nilai kosong pada fitur penting akan diisi dengan model yang lebih baik. Proses pengisian nilai kosong ini ditunjukkan pada **Gambar 3.3.** Terakhir, akan dilakukan *Train-Validation Split* yang akan membagi *dataset* menjadi dua bagian yaitu 80% data latih dan 20% data validasi.



Gambar 3.3 Proses pengisian nilai kosong, tanda panah menunjukkan proses pelatihan model dengan *input* fitur-fitur untuk mengisi fitur tujuan pada *output*

3.6 *Modelling*

Strategi *modelling* yang dipakai adalah *Weighted Ensemble*, dimana beberapa model *machine learning* tradisional dan *deep learning* akan dilatih dalam tumpukkan (*stack*) pertama, lalu akan dilatih model *Weighted Ensemble L2* dari semua model di tumpukkan pertama. Selanjutnya, output dari model-model di tumpukkan pertama akan menjadi input dari model-model di tumpukkan kedua (*stacking*). Setelah itu akan dilatih model *Weighted Ensemble L3* dari semua model di tumpukkan kedua. Semua model yang dilatih (termasuk model-model pada tumpukkan) akan diberi peringkat dan disusun dalam papan peringkat sesuai dengan skor validasinya.

3.7 *Deployment*

Deployment dilakukan menggunakan Python dengan *package* Flask yang berfungsi sebagai *web framework*. Flask dapat digunakan untuk membuat *website* dengan mengimplementasikan model yang dihasilkan. Layanan yang digunakan untuk *deployment* adalah Google Cloud Run, layanan ini menyediakan *server* untuk *website*, sehingga *website* dapat diakses dengan Uniform Resource Locator (URL) tertentu. *Large Language Model* akan dipakai sebagai fitur penambah deskripsi setelah melakukan prediksi pada model.

Gambar 3.4 Alur *deployment*

BAB IV

ANALISIS

4.1 Hasil *Preprocessing* Data

Setelah melalui *preprocessing* data yaitu *feature selection*, penanganan *outlier*, *feature engineering*, normalisasi data, pengisian nilai kosong, dan pemisahan data latih serta data validasi. *Preprocessing* data pertama yang dilakukan adalah *feature selection*, dimana fitur dengan nilai kosong terbanyak akan dihapus. **Tabel 4.1.** menunjukkan setiap fitur yang diurutkan berdasarkan banyak data yang kosong, fitur yang dihapus ditunjukkan dengan warna merah.

Tabel 4.1. Banyak data kosong terurut setiap fitur

Fitur	Banyak Nilai Kosong	Persentase Nilai Kosong
tahun_kelahiran	0	0.000%
pembelian_web	345	0.045%
pembelian_toko	355	0.046%
pembelian_diskon	357	0.047%
belanja_buah	367	0.048%
terakhir_belanja	373	0.049%
belanja_daging	373	0.049%
belanja_ikan	389	0.051%
keluhan	389	0.051%
pendidikan	393	0.051%
pendapatan	393	0.051%
status_pernikahan	394	0.052%
jumlah_anak_balita	399	0.052%
belanja_kue	401	0.052%
jumlah_anak_remaja	414	0.054%
tanggal_menjadi_anggota	5511	0.722%

Tabel 4.2. menunjukkan hasil *preprocessing* dari *dataset* mentah untuk *feature engineering* dengan Label Encoder untuk data kategorik. Data yang salah ditunjukkan dengan warna merah pada tabel, data yang salah tersebut akan diubah menjadi label dengan modus fitur. Selain itu, fitur tahun kelahiran akan diubah menjadi umur.

