Kopekan UAS Matkul Feature Engineering Open-Book

Lathif Ramadhan (5231811022)

Cara Kerja Model Machine Learning Secara Umum

1. **Pengumpulan Data**
   Data dikumpulkan dari berbagai sumber untuk menjadi dasar pelatihan model. Misalnya, data bisa berupa angka, teks, gambar, atau audio.
2. **Preprocessing Data**
   Data yang dikumpulkan biasanya tidak langsung bisa digunakan. Tahap preprocessing mencakup:
   - **Membersihkan data** (handling missing values, outlier, dsb.).
   - **Mengubah format data** (contoh: encoding variabel kategorikal menjadi numerik).
   - **Normalisasi atau standarisasi** untuk memastikan semua fitur berada dalam skala yang sama.
3. **Feature Engineering**
   Model machine learning sangat bergantung pada kualitas fitur. Di sinilah Feature Engineering berperan, mencakup:
   - **Feature Selection**: Memilih fitur paling relevan untuk meningkatkan performa model.
   - **Feature Transformation**: Mengubah data agar lebih cocok untuk algoritma.
   - **Feature Creation**: Menciptakan fitur baru dari data yang ada.
4. **Membagi Data**
   Data dibagi menjadi:
   - **Training set**: Untuk melatih model.
   - **Validation set** (opsional): Untuk menyesuaikan hyperparameter.
   - **Testing set**: Untuk mengevaluasi performa model.
5. **Melatih Model**
   Algoritma machine learning dilatih menggunakan data. Ini berarti model akan "belajar" pola dari data training dengan meminimalkan error atau loss.
6. **Evaluasi Model**
   Model yang sudah dilatih diuji pada data testing. Beberapa metrik evaluasi yang digunakan meliputi akurasi, precision, recall, F1 score, atau mean squared error (MSE), tergantung pada jenis masalah.
7. **Prediksi atau Deployment**
   Setelah model menunjukkan performa yang baik, model dapat digunakan untuk membuat prediksi pada data baru.

Berikut adalah **metode Feature Engineering** yang umum digunakan dalam machine learning, mencakup langkah-langkah dari pemilihan fitur hingga transformasi, pembuatan fitur, dan penanganan data:

A. Feature Selection

Feature Selection adalah proses memilih fitur (kolom dalam dataset) yang paling relevan atau penting untuk membantu model machine learning membuat prediksi yang akurat. Tujuannya adalah:

- Meningkatkan akurasi model.
- Mengurangi kompleksitas model.
- Menghindari overfitting (model terlalu “hafal” data training).

Ada tiga pendekatan utama dalam pemilihan fitur: **Metode Statistik**, **Model-based Methods**, dan **Dimensional Reduction**.

2. Metode Statistik

Digunakan untuk menganalisis hubungan antara fitur dan target berdasarkan perhitungan matematis.

a. Correlation Analysis (Analisis Korelasi)

- **Kegunaan:** Untuk fitur numerik, mengukur seberapa kuat hubungan antara fitur dan target.
- **Contoh:** Korelasi Pearson digunakan untuk mengukur hubungan linier.
  - Korelasi positif (+): Jika satu fitur meningkat, target juga meningkat.
  - Korelasi negatif (-): Jika satu fitur meningkat, target menurun.
  - Korelasi nol (0): Tidak ada hubungan.

b. Chi-Square Test

- **Kegunaan:** Untuk fitur kategorikal dan target kategorikal.
- **Cara Kerja:** Mengukur apakah ada hubungan signifikan antara dua variabel kategorikal.
  - Contoh: Apakah jenis kelamin (kategori) berhubungan dengan pembelian produk (kategori).

c. ANOVA (Analysis of Variance)

- **Kegunaan:** Untuk fitur kategorikal dan target numerik.
- **Cara Kerja:** Mengukur apakah rata-rata target berbeda secara signifikan di antara kategori fitur.
  - Contoh: Apakah rata-rata penghasilan (numerik) berbeda berdasarkan tingkat pendidikan (kategori).

3. Model-based Methods

Menggunakan algoritma machine learning untuk menentukan fitur yang penting berdasarkan pelatihan model.

a. Feature Importance (Pentingnya Fitur)

- **Kegunaan:** Menilai fitur mana yang berkontribusi paling besar terhadap prediksi model.

- **Cara Kerja:**
  - Model seperti *Random Forest* atau *Gradient Boosting* menghitung pentingnya fitur berdasarkan pengaruhnya terhadap hasil prediksi.
  - Hasilnya berupa skor pentingnya fitur (importance score).

b. LASSO (Least Absolute Shrinkage and Selection Operator)

- **Kegunaan:** Untuk regresi linier, memilih fitur dengan koefisien terbesar.
- **Cara Kerja:**
  - LASSO menambahkan penalti ke fitur yang tidak relevan, sehingga koefisiennya menjadi nol.
  - Fitur dengan koefisien nol akan dihapus dari model.

4. Dimensional Reduction (Reduksi Dimensi)

Bertujuan untuk mengurangi jumlah fitur tanpa kehilangan informasi yang signifikan. Cocok untuk dataset dengan banyak kolom (dimensi tinggi).

a. Principal Component Analysis (PCA)

- **Kegunaan:** Mengubah data ke dalam dimensi yang lebih sedikit sambil mempertahankan informasi maksimum.
- **Cara Kerja:**
  1. PCA membuat kombinasi linier baru dari fitur-fitur awal.
  2. Kombinasi tersebut disebut *principal components*.
  3. Hanya beberapa *principal components* dengan varian terbesar yang dipilih.
- **Contoh:** Dataset dengan 100 fitur bisa direduksi menjadi 10 fitur menggunakan PCA.

b. t-SNE (t-Distributed Stochastic Neighbor Embedding) dan UMAP (Uniform Manifold Approximation and Projection)

- **Kegunaan:** Untuk visualisasi data berdimensi tinggi.
- **Cara Kerja:** Mengelompokkan data berdasarkan pola-pola tertentu, sehingga mudah dilihat pada grafik 2D atau 3D.

Kenapa Feature Selection Penting?

1. **Menghemat Waktu Komputasi**: Mengurangi jumlah fitur berarti proses pelatihan lebih cepat.
2. **Mengurangi Kebisingan Data**: Fitur yang tidak relevan dapat menyesatkan model.
3. **Meningkatkan Interpretabilitas**: Dengan fitur yang lebih sedikit, hasil model lebih mudah dipahami.

B. Feature Transformation

Transformasi fitur adalah proses mengubah atau memodifikasi data agar lebih cocok untuk digunakan dalam model machine learning. Hal ini penting karena model bekerja lebih baik jika data berada dalam format yang sesuai dan distribusi yang seragam.

1. Skalabilitas Data (Data Scaling)

Skalabilitas data memastikan semua fitur berada dalam skala yang seragam. Tanpa ini, fitur dengan nilai besar bisa mendominasi proses pelatihan model.

a. Standardization (Standarisasi)
Mengubah data sehingga memiliki rata-rata (mean) 0 dan standar deviasi 1. **Kapan digunakan?**Ketika fitur memiliki distribusi Gaussian atau model yang sensitif terhadap skala (misalnya: SVM, KNN, PCA).

- **Rumus:**

$$z = \frac{x - \mu}{\sigma}$$

Di mana:

- x: Nilai fitur.
- μ: Mean (rata-rata).
- σ: Standar deviasi.

- **Contoh:**
  Jika kolom umur memiliki nilai rata-rata 30 dan standar deviasi 5, nilai 40 akan menjadi:

$$z = \frac{40 - 30}{5} = 2$$

Artinya, 40 berada 2 standar deviasi di atas rata-rata.

b. Normalization (Normalisasi)
Mengubah data agar berada dalam rentang tertentu, biasanya 0 hingga 1.

Kapan digunakan? Cocok untuk metode seperti neural networks atau algoritma berbasis jarak (KNN, K-Means).

- **Rumus:**

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

Di mana:

- $x_{\min}$: Nilai minimum dalam kolom.
- $x_{\max}$: Nilai maksimum dalam kolom.

- **Contoh:**
  Jika tinggi badan berada dalam rentang 150-200 cm, tinggi 175 akan dinormalisasi menjadi:

$$x' = \frac{175 - 150}{200 - 150} = 0.5$$

2. Transformasi Non-linear

Transformasi ini digunakan untuk mengubah bentuk distribusi data agar lebih simetris atau lebih sesuai untuk model.

a. Log Transformation
Mengubah data dengan mengambil logaritma untuk mengatasi data yang sangat skewed (tidak simetris). **Kapan digunakan?** Untuk fitur dengan rentang nilai yang luas atau outlier besar.

- **Rumus:**

$$x' = \log(x + 1)$$

(Penambahan 1 untuk menghindari logaritma dari 0).

Contoh: Data pendapatan yang bervariasi dari 10 ribu hingga 1 juta. Dengan log transformasi, perbedaan ekstremnya menjadi lebih kecil.

b. Square Root Transformation
Mengambil akar kuadrat untuk mengurangi variabilitas atau skala fitur. **Kapan digunakan?**
Untuk data dengan outlier moderat.

Rumus:

$$x' = \sqrt{x}$$

Contoh:
Jika nilai data adalah 16, transformasinya menjadi 4 ($\sqrt{16} = 4$).

c. Exponential Transformation
Menggunakan eksponensial untuk meningkatkan skala fitur. **Kapan digunakan?**
Ketika fitur memiliki distribusi kecil dan ingin diperbesar.

3. Encoding (Mengubah Variabel Kategorikal)

Encoding diperlukan karena algoritma machine learning tidak dapat langsung memahami data kategorikal.

a. One-Hot Encoding
Mengubah setiap kategori menjadi kolom biner (0 atau 1). **Kapan digunakan?** Ketika kategori tidak memiliki urutan alami (contoh: warna, jenis kelamin). **Contoh:** Kolom “warna” dengan nilai ["merah", "biru", "hijau"] akan menjadi:

merah	biru	hijau
1	0	0
0	1	0

b. Ordinal Encoding

Mengubah kategori dengan urutan alami menjadi angka. **Kapan digunakan?** Ketika ada urutan yang logis (contoh: pendidikan: SD < SMP < SMA). **Contoh:** Kolom “pendidikan” akan menjadi:

SD: 1 SMP: 2 SMA: 3

c. Binarization
Mengubah data numerik menjadi biner berdasarkan nilai ambang (threshold). **Contoh:**
Jika threshold adalah 50, nilai lebih besar atau sama dengan 50 menjadi 1, sedangkan nilai di bawahnya menjadi 0.

Transformasi fitur membantu data lebih sesuai dengan algoritma, memastikan performa model lebih baik dan stabil.

C. Feature Creation

Feature Creation adalah proses menciptakan fitur baru dari data yang sudah ada. **Tujuannya** adalah untuk memberikan model informasi tambahan yang lebih relevan sehingga prediksinya menjadi lebih akurat. Metode ini sering kali bergantung pada kreativitas, pemahaman domain, dan logika analisis data.

1. Arithmetic Transformations (Transformasi Aritmatika)

Melakukan operasi matematika antar kolom dalam dataset untuk menciptakan fitur baru. **Contoh:**

- **Penjumlahan:** Dataset memiliki kolom *pendapatan* dan *pengeluaran*. Anda bisa menambahkan kolom baru bernama *sisa uang*:
  sisa uang=pendapatan−pengeluaran
- **Perkalian:** Jika ada kolom *panjang* dan *lebar*, Anda bisa menciptakan fitur *luas*: *luas=panjang×lebar*
- **Rasio:** Membuat rasio dari dua fitur, misalnya:

$$\text{rasio konsumsi} = \frac{\text{pengeluaran}}{\text{pendapatan}}$$

Kenapa penting? Transformasi ini sering kali memberikan insight tambahan bagi model. Contoh: *rasio konsumsi* lebih relevan daripada hanya melihat *pengeluaran* atau *pendapatan* saja.

2. Agregasi (Aggregation)

Menggabungkan data dalam grup tertentu menggunakan fungsi statistik seperti *mean*, *sum*, *min*, *max*, atau *count*. **Contoh:** Dataset toko online memiliki data transaksi dengan beberapa kolom: *ID pelanggan*, *total belanja*, dan *tanggal transaksi*. Anda bisa membuat fitur baru:

- **Rata-rata belanja pelanggan:** Mengelompokkan data berdasarkan *ID pelanggan* dan menghitung rata-rata belanja mereka.
- **Total transaksi pelanggan:** Menghitung jumlah transaksi yang dilakukan setiap pelanggan (fungsi *count*).

ID Pelanggan	Total Belanja	Rata-rata Belanja	Total Transaksi
A001	1,000,000	1,250,000	2
A002	500,000	750,000	2

Kenapa penting?
Agregasi memberikan informasi tingkat makro dari data granular. Ini sangat membantu terutama untuk analisis perilaku pelanggan atau data berbasis waktu.

3. Feature Interaction (Interaksi Fitur)

Menggabungkan dua atau lebih fitur untuk menciptakan fitur baru yang lebih informatif. **Contoh:**

- Dataset properti memiliki kolom *jumlah kamar* dan *luas bangunan*. Anda bisa membuat fitur baru:

$$\text{luas per kamar} = \frac{\text{luas bangunan}}{\text{jumlah kamar}}$$

  Fitur ini memberikan informasi tentang seberapa luas rata-rata setiap kamar.
- Untuk data e-commerce, Anda bisa membuat:

$$\text{pendapatan per transaksi} = \frac{\text{total pendapatan}}{\text{jumlah transaksi}}$$

Kenapa penting? Interaksi fitur sering kali menggambarkan hubungan yang tidak langsung terlihat di data awal. Misalnya, *luas per kamar* memberikan insight yang lebih daripada hanya *luas bangunan*.

4. Time-based Features (Fitur Berbasis Waktu)

Menciptakan fitur baru dari data waktu untuk menangkap pola musiman atau temporal. **Contoh:**

Dataset dengan kolom *tanggal transaksi* dapat diolah menjadi:

- **Hari dalam seminggu:** Apakah transaksi terjadi pada Senin, Selasa, dll.
- **Bulan:** Apakah transaksi terjadi di bulan Januari, Februari, dll.
- **Musim:** Apakah transaksi terjadi di musim panas atau musim hujan.
- **Jam transaksi:** Apakah transaksi terjadi di pagi hari, siang, atau malam.

Tanggal Transaksi	Hari	Bulan	Musim
01/01/2023	Minggu	Januari	Musim Hujan

Kenapa penting?
Banyak data menunjukkan pola musiman. Contoh: transaksi online cenderung lebih banyak pada akhir pekan atau saat musim liburan. Dengan fitur berbasis waktu, model dapat memahami pola ini.

5. Text Features (Fitur dari Data Teks)

Menciptakan fitur dari data teks untuk memahami informasi yang terkandung di dalamnya. **Metode:**

- **TF-IDF (Term Frequency-Inverse Document Frequency):**
  Mengukur seberapa penting sebuah kata dalam dokumen relatif terhadap kumpulan dokumen. Kata yang sering muncul di dokumen tertentu tetapi jarang di dokumen lain akan memiliki skor TF-IDF tinggi.
- **Word Embeddings (Representasi Kata):**
  Mengubah teks menjadi representasi vektor menggunakan model seperti Word2Vec, GloVe, atau BERT. Ini membantu menangkap konteks semantik dari kata-kata.
- **n-grams:** Menggunakan kombinasi kata-kata dalam teks. Contoh:
  1- *gram (unigram):* "saya belajar".
  2- *gram (bigram):* "saya belajar", "belajar keras".

Contoh: Dari teks ulasan pelanggan: "Makanannya enak sekali."

- Fitur *jumlah kata*: 3.
- Fitur *TF-IDF*: Skor berdasarkan pentingnya kata seperti “enak” atau “makanan”.

Kenapa penting? Data teks sering kali sulit diolah langsung. Dengan membuat fitur seperti TF-IDF atau word embeddings, data teks menjadi lebih informatif untuk model.

Kesimpulan: Feature Creation adalah langkah yang sangat penting dalam machine learning. Dengan menciptakan fitur baru dari data yang ada, kita bisa membantu model memahami pola yang lebih kompleks. Teknik-teknik ini sering kali mengandalkan pemahaman domain, analisis data yang mendalam, dan logika kreatif.