Tabel 4.2. Hasil *feature engineering*

tahun_kelahiran	pendidikan	status_pernikahan
1892	5	5
1899	SMP	Sendiri
1900	SMA	Rencana Menikah
1902	Sarjana	Menikah

→

umur	pendidikan	status_pernikahan
132	2	1
125	0	0
124	1	1
122	2	2

tahun_kelahiran	pendidikan	status_pernikahan
1892	5	5
1899	SMP	Sendiri
1900	SMA	Rencana Menikah
…	Magister	Cerai
2009	Doktor	Cerai Mati

→

umur	pendidikan	status_pernikahan
132	2	1
125	0	0
124	1	1
…	3	3
15	4	4

Semua fitur telah menjadi jenis data numerik, sehingga dapat dilakukan penanganan *outlier* untuk menangani nilai yang tidak wajar. Fitur yang ditangani adalah fitur umur dengan batas atas *outlier* 100 tahun dan pendapatan dengan batas atas 3.329×10^8.

Tabel 4.3. Hasil penanganan *outlier*

	Umur	Pendapatan
Mean	56.193	1.154×10^8
Std	11.788	4.741×10^7
Min	15	2.999×10^6
Q1	48	8.143×10^7
Q2	56	1.172×10^8
Q3	65	1.506×10^8
Max	132	1.306×10^9

→

	Umur	Pendapatan
Mean	56.152	1.151×10^8
Std	11.600	4.330×10^7
Min	15	2.999×10^6
Q1	48	8.143×10^7
Q2	56	1.172×10^8
Q3	65	1.506×10^8
Max	100	3.329×10^8

Berikutnya dilakukan normalisasi data, sehingga setiap fitur memiliki peran yang sama besar dengan fitur lain, jenis normalisasi ditentukan dari distribusi setiap fitur yang dapat dilihat pada **Tabel 4.4.**

Tabel 4.4. Normalisasi Data pada Fitur

Fitur	Bentuk Distribusi	Jenis Normalisasi
umur	Normal	Standardization
pendidikan	Normal	Standardization
status_pernikahan	Normal	Standardization
pendapatan	Normal	Standardization
jumlah_anak_balita	Eksponensial	Min-Max Scaler
jumlah_anak_remaja	Eksponensial	Min-Max Scaler
terakhir_belanja	Normal	Standardization
belanja_buah	Eksponensial	Min-Max Scaler
belanja_daging	Eksponensial	Min-Max Scaler
belanja_ikan	Eksponensial	Min-Max Scaler
belanja_kue	Eksponensial	Min-Max Scaler
pembelian_diskon	Eksponensial	Min-Max Scaler
pembelian_web	Normal	Standardization
pembelian_toko	Normal	Standardization
keluhan	-	Min-Max Scaler

Setelah data dinormalisasi, selanjutnya setiap nilai kosong pada fitur akan diisi dengan model terbaik dari proses penumpukkan model-model *machine learning* tradisional dan *deep learning* menjadi model *Weighted Ensemble* yang ditunjukkan pada **Tabel 4.5**.

Tabel 4.5. Model pengisian nilai kosong

Fitur	Model	Metriks	Skor Validasi
keluhan	WeightedEnsemble_L2	Akurasi	99.72%
pembelian_diskon	WeightedEnsemble_L2	RMSE	0.0551
status_pernikahan	WeightedEnsemble_L2	Akurasi	83.86%
jumlah_anak_balita	WeightedEnsemble_L2	RMSE	0.1429
pembelian_toko	WeightedEnsemble_L2	RMSE	0.4872
jumlah_anak_remaja	WeightedEnsemble_L2	RMSE	0.1326
pendidikan	WeightedEnsemble_L2	Akurasi	86.25%
terakhir_belanja	WeightedEnsemble_L2	RMSE	0.6875
pembelian_web	WeightedEnsemble_L2	RMSE	0.5186
belanja_ikan	WeightedEnsemble_L2	RMSE	0.0774
belanja_kue	WeightedEnsemble_L2	RMSE	0.0548
belanja_buah	WeightedEnsemble_L2	RMSE	0.0702
belanja_daging	WeightedEnsemble_L2	RMSE	0.0461
pendapatan	WeightedEnsemble_L2	RMSE	0.2922

Setelah melalui *feature selection*, pengisian nilai kosong, dan normalisasi data. *Dataset* akan dibagi menjadi 80% data latih dan 20% data validasi. Setelah itu, model siap dilatih menggunakan data latih, sekaligus menghitung skor data validasi.