4. Handling Missing Values

Ketika bekerja dengan data, sering kali kita menemukan **missing values** (data yang hilang). Hal ini bisa terjadi karena kesalahan pencatatan, data tidak tersedia, atau masalah teknis lainnya. **Mengatasi data yang hilang** adalah langkah penting karena:

1. Model machine learning tidak dapat bekerja dengan baik jika ada data kosong.
2. Missing values dapat menyebabkan hasil analisis menjadi bias atau salah.

Metode Mengatasi Missing Values

1. Imputation (Pengisian Nilai) Metode ini mengisi data yang hilang dengan nilai yang relevan, seperti rata-rata, median, atau modus.

a. Mean Imputation (Rata-rata)

- **Cara Kerja:** Mengisi nilai yang hilang dengan rata-rata dari kolom tersebut.
- **Contoh:** Dataset dengan kolom *umur*:

ID	Umur
1	25
2	30
3	-
4	35

Rata-rata umur:

$$\text{Mean} = \frac{25 + 30 + 35}{3} = 30$$

Data yang hilang diisi menjadi:

ID	Umur
3	30

b. Median Imputation (Median)

- **Cara Kerja:** Mengisi nilai hilang dengan median (nilai tengah) dari kolom tersebut.
- **Kapan digunakan:** Jika data memiliki outlier, median lebih baik daripada rata-rata.
- **Contoh:** Jika umur adalah [25, 30, 100 (outlier)], median adalah 30 (bukan rata-rata 51.6).

c. Mode Imputation (Modus)

- **Cara Kerja:** Mengisi nilai hilang dengan modus (nilai yang paling sering muncul).
- **Kapan digunakan:** Cocok untuk data kategorikal.
- **Contoh:** Jika kolom *warna* memiliki nilai ["merah", "biru", "merah"], nilai hilang diisi dengan "merah".

2. Forward Fill/Backward Fill Metode ini sering digunakan untuk data **time series** (berbasis waktu).

a. Forward Fill

- **Cara Kerja:** Mengisi nilai hilang dengan nilai terakhir yang tersedia sebelumnya.
- **Contoh:** Data penjualan harian:

Tanggal	Penjualan
01-01	100
02-01	-
03-01	120

Dengan forward fill:

Tanggal	Penjualan
02-01	100

b. Backward Fill Mengisi nilai hilang dengan nilai berikutnya yang tersedia.

Contoh: Data sama seperti sebelumnya, dengan backward fill:

Tanggal	Penjualan
02-01	120

Kapan digunakan? Data yang bersifat sekuensial, seperti time series, cocok menggunakan metode ini.

3. Model-Based Imputation Pendekatan canggih menggunakan model machine learning untuk memprediksi nilai yang hilang. **Cara Kerja:**

1. Identifikasi kolom dengan missing values.
2. Gunakan kolom lain dalam dataset untuk melatih model prediksi.
3. Prediksi nilai yang hilang dengan model ini.

Contoh: Dataset dengan kolom *umur* (hilang) dan *pendapatan*. Model regresi bisa dilatih menggunakan *pendapatan* untuk memprediksi nilai umur yang hilang.

ID	Pendapatan	Umur
1	50 juta	25
2	70 juta	-
3	80 juta	35

Hasil prediksi dari model regresi bisa mengisi nilai umur yang hilang untuk ID 2.

- **Kelebihan:** Lebih akurat karena memanfaatkan informasi lain dalam dataset.
- **Kekurangan:** Membutuhkan waktu lebih lama dan sumber daya tambahan.

Kapan Menggunakan Metode Tertentu?

- **Mean/Median/Mode Imputation:** Cepat, cocok untuk data kecil atau data tanpa pola kompleks.
- **Forward Fill/Backward Fill:** Cocok untuk data time series.
- **Model-Based Imputation:** Ideal untuk dataset besar dengan pola kompleks, tetapi memerlukan sumber daya lebih.

Kesimpulan Mengatasi missing values adalah langkah penting untuk memastikan data bersih dan model machine learning dapat bekerja dengan optimal. Metode yang dipilih harus disesuaikan dengan jenis data dan konteks analisis.

5. Handling Outliers (Penanganan Outlier)

Outlier adalah nilai data yang sangat berbeda atau ekstrem dibandingkan dengan sebagian besar data lainnya. Outlier ini bisa berdampak negatif pada model machine learning karena dapat menyebabkan model menjadi bias atau tidak akurat. **Tujuan Penanganan Outlier**:

- Mengurangi dampak outlier yang dapat merusak analisis atau prediksi.
- Membuat model lebih robust (tahan terhadap gangguan atau ketidaknormalan dalam data).

Metode Umum dalam Penanganan Outlier

1. Winsorization Winsorization adalah metode untuk mengubah nilai ekstrem (outlier) menjadi batas tertentu, yaitu nilai yang lebih moderat agar tidak terlalu mempengaruhi hasil analisis. **Bagaimana Cara Kerjanya?**

- Misalnya, jika ada data yang lebih besar dari nilai tertentu (misalnya 100), maka nilai tersebut akan diganti dengan 100. Begitu juga untuk data yang terlalu kecil (misalnya kurang dari -100), akan diganti menjadi -100.
- Tujuannya adalah untuk membatasi nilai ekstrim dalam batas yang masuk akal tanpa menghapusnya sepenuhnya.

Kapan Digunakan? Saat Anda ingin mempertahankan semua data dalam dataset dan hanya mengurangi dampak outlier yang sangat ekstrem tanpa menghilangkannya.

2. Clipping Clipping adalah metode yang sangat mirip dengan Winsorization, tetapi dengan sedikit perbedaan. Dalam clipping, Anda "memotong" atau membuang nilai-nilai yang berada di luar batas yang telah ditentukan. **Bagaimana Cara Kerjanya?**

- Jika data memiliki nilai yang lebih besar dari threshold (misalnya 100), maka nilai tersebut akan dipotong atau diganti dengan nilai maksimal, yaitu 100.
- Demikian pula untuk nilai yang lebih kecil dari batas minimum (misalnya -100), maka nilai tersebut akan diganti dengan -100.

Kapan Digunakan? Digunakan ketika Anda ingin menghindari nilai ekstrem dalam data yang bisa memengaruhi hasil model dan tidak ingin membiarkan data tersebut tetap berada dalam dataset.

3. Transformation (Transformasi)

Metode transformasi digunakan untuk mengubah skala atau distribusi data sehingga outlier memiliki dampak yang lebih kecil terhadap model. Ada beberapa jenis transformasi yang bisa digunakan:

Log Transformation: Jika data memiliki distribusi yang sangat miring (skewed), misalnya banyak nilai besar yang jauh lebih besar dari nilai lainnya, kita bisa menggunakan **logaritma** untuk memperkecil jarak antara nilai-nilai tersebut. Contoh: Jika nilai dataset berupa penghasilan yang sangat bervariasi, logaritma penghasilan akan meratakan distribusi dan mengurangi dampak dari penghasilan yang sangat besar.

Square Root Transformation:

- Sama seperti logaritma, tetapi lebih sederhana. Digunakan untuk mengurangi efek nilai besar (outlier) pada data yang memiliki distribusi skewed.
- Misalnya, untuk data yang memiliki kuadrat atau jarak yang sangat besar, square root akan membuat distribusi lebih normal.

Exponential Transformation:

- Dapat digunakan untuk mengubah data yang sangat terdistribusi rendah menjadi lebih tersebar.
- Penggunaan transformasi ini bisa mengurangi pengaruh data yang lebih besar dan menyeimbangkan distribusi data.

Mengapa Penanganan Outlier Itu Penting? Outlier dapat menyebabkan model untuk:

- Menjadi terlalu sensitif terhadap data yang ekstrem, yang bisa mengarah pada overfitting.
- Mengabaikan pola utama dalam data.
- Menyebabkan model tidak memprediksi dengan baik, terutama dalam model yang sensitif terhadap data ekstrem seperti regresi linier.

Dengan mengelola outlier menggunakan metode di atas, kita bisa memastikan bahwa model machine learning akan bekerja dengan lebih baik dan lebih stabil.

Kapan Harus Menggunakan Setiap Metode?

- **Winsorization** dan **Clipping** cocok digunakan saat Anda ingin mempertahankan data lengkap tanpa menghapus data yang sangat ekstrim. Anda hanya ingin menurunkan dampaknya.
- **Transformation** cocok digunakan saat distribusi data Anda sangat miring (skewed) dan perlu distabilkan untuk memperbaiki model, terutama untuk algoritma yang sangat sensitif terhadap distribusi data, seperti regresi atau algoritma berbasis jarak (contoh: k-Nearest Neighbors).

6. Feature Augmentation (Augmentasi Fitur)

Feature Augmentation adalah proses menambahkan informasi baru ke dalam dataset untuk meningkatkan kualitas model. Dengan menambahkan fitur baru, model dapat menangkap pola yang lebih baik dan menghasilkan prediksi yang lebih akurat.

Pada dasarnya, augmentasi fitur **bertujuan** untuk memperkaya dataset yang ada dengan informasi tambahan yang relevan sehingga model dapat belajar lebih banyak dan lebih efektif.

Metode Umum dalam Feature Augmentation

1. External Data Integration (Integrasi Data Eksternal)

External Data Integration adalah teknik untuk menambahkan data dari sumber eksternal yang relevan dengan masalah yang sedang dianalisis. Data ini bisa berasal dari berbagai sumber yang tidak ada dalam dataset asli tetapi dapat memberikan wawasan lebih untuk meningkatkan performa model.

Contoh:

- **Data Cuaca**: Misalnya, jika kita ingin menganalisis penjualan produk tertentu, data cuaca seperti suhu atau curah hujan dapat memengaruhi penjualan. Produk seperti payung atau jas hujan biasanya terjual lebih banyak saat hujan atau cuaca dingin. Dengan menambahkan data cuaca ke dalam dataset penjualan, model bisa lebih akurat dalam memprediksi penjualan.
- **Data Pasar**: Dalam analisis saham atau pasar, data eksternal seperti nilai tukar mata uang, indeks harga saham, atau data ekonomi lainnya bisa membantu memprediksi tren pasar.
- **Data Sosial Media**: Untuk analisis sentimen atau perilaku pelanggan, menambahkan data dari media sosial (misalnya Twitter, Facebook) dapat memberikan perspektif baru yang lebih luas.

Kapan Digunakan? Saat ada informasi tambahan yang relevan dari luar dataset yang dapat memperbaiki hasil model. Ini sangat bermanfaat untuk model yang memerlukan kontekstualisasi atau data lebih dari sekadar yang ada di dataset asli.

Bagaimana Cara Mengintegrasikannya?

- Data eksternal bisa digabungkan dengan dataset yang ada menggunakan ID yang sama (misalnya ID produk, ID pelanggan, atau tanggal yang sama).
- Setelah data eksternal ditambahkan, pastikan untuk menyeimbangkan dan membersihkan data tersebut agar konsisten dengan data yang sudah ada.

2. Domain-Specific Features (Fitur Khusus Berdasarkan Domain)

Fitur khusus ini dibuat berdasarkan pemahaman tentang domain atau bidang yang sedang dianalisis. Menggunakan pengetahuan domain dapat membantu dalam membuat fitur yang lebih informatif dan relevan untuk model.

Contoh:

- **Industri Kesehatan**: Jika dataset berisi data pasien, kita bisa membuat fitur baru berdasarkan pengetahuan medis, seperti "risiko penyakit jantung" yang dihitung berdasarkan faktor-faktor seperti usia, jenis kelamin, tekanan darah, dan riwayat keluarga.
- **E-commerce**: Dalam analisis pembelian produk online, kita bisa membuat fitur seperti "jumlah produk dalam keranjang" atau "frekuensi pembelian sebelumnya" yang dapat meningkatkan prediksi terkait perilaku pembelian pelanggan.
- **Data Transportasi**: Jika kita memiliki data perjalanan, kita bisa membuat fitur baru seperti "waktu tempuh" atau "kepadatan lalu lintas" berdasarkan waktu atau lokasi tertentu.

Kapan Digunakan? Saat ada pemahaman atau pengetahuan khusus yang dapat memperkaya dataset dan memberikan konteks tambahan yang berguna untuk model. Fitur domain-spesifik sangat bermanfaat untuk model yang membutuhkan analisis yang lebih mendalam dalam konteks tertentu.

Bagaimana Cara Membuatnya?

- Fitur domain-spesifik sering kali melibatkan pemrosesan data yang lebih rumit, seperti perhitungan statistik atau penggabungan beberapa variabel untuk menciptakan metrik baru yang relevan.
- Anda bisa mengkombinasikan berbagai variabel yang ada dalam dataset dan membuat fitur baru berdasarkan pemahaman Anda tentang bagaimana fitur-fitur tersebut berhubungan dalam konteks dunia nyata.

Mengapa Feature Augmentation Itu Penting?

- **Memperbaiki Kinerja Model**: Menambahkan fitur yang relevan dapat membantu model menangkap informasi lebih banyak, sehingga dapat meningkatkan akurasi prediksi atau analisis.
- **Menyediakan Wawasan yang Lebih Dalam**: Dengan menggunakan data eksternal atau fitur berdasarkan pengetahuan domain, model dapat bekerja dengan informasi yang lebih kontekstual dan mendalam.
- **Meningkatkan Generalisasi**: Model yang menggunakan augmentasi fitur cenderung lebih generalis, artinya model lebih mampu menangani data baru yang belum pernah dilihat sebelumnya.

Kapan Harus Menggunakan Feature Augmentation?

- **Data yang Terbatas**: Jika dataset yang Anda miliki tidak cukup kaya atau beragam, augmentasi fitur dapat membantu untuk menambah variasi data dan memberikan lebih banyak informasi kepada model.
- **Meningkatkan Performa**: Jika model Anda belum memberikan hasil yang baik, augmentasi fitur bisa menjadi cara untuk meningkatkan kinerja dengan memberikan fitur yang lebih relevan atau kontekstual.
- **Fitur yang Terlalu Sederhana**: Jika fitur dalam dataset Anda terlalu sederhana atau kurang menggambarkan hubungan antar data, membuat fitur baru bisa membantu model mempelajari pola yang lebih kompleks.

Kesimpulan; Feature Augmentation adalah cara untuk memperkaya dataset dengan informasi tambahan yang dapat membantu model belajar lebih baik. Dengan menambahkan data eksternal yang relevan atau membuat fitur baru berdasarkan pengetahuan domain, Anda memberi model lebih banyak konteks yang dapat meningkatkan performa dan akurasi prediksi. Selalu pastikan bahwa fitur yang ditambahkan memang relevan dan dapat memberikan nilai tambah, karena menambah fitur yang tidak relevan bisa menambah kompleksitas tanpa meningkatkan hasil.

7. Advanced Encoding Techniques for Categorical Data

Data kategorikal adalah jenis data yang terdiri dari kategori atau label, seperti jenis kelamin (pria/wanita), status perkawinan (menikah/belum menikah), atau warna produk (merah/hijau/kuning). Banyak model machine learning, seperti regresi atau pohon keputusan, memerlukan data numerik, sehingga data kategorikal perlu diubah menjadi bentuk numerik sebelum dapat digunakan dalam model.

Selain metode encoding dasar seperti **One-Hot Encoding**, ada beberapa metode lanjutan yang lebih canggih, yang sering digunakan untuk mengatasi masalah encoding pada data kategorikal, terutama ketika data tersebut memiliki banyak kategori.

Metode Umum dalam Advanced Encoding

1. Frequency Encoding (Pengkodean Frekuensi)

Frequency Encoding mengganti kategori dengan frekuensi kemunculan kategori tersebut dalam dataset. Setiap kategori akan diubah menjadi angka yang mewakili seberapa sering kategori tersebut muncul dalam data. **Contoh**: Misalkan kita memiliki fitur warna_produk dengan kategori merah, hijau, dan biru. Jika dalam dataset terdapat 100 produk dengan warna merah, 50 hijau, dan 30 biru, maka: *merah akan diganti dengan 100, hijau akan diganti dengan 50, biru akan diganti dengan 30*.

Kapan Digunakan? Digunakan ketika kategori memiliki distribusi yang tidak merata, dan frekuensi kategori tersebut memberikan informasi penting yang bisa dimanfaatkan oleh model. Metode ini sangat berguna ketika jumlah kategori sangat banyak dan kita ingin mengurangi dimensi data tanpa mengorbankan informasi penting.

Kelebihan:

- Simpel dan mudah diterapkan.
- Mengurangi dimensionalitas jika dibandingkan dengan metode One-Hot Encoding, karena tidak menghasilkan banyak kolom.

Kekurangan:

- Dapat menyebabkan masalah **overfitting** jika kategori yang jarang muncul memiliki frekuensi yang tinggi, atau sebaliknya.
- Tidak mempertimbangkan hubungan antara kategori dan target secara langsung, hanya berdasarkan frekuensi kemunculannya.

2. Target Encoding (Pengkodean Target)

Target Encoding mengganti kategori dengan rata-rata nilai target (output) untuk setiap kategori. Dengan kata lain, untuk setiap kategori, kita menghitung nilai rata-rata target untuk data yang masuk ke dalam kategori tersebut, dan mengganti kategori tersebut dengan rata-rata tersebut. **Contoh**:

Misalkan kita memiliki fitur kota (kategori: Jakarta, Surabaya, Bandung) dan targetnya adalah pendapatan (nilai kontinu). Jika rata-rata pendapatan untuk Jakarta adalah 10 juta, Surabaya 8 juta, dan Bandung 6 juta, maka:

Jakarta akan diganti dengan 10 juta, Surabaya akan diganti dengan 8 juta, Bandung akan diganti dengan 6 juta.

Kapan Digunakan? Metode ini sangat berguna untuk data kategorikal dengan hubungan yang kuat antara kategori dan target. Misalnya, dalam analisis harga rumah, kategori seperti lokasi atau jenis rumah sangat memengaruhi harga, sehingga target encoding bisa memberikan wawasan yang lebih baik.

Kelebihan:

- Dapat menangkap hubungan antara kategori dan target, yang meningkatkan prediksi model.
- Mengurangi dimensi dataset karena tidak menghasilkan banyak kolom.

Kekurangan:

- Dapat menyebabkan **overfitting** karena model bisa terlalu bergantung pada nilai rata-rata target untuk kategori tertentu, terutama jika dataset kecil.
- Memerlukan teknik validasi yang hati-hati, seperti **cross-validation**, untuk menghindari kebocoran data.

3. Leave-One-Out Encoding (Pengkodean Leave-One-Out)

Leave-One-Out Encoding adalah versi yang lebih stabil dari **Target Encoding**. Alih-alih mengganti kategori dengan rata-rata target dari semua data dalam kategori tersebut, Leave-One-Out Encoding mengganti kategori dengan rata-rata target, namun **mengabaikan nilai target dari observasi yang sedang diproses**. Jadi, ketika kita mengubah kategori, kita mengabaikan observasi itu sendiri agar tidak ada kebocoran informasi.

Contoh: Misalkan kita memiliki fitur kota dan target pendapatan dengan kategori Jakarta, Surabaya, dan Bandung. Untuk Jakarta, alih-alih menggunakan rata-rata seluruh pendapatan yang ada di Jakarta, kita akan menghitung rata-rata pendapatan di Jakarta tetapi mengabaikan observasi yang sedang diproses.

Kapan Digunakan? Digunakan saat Anda ingin menggunakan Target Encoding tetapi juga ingin menghindari masalah **overfitting** atau kebocoran data (data leakage), yang bisa terjadi jika model melihat data target pada saat pelatihan.

Kelebihan:

- Lebih stabil dibandingkan Target Encoding karena mengurangi risiko overfitting.
- Membantu model menghindari **data leakage**, terutama dalam kasus validasi dan prediksi.

Kekurangan:

- Lebih kompleks dibandingkan Target Encoding.
- Memerlukan perhitungan yang lebih rumit dan mungkin lebih memakan waktu untuk dataset besar.

Kapan Harus Menggunakan Advanced Encoding Techniques?

- **Frekuensi Kategori Tidak Merata**: Jika dataset memiliki kategori dengan frekuensi yang sangat tidak merata (misalnya, beberapa kategori muncul sangat sering, sementara yang lain sangat jarang), maka **Frequency Encoding** bisa menjadi solusi yang baik.
- **Hubungan Kuat Antara Kategori dan Target**: Jika kategori dalam fitur sangat berhubungan dengan target, maka **Target Encoding** atau **Leave-One-Out Encoding** dapat memberikan hasil yang lebih baik karena keduanya mempertimbangkan hubungan tersebut.

Kesimpulan: Advanced Encoding Techniques for Categorical Data adalah metode lanjutan untuk mengubah data kategorikal menjadi bentuk numerik yang lebih berguna untuk model machine learning. **Frequency Encoding** menggantikan kategori dengan frekuensinya, **Target Encoding** menggantikan kategori dengan rata-rata target, dan **Leave-One-Out Encoding** adalah versi yang lebih stabil dari Target Encoding yang menghindari kebocoran data. Semua metode ini dapat meningkatkan kinerja model, tetapi harus diterapkan dengan hati-hati agar tidak menyebabkan overfitting atau masalah lain.

8. Dimensionality Reduction (Reduksi Dimensi)

Dimensionality Reduction adalah teknik yang digunakan untuk mengurangi jumlah fitur atau kolom dalam dataset tanpa kehilangan informasi penting. **Tujuan utama** dari teknik ini adalah untuk mempermudah analisis dan mencegah **overfitting**, yaitu masalah ketika model terlalu kompleks dan gagal generalisasi pada data yang tidak terlihat sebelumnya.

Reduksi dimensi sangat berguna ketika dataset memiliki banyak fitur, yang sering kali menyebabkan:

- Waktu pelatihan yang lama.
- Model yang lebih kompleks dan rentan terhadap overfitting.
- Kesulitan dalam menganalisis atau visualisasi data.

Ada beberapa metode yang umum digunakan untuk reduksi dimensi, dan masing-masing memiliki cara berbeda dalam memilih fitur yang harus dipertahankan dan yang harus dibuang.

Metode Umum dalam Dimensionality Reduction

1. Variance Threshold (Ambang Variansi)

Variance Threshold adalah teknik yang sederhana di mana kita menghapus fitur yang memiliki variansi sangat rendah. Fitur dengan variansi rendah (misalnya, hampir semua nilainya sama) tidak memberikan banyak informasi untuk model. Oleh karena itu, fitur-fitur ini dapat diabaikan, dan hanya fitur dengan variansi tinggi yang dipertahankan.

Contoh: Misalkan kita memiliki fitur warna_produk dengan kategori merah, hijau, biru. Jika hampir semua data dalam fitur tersebut memiliki nilai yang sama (misalnya, hampir semua produk berwarna merah), maka fitur ini memiliki variansi rendah dan tidak banyak memberikan informasi yang berbeda, sehingga bisa dihapus.

Kapan Digunakan? Digunakan ketika kita tahu bahwa fitur dengan variansi rendah tidak akan membantu model untuk membedakan pola, seperti fitur yang hampir selalu memiliki nilai yang sama.

Kelebihan:

- Metode yang cepat dan sederhana.
- Mengurangi jumlah fitur dengan cara yang sangat efektif, tanpa memerlukan banyak perhitungan.

Kekurangan:

- Hanya mempertimbangkan variansi antar fitur, tanpa memperhitungkan relevansi fitur terhadap target atau hubungan antar fitur.
- Mungkin ada fitur yang penting meskipun memiliki variansi rendah, jadi perlu hati-hati dalam menerapkannya.

2. Feature Clustering (Pengelompokan Fitur)

Feature Clustering adalah teknik yang menggabungkan fitur yang sangat berkorelasi menjadi satu fitur tunggal. Ketika dua atau lebih fitur memiliki korelasi yang tinggi (misalnya, kedua fitur tersebut mengandung informasi yang hampir sama), kita dapat menggabungkannya menjadi satu fitur yang mewakili keduanya. Ini akan mengurangi jumlah fitur tanpa mengorbankan banyak informasi.

Contoh: Misalkan kita memiliki dua fitur, tinggi_badan (height) dan berat_badan (weight), yang memiliki korelasi tinggi dengan kategori tertentu (misalnya, orang yang tinggi biasanya juga lebih berat). Kita dapat menggabungkan kedua fitur ini menjadi satu fitur yang menggambarkan ukuran tubuh, seperti **Indeks Massa Tubuh (IMT/BMI)**, yang merupakan kombinasi dari tinggi dan berat badan.

Kapan Digunakan? Digunakan ketika ada fitur yang saling berkorelasi dan informasi yang diberikan oleh fitur-fitur tersebut hampir sama, sehingga tidak perlu mempertahankan keduanya.

Kelebihan:

- Mengurangi redundansi dalam dataset.
- Mempertahankan informasi yang relevan dengan cara mengurangi jumlah fitur.

Kekurangan:

- Mungkin ada beberapa informasi spesifik yang hilang setelah penggabungan, meskipun penggabungan dilakukan dengan hati-hati.
- Teknik ini membutuhkan analisis yang lebih mendalam untuk mengidentifikasi fitur yang relevan untuk digabungkan.

3. Autoencoders

Autoencoders adalah teknik berbasis **neural network** yang digunakan untuk reduksi dimensi. Autoencoder terdiri dari dua bagian utama: encoder dan decoder. Encoder mengubah data input menjadi representasi berdimensi lebih rendah (dikenal sebagai kode atau encoding), sementara decoder mencoba untuk merekonstruksi data asli dari encoding tersebut. **Tujuan** dari autoencoder adalah untuk belajar representasi yang lebih kompak dari data, sehingga dimensi data bisa dikurangi.

Proses:

- Data input masuk ke bagian **encoder** dan diubah menjadi representasi berdimensi lebih rendah.
- Representasi ini kemudian diteruskan ke bagian **decoder**, yang mencoba merekonstruksi data asli berdasarkan representasi berdimensi rendah tersebut.
- Selama pelatihan, model belajar bagaimana menyimpan informasi penting dalam representasi berdimensi rendah, dan membuang informasi yang kurang relevan.

Kapan Digunakan?

- Digunakan untuk dataset besar dengan dimensi tinggi, seperti gambar atau data sensor.
- Cocok digunakan ketika kita ingin mengurangi dimensi tetapi tetap mempertahankan informasi sebanyak mungkin.

Kelebihan:

- Dapat mengurangi dimensi data secara signifikan tanpa kehilangan informasi penting.
- Mampu menangani data yang sangat kompleks, seperti gambar atau urutan waktu.

Kekurangan:

- Memerlukan pelatihan yang lebih lama dan sumber daya komputasi yang lebih besar, karena menggunakan jaringan saraf.
- Kadang-kadang bisa terlalu kompleks untuk dataset yang lebih kecil atau sederhana.

Kapan Harus Menggunakan Dimensionality Reduction?

- **Overfitting**: Jika model terlalu kompleks dan cenderung mempelajari noise atau detail yang tidak penting dari data (overfitting), maka reduksi dimensi dapat membantu dengan mengurangi jumlah fitur.
- **Kinerja Model**: Jika model berjalan lambat karena terlalu banyak fitur, mengurangi dimensi dapat mempercepat proses pelatihan dan prediksi.
- **Visualisasi**: Jika tujuannya adalah untuk memahami data atau menampilkan data dalam bentuk yang lebih mudah dipahami (misalnya dalam 2D atau 3D), reduksi dimensi memungkinkan visualisasi data berdimensi tinggi.

Kesimpulan

Dimensionality Reduction (Reduksi Dimensi) adalah proses untuk mengurangi jumlah fitur dalam dataset, yang bertujuan untuk mencegah **overfitting**, mempercepat pelatihan model, dan mempermudah analisis data. Ada beberapa metode untuk melakukannya, seperti **Variance Threshold**, yang menghapus fitur dengan variansi rendah, **Feature Clustering**, yang menggabungkan fitur yang berkorelasi tinggi, dan **Autoencoders**, yang menggunakan teknik neural network untuk mengubah data ke representasi berdimensi lebih rendah.

Label Encoding

Label Encoding adalah teknik untuk mengubah fitur kategorikal menjadi angka. Setiap kategori unik diubah menjadi angka yang sesuai, tanpa memperhatikan urutan atau peringkat antar kategori. Contohnya, jika kita memiliki kategori warna seperti "Merah", "Biru", dan "Hijau", Label Encoding bisa mengubahnya menjadi angka 0, 1, dan 2, atau dalam urutan lainnya, meskipun tidak ada hubungan urutan antara warna-warna tersebut. Contoh Label Encoding:

Merah → 0 Biru → 1 Hijau → 2

Ordinal Encoding

Ordinal Encoding adalah metode encoding yang serupa, tetapi digunakan khusus untuk data kategorikal yang memiliki urutan atau ranking. Misalnya, jika kita memiliki kategori "Rendah", "Sedang", dan "Tinggi", kita bisa mengubahnya menjadi angka 0, 1, dan 2 karena ada urutan yang jelas dalam kategori tersebut. Contoh Ordinal Encoding:

Rendah → 0 Sedang → 1 Tinggi → 2

Perbedaan Kunci:

- **Label Encoding** tidak memedulikan urutan dalam kategori, sedangkan **Ordinal Encoding** digunakan ketika kategori memiliki urutan atau ranking yang jelas.
- Label Encoding umumnya digunakan untuk fitur kategorikal nominal (tanpa urutan), sementara Ordinal Encoding digunakan untuk fitur kategorikal ordinal (dengan urutan).

Kesimpulan:

Meskipun keduanya mengubah kategori menjadi angka, **Ordinal Encoding** biasanya merujuk pada encoding untuk data dengan urutan yang jelas, sementara **Label Encoding** lebih umum dan dapat digunakan pada data dengan atau tanpa urutan. Namun, dalam praktik, istilah **Label Encoding** sering digunakan secara luas dan mencakup **Ordinal Encoding** ketika urutan dianggap relevan.

---

1. Binning (Discretization)

Binning adalah teknik dalam feature engineering yang digunakan untuk mengubah fitur kontinu (numerik) menjadi kategori diskrit. Tujuannya adalah untuk menyederhanakan data dan mempermudah pemodelan, atau untuk mengurangi dampak noise pada data. Teknik ini sangat berguna ketika fitur numerik memiliki banyak nilai yang sangat bervariasi, dan kita ingin mengelompokkan nilai-nilai tersebut menjadi beberapa kategori.

Proses Binning:

- **Equal Width Binning**: Membagi rentang nilai fitur ke dalam interval yang memiliki lebar yang sama. Misalnya, jika nilai fitur berkisar antara 0 dan 100, kita bisa membaginya menjadi 5 bin dengan rentang 20 (misalnya 0-20, 21-40, 41-60, dst.).
- **Equal Frequency Binning**: Membagi data ke dalam bin yang masing-masing memiliki jumlah data yang sama. Misalnya, jika kita memiliki 100 data, kita bisa membagi data menjadi 5 bin dengan masing-masing 20 data.
- **Custom Binning**: Membuat bin dengan interval yang disesuaikan berdasarkan pengetahuan domain atau pemahaman terhadap data. Misalnya, dalam konteks usia, kita bisa membuat bin seperti "0-18", "19-35", "36-60", "60+".

Kelebihan Binning:

- Membantu mereduksi variabilitas yang terlalu tinggi pada data numerik.
- Membuat model lebih stabil dengan mengelompokkan nilai-nilai yang mirip.
- Bisa meningkatkan interpretabilitas model, terutama jika model yang digunakan sulit untuk menangani fitur kontinu.

Kekurangan Binning:

- Kehilangan informasi rinci karena data diubah menjadi bin yang lebih besar.
- Bisa mengurangi performa model, terutama jika bin yang digunakan terlalu kasar atau tidak relevan.

2. Scaling (Penormalan dan Standarisasi Data)

Scaling adalah teknik untuk mengubah skala fitur sehingga data memiliki nilai yang lebih seragam, memudahkan pembelajaran model, dan meningkatkan performa model yang peka terhadap skala fitur, seperti **k-Nearest Neighbors (k-NN)** atau **Gradient Descent**.

Jenis-jenis Scaling:

- **Standardization (Z-score normalization)**: Mengubah data sehingga memiliki distribusi dengan mean 0 dan standar deviasi 1. Formula standar untuk standardization adalah:

$$Z = \frac{X - \mu}{\sigma}$$

Di mana:

- X adalah nilai fitur,
- μ adalah mean dari fitur,
- σ adalah standar deviasi fitur.

Standardization sangat berguna ketika data memiliki distribusi normal atau hampir normal.

- **Normalization (Min-Max scaling)**: Mengubah data sehingga berada dalam rentang tertentu, biasanya antara 0 dan 1. Formula untuk normalisasi adalah:

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

Di mana:

- X_{min} dan X_{max} adalah nilai minimum dan maksimum fitur.