4.2 Hasil Analisis

Kolom target yaitu jumlah promosi yang dibutuhkan agar pelanggan menerima promosi akan diprediksi menggunakan model *Weighted Ensemble*. Semua model yang dipakai untuk membentuk model *Weighted Ensemble* dilatih baik semua model pada tumpukkan pertama maupun pada tumpukkan kedua, sehingga menghasilkan papan peringkat model pada **Tabel 4.6.** yang diurutkan berdasarkan skor validasi.

Tabel 4.6. Papan Peringkat Model

Model	Skor Validasi (F1 Macro)	Waktu Pelatihan Total (detik)	Waktu Pelatihan Model (detik)
WeightedEnsemble_L3	**0.818968**	7559.727	3.492
XGBoost_L2	0.818968	7556.235	401.532
FastAI_L2	0.815941	7288.649	133.947
CatBoost_L2	0.811777	8661.878	1507.176
RandomForest_L2	0.810208	7172.099	17.397
…	…	…	…
CatBoost_L1	0.762020	1244.550	1244.550
XGBoost_L1	0.757990	162.296	162.296

FastAI_L1	0.697313	87.969	87.969
NeuralNetwork_L1	0.693847	317.921	317.921
KNN_L1	0.557770	**0.013**	**0.013**

Model terbaik yang didapat adalah Weighted Ensemble L3 dengan skor validasi F1 Macro sebesar 81.8968%. Model dengan waktu pelatihan tercepat didapatkan oleh K-Nearest Neighbor, namun dengan akurasi yang rendah yaitu hanya 55.7770% saja. Oleh karena itu, dipilih model Weighted Ensemble L3 sebagai model utama yang dapat digunakan untuk *deployment*.

4.4 Hasil *Deployment*

Model terbaik dari hasil analisis dipilih untuk diimplementasikan sebagai sebuah *website*, yang dapat dilihat pada **Gambar 4.1.**

Gambar 4.1 Tampilan *deployment website*
https://aigeniuses.jemmyfebryan.site

Dengan implementasi *website* tersebut, para pelaku industri atau pemerintah dapat mengintegrasikan sistem ini dalam pelayanan ritel industri, maupun untuk menganalisis perilaku masyarakat Indonesia yang berbelanja di toko-toko ritel.

BAB V

KESIMPULAN

5.1 Kesimpulan

Transformasi teknologi digital telah mengubah perekonomian global secara signifikan, terutama dalam industri ritel yang kini menghadapi persaingan ketat dan pergeseran perilaku konsumen. Penggunaan teknologi seperti komputasi awan dan kecerdasan buatan, khususnya *machine learning*, telah menjadi alat vital dalam meningkatkan efisiensi operasional dan personalisasi layanan pelanggan. Melalui pengumpulan dan analisis data yang komprehensif, industri ritel dapat memahami kebutuhan dan preferensi pelanggan dengan lebih baik, memungkinkan mereka untuk merancang strategi pemasaran yang lebih efektif dan tepat sasaran. Hal ini tidak hanya meningkatkan kepuasan pelanggan tetapi juga mendorong retensi dan peningkatan keuntungan. Penelitian ini menunjukkan bahwa penggunaan bahasa pemrograman Python untuk memproses data demografi dan perilaku pembelian pelanggan dapat menghasilkan model prediksi yang akurat. Model terbaik yang didapat dalam penelitian ini adalah Weighted Ensemble L3 dengan skor validasi F1 Macro sebesar 81.8968%. Model prediktif yang dihasilkan memungkinkan identifikasi pola pembelian dan respons pelanggan terhadap promosi, yang membantu dalam pengambilan keputusan yang lebih cerdas dan strategis. Inovasi ini, termasuk pengembangan *website* yang terintegrasi dengan teknologi *machine learning*, diharapkan memberikan nilai tambah yang signifikan bagi industri ritel dalam merencanakan serta melaksanakan strategi pemasaran dan promosi yang lebih efektif dan efisien.