Normalization cocok untuk data yang memiliki rentang nilai yang sangat besar atau tidak terdistribusi secara normal, dan ketika kita ingin memastikan bahwa fitur-fitur berada dalam rentang yang seragam.

Kelebihan Scaling:

- Membantu model yang berbasis jarak (misalnya k-NN, SVM) untuk bekerja lebih baik, karena model-model ini sangat peka terhadap perbedaan skala antar fitur.
- Mengurangi potensi bias yang disebabkan oleh skala fitur yang sangat bervariasi.
- Diperlukan oleh beberapa model yang menggunakan algoritma optimasi berbasis gradien.

Kekurangan Scaling:

- Tidak selalu meningkatkan performa model, terutama jika model sudah cukup robust terhadap skala data, seperti pohon keputusan.
- Dalam beberapa kasus, bisa menyebabkan hilangnya informasi yang dibawa oleh skala fitur asli.

3. Date Extraction (Ekstraksi Fitur Waktu)

Date Extraction adalah proses mengambil informasi penting dari data bertipe waktu (datetime) dan mengubahnya menjadi fitur yang lebih bermakna untuk model. Misalnya, jika kita memiliki data waktu transaksi, kita bisa mengekstrak komponen-komponen seperti **tahun**, **bulan**, **hari**, **hari dalam minggu**, atau **jam** untuk membantu model memprediksi pola yang mungkin ada di dalam data tersebut.

Contoh Ekstraksi Fitur Waktu:

- **Year**: Mengambil tahun dari tanggal (misalnya, 2023).
- **Month**: Mengambil bulan dari tanggal (misalnya, 01-12).
- **Day of Week**: Mengambil hari dalam seminggu (misalnya, Senin = 0, Selasa = 1, dst.).
- **Day of Month**: Mengambil hari dalam bulan (misalnya, 1, 2, 3, dst.).
- **Hour/Minute/Second**: Mengambil waktu dalam jam, menit, atau detik.
- **Weekend Indicator**: Menandai apakah tanggal tersebut jatuh pada akhir pekan (misalnya, Sabtu atau Minggu).

Kelebihan Date Extraction:

- Memudahkan analisis berdasarkan pola musiman, tren mingguan, atau bahkan tren harian.
- Dapat memberikan insight tambahan yang penting untuk model, terutama dalam konteks bisnis atau prediksi berbasis waktu (seperti penjualan, lalu lintas situs web, dll).

Kekurangan Date Extraction:

- Mungkin memperkenalkan banyak fitur baru, yang dapat menambah kompleksitas model.
- Tidak selalu relevan jika data waktu tidak menunjukkan pola musiman atau tren.

4. Feature Split (Pemisahan Fitur)

Feature Split adalah teknik untuk memisahkan satu fitur menjadi beberapa fitur berdasarkan komponen-komponennya. Teknik ini berguna ketika sebuah fitur mengandung informasi yang dapat dipecah untuk membantu model belajar lebih baik.

Contoh Feature Split:

- **Pemisahan Alamat (Street, City, State, Zip)**: Jika kita memiliki kolom alamat yang berisi data seperti "Jalan Merdeka 10, Jakarta, 12345", kita bisa memisahkan alamat tersebut menjadi beberapa kolom: "Street", "City", "State", dan "Zip".
- **Pemisahan Tanggal (Year, Month, Day)**: Jika kita memiliki kolom tanggal yang berisi data seperti "2023-01-15", kita bisa memisahkannya menjadi tiga kolom: "Year", "Month", dan "Day".
- **Pemisahan Nama**: Jika kolom nama berisi data seperti "John Doe", kita bisa memisahkan nama menjadi dua kolom: "First Name" dan "Last Name".

Kelebihan Feature Split:

- Mempermudah model untuk menangkap informasi yang lebih detail.
- Dapat menghasilkan fitur yang lebih mudah dipahami dan lebih relevan untuk analisis.

- Memungkinkan penggunaan model yang lebih kuat untuk tipe data yang terpisah (misalnya, fitur geografis atau waktu).

Kekurangan Feature Split:

- Dapat meningkatkan jumlah fitur dalam dataset, yang bisa memperlambat pelatihan model.
- Kadang-kadang terlalu banyak pemisahan bisa membuat model terlalu rumit tanpa menambah nilai signifikan.

Kesimpulan:

- **Binning** adalah cara untuk mengelompokkan nilai kontinu menjadi kategori yang lebih mudah dikelola.
- **Scaling** digunakan untuk mengubah data agar berada pada skala yang seragam, yang sangat penting bagi model yang peka terhadap skala fitur.
- **Date Extraction** berfokus pada memecah informasi waktu menjadi komponen yang lebih berguna, seperti tahun, bulan, dan hari.
- **Feature Split** membantu untuk memisahkan satu fitur menjadi beberapa bagian yang lebih bermakna bagi model.

Semua teknik ini bertujuan untuk meningkatkan kualitas dan prediktabilitas model dengan mengubah atau memanipulasi data agar lebih cocok dengan algoritma yang digunakan.

PCA

PCA (Principal Component Analysis) dalam Feature Engineering

Principal Component Analysis (PCA) adalah teknik yang digunakan dalam **Feature Engineering** untuk mengurangi dimensi data. **Tujuannya** adalah untuk menyederhanakan data tanpa kehilangan terlalu banyak informasi penting. PCA memungkinkan kita untuk mengubah sekumpulan fitur asli menjadi fitur baru yang lebih kecil, tetapi tetap mempertahankan sebagian besar variabilitas data. Dalam konteks analisis data, PCA sangat berguna ketika kita memiliki banyak fitur yang saling berkorelasi dan ingin mereduksi kompleksitasnya.

PCA sering digunakan dalam konteks **reduksi dimensi** untuk mencegah **overfitting** dan **mempermudah analisis data**.

Konsep Dasar PCA:

PCA adalah teknik matematika yang bekerja dengan mengubah data asli (yang mungkin memiliki banyak fitur) menjadi kumpulan fitur yang lebih kecil, yang disebut **komponen utama (principal components)**. Komponen utama ini adalah kombinasi linier dari fitur asli, dan mereka diurutkan berdasarkan seberapa banyak **variansi** data yang mereka jelaskan.

- **Komponen pertama (PC1)** adalah kombinasi dari fitur asli yang menjelaskan variansi terbesar dalam data.
- **Komponen kedua (PC2)** adalah kombinasi linier dari fitur yang menjelaskan variansi terbesar kedua, dan seterusnya.

Secara visual, PCA berusaha mencari **garis atau bidang** di ruang data yang dapat mewakili sebagian besar informasi dalam data, namun dengan dimensi yang lebih rendah.

Langkah-langkah dalam PCA:

1. **Menghitung Matriks Covariance:** Langkah pertama dalam PCA adalah menghitung matriks covariance dari data. Matriks covariance ini menunjukkan bagaimana fitur-fitur dalam data saling berkorelasi satu sama lain. Jika dua fitur memiliki covariance yang tinggi, itu berarti mereka saling berhubungan secara kuat.

Matriks covariance dihitung dengan rumus berikut:

$$\text{Cov}(X,Y) = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})$$

Di mana X_i dan Y_i adalah nilai dari dua fitur yang berbeda, dan $\overline{X}$ dan $\overline{Y}$ adalah rata-rata dari fitur tersebut.

2. **Menghitung Vektor dan Nilai Eigen:** Setelah mendapatkan matriks covariance, langkah selanjutnya adalah mencari **vektor eigen (eigenvectors)** dan **nilai eigen (eigenvalues)**. Vektor eigen menunjukkan arah baru (komponen utama), sementara nilai eigen mengindikasikan seberapa penting arah tersebut dalam menjelaskan variansi data.

Vektor eigen adalah arah yang menunjukkan "arah" baru dalam ruang data, sementara nilai eigen menggambarkan seberapa banyak informasi (variansi) yang terkandung dalam arah tersebut.

3. **Memilih Komponen Utama:** Berdasarkan nilai eigen, kita memilih komponen utama yang memiliki nilai eigen terbesar, karena komponen ini menjelaskan variansi terbesar dalam data. Komponen utama pertama (PC1) adalah yang memiliki nilai eigen terbesar, yang berarti dia menjelaskan sebagian besar variasi dalam data. Komponen kedua (PC2) menjelaskan sebagian besar variasi yang tersisa, dan seterusnya.

Proses ini membantu kita untuk menentukan berapa banyak komponen utama yang perlu dipertahankan untuk menjaga informasi penting dari data. Biasanya, kita memilih komponen utama dengan nilai eigen terbesar sampai mencapai **jumlah variansi yang diinginkan** (misalnya, 90% atau 95%).

4. **Transformasi Data:** Setelah memilih komponen utama, kita dapat mentransformasikan data asli ke ruang yang baru menggunakan vektor eigen. Setiap data akan diproyeksikan ke komponen utama yang telah dipilih, menghasilkan dataset dengan dimensi yang lebih rendah.

Proses ini mengubah data menjadi representasi yang lebih sederhana, dengan lebih sedikit fitur namun tetap mempertahankan banyak informasi penting.

Manfaat PCA dalam Feature Engineering:

1. **Mengurangi Dimensi:** Salah satu manfaat utama PCA adalah kemampuannya untuk mereduksi dimensi dataset. Ini sangat berguna ketika kita bekerja dengan dataset yang memiliki banyak fitur (dimensi tinggi), yang bisa menyebabkan masalah seperti **overfitting** dan **kompleksitas model** yang tinggi.
2. **Mengurangi Kolinearitas:** Dalam banyak kasus, fitur dalam data bisa sangat berkorelasi satu sama lain (kolinearitas). Kolineritas dapat menyebabkan masalah dalam model statistik dan membuat interpretasi model menjadi sulit. PCA mengurangi kolinearitas dengan menciptakan komponen baru yang tidak saling berkorelasi.
3. **Meningkatkan Kinerja Model:** Dengan mengurangi dimensi data dan menghilangkan noise yang tidak relevan, PCA dapat membantu meningkatkan kinerja model, terutama model yang sensitif terhadap dimensi data yang tinggi, seperti **k-Nearest Neighbors (k-NN)** dan **SVM**.
4. **Visualisasi:** PCA memungkinkan kita untuk mereduksi dimensi data menjadi dua atau tiga komponen utama, yang kemudian dapat divisualisasikan dalam bentuk plot untuk memahami pola dalam data. Ini sangat berguna dalam analisis eksplorasi data (EDA).

Contoh Penggunaan PCA dalam Feature Engineering:

Misalkan kita memiliki dataset dengan banyak fitur, seperti data keuangan dengan ratusan kolom (misalnya pendapatan, pengeluaran, laba, utang, dll.). PCA dapat digunakan untuk mereduksi dimensi data tersebut dengan mengubah fitur asli menjadi beberapa komponen utama.

- **Langkah 1**: Hitung matriks covariance untuk melihat bagaimana fitur-fitur tersebut berhubungan satu sama lain.
- **Langkah 2**: Temukan vektor eigen dan nilai eigen untuk mengetahui arah mana yang menjelaskan variasi terbesar dalam data.
- **Langkah 3**: Pilih komponen utama berdasarkan nilai eigen terbesar.
- **Langkah 4**: Transformasikan data ke ruang baru dengan dimensi yang lebih rendah.

Setelah menggunakan PCA, kita mungkin hanya perlu beberapa komponen utama untuk menggambarkan sebagian besar variasi dalam data, sehingga model yang dilatih pada data yang sudah direduksi dimensi ini menjadi lebih cepat dan lebih mudah diinterpretasikan.

Kelebihan dan Kekurangan PCA:

Kelebihan:

1. **Mengurangi Overfitting**: Dengan mereduksi dimensi, PCA dapat membantu mengurangi overfitting, terutama ketika dataset memiliki banyak fitur.
2. **Menangani Multikolinearitas**: PCA membantu mengatasi multikolinearitas dengan membuat komponen yang tidak saling berkorelasi.
3. **Meningkatkan Kecepatan dan Efisiensi Model**: Dengan mengurangi jumlah fitur yang perlu diproses, model akan lebih cepat dalam pelatihan dan prediksi.
4. **Visualisasi yang Lebih Baik**: PCA memungkinkan kita untuk mereduksi dimensi menjadi dua atau tiga komponen utama yang dapat divisualisasikan dengan lebih mudah.

Kekurangan:

1. **Kehilangan Interpretabilitas**: Komponen utama yang dihasilkan oleh PCA adalah kombinasi linier dari fitur asli, sehingga mereka seringkali sulit untuk diinterpretasikan secara langsung.
2. **Tidak Selalu Membantu dalam Semua Kasus**: Jika data sudah cukup sederhana atau tidak memiliki banyak fitur yang saling berkorelasi, PCA mungkin tidak memberikan keuntungan yang besar.
3. **Bergantung pada Asumsi Linearitas**: PCA mengasumsikan hubungan linear antar fitur. Jadi, jika data memiliki hubungan non-linear, PCA mungkin tidak efektif.

Kesimpulan:

PCA adalah alat yang sangat berguna dalam feature engineering, terutama untuk mengurangi dimensi data yang besar, menghilangkan kolinearitas, dan mempercepat pelatihan model. Dengan menggunakan PCA, kita dapat mempertahankan informasi penting dari data namun dengan jumlah fitur yang lebih sedikit, sehingga lebih efisien dan mudah dikelola. Namun, kita perlu hati-hati dalam memilih komponen utama yang tepat dan menyadari bahwa interpretabilitas model bisa berkurang karena komponen utama merupakan kombinasi dari banyak fitur asli.

KASUS SOAL JAWABAN

Kasus Harga Rumah

Kasus 1: Imputasi dan Data Cleansing pada Data Harga Rumah

Dataset ini berisi informasi mengenai harga rumah yang dijual, lokasi, ukuran rumah, jumlah kamar tidur, dan fitur lainnya. Beberapa data dalam dataset ini hilang atau memiliki nilai yang tidak valid, yang dapat memengaruhi hasil analisis.

Soal 1: Imputasi untuk Missing Values

Dalam dataset harga rumah, beberapa kolom memiliki nilai yang hilang, seperti kolom "jumlah kamar tidur" dan "ukuran rumah". Jelaskan bagaimana Anda akan menangani missing values dalam dataset ini. Sebutkan metode imputasi yang bisa digunakan dan alasan Anda memilih metode tersebut.

Jawaban: Untuk menangani missing values pada kolom "jumlah kamar tidur" dan "ukuran rumah", kita bisa menggunakan beberapa metode imputasi, tergantung pada jenis data dan distribusi nilai yang hilang. Beberapa metode imputasi yang bisa dipertimbangkan adalah:

1. **Imputasi dengan Rata-rata (Mean) atau Median**: Untuk kolom **"jumlah kamar tidur"** yang merupakan data numerik, kita bisa mengisi nilai yang hilang dengan rata-rata (mean) atau median dari kolom tersebut. Penggunaan rata-rata cocok jika data terdistribusi normal, tetapi jika data memiliki distribusi yang miring atau ada beberapa nilai ekstrim (outliers), lebih baik menggunakan median untuk menghindari bias.

2. **Imputasi berdasarkan Prediksi (Model-based Imputation)**: Jika kita ingin menggunakan metode yang lebih canggih, kita bisa menggunakan algoritma seperti regresi atau Random Forest untuk memprediksi nilai yang hilang berdasarkan fitur lainnya (misalnya, harga rumah dan lokasi).

Metode ini akan lebih akurat jika banyak fitur terkait yang dapat membantu memprediksi nilai yang hilang.

3. **Penghapusan Data yang Hilang**: Jika jumlah data yang hilang tidak banyak (misalnya hanya 5-10% dari dataset), kita bisa memilih untuk menghapus baris yang mengandung nilai yang hilang. Namun, penghapusan data sebaiknya dilakukan hanya jika kehilangan data tidak akan mempengaruhi analisis secara signifikan.

Soal 2: Mengatasi Duplikasi dan Data Tidak Valid

Dalam dataset ini terdapat beberapa entri duplikat dan beberapa nilai yang tampaknya tidak valid, seperti harga rumah yang lebih rendah dari biaya konstruksi atau ukuran rumah yang sangat kecil. Bagaimana cara Anda menangani data duplikat dan nilai yang tidak valid?