5.2 Saran

Industri ritel harus terus berinvestasi dalam pengembangan dan pembaharuan teknologi serta meningkatkan keterampilan tenaga kerja mereka, untuk mencapai manfaat maksimal dari teknologi digital dan *machine learning*. Selain itu, penting untuk melakukan pelatihan berkelanjutan bagi karyawan agar mereka dapat memanfaatkan teknologi ini secara efektif. Perusahaan juga perlu mengadopsi pendekatan yang lebih *data-driven* dalam pengambilan keputusan, dengan memperkuat sistem pengumpulan dan analisis data pelanggan. Kolaborasi dengan pihak akademis dan penyedia teknologi juga bisa dilakukan untuk memastikan bahwa inovasi yang diterapkan selalu berada di garis depan perkembangan industri. Dengan demikian, fleksibilitas dan adaptabilitas dalam strategi pemasaran dan promosi akan membantu perusahaan tetap kompetitif dalam menghadapi perubahan perilaku konsumen dan dinamika pasar.

DAFTAR PUSTAKA

Ackermann, K., Walsh, J., Unánue, A., Naveed, H., Rivera, A., Lee, S., Bennett, J., Defoe, M., Cody, C., Haynes, L., & Ghani, R. (2018). Deploying Machine Learning Models for Public Policy: A Framework. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. https://doi.org/10.1145/3219819.3219911.

Awodele, O., & Jegede, O. (2009). Neural Networks and Its Application in Engineering. . https://doi.org/10.28945/3317.

Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

Dehghani, M., Djolonga, J., Mustafa, B., Padlewski, P., Heek, J., Gilmer, J., Steiner, A., Caron, M., Geirhos, R., Alabdulmohsin, I., Jenatton, R., Beyer, L., Tschannen, M., Arnab, A., Wang, X., Riquelme, C., Minderer, M., Puigcerver, J., Evci, U., Kumar, M., Steenkiste, S., Elsayed, G., Mahendran, A., Yu, F., Oliver, A., Huot, F., Bastings, J., Collier, M., Gritsenko, A., Birodkar, V., Vasconcelos, C., Tay, Y., Mensink, T., Kolesnikov, A., Paveti'c, F., Tran, D., Kipf, T., Luvci'c, M., Zhai, X., Keysers, D., Harmsen, J., & Houlsby, N. (2023). Scaling Vision Transformers to 22 Billion Parameters. , 7480-7512. https://doi.org/10.48550/arXiv.2302.05442.

Fader, P. S., & Hardie, B. G. (2018). How to project customer churn and loyalty with machine learning. Journal of Interactive Marketing, 43, 45-57.

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. Machine Learning, 63(1), 3-42.

Granitto, P., Verdes, P., & Ceccatto, H. (2005). Neural network ensembles: evaluation of aggregation algorithms. ArXiv, abs/cs/0502006. https://doi.org/10.1016/j.artint.2004.09.006.

Gupta, S., Lehmann, D. R., & Stuart, J. A. (2004). Valuing customers. Journal of marketing research, 41(1), 7-18.

Han, S. L., & Yoon, S. J. (2015). Customer behavior analysis for personalized services. Journal of Software, 10(8), 1005-1011.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction. Springer Science & Business Media.

Howard, J., & Gugger, S. (2020). fastai: A Layered API for Deep Learning. Inf., 11, 108. https://doi.org/10.3390/info11020108.

Janiesch, C., Zschech, P. & Heinrich, K. Machine learning and deep learning. Electron Markets 31, 685–695 (2021). https://doi.org/10.1007/s12525-021-00475-2

Klaise, J., Looveren, A., Cox, C., Vacanti, G., & Coca, A. (2020). Monitoring and explainability of models in production. ArXiv, abs/2007.06299.

Kotler, P., & Armstrong, G. (2017). Principles of Marketing (17th Global Edition). Pearson Education Limited.

Kruse, R., Borgelt, C., Braune, C., Mostaghim, S., & Steinbrecher, M. (2016). Introduction to Neural Networks. , 9-13. https://doi.org/10.1007/978-1-4471-7296-3_2.