Jawaban:

1. **Mengatasi Data Duplikat**: Data duplikat harus diidentifikasi dan dihapus agar tidak memengaruhi hasil analisis. Misalnya, jika terdapat dua entri yang sama persis (harga, lokasi, jumlah kamar tidur, dll.), kita dapat menghapus salah satunya. Jika terdapat duplikat dengan sedikit perbedaan, kita bisa memutuskan untuk menggabungkannya dengan menggunakan nilai rata-rata atau modus, tergantung pada jenis data yang ada.

2. **Menangani Data Tidak Valid**:

- Untuk harga rumah yang tidak realistis (misalnya harga yang lebih rendah dari biaya konstruksi), kita bisa menggunakan teknik **Winsorization** atau **Clipping**. Dalam hal ini, kita menetapkan batas bawah yang wajar untuk harga rumah dan memotong nilai yang lebih rendah dari batas tersebut.
- Untuk ukuran rumah yang sangat kecil atau tidak wajar (misalnya, rumah dengan ukuran kurang dari 10 meter persegi), kita juga bisa memotong atau mengganti nilai tersebut dengan nilai yang lebih realistis menggunakan teknik imputasi berbasis nilai rata-rata atau menggunakan nilai berdasarkan lokasi dan tipe rumah.

Soal 3: Mengatasi Outliers

Terdapat beberapa rumah dengan harga yang jauh lebih tinggi atau lebih rendah dibandingkan dengan mayoritas harga rumah di dataset ini. Bagaimana Anda akan menangani outliers dalam dataset harga rumah ini agar tidak memengaruhi model secara negatif?

Jawaban: Untuk menangani outliers dalam dataset harga rumah, ada beberapa metode yang bisa digunakan:

1. **Winsorization**: Jika terdapat harga rumah yang sangat tinggi atau sangat rendah dibandingkan mayoritas data, kita dapat menggunakan Winsorization untuk mengganti nilai ekstrem tersebut dengan nilai batas tertentu. Misalnya, harga rumah yang lebih tinggi dari persentil ke-95 atau lebih rendah dari persentil ke-5 dapat diganti dengan nilai pada persentil tersebut.

2. **Clipping**: Clipping adalah metode yang serupa dengan Winsorization, namun lebih sederhana. Dalam clipping, kita akan memotong nilai yang berada di luar rentang yang wajar (misalnya, harga rumah lebih rendah dari 50 juta atau lebih tinggi dari 10 miliar) dan menggantinya dengan batas atas atau bawah yang telah ditentukan.

3. **Transformasi Data**: Untuk mengurangi dampak outliers, kita bisa menerapkan transformasi non-linear seperti **log transformation** pada kolom harga rumah. Transformasi ini akan membantu mengurangi pengaruh outliers dengan merubah distribusi data agar lebih mendekati distribusi normal.

4. **Identifikasi Outliers**: Outliers bisa diidentifikasi menggunakan metode statistik seperti **z-score** (nilai yang lebih besar dari 3 atau kurang dari -3 dianggap outlier) atau **IQR (Interquartile Range)**. Setelah mengidentifikasi outliers, kita bisa memutuskan apakah akan menghapus, mengganti, atau mentransformasikan data outlier tersebut.

Ringkasan Kasus 1:

1. **Imputasi Missing Values**: Gunakan rata-rata, median, atau model untuk mengisi nilai yang hilang tergantung pada jenis data.
2. **Mengatasi Duplikasi dan Data Tidak Valid**: Hapus duplikat atau gabungkan data yang serupa, dan gunakan teknik seperti Winsorization atau clipping untuk nilai yang tidak valid.
3. **Mengatasi Outliers**: Gunakan Winsorization, clipping, atau transformasi data (seperti log transformation) untuk mengurangi dampak outliers.

Kasus 2: Binning dan Encoding pada Data Harga Rumah

Dataset ini berisi informasi harga rumah yang dijual, lokasi, ukuran rumah, dan jumlah kamar tidur. Tugas Anda adalah melakukan binning pada beberapa fitur untuk mengubah data numerik menjadi kategori, serta mengaplikasikan teknik encoding untuk fitur kategorikal yang ada dalam dataset ini.

Soal 1: Binning pada Harga Rumah

Dalam dataset harga rumah, harga rumah dicatat sebagai nilai numerik. Namun, Anda ingin mengelompokkan harga rumah ke dalam beberapa kategori, seperti “Mahal”, “Sedang”, dan “Murah”, agar lebih mudah untuk analisis dan pemodelan.

Jelaskan apa yang dimaksud dengan **binning**, bagaimana Anda akan mengelompokkan harga rumah dalam dataset ini, dan metode apa yang akan Anda gunakan untuk menentukan batasan setiap kategori harga rumah.

Jawaban: Binning adalah proses mengubah data numerik menjadi kategori dengan membaginya ke dalam beberapa interval atau kelompok. Dalam hal ini, kita ingin mengubah harga rumah yang berupa nilai kontinu menjadi beberapa kategori, seperti “Mahal”, “Sedang”, dan “Murah”.

Langkah-langkah dalam proses binning harga rumah:

1. **Menentukan Interval atau Batas**: Untuk menentukan kategori harga rumah, kita harus menentukan batasan untuk masing-masing kategori. Misalnya, kita dapat menggunakan **persentil** (seperti persentil ke-25, ke-50, dan ke-75) untuk membagi harga rumah menjadi tiga kategori.

- **“Murah”**: Rumah dengan harga lebih rendah dari persentil ke-25.
- **“Sedang”**: Rumah dengan harga antara persentil ke-25 dan ke-75.
- **“Mahal”**: Rumah dengan harga lebih tinggi dari persentil ke-75.

2. **Metode Binning**:
    - **Equal Width Binning**: Membagi rentang harga rumah menjadi beberapa interval dengan lebar yang sama. Misalnya, kita bisa membagi rentang harga dari yang terendah hingga tertinggi menjadi tiga bagian yang sama.
    - **Equal Frequency Binning**: Membagi harga rumah menjadi beberapa kategori berdasarkan distribusi frekuensinya, sehingga setiap kategori mengandung jumlah data yang sama.
    - **Custom Binning**: Menggunakan pemahaman domain atau informasi eksternal untuk menentukan batas kategori yang lebih sesuai dengan konteks analisis harga rumah.
3. **Penerapan Binning**: Setelah menentukan kategori, harga rumah dapat dipetakan ke dalam kategori yang sesuai, dan kategori ini akan digunakan sebagai fitur baru dalam model analisis atau prediksi.

Soal 2: Encoding pada Fitur Kategorikal

Dalam dataset harga rumah, terdapat beberapa fitur kategorikal seperti “lokasi” dan “jenis rumah” (misalnya rumah tunggal, apartemen, dll.). Jelaskan bagaimana Anda akan mengaplikasikan teknik **encoding** pada fitur kategorikal tersebut dan pilih metode encoding yang sesuai.

Jawaban: Encoding adalah proses mengubah fitur kategorikal (misalnya, lokasi atau jenis rumah) menjadi bentuk numerik yang dapat diproses oleh algoritma machine learning. Berikut adalah beberapa metode encoding yang dapat diterapkan pada fitur kategorikal dalam dataset harga rumah:

1. **One-Hot Encoding**:

- **Penjelasan**: One-Hot Encoding adalah teknik untuk mengubah fitur kategorikal menjadi kolom biner (0 atau 1) untuk setiap kategori yang ada. Jika terdapat kategori "lokasi" yang memiliki nilai seperti "Jakarta", "Bandung", dan "Surabaya", kita akan membuat tiga kolom baru yang masing-masing menunjukkan apakah rumah berada di Jakarta, Bandung, atau Surabaya dengan nilai 0 atau 1.
- **Kapan Digunakan**: Teknik ini cocok digunakan jika fitur kategorikal tidak memiliki urutan atau hubungan numerik yang signifikan antara kategori. Misalnya, kolom "lokasi" yang hanya menunjukkan lokasi geografis yang berbeda.

2. **Label Encoding**:

- **Penjelasan**: Label Encoding adalah teknik mengubah kategori menjadi angka yang mewakili urutan kategori. Misalnya, jika kategori "jenis rumah" terdiri dari "Rumah Tunggal", "Apartemen", dan "Rumah Tepi Laut", kita dapat mengkodekan mereka sebagai 0, 1, dan 2. Ini berguna jika kategori memiliki hubungan urutan atau tingkat tertentu (misalnya, ukuran rumah dari kecil ke besar).
- **Kapan Digunakan**: Teknik ini cocok untuk fitur yang memiliki urutan logis antara kategori, seperti "tingkat kepuasan" yang dapat diurutkan dari "Rendah" (0) hingga "Tinggi" (2).

3. **Target Encoding (Mean Encoding)**:

- **Penjelasan**: Target Encoding mengubah kategori menjadi nilai rata-rata dari target (misalnya, harga rumah) untuk setiap kategori. Misalnya, untuk setiap lokasi, kita bisa mengganti lokasi tersebut dengan rata-rata harga rumah di lokasi tersebut. Teknik ini sering digunakan ketika ada banyak kategori dan ingin mengurangi dimensi, serta memberikan informasi lebih lanjut yang relevan dengan target.
- **Kapan Digunakan**: Target Encoding digunakan ketika kita ingin mengencode kategori dengan mempertimbangkan hubungan antara fitur dan target. Teknik ini berguna untuk fitur yang memiliki banyak kategori dan kita ingin mengoptimalkan model prediktif.

4. **Binary Encoding**:

- **Penjelasan**: Binary Encoding adalah kombinasi dari Label Encoding dan One-Hot Encoding, yang mengkodekan kategori menjadi representasi biner. Teknik ini bisa berguna ketika jumlah kategori terlalu banyak untuk menggunakan One-Hot Encoding karena dapat menciptakan banyak kolom.
- **Kapan Digunakan**: Teknik ini digunakan saat kita memiliki banyak kategori dan tidak ingin menggunakan One-Hot Encoding yang berpotensi menyebabkan masalah **curse of dimensionality**.

Soal 3: Kapan Harus Menggunakan Binning dan Encoding?

Dalam konteks analisis harga rumah, jelaskan kapan sebaiknya Anda menggunakan teknik **binning** dan **encoding** pada fitur dalam dataset.

Jawaban:

1. **Penggunaan Binning**:

- Binning berguna ketika kita ingin mengubah data numerik menjadi kategori untuk mempermudah interpretasi atau analisis lebih lanjut. Misalnya, jika kita ingin mengklasifikasikan harga rumah ke dalam kategori seperti “Mahal”, “Sedang”, dan “Murah”, maka kita akan menggunakan binning pada kolom harga rumah.
- Binning juga berguna untuk mengurangi dampak **outliers** dengan membatasi rentang nilai yang terlalu ekstrim, sehingga model yang dibangun lebih stabil dan tidak terlalu dipengaruhi oleh data ekstrem.

2. **Penggunaan Encoding**:

- Encoding digunakan untuk mengubah fitur kategorikal menjadi bentuk numerik agar dapat dimasukkan ke dalam model machine learning. Misalnya, untuk fitur “lokasi” yang berisi kategori seperti “Jakarta”, “Bandung”, dan “Surabaya”, kita harus menggunakan encoding untuk mengubahnya menjadi representasi numerik, seperti menggunakan One-Hot Encoding atau Label Encoding.

- Encoding juga diperlukan saat kita ingin memanfaatkan hubungan antara fitur kategorikal dan target (misalnya, dengan menggunakan Target Encoding) atau ketika fitur kategorikal memiliki banyak kategori yang perlu dipertimbangkan dengan cara yang lebih efisien.

Ringkasan Kasus 2:

- **Binning** digunakan untuk mengelompokkan data numerik ke dalam kategori, seperti mengelompokkan harga rumah menjadi "Mahal", "Sedang", dan "Murah".
- **Encoding** digunakan untuk mengubah fitur kategorikal menjadi representasi numerik, menggunakan teknik seperti One-Hot Encoding, Label Encoding, atau Target Encoding.

Kasus 3: Feature Selection dan Transformation pada Data Harga Rumah

Dataset ini berisi informasi tentang harga rumah yang dijual, lokasi, ukuran rumah, jumlah kamar tidur, usia rumah, dan fitur lainnya. Tugas Anda adalah memilih fitur yang paling relevan untuk memprediksi harga rumah dan melakukan transformasi pada fitur yang dibutuhkan untuk meningkatkan kualitas model prediksi.

Soal 1: Feature Selection pada Harga Rumah

Dalam dataset harga rumah, terdapat banyak fitur yang mungkin tidak semuanya berkontribusi signifikan terhadap model prediksi harga rumah. Jelaskan apa yang dimaksud dengan **Feature Selection**, bagaimana Anda akan melakukan seleksi fitur pada dataset harga rumah ini, dan metode apa yang akan Anda gunakan untuk memilih fitur yang paling relevan.

Jawaban: Feature Selection adalah proses memilih subset fitur yang paling relevan dan berkontribusi besar terhadap prediksi model, sambil menghapus fitur-fitur yang tidak relevan atau redundan. Hal ini dilakukan untuk meningkatkan kinerja model dan mengurangi **overfitting**.

Langkah-langkah dalam proses feature selection untuk harga rumah:

1. **Memahami Fitur yang Ada**: Sebelum melakukan seleksi fitur, kita perlu memahami hubungan antara fitur dan target (harga rumah). Fitur seperti lokasi, ukuran rumah, dan jumlah kamar tidur kemungkinan besar memiliki hubungan langsung dengan harga rumah. Namun, fitur seperti ID rumah atau alamat mungkin tidak memberikan kontribusi yang signifikan terhadap prediksi.
2. **Metode Feature Selection**:
    - **Filter Method**: Menggunakan statistik untuk menilai relevansi fitur terhadap target. Misalnya, menggunakan **correlation matrix** untuk melihat korelasi antara fitur numerik (seperti ukuran rumah atau usia rumah) dengan harga rumah. Fitur dengan korelasi rendah terhadap harga rumah dapat dihapus.
    - **Wrapper Method**: Menggunakan algoritma machine learning untuk memilih subset fitur yang optimal dengan mengevaluasi kinerja model berdasarkan kombinasi fitur. Teknik seperti **Recursive Feature Elimination (RFE)** bisa digunakan untuk memilih fitur terbaik.
    - **Embedded Method**: Menggunakan metode yang secara langsung memilih fitur selama pelatihan model. Contohnya, menggunakan **decision tree** atau **Lasso Regression**, di mana model akan memberikan bobot atau koefisien untuk setiap fitur, dan fitur dengan koefisien sangat kecil bisa diabaikan.
3. **Penerapan**: Setelah menggunakan metode feature selection, kita akan mendapatkan subset fitur yang relevan (misalnya ukuran rumah, lokasi, dan jumlah kamar tidur) yang dapat digunakan dalam model untuk memprediksi harga rumah.

Soal 2: Feature Transformation pada Harga Rumah

Setelah melakukan seleksi fitur, beberapa fitur mungkin perlu ditransformasi untuk meningkatkan kualitas model atau untuk memudahkan pemodelan. Jelaskan apa yang dimaksud dengan **Feature Transformation**, bagaimana Anda akan melakukan transformasi pada fitur dalam dataset harga rumah, dan berikan contoh teknik transformasi yang dapat diterapkan.

Jawaban: Feature Transformation adalah proses mengubah fitur yang ada menjadi bentuk yang lebih sesuai untuk digunakan dalam model. Transformasi ini bertujuan untuk meningkatkan kualitas data dan membuat model lebih efisien dalam mempelajari pola dari data.