Kuang, W., Qian, B., Li, Z., Chen, D., Gao, D., Pan, X., Xie, Y., Li, Y., Ding, B., & Zhou, J. (2023). FederatedScope-LLM: A Comprehensive Package for Fine-tuning Large Language Models in Federated Learning. ArXiv, abs/2309.00363.

Kumar, V., & Reinartz, W. (2016). Customer relationship management: Concept, strategy, and tools. Springer.

LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. Nature 521, 436–444 (2015). https://doi.org/10.1038/nature14539

Li, L., Hu, Q., Wu, X., & Yu, D. (2014). Exploration of classification confidence in ensemble learning. *Pattern Recognit.*, 47, 3120-3131. https://doi.org/10.1016/j.patcog.2014.03.021.

Lomborg, S., & Bechmann, A. (2014). Using APIs for Data Collection on Social Media. The Information Society, 30, 256 - 265. https://doi.org/10.1080/01972243.2014.915276.

Mao, S., Lin, W., Jiao, L., Gou, S., & Chen, J. (2021). End-to-End Ensemble Learning by Exploiting the Correlation Between Individuals and Weights. IEEE Transactions on Cybernetics, 51, 2835-2846. https://doi.org/10.1109/TCYB.2019.2931071.

Najafabadi, M., Villanustre, F., Khoshgoftaar, T., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. Journal of Big Data, 2. https://doi.org/10.1186/s40537-014-0007-7.

Oymak, S., Rawat, A., Soltanolkotabi, M., & Thrampoulidis, C. (2023). On the Role of Attention in Prompt-tuning. , 26724-26768. https://doi.org/10.48550/arXiv.2306.03435.

Paleyes, A., Urma, R., & Lawrence, N. (2020). Challenges in Deploying Machine Learning: A Survey of Case Studies. ACM Computing Surveys, 55, 1 - 29. https://doi.org/10.1145/3533378.

Shrestha, A., & Mahmood, A. (2019). Review of Deep Learning Algorithms and Architectures. IEEE Access, 7, 53040-53065. https://doi.org/10.1109/ACCESS.2019.2912200.

Tu, J., & Tu, J. (1996). Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes.. Journal of clinical epidemiology, 49 11, 1225-31. https://doi.org/10.1016/S0895-4356(96)00002-9.

Wang, Y., Li, X., & Li, H. (2010). CRM based on data mining: A case study of the bank marketing. In 2010 International Conference on Computer and Information Application (ICCIA 2010) (Vol. 2, pp. V2-73). IEEE.

Wu, L., Zheng, Z., Qiu, Z., Wang, H., Gu, H., Shen, T., Qin, C., Zhu, C., Zhu, H., Liu, Q., Xiong, H., & Chen, E. (2023). A Survey on Large Language Models for Recommendation. ArXiv, abs/2305.19860. https://doi.org/10.48550/arXiv.2305.19860.

Zhang, L., & Zhou, W. (2011). Sparse ensembles using weighted combination methods based on linear programming. *Pattern Recognit.*, 44, 97-106. https://doi.org/10.1016/j.patcog.2010.07.021.

Zhao, W., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J., & Wen, J. (2023). A Survey of Large Language Models. ArXiv, abs/2303.18223.

https://doi.org/10.1016/j.autcon.2020.103155

https://doi.org/10.26740/jupe.v8n3.p86-94

https://doi.org/10.1016/j.jretai.2020.10.006

https://journal.ikopin.ac.id/index.php/coopetition/article/view/3076/2559

https://opac.atmaluhur.ac.id/uploaded_files/temporary/DigitalCollection/ODljY2E4ODIyODViZjFkODgzNDUxYWZlNWFhZmY2MGE5MDc0ZDVmYw==.pdf

(PDF) Ekonomi Digital: Transformasi Bisnis di Era Digital (researchgate.net)

https://doi.org/10.1016/j.ijresmar.2018.12.002

https://repository.upy.ac.id/4945/1/FullBook%20Implementasi%20Artificial%20Intelligence%20%28AI%29%20dalam%20Kehidupan.pdf

https://www.mdpi.com/2071-1050/9/11/2008

10.1109/ACCESS.2018.2841987

Machine Learning Algorithms - A Review (researchgate.net)