Langkah-langkah dalam proses feature transformation untuk harga rumah:

1. **Mengidentifikasi Fitur yang Perlu Ditrasnformasi**: Beberapa fitur dalam dataset harga rumah mungkin memerlukan transformasi karena distribusi mereka yang tidak normal atau karena mereka mempengaruhi performa model. Contoh fitur yang sering memerlukan transformasi adalah harga rumah (karena sering memiliki distribusi yang sangat miring) atau ukuran rumah (dalam beberapa kasus).
2. **Metode Feature Transformation**:
    - **Log Transformation**: Digunakan untuk fitur yang memiliki distribusi yang sangat miring atau skewed, seperti harga rumah. Misalnya, jika harga rumah memiliki nilai ekstrim atau distribusi yang tidak normal, kita bisa menerapkan logaritma pada harga rumah untuk membuat distribusinya lebih mendekati normal.
    - **Square Root Transformation**: Digunakan untuk mengurangi efek outlier pada fitur numerik yang sangat besar, seperti ukuran rumah yang sangat besar. Dengan mengaplikasikan akar kuadrat pada fitur ini, nilai-nilai besar akan lebih tereduksi dan tidak mendominasi model.
    - **Normalization/Standardization**: Digunakan untuk fitur yang memiliki skala yang berbeda. Misalnya, ukuran rumah dalam meter persegi dan harga rumah dalam juta bisa dinormalisasi untuk memastikan bahwa kedua fitur berada dalam skala yang sama, menghindari fitur dengan skala yang lebih besar mendominasi model. **Standardization** (z-score normalization) sering digunakan untuk membuat distribusi fitur memiliki rata-rata 0 dan deviasi standar 1.
    - **One-Hot Encoding**: Untuk fitur kategorikal seperti lokasi atau jenis rumah, kita dapat mengaplikasikan **one-hot encoding** untuk mengubahnya menjadi format numerik yang bisa diproses oleh model machine learning. Setiap kategori akan menjadi kolom

terpisah dengan nilai 1 atau 0, yang menunjukkan apakah rumah tersebut berada di lokasi atau memiliki jenis rumah tertentu.

3. **Penerapan**: Setelah menerapkan transformasi pada fitur yang diperlukan, kita akan mendapatkan fitur yang lebih siap digunakan untuk pemodelan. Misalnya, harga rumah yang sudah tertransformasi dengan logaritma atau ukuran rumah yang telah dinormalisasi akan membantu model lebih mudah dalam belajar dan meningkatkan akurasi prediksi.

Soal 3: Kapan Harus Menggunakan Feature Selection dan Transformation?

Jelaskan kapan sebaiknya Anda menggunakan **Feature Selection** dan **Feature Transformation** pada dataset harga rumah. Apa alasan di balik pemilihan teknik-teknik tersebut?

Jawaban:

1. **Feature Selection**:
    - **Kapan Digunakan**: Feature selection digunakan ketika kita ingin mengurangi jumlah fitur yang digunakan dalam model untuk mencegah **overfitting** dan meningkatkan **efisiensi** model. Teknik ini diterapkan ketika ada banyak fitur dalam dataset yang mungkin tidak semuanya relevan atau berguna untuk memprediksi harga rumah.
    - **Alasan Penggunaan**: Misalnya, fitur seperti ID rumah atau alamat tidak memberikan informasi yang cukup untuk mempengaruhi harga rumah, sehingga fitur-fitur tersebut bisa dihapus. Selain itu, memilih fitur yang paling relevan akan mempercepat proses pelatihan model dan meningkatkan interpretasi hasil.
2. **Feature Transformation**:
    - **Kapan Digunakan**: Feature transformation diterapkan ketika kita menemukan bahwa beberapa fitur memiliki distribusi yang tidak normal, memiliki skala yang berbeda, atau membutuhkan representasi numerik agar bisa dimasukkan ke dalam model machine learning.
    - **Alasan Penggunaan**: Misalnya, harga rumah sering kali memiliki distribusi yang sangat miring, yang bisa mempengaruhi kinerja model, sehingga transformasi seperti logaritma bisa digunakan. Selain itu, untuk fitur kategorikal, kita perlu melakukan encoding agar model dapat memahami dan mengolah fitur-fitur tersebut dengan benar.

Ringkasan Kasus 3:

- **Feature Selection** digunakan untuk memilih subset fitur yang paling relevan dengan target dan mengurangi risiko overfitting.
- **Feature Transformation** digunakan untuk mengubah fitur menjadi bentuk yang lebih sesuai untuk model, seperti transformasi logaritma untuk mengatasi distribusi miring atau encoding untuk fitur kategorikal.

Kasus 4: Feature Creation dan Augmentation pada Data Harga Rumah

Dataset ini berisi informasi tentang harga rumah yang dijual, lokasi, ukuran rumah, jumlah kamar tidur, usia rumah, dan beberapa fitur lainnya. Tugas Anda adalah menciptakan fitur baru yang dapat membantu model dalam memprediksi harga rumah secara lebih akurat dan menambah informasi tambahan yang relevan untuk meningkatkan kinerja model.

Soal 1: Feature Creation pada Harga Rumah

Apa yang dimaksud dengan **Feature Creation**, dan bagaimana Anda akan menciptakan fitur baru yang relevan dalam dataset harga rumah untuk membantu prediksi harga rumah lebih akurat? Berikan contoh fitur baru yang dapat Anda buat berdasarkan informasi yang ada.

Jawaban: Feature Creation adalah proses menciptakan fitur baru yang tidak ada dalam dataset asli tetapi dapat membantu meningkatkan prediksi model dengan menggabungkan atau mengolah informasi dari fitur yang sudah ada.

Langkah-langkah dalam proses feature creation untuk harga rumah:

1. **Memahami Fitur yang Ada**: Dalam dataset harga rumah, beberapa fitur seperti lokasi, ukuran rumah, dan jumlah kamar tidur mungkin sudah cukup informatif. Namun, kita bisa menciptakan fitur baru yang mungkin lebih bermakna atau memiliki hubungan yang lebih kuat dengan harga rumah.
2. **Contoh Feature Creation**:
    - **Harga per Meter Persegi**: Fitur ini dibuat dengan membagi harga rumah dengan ukuran rumah (dalam meter persegi). Harga per meter persegi bisa menjadi indikator penting tentang harga rumah yang lebih proporsional berdasarkan ukuran rumah. Ini dapat membantu model untuk memahami harga rumah dalam konteks ukuran yang lebih adil.
    - **Usia Rumah Terhadap Tahun**: Fitur ini bisa dibuat dengan mengurangi tahun pembuatan rumah dengan tahun saat ini, untuk mendapatkan usia rumah. Usia rumah sering menjadi faktor penting dalam menentukan harga rumah, di mana rumah yang lebih tua atau lebih baru biasanya memiliki harga yang berbeda.
    - **Fitur Lokasi Tertentu**: Jika ada informasi tentang lokasi, kita bisa menciptakan fitur baru berdasarkan kategori lokasi tertentu, misalnya "dekat sekolah" atau "dekat pusat perbelanjaan". Lokasi adalah faktor penting dalam menentukan harga rumah, dan mengkategorikannya bisa memberikan informasi tambahan yang membantu model.
3. **Alasan Penggunaan Feature Creation**: Dengan fitur-fitur baru ini, model prediksi harga rumah bisa lebih mendalam dalam memahami pola yang ada, seperti harga rumah yang lebih tinggi untuk rumah baru atau di lokasi yang strategis, dan harga yang lebih rendah per meter persegi untuk rumah dengan ukuran sangat besar.

Soal 2: Feature Augmentation pada Harga Rumah

Jelaskan apa yang dimaksud dengan **Feature Augmentation** dan bagaimana Anda akan menambahkan data eksternal atau informasi tambahan untuk meningkatkan model prediksi harga rumah. Sebutkan beberapa contoh data eksternal atau informasi tambahan yang dapat digunakan untuk augmentasi fitur.

Jawaban: Feature Augmentation adalah proses menambahkan data atau informasi eksternal yang relevan ke dalam dataset untuk meningkatkan kualitas model dan prediksi. Augmentasi fitur membantu model dengan memberikan lebih banyak konteks atau variabilitas dalam data yang dapat digunakan untuk memperbaiki akurasi prediksi.

Langkah-langkah dalam proses feature augmentation untuk harga rumah:

1. **Memahami Kebutuhan Fitur Eksternal**: Kadang-kadang, data yang ada dalam dataset tidak cukup untuk memodelkan variabel target (misalnya, harga rumah). Oleh karena itu, data eksternal yang relevan perlu ditambahkan untuk memberikan informasi lebih lanjut yang mungkin mempengaruhi harga rumah.

2. **Contoh Feature Augmentation**:

    - **Data Eksternal tentang Perkembangan Ekonomi**: Data tentang perkembangan ekonomi di wilayah tertentu, seperti **pendapatan rata-rata** atau tingkat pengangguran, bisa ditambahkan. Kondisi ekonomi di suatu wilayah sering kali berdampak langsung pada harga rumah. Misalnya, di wilayah dengan tingkat pendapatan tinggi, harga rumah mungkin lebih mahal.
    - **Data Cuaca atau Iklim**: Informasi tentang iklim atau cuaca, seperti jumlah hujan per tahun atau suhu rata-rata di wilayah tempat rumah berada, dapat mempengaruhi harga rumah. Rumah di daerah dengan iklim yang lebih stabil atau nyaman mungkin memiliki harga yang lebih tinggi.
    - **Data Infrastruktur dan Aksesibilitas**: Menambahkan informasi mengenai **akses transportasi** atau **infrastruktur terdekat** seperti stasiun kereta, jalan tol, atau bandara juga bisa menjadi fitur yang relevan. Rumah yang terletak dekat dengan akses transportasi yang baik mungkin memiliki harga yang lebih tinggi.

3. **Penerapan Feature Augmentation**: Setelah menambahkan data eksternal yang relevan, dataset harga rumah akan memiliki lebih banyak informasi yang dapat membantu model membuat prediksi yang lebih akurat. Misalnya, menambahkan data pendapatan rata-rata atau aksesibilitas transportasi bisa membuat model lebih cerdas dalam mengidentifikasi harga rumah berdasarkan faktor eksternal.

4. **Keuntungan Feature Augmentation**: Feature augmentation meningkatkan konteks data dan memberikan model lebih banyak perspektif untuk memprediksi harga rumah dengan lebih akurat. Dengan informasi eksternal tambahan, model bisa memprediksi harga rumah berdasarkan faktor yang sebelumnya tidak ada dalam dataset asli.

Soal 3: Perbedaan antara Feature Creation dan Feature Augmentation

Jelaskan perbedaan antara **Feature Creation** dan **Feature Augmentation** dalam konteks dataset harga rumah. Kapan sebaiknya Anda memilih untuk menggunakan feature creation dibandingkan dengan feature augmentation?

Jawaban:

1. **Feature Creation**:
    - **Pengertian**: Feature creation adalah proses menciptakan fitur baru dari fitur yang sudah ada dalam dataset. Fitur baru ini dihasilkan dengan cara memanipulasi atau mengkombinasikan data yang sudah ada.
    - **Contoh dalam Harga Rumah**: Membuat fitur baru seperti "harga per meter persegi" atau "usia rumah" yang dihitung dari informasi yang ada (harga rumah dan ukuran rumah).
    - **Kapan Digunakan**: Feature creation digunakan ketika kita ingin menciptakan fitur baru yang lebih representatif atau lebih bermakna berdasarkan data yang sudah ada. Hal ini sering dilakukan ketika fitur yang ada memiliki potensi untuk memberikan informasi yang lebih baik jika diproses atau dimodifikasi.

2. **Feature Augmentation**:
    - **Pengertian**: Feature augmentation adalah proses menambahkan data eksternal atau informasi tambahan yang tidak ada dalam dataset asli. Data tambahan ini bisa berasal dari sumber eksternal atau data lain yang relevan.
    - **Contoh dalam Harga Rumah**: Menambahkan data eksternal seperti pendapatan rata-rata atau akses transportasi untuk meningkatkan konteks dalam memprediksi harga rumah.
    - **Kapan Digunakan**: Feature augmentation digunakan ketika dataset yang ada dirasa kurang informasi atau tidak lengkap. Ini dilakukan untuk memberikan lebih banyak konteks atau variabilitas pada model agar dapat membuat prediksi yang lebih akurat.

3. **Perbedaan Utama**:
    - **Feature Creation** bekerja dengan data yang sudah ada dan mengubah atau menggabungkan informasi yang sudah tersedia untuk menciptakan fitur baru.
    - **Feature Augmentation** bekerja dengan menambahkan informasi baru dari sumber eksternal yang relevan untuk memberikan lebih banyak konteks kepada model.

Ringkasan Kasus 4:

1. **Feature Creation** menciptakan fitur baru berdasarkan fitur yang sudah ada dalam dataset untuk meningkatkan model. Contoh: membuat fitur "harga per meter persegi" dari harga dan ukuran rumah.
2. **Feature Augmentation** menambahkan informasi eksternal yang relevan untuk meningkatkan akurasi model. Contoh: menambahkan data tentang pendapatan rata-rata atau akses transportasi di wilayah tempat rumah berada.

Kasus Toko Online

Kasus 1: Imputasi dan Data Cleansing pada Dataset Toko Online

Anda diberi dataset transaksi dari sebuah toko online yang mencakup informasi berikut:

- **ID Transaksi**: Nomor unik setiap transaksi.

- **Tanggal Transaksi**: Tanggal ketika transaksi dilakukan.
- **ID Pelanggan**: Nomor unik untuk setiap pelanggan.
- **Jumlah Produk Dibeli**: Total jumlah produk yang dibeli dalam satu transaksi.
- **Total Pembayaran**: Total jumlah uang yang dibayar oleh pelanggan untuk transaksi tersebut.
- **Metode Pembayaran**: Jenis metode pembayaran yang digunakan (misalnya, kartu kredit, transfer bank, e-wallet).

Namun, dataset ini memiliki beberapa masalah, seperti nilai yang hilang, duplikasi data, dan outlier. Anda diminta untuk melakukan data cleansing dan menjelaskan prosesnya secara rinci.

Soal 1: Identifikasi Nilai yang Hilang

Apa yang dimaksud dengan nilai yang hilang dalam dataset? Bagaimana Anda mengidentifikasi nilai yang hilang pada dataset toko online? Jelaskan cara menangani nilai yang hilang tersebut.

Jawaban:

1. **Pengertian Nilai yang Hilang**: Nilai yang hilang adalah data yang tidak tersedia untuk suatu entri pada satu atau lebih kolom dalam dataset. Hal ini bisa terjadi karena kesalahan dalam pencatatan data, kendala teknis, atau kelalaian pengguna.

2. **Cara Identifikasi**:

    - Dalam dataset toko online, nilai yang hilang dapat dikenali dengan adanya entri kosong pada kolom tertentu, seperti ID pelanggan, jumlah produk yang dibeli, atau total pembayaran.
    - Biasanya, nilai yang hilang dilambangkan dengan simbol khusus seperti NaN, NULL, atau sel kosong.

3. **Cara Menangani Nilai yang Hilang**:

    - **Imputasi**: Mengganti nilai yang hilang dengan nilai lain yang masuk akal berdasarkan analisis.

        - **Kolom Tanggal Transaksi**: Jika ada tanggal yang hilang, kita bisa menghapus baris tersebut karena tanggal sangat penting untuk analisis.
        - **Kolom Jumlah Produk Dibeli**: Jika ada nilai yang hilang, menggantinya dengan nilai rata-rata atau median dari jumlah produk yang dibeli.
        - **Kolom Metode Pembayaran**: Jika metode pembayaran hilang, menggantinya dengan modus (metode pembayaran yang paling sering digunakan).

    - **Penghapusan Data**: Jika nilai yang hilang terlalu banyak (>30% dari total data) atau tidak dapat diimputasi, baris atau kolom tersebut bisa dihapus.

Soal 2: Penanganan Data Duplikasi

Apa itu data duplikasi? Bagaimana Anda menangani data duplikasi pada dataset toko online? Berikan contoh kasus duplikasi dalam dataset ini.

Jawaban:

1. **Pengertian Data Duplikasi**: Data duplikasi terjadi ketika ada entri yang sama persis dengan entri lain dalam dataset, baik sebagian maupun seluruh kolomnya. Hal ini dapat menyebabkan analisis menjadi tidak akurat karena data dihitung lebih dari sekali.

2. **Contoh Duplikasi dalam Dataset**:

    - **Duplikasi Penuh**: Dua baris data memiliki ID transaksi, ID pelanggan, jumlah produk dibeli, total pembayaran, dan metode pembayaran yang sama.
    - **Duplikasi Parsial**: Dua baris data memiliki ID transaksi yang sama, tetapi beberapa informasi seperti total pembayaran berbeda.

3. **Cara Menangani**:

    - **Identifikasi Duplikasi**: Mencari duplikasi dengan membandingkan setiap baris data menggunakan seluruh kolom atau kolom kunci (misalnya, ID transaksi).
    - **Menghapus Duplikasi**: Jika duplikasi ditemukan, salah satu baris bisa dihapus karena data tersebut redundant.
    - **Verifikasi Data**: Jika duplikasi parsial ditemukan, lakukan validasi dengan sumber data asli untuk memastikan mana data yang benar.

Soal 3: Identifikasi dan Penanganan Outlier

Apa yang dimaksud dengan outlier? Bagaimana Anda mengidentifikasi dan menangani outlier pada dataset toko online? Berikan contoh outlier pada kolom "Total Pembayaran".

Jawaban:

1. **Pengertian Outlier**: Outlier adalah data yang nilainya sangat berbeda dari data lainnya dalam dataset, baik terlalu tinggi maupun terlalu rendah. Hal ini dapat disebabkan oleh kesalahan pencatatan atau data yang memang unik.

2. **Contoh Outlier pada Kolom Total Pembayaran**:

    - Transaksi dengan total pembayaran sebesar **Rp 0**: Hal ini tidak mungkin terjadi karena setiap pembelian harus memiliki pembayaran.
    - Transaksi dengan total pembayaran sebesar **Rp 1.000.000.000**: Nilai ini jauh lebih tinggi daripada transaksi lainnya.

3. **Cara Identifikasi**:

    - Menggunakan statistik deskriptif seperti **IQR (Interquartile Range)** untuk melihat data yang berada di luar batas normal.
    - Membuat plot seperti boxplot untuk mengidentifikasi nilai-nilai ekstrem.

4. **Cara Menangani**:

- **Verifikasi Data**: Periksa ulang data dengan sumber aslinya untuk memastikan apakah nilai tersebut valid atau salah.
- **Transformasi Data**: Jika outlier valid tetapi terlalu ekstrem, kita dapat menggunakan transformasi logaritmik untuk mengurangi dampaknya.
- **Penghapusan Data**: Jika outlier ternyata hasil kesalahan pencatatan dan tidak dapat diperbaiki, data tersebut bisa dihapus.

Soal 4: Langkah Data Cleansing Secara Keseluruhan

Jelaskan langkah-langkah yang harus dilakukan untuk melakukan data cleansing secara keseluruhan pada dataset toko online.

Jawaban: Langkah-langkah data cleansing pada dataset toko online meliputi:

1. **Identifikasi dan Penanganan Nilai yang Hilang**:
   - Mengganti nilai yang hilang dengan rata-rata, median, atau modus, sesuai kolomnya.
   - Menghapus baris atau kolom dengan nilai yang hilang jika persentase missing value terlalu besar.
2. **Identifikasi dan Penanganan Data Duplikasi**:
   - Mencari baris yang sama persis (duplikasi penuh) atau sebagian (duplikasi parsial).
   - Menghapus duplikasi untuk mencegah penghitungan ganda.
3. **Identifikasi dan Penanganan Outlier**:
   - Mencari nilai-nilai ekstrem menggunakan metode statistik seperti IQR atau z-score.
   - Memutuskan apakah outlier valid atau kesalahan, lalu menanganinya sesuai konteks.
4. **Validasi dan Verifikasi Data**:
   - Memastikan setiap kolom memiliki nilai yang masuk akal dan sesuai dengan konteks.
   - Misalnya, total pembayaran tidak boleh negatif, dan jumlah produk dibeli harus berupa angka bulat positif.
5. **Pemeriksaan Konsistensi Data**:
   - Memastikan konsistensi format (misalnya, format tanggal seragam).
   - Memastikan semua ID transaksi dan ID pelanggan unik.
6. **Penyimpanan Dataset yang Sudah Dibersihkan**:
   - Dataset yang sudah melalui proses data cleansing disimpan dalam file terpisah untuk mencegah hilangnya perubahan yang telah dilakukan.

Soal 5: Pentingnya Data Cleansing dalam Analisis

Mengapa data cleansing sangat penting dalam analisis data? Jelaskan konsekuensi jika data cleansing tidak dilakukan pada dataset toko online.

Jawaban:

1. **Pentingnya Data Cleansing**:
   - Data cleansing memastikan bahwa data yang digunakan bebas dari kesalahan, lengkap, dan dapat dipercaya.
   - Proses ini membantu meningkatkan kualitas analisis dan akurasi model prediksi.
2. **Konsekuensi Jika Tidak Dilakukan**:
   - **Nilai yang Hilang**: Analisis statistik atau algoritma machine learning mungkin gagal jika ada nilai yang hilang dalam dataset.
   - **Data Duplikasi**: Duplikasi data dapat menyebabkan overestimation atau hasil analisis yang bias.
   - **Outlier**: Outlier yang tidak ditangani dapat mendistorsi hasil analisis dan menyebabkan keputusan yang salah.
   - **Konsistensi Format**: Format yang tidak konsisten dapat menyebabkan kesalahan dalam interpretasi data atau kegagalan dalam pemrosesan data otomatis.

Dengan data yang bersih, hasil analisis akan lebih akurat dan dapat diandalkan untuk pengambilan keputusan.

Kasus 2: Binning dan Encoding pada Dataset Toko Online

Anda diberikan dataset toko online yang mencakup informasi berikut:

1. **ID Transaksi**: Nomor unik setiap transaksi.
2. **Tanggal Transaksi**: Tanggal ketika transaksi dilakukan.
3. **ID Pelanggan**: Nomor unik untuk setiap pelanggan.
4. **Usia Pelanggan**: Usia pelanggan dalam tahun.
5. **Kategori Produk**: Kategori dari produk yang dibeli (misalnya, elektronik, pakaian, makanan).
6. **Jumlah Produk Dibeli**: Jumlah total produk dalam satu transaksi.
7. **Total Pembayaran**: Total jumlah uang yang dibayarkan dalam satu transaksi.
8. **Metode Pembayaran**: Metode pembayaran yang digunakan (misalnya, kartu kredit, transfer bank, e-wallet).

Anda diminta untuk melakukan proses binning dan encoding untuk mendukung analisis data lebih lanjut.

Soal 1: Pengertian dan Contoh Binning

a. Apa yang dimaksud dengan binning dalam analisis data?
b. Berikan contoh bagaimana binning dapat diterapkan pada kolom "Usia Pelanggan" dalam dataset toko online.

Jawaban:
a. **Pengertian Binning**:
Binning adalah proses mengelompokkan data numerik menjadi beberapa kategori (bin) untuk

menyederhanakan analisis atau menemukan pola tertentu. Misalnya, usia pelanggan yang memiliki nilai beragam dapat dikelompokkan menjadi kategori seperti "remaja", "dewasa", dan "lanjut usia".

b. **Contoh Binning pada Usia Pelanggan**:
Kolom "Usia Pelanggan" dapat dikelompokkan sebagai berikut:

1. 0-17 tahun → **Remaja**
2. 18-35 tahun → **Dewasa Muda**
3. 36-50 tahun → **Dewasa**
4. 51 tahun ke atas → **Lanjut Usia**

Dengan binning ini, analisis dapat lebih mudah difokuskan pada perilaku pelanggan berdasarkan kelompok usia.

Soal 2: Pendekatan dalam Binning

a. Jelaskan dua pendekatan yang dapat digunakan dalam proses binning.
b. Pilih salah satu pendekatan dan terapkan pada kolom "Total Pembayaran".

Jawaban:
a. **Pendekatan Binning**:

1. **Equal-width Binning (Binning dengan Lebar yang Sama)**:
    - Data numerik dibagi menjadi beberapa kelompok (bin) dengan interval yang sama.
    - Contoh: Jika rentang total pembayaran adalah 0–10.000, dan dibagi menjadi 5 bin, maka interval setiap bin adalah 2.000 (0–2.000, 2.001–4.000, dst.).
2. **Equal-frequency Binning (Binning dengan Frekuensi yang Sama)**:
    - Data dibagi menjadi beberapa kelompok (bin) sehingga setiap kelompok memiliki jumlah data yang sama.
    - Contoh: Dalam dataset dengan 1.000 transaksi, setiap bin berisi 200 transaksi.

b. **Penerapan pada Kolom "Total Pembayaran" dengan Equal-width Binning**:
Misalnya, total pembayaran memiliki rentang dari Rp 0 hingga Rp 10.000.000. Data dapat dikelompokkan menjadi:

1. Rp 0 – Rp 2.500.000 → **Bin 1 (Rendah)**
2. Rp 2.500.001 – Rp 5.000.000 → **Bin 2 (Sedang)**
3. Rp 5.000.001 – Rp 7.500.000 → **Bin 3 (Tinggi)**
4. Rp 7.500.001 – Rp 10.000.000 → **Bin 4 (Sangat Tinggi)**

Soal 3: Pengertian dan Contoh Encoding

a. Apa yang dimaksud dengan encoding dalam analisis data?
b. Berikan contoh bagaimana encoding dapat diterapkan pada kolom "Kategori Produk" dalam dataset toko online.

Jawaban:
a. **Pengertian Encoding**:
Encoding adalah proses mengubah data kategorikal menjadi format numerik agar dapat digunakan dalam analisis atau algoritma machine learning. Proses ini bertujuan untuk membuat data lebih terstruktur dan dapat dihitung secara matematis.

b. **Contoh Encoding pada Kategori Produk**:
Kolom "Kategori Produk" memiliki kategori seperti "Elektronik", "Pakaian", dan "Makanan".

1. **Label Encoding**:
    - Setiap kategori diberikan nilai numerik unik.
    - Elektronik → 0, Pakaian → 1, Makanan → 2.
2. **One-Hot Encoding**:
    - Setiap kategori diubah menjadi kolom biner (0 atau 1).
    - Elektronik → [1, 0, 0], Pakaian → [0, 1, 0], Makanan → [0, 0, 1].

Soal 4: Proses Encoding pada Kolom Metode Pembayaran

a. Bagaimana Anda menentukan metode encoding yang tepat untuk kolom "Metode Pembayaran"?
b. Jelaskan langkah-langkah penerapan encoding untuk kolom tersebut.

Jawaban:
a. **Pemilihan Metode Encoding**:

1. Jika kolom "Metode Pembayaran" hanya digunakan untuk analisis deskriptif atau visualisasi, **label encoding** bisa digunakan.
2. Jika kolom ini digunakan dalam model machine learning, **one-hot encoding** lebih disarankan untuk menghindari asumsi urutan pada nilai numerik.

b. **Langkah-Langkah Encoding**:

1. Identifikasi kategori unik dalam kolom "Metode Pembayaran" (misalnya, kartu kredit, transfer bank, e-wallet).
2. Pilih metode encoding yang sesuai:
    - Untuk label encoding, setiap kategori diberi nilai numerik.
    - Untuk one-hot encoding, buat kolom baru untuk setiap kategori, dengan nilai 1 jika transaksi menggunakan kategori tersebut, dan 0 jika tidak.
3. Simpan hasil encoding dalam dataset yang sudah dimodifikasi.

Soal 5: Manfaat dan Tantangan dalam Binning dan Encoding

a. Apa manfaat binning dan encoding dalam analisis data?
b. Apa tantangan yang dapat muncul saat melakukan binning dan encoding?

Jawaban:
a. **Manfaat**:

1. **Binning**:
    - Menyederhanakan data numerik untuk analisis yang lebih mudah dipahami.
    - Membantu dalam pengelompokan data untuk menemukan pola tertentu (misalnya, usia pelanggan).
2. **Encoding**:
    - Mengubah data kategorikal menjadi format yang dapat digunakan dalam analisis statistik atau algoritma machine learning.
    - Mempermudah interpretasi data secara kuantitatif.

b. **Tantangan**:

1. **Binning**:
    - Rentang bin yang terlalu besar atau kecil dapat mengaburkan pola dalam data.
    - Pemilihan jumlah bin yang optimal membutuhkan pertimbangan khusus.
2. **Encoding**:
    - Label encoding dapat menyebabkan masalah jika algoritma menganggap kategori memiliki hubungan ordinal (urutan).
    - One-hot encoding dapat menyebabkan ledakan dimensi jika jumlah kategori terlalu banyak, sehingga memerlukan lebih banyak memori dan waktu pemrosesan.

Soal 6: Studi Kasus

Dataset toko online menunjukkan bahwa pelanggan dengan total pembayaran tinggi sering menggunakan metode pembayaran tertentu. Bagaimana Anda dapat menghubungkan binning pada "Total Pembayaran" dengan encoding pada "Metode Pembayaran" untuk mendapatkan wawasan yang lebih baik?

Jawaban:

1. Lakukan binning pada kolom "Total Pembayaran" untuk mengelompokkan transaksi berdasarkan kategori seperti rendah, sedang, tinggi, dan sangat tinggi.
2. Lakukan encoding pada kolom "Metode Pembayaran" menggunakan one-hot encoding sehingga setiap metode pembayaran dapat dihitung frekuensinya.
3. Analisis hubungan antara kategori total pembayaran dengan metode pembayaran menggunakan tabel silang (cross-tabulation) atau visualisasi, seperti stacked bar chart.
4. Wawasan yang didapat:
    - Apakah pelanggan dengan total pembayaran tinggi lebih cenderung menggunakan metode tertentu (misalnya, e-wallet)?
    - Informasi ini dapat digunakan untuk strategi pemasaran atau promosi metode pembayaran tertentu.

Kasus 3: Feature Selection dan Transformation pada Dataset Toko Online

Anda diberikan dataset toko online yang mencakup informasi berikut:

1. **ID Transaksi**: Nomor unik setiap transaksi.
2. **Tanggal Transaksi**: Tanggal ketika transaksi dilakukan.
3. **ID Pelanggan**: Nomor unik untuk setiap pelanggan.
4. **Usia Pelanggan**: Usia pelanggan dalam tahun.
5. **Kategori Produk**: Kategori dari produk yang dibeli (misalnya, elektronik, pakaian, makanan).
6. **Jumlah Produk Dibeli**: Jumlah total produk dalam satu transaksi.
7. **Total Pembayaran**: Total jumlah uang yang dibayarkan dalam satu transaksi.
8. **Metode Pembayaran**: Metode pembayaran yang digunakan (misalnya, kartu kredit, transfer bank, e-wallet).
9. **Rating Pelanggan**: Penilaian pelanggan terhadap transaksi (skala 1–5).

Anda diminta untuk melakukan feature selection dan transformation untuk mendukung analisis data lebih lanjut.

Soal 1: Pengertian Feature Selection

a. Apa yang dimaksud dengan feature selection?
b. Mengapa feature selection penting dalam analisis data?

Jawaban:
a. **Pengertian Feature Selection**:
Feature selection adalah proses memilih fitur (variabel) yang paling relevan atau memiliki pengaruh signifikan terhadap analisis atau model prediksi. Tujuan utamanya adalah menyederhanakan dataset tanpa kehilangan informasi penting.

b. **Pentingnya Feature Selection**:

1. **Meningkatkan Efisiensi**: Mengurangi jumlah fitur dapat mempercepat proses analisis atau pelatihan model.
2. **Meningkatkan Akurasi**: Menghilangkan fitur yang tidak relevan atau redundan dapat meningkatkan performa model.
3. **Mempermudah Interpretasi**: Dataset yang lebih sederhana memudahkan pemahaman dan interpretasi hasil analisis.

Soal 2: Pemilihan Fitur yang Relevan

a. Dari dataset yang diberikan, fitur mana yang dianggap kurang relevan untuk analisis perilaku pelanggan? Jelaskan alasannya.
b. Bagaimana Anda memilih fitur yang paling penting untuk menganalisis pengaruh kategori produk terhadap total pembayaran?

Jawaban:
a. **Fitur Kurang Relevan**:

1. **ID Transaksi** dan **ID Pelanggan** dianggap kurang relevan karena tidak memberikan informasi langsung tentang perilaku atau pola pelanggan.
2. **Tanggal Transaksi** mungkin relevan dalam analisis temporal, tetapi untuk analisis perilaku, dapat diubah menjadi fitur seperti "Hari dalam Seminggu" atau "Bulan".

b. **Memilih Fitur Penting**:
Untuk menganalisis pengaruh kategori produk terhadap total pembayaran, fitur yang relevan adalah:

1. **Kategori Produk**: Variabel independen yang menunjukkan jenis produk.
2. **Total Pembayaran**: Variabel dependen yang mencerminkan hasil transaksi.
3. Fitur lain seperti **Usia Pelanggan** atau **Metode Pembayaran** mungkin digunakan sebagai variabel pendukung jika diperlukan.

Soal 3: Transformasi Fitur

a. Apa yang dimaksud dengan transformasi fitur?
b. Berikan contoh transformasi fitur pada kolom "Tanggal Transaksi".