Machine learning and deep learning | Electronic Markets (springer.com)

https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=ea1c116ff0700b9250ef01315a94366ce8bf753c

https://www.researchgate.net/publication/353338909_Machine_Learning_Teori_Studi_Kasus_dan_Implementasi_Menggunakan_Python

https://ejurnal.politeknikpratama.ac.id/index.php/jupiman/article/view/2739/2591

https://doi.org/10.1109/R10-HTC57504.2023.10461930

https://www.sciencedirect.com/science/article/pii/S0167865520302981?casa_token=IKUgpUnMEvYAAAAA:OAIptVfEjHon8sxkL_5IGy8ONWwqInpEt76aD6tapQW-ptNv9ngcV8UfMIkkmm0vpyrjEzYu7Q

https://www.comp.hkbu.edu.hk/~cib/2018/Aug/iib_vol19no1.pdf

https://doi.org/10.1109/IEMECONX.2019.8877011

https://doi.org/10.1109/ICTACS56270.2022.9988247

https://link.springer.com/chapter/10.1007/978-3-030-03493-1_32

https://doi.org/10.1016/j.dss.2012.01.016

https://doi.org/10.1016/j.asoc.2023.111193

https://doi.org/10.48550/arXiv.2207.06084

https://doi.org/10.1016/j.compeleceng.2013.11.024

https://books.google.co.id/books?hl=en&lr=&id=X5ySEAAAQBAJ&oi=fnd&pg=PT10&dq=Aur%C3%A9lien+G%C3%A9ron+covers+imputation+techniques+such+as+filling+missing+values+with+mean,+median,+mode,+or+using+more+advanced+imputation+techniques+based+on+models&ots=yC_vve-5xN&sig=e08GbLp-TOk0Iltlg5X82ZtY26c&redir_esc=y#v=onepage&q&f=false

https://books.google.co.id/books?hl=en&lr=&id=1-4lDQAAQBAJ&oi=fnd&pg=PP1&dq=Andreas+C.+M%C3%BCller+%26+Sarah+Guido&ots=29hRJIQKYR&sig=WBHTsf3PbfoBT6dLAuYvp2WRuq4&redir_esc=y#v=onepage&q=Andreas%20C.%20M%C3%BCller%20%26%20Sarah%20Guido&f=false

https://books.google.co.id/books?hl=en&lr=&id=zLHGEAAAQBAJ&oi=fnd&pg=PA22&dq=Kombinasi+dari+beberapa+teknik+Feature+Selection+sering+digunakan+untuk+memperoleh+subset+fitur+yang+optimal+sesuai+dengan+kebutuhan+dan+karakteristik+dataset&ots=tzMjdsSQK9&sig=kexVusIL2uOQj9nSDc5eS0lLXYo&redir_esc=y#v=onepage&q&f=false

https://doi.org/10.1016/j.petrol.2018.11.067

doi:10.4249/scholarpedia.1883

https://doi.org/10.3390/electronics12214551

https://www.researchgate.net/profile/Ingmar-Nitze/publication/275641579_COMPARISON_OF_MACHINE_LEARNING_ALGORITHMS_RANDOM_FOREST_ARTIFICIAL_NEURAL_NETWORK_AND_SUPPORT_VECTOR_MACHINE_TO_MAXIMUM_LIKELIHOOD_FOR_SUPERVISED_CROP_TYPE_CLASSIFICATION/links/5541238e0cf2b790436bc791/COMPARISON-OF-MACHINE-LEARNING-ALGORITHMS-RANDOM-FOREST-ARTIFICIAL-NEURAL-NETWORK-AND-SUPPORT-VECTOR-MACHINE-TO-MAXIMUM-LIKELIHOOD-FOR-SUPERVISED-CROP-TYPE-CLASSIFICATION.pdf

https://doi.org/10.48550/arXiv.2309.00363.