Jawaban:
a. **Pengertian Transformasi Fitur**:
Transformasi fitur adalah proses mengubah format atau skala data agar lebih sesuai untuk analisis atau model prediksi. Contohnya termasuk normalisasi, standarisasi, atau ekstraksi informasi baru dari fitur yang ada.

b. **Contoh Transformasi pada "Tanggal Transaksi"**:
Daripada menggunakan tanggal mentah, informasi berikut dapat diekstraksi:

1. **Hari dalam Seminggu**: Untuk melihat apakah transaksi lebih banyak terjadi pada hari kerja atau akhir pekan.
2. **Bulan**: Untuk melihat tren musiman, misalnya, peningkatan transaksi selama bulan liburan.

Soal 4: Pentingnya Standarisasi

a. Apa tujuan standarisasi dalam analisis data?
b. Mengapa kolom "Usia Pelanggan" dan "Total Pembayaran" mungkin perlu distandarisasi?

Jawaban:
a. **Tujuan Standarisasi**:
Standarisasi bertujuan untuk mengubah data numerik menjadi skala yang seragam, sehingga semua fitur memiliki kontribusi yang setara dalam analisis atau model.

b. **Standarisasi Kolom "Usia Pelanggan" dan "Total Pembayaran"**:

1. **Usia Pelanggan**: Perbedaan skala antara usia pelanggan (dalam puluhan) dan total pembayaran (dalam ribuan atau jutaan) dapat menyebabkan model memberikan bobot yang tidak seimbang.
2. **Total Pembayaran**: Standarisasi membantu mengurangi pengaruh fitur dengan rentang nilai besar yang dapat mendominasi analisis.

Soal 5: Teknik Feature Selection

a. Sebutkan tiga teknik yang dapat digunakan untuk feature selection.
b. Bagaimana teknik correlation analysis dapat diterapkan pada dataset toko online?

Jawaban:
a. **Tiga Teknik Feature Selection**:

1. **Filter Method**: Menggunakan statistik seperti korelasi untuk menentukan relevansi fitur.
2. **Wrapper Method**: Menggunakan algoritma pembelajaran mesin untuk mengevaluasi kombinasi fitur.
3. **Embedded Method**: Mengintegrasikan proses seleksi fitur dalam pelatihan model (misalnya, regularisasi).

b. **Penerapan Correlation Analysis**:

1. Analisis korelasi dapat digunakan untuk mengukur hubungan antara fitur numerik, seperti "Jumlah Produk Dibeli" dan "Total Pembayaran".
2. Fitur dengan korelasi rendah terhadap target (misalnya, "Total Pembayaran") dapat dihapus karena kurang relevan.

Soal 6: Kombinasi Feature Selection dan Transformation

Anda ingin menganalisis pengaruh metode pembayaran terhadap total pembayaran, tetapi data Anda memiliki kategori yang banyak. Bagaimana Anda menggabungkan feature selection dan transformation untuk menyederhanakan analisis?

Jawaban:

1. **Feature Selection**:
    - Pilih fitur "Metode Pembayaran" dan "Total Pembayaran" sebagai fokus analisis.
    - Jika ada fitur pendukung seperti "Kategori Produk", pertimbangkan relevansinya dengan tujuan analisis.
2. **Feature Transformation**:
    - Lakukan encoding pada "Metode Pembayaran" untuk mengubah kategori menjadi representasi numerik (misalnya, one-hot encoding).
    - Lakukan binning pada "Total Pembayaran" untuk mengelompokkan transaksi menjadi kategori seperti rendah, sedang, dan tinggi.
3. **Langkah Analisis**:
    - Hubungkan hasil encoding "Metode Pembayaran" dengan kategori "Total Pembayaran" untuk menemukan pola penggunaan metode pembayaran pada setiap kategori transaksi.

Soal 7: Tantangan dalam Feature Selection dan Transformation

a. Apa tantangan utama dalam feature selection?
b. Apa tantangan yang dapat muncul saat melakukan transformasi data kategori menjadi numerik?

Jawaban:
a. **Tantangan Feature Selection**:

1. Memilih fitur yang relevan tanpa menghilangkan informasi penting.
2. Mengatasi fitur yang saling berkorelasi tinggi (multikolinearitas).
3. Mengidentifikasi fitur tersembunyi yang mungkin memiliki pengaruh signifikan terhadap target.

b. **Tantangan Transformasi Data Kategori**:

1. Encoding kategori dengan jumlah yang sangat banyak dapat menyebabkan ledakan dimensi.
2. Risiko memberikan bobot yang tidak seimbang pada kategori tertentu jika distribusi data tidak merata.

Kasus 4: Feature Creation dan Augmentation pada Dataset Toko Online

Anda diberikan dataset toko online yang mencakup informasi berikut:

1. **ID Transaksi**: Nomor unik setiap transaksi.
2. **Tanggal Transaksi**: Tanggal ketika transaksi dilakukan.
3. **ID Pelanggan**: Nomor unik untuk setiap pelanggan.
4. **Usia Pelanggan**: Usia pelanggan dalam tahun.
5. **Kategori Produk**: Kategori produk yang dibeli (misalnya, elektronik, pakaian, makanan).
6. **Jumlah Produk Dibeli**: Jumlah total produk dalam satu transaksi.
7. **Total Pembayaran**: Total jumlah uang yang dibayarkan dalam satu transaksi.
8. **Metode Pembayaran**: Metode pembayaran yang digunakan (misalnya, kartu kredit, transfer bank, e-wallet).
9. **Rating Pelanggan**: Penilaian pelanggan terhadap transaksi (skala 1–5).

Tujuan Anda adalah menciptakan fitur baru yang relevan dan memperluas dataset untuk analisis lebih lanjut.

Soal 1: Pengertian Feature Creation dan Augmentation

a. Apa yang dimaksud dengan feature creation dan feature augmentation?
b. Berikan satu contoh untuk masing-masing dalam konteks dataset toko online.

Jawaban:
a. **Pengertian Feature Creation dan Feature Augmentation**:

1. **Feature Creation** adalah proses menciptakan fitur baru dari fitur yang sudah ada untuk memberikan informasi tambahan.
2. **Feature Augmentation** adalah proses memperluas dataset dengan menambahkan data eksternal yang relevan atau menggabungkan informasi dari berbagai sumber.

b. **Contoh dalam Dataset Toko Online**:

1. **Feature Creation**: Membuat fitur baru seperti "Waktu Transaksi" dari kolom **Tanggal Transaksi** untuk mengetahui apakah transaksi dilakukan pagi, siang, atau malam hari.
2. **Feature Augmentation**: Menambahkan data cuaca berdasarkan tanggal dan lokasi transaksi untuk menganalisis pengaruh cuaca terhadap jumlah transaksi.

Soal 2: Identifikasi Fitur Baru

Dari dataset yang diberikan, buat dua fitur baru berdasarkan kolom yang ada dan jelaskan tujuan pembuatannya.

Jawaban:

1. **Fitur "Frekuensi Belanja Pelanggan"**:
    - Dihitung sebagai jumlah transaksi yang dilakukan oleh pelanggan dalam jangka waktu tertentu (misalnya, per bulan).
    - **Tujuan**: Untuk menganalisis pola belanja pelanggan dan mengidentifikasi pelanggan loyal.
2. **Fitur "Rata-rata Pembayaran per Produk"**:
    - Dihitung dengan membagi **Total Pembayaran** dengan **Jumlah Produk Dibeli**.
    - **Tujuan**: Untuk mengetahui nilai rata-rata produk yang dibeli dan membandingkan kategori produk dengan rata-rata pembayaran.

Soal 3: Kapan Feature Augmentation Dibutuhkan?

a. Dalam situasi apa feature augmentation diperlukan dalam analisis data?
b. Sebutkan dua sumber data eksternal yang relevan untuk augmentasi dataset toko online dan jelaskan bagaimana data tersebut dapat digunakan.

Jawaban:
a. **Kapan Feature Augmentation Dibutuhkan**:
Feature augmentation diperlukan ketika data yang ada tidak cukup untuk menjawab pertanyaan analisis, memprediksi hasil, atau memberikan wawasan yang lebih dalam.

b. **Sumber Data Eksternal**:

1. **Data Demografi**: Informasi seperti pendapatan rata-rata atau tingkat pendidikan di area tertentu dapat digunakan untuk memahami profil pelanggan berdasarkan lokasi mereka.
2. **Data Tren Musiman**: Data seperti periode liburan atau hari besar dapat digunakan untuk mengidentifikasi pola musiman dalam transaksi.

Soal 4: Tantangan dalam Feature Creation

a. Apa tantangan utama yang sering dihadapi dalam menciptakan fitur baru?
b. Bagaimana cara mengatasi tantangan tersebut?

Jawaban:
a. **Tantangan dalam Feature Creation**:

1. Sulit mengidentifikasi fitur yang benar-benar relevan dan berdampak pada analisis.

2. Membutuhkan waktu dan pemahaman domain untuk menciptakan fitur yang bermakna.
3. Risiko overfitting jika fitur baru terlalu kompleks atau tidak relevan.

b. **Cara Mengatasi Tantangan**:

1. Lakukan eksplorasi data mendalam untuk memahami hubungan antar fitur.
2. Libatkan ahli domain untuk mendapatkan wawasan yang relevan.
3. Validasi fitur baru dengan metode statistik atau uji coba pada model untuk memastikan kegunaannya.

Soal 5: Transformasi Fitur Baru

Anda telah membuat fitur "Waktu Transaksi" yang mengkategorikan transaksi menjadi pagi, siang, dan malam. Jelaskan bagaimana fitur ini dapat digunakan untuk analisis dan keputusan bisnis.

Jawaban:
Penggunaan Fitur "Waktu Transaksi":

1. **Analisis Pola Belanja**: Mengetahui waktu transaksi paling ramai untuk menentukan jam operasional atau promosi waktu tertentu (time-limited sales).
2. **Strategi Pemasaran**: Mengarahkan kampanye iklan pada waktu transaksi paling aktif untuk meningkatkan efektivitas.
3. **Efisiensi Operasional**: Mengoptimalkan alokasi sumber daya seperti staf dan logistik berdasarkan waktu transaksi.

Soal 6: Contoh Feature Augmentation

Sebuah toko online ingin mengetahui pengaruh harga bahan bakar terhadap total pembayaran. Bagaimana Anda dapat menambahkan data eksternal untuk melakukan analisis ini?

Jawaban:

1. **Sumber Data**: Gunakan data harga bahan bakar harian dari lembaga resmi.
2. **Proses Augmentasi**: Gabungkan data harga bahan bakar dengan dataset transaksi berdasarkan **Tanggal Transaksi**.
3. **Analisis**: Lakukan analisis korelasi atau regresi untuk mengukur pengaruh fluktuasi harga bahan bakar terhadap total pembayaran pelanggan, terutama untuk kategori produk tertentu seperti kebutuhan sehari-hari.

Soal 7: Evaluasi Fitur Baru

Anda telah menciptakan beberapa fitur baru dari dataset toko online. Bagaimana Anda mengevaluasi apakah fitur-fitur baru tersebut memberikan nilai tambah pada analisis atau model prediksi?

Jawaban:

1. **Analisis Statistik**: Gunakan statistik deskriptif untuk memahami distribusi fitur baru dan relevansinya terhadap variabel target.
2. **Uji Korelasi**: Lakukan analisis korelasi antara fitur baru dan target untuk mengevaluasi hubungan yang signifikan.

SOAL UTS KEMARIN SOB

SOAL WAJIB

1. Jelaskan perbedaan antara feature extraction dan feature construction dalam feature transformation. Berikan masing-masing contohnya.
Jawaban:

- **Feature Extraction** adalah proses mengambil informasi penting dari fitur asli untuk membuat representasi yang lebih sederhana dan relevan. Tujuan utamanya adalah mengurangi dimensi data tanpa kehilangan informasi penting.
  Contoh: Dari data gambar, kita dapat mengekstrak fitur seperti jumlah piksel berwarna merah atau tingkat kecerahan rata-rata. Dalam teks, kita bisa mengekstrak jumlah kata kunci atau frekuensi istilah tertentu.

- **Feature Construction** adalah proses membuat fitur baru berdasarkan kombinasi, transformasi, atau manipulasi fitur yang sudah ada. Tujuan utamanya adalah menambah informasi untuk meningkatkan kinerja model.
  Contoh: Dalam data toko online, kita dapat membuat fitur baru seperti "rasio penghasilan terhadap jumlah transaksi" untuk mengetahui pengeluaran rata-rata pelanggan.

---

2. Bagaimana kita bisa melakukan feature extraction dan feature construction terhadap suatu data? Beri contoh dengan data bebas.
Jawaban:

- **Feature Extraction:**
  Misalnya, pada data pelanggan toko online yang memiliki fitur "umur," "penghasilan bulanan," dan "jumlah transaksi," kita dapat mengekstrak fitur seperti "kategori umur" (anak-anak, dewasa muda, dewasa, lansia) dengan menggunakan batasan usia tertentu.

- **Feature Construction:**
  Dengan data yang sama, kita bisa membuat fitur baru, seperti "rasio pengeluaran per transaksi" yang diperoleh dari penghasilan bulanan dibagi dengan jumlah transaksi bulanan. Ini membantu memahami pola pengeluaran pelanggan.

SOAL PILIHAN

Kasus 1:

1. Sebutkan dan jelaskan cara sederhana untuk melakukan feature extraction dari data ini.
Jawaban:
Beberapa cara sederhana untuk melakukan feature extraction dari data "Umur," "Penghasilan bulanan," dan "Jumlah transaksi per bulan":

- **Mengelompokkan umur:** Mengubah data umur menjadi kategori seperti "anak-anak" (0-17), "dewasa muda" (18-35), "dewasa" (36-60), dan "lansia" (>60).

- **Mengelompokkan penghasilan:** Membagi penghasilan ke dalam kelompok seperti "rendah" (<5 juta), "sedang" (5-15 juta), dan "tinggi" (>15 juta).

- **Mengelompokkan transaksi:** Membuat kategori frekuensi, seperti "jarang" (1-5 transaksi/bulan), "sedang" (6-15 transaksi/bulan), dan "sering" (>15 transaksi/bulan).

Cara ini menyederhanakan data numerik menjadi data kategorikal untuk analisis yang lebih mudah.

2. Cobalah buat satu fitur baru (feature construction) yang dapat memberikan informasi lebih banyak tentang pelanggan.
Jawaban:
Kita dapat membuat fitur baru, seperti "rasio pengeluaran per transaksi," yang dihitung dengan membagi "penghasilan bulanan" dengan "jumlah transaksi per bulan." Fitur ini memberikan informasi tentang rata-rata pengeluaran pelanggan untuk setiap transaksi, yang dapat membantu menganalisis pola belanja.

Kasus 2:

1. Lakukan feature construction dengan membuat satu fitur baru yang dapat menunjukkan "kepadatan" ruangan dalam apartemen (rasio luas terhadap jumlah total kamar).
Jawaban:
Fitur "kepadatan" dapat dihitung dengan rumus:
Kepadatan Ruangan=Luas unit apartemenJumlah kamar tidur+Jumlah kamar mandi\text{Kepadatan Ruangan} = \frac{\text{Luas unit apartemen}}{\text{Jumlah kamar tidur} + \text{Jumlah kamar mandi}}Kepadatan Ruangan=Jumlah kamar tidur+Jumlah kamar mandiLuas unit apartemen

Fitur "kepadatan" dapat dihitung dengan rumus:

$$\text{Kepadatan Ruangan} = \frac{\text{Luas unit apartemen}}{\text{Jumlah kamar tidur}+\text{Jumlah kamar mandi}}$$

Misalnya, jika luas apartemen adalah 60 m^2, jumlah kamar tidur adalah 2, dan jumlah kamar mandi adalah 1, maka:

$$\text{Kepadatan Ruangan} = \frac{60}{2+1} = 20\,\text{m}^2/\text{kamar}$$

2. Jelaskan mengapa fitur kepadatan ini mungkin berguna untuk memprediksi harga sewa apartemen.
Jawaban:
Fitur kepadatan memberikan gambaran tentang efisiensi penggunaan ruang di apartemen. Apartemen dengan kepadatan rendah (luas per kamar lebih besar) cenderung lebih nyaman dan memiliki harga sewa lebih tinggi. Sebaliknya, apartemen dengan kepadatan tinggi mungkin terasa sempit dan memiliki harga sewa lebih rendah. Dengan demikian, fitur ini membantu model memprediksi harga sewa berdasarkan kenyamanan yang dirasakan oleh penyewa potensial.