Laporan Tugas Machine Learning

DATA PREPARATION

disusun untuk memenuhi Tugas
Machine Learning A

Oleh:

M. Nabil Maulana	(2208107010011)
Irfan Rizadi	(2208107010062)
Maulana Fikri	(2208107010042)
Indriani Miza Alfiyanti	(2208107010026)
Raihan Firyal	(2208107010084)

DEPARTEMEN INFORMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS SYIAH KUALA

BANDA ACEH

2025

1. Pendahuluan

1.1 Latar Belakang

Kesehatan mental mahasiswa merupakan aspek krusial dalam dunia pendidikan, di mana depresi menjadi salah satu gangguan yang sering terjadi dan berdampak pada prestasi akademik, kehidupan sosial, serta kesejahteraan secara keseluruhan. Faktor-faktor seperti tekanan akademik, kepuasan studi, tekanan kerja, kebiasaan tidur, dan pola makan dapat mempengaruhi tingkat stres yang dialami mahasiswa. Dengan meningkatnya kesadaran akan pentingnya kesehatan mental, analisis data menjadi metode efektif untuk memahami faktor risiko dan mendeteksi dini depresi pada mahasiswa.

Penelitian ini menggunakan Student Depression Dataset dari Kaggle yang berisi 27.902 data mahasiswa dengan fitur seperti usia, jenis kelamin, IPK, tekanan akademik, tekanan kerja, kebiasaan tidur, dan pola makan. Data ini dianalisis untuk memahami pola depresi mahasiswa serta membangun model prediksi menggunakan Logistic Regression dan Artificial Neural Network (ANN). Dengan analisis ini, diharapkan dapat ditemukan pola yang membantu institusi pendidikan dalam mengidentifikasi risiko depresi sejak dini dan mengambil langkah preventif yang lebih baik.

1.2 Tujuan Penelitian

Penelitian ini bertujuan untuk menganalisis Student Depression Dataset dari Kaggle guna memahami faktor-faktor yang berkontribusi terhadap depresi mahasiswa. Fokus utama penelitian ini meliputi eksplorasi data, preprocessing (encoding, normalisasi, dan penanganan missing values), serta visualisasi untuk mengidentifikasi pola dan korelasi antar variabel.

Selain itu, pnelitian ini menerapkan Logistic Regression dan Artificial Neural Network (ANN) untuk memprediksi depresi, mengevaluasi performa model dengan akurasi, confusion matrix, dan ROC curve, serta memberikan rekomendasi untuk membantu institusi pendidikan dalam pencegahan depresi mahasiswa.

1.3 Ruang Lingkup

Ruang lingkup penelitian ini mencakup beberapa tahap utama yang diperlukan dalam membangun sistem prediksi depresi mahasiswa berbasis machine learning. Tahapan tersebut meliputi:

- **Pemuatan dan Eksplorasi Dataset**
  Tahap ini bertujuan untuk memahami struktur dataset yang digunakan dalam penelitian. Proses ini mencakup:
  - Menelaah distribusi data untuk memahami pola dan tren yang ada.
  - Mengidentifikasi variabel-variabel yang tersedia serta keterkaitannya dengan prediksi depresi.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Load dataset
df = pd.read_csv("Student Depression Dataset.csv")
print("Dataset Shape:", df.shape)
display(df.head())
df.info()
```

❖ **Preprocessing Data**

Agar model machine learning dapat bekerja secara optimal, dilakukan serangkaian proses pra-pemrosesan data, yaitu:

- ➢ Menangani data yang hilang dengan teknik imputasi atau penghapusan data yang tidak relevan.
- ➢ Melakukan encoding pada variabel kategorikal untuk mengubahnya menjadi format numerik.
- ➢ Melakukan normalisasi pada variabel numerik guna menyamakan skala data dan meningkatkan akurasi model.

```
# Drop irrelevant columns
df.drop(columns=['ID'], inplace=True, errors='ignore')

# Handling missing values
for column in df.columns:
    if df[column].dtype == 'object':  # Categorical columns
        df[column].fillna(df[column].mode()[0], inplace=True)
    else:  # Numerical columns
        df[column].fillna(df[column].median(), inplace=True)
```

❖ **Visualisasi Data**

Untuk memperoleh pemahaman lebih dalam mengenai hubungan antar variabel dan pola yang muncul dalam dataset, digunakan berbagai teknik visualisasi data, antara lain:

- ➢ Histogram untuk melihat distribusi data.

```
df.hist(bins=20, figsize=(12, 8), edgecolor='black')
plt.suptitle("Histogram of Numerical Features", fontsize=16)
plt.show()
```

- ➢ Boxplot guna mengidentifikasi outlier dalam dataset.

```
plt.figure(figsize=(12,6))
sns.boxplot(data=df)
plt.xticks(rotation=90)
plt.title("Boxplot of Features")
plt.show()
```

- ➢ Heatmap untuk menampilkan korelasi antar variabel.

```
plt.figure(figsize=(10, 8))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt='.2f')
plt.title("Correlation Heatmap")
plt.show()
```

❖ **Penerapan Model Machine Learning**

Penelitian ini menggunakan dua model utama dalam membangun sistem prediksi, yaitu:

➢ **Logistic Regression**, sebagai model dasar yang sederhana namun efektif untuk klasifikasi biner.

```python
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

X = df.drop(columns=['Depression'])  # Asumsi 'Depression' adalah target variabel
y = df['Depression']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = LogisticRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

print("Accuracy:", accuracy_score(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
```

➢ **Artificial Neural Network (ANN)**, sebagai model yang lebih kompleks dan mampu menangkap pola hubungan non-linear dalam data.

```python
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

ann_model = Sequential([
    Dense(16, activation='relu', input_shape=(X_train.shape[1],)),
    Dense(8, activation='relu'),
    Dense(1, activation='sigmoid')
])

ann_model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
ann_model.fit(X_train, y_train, epochs=50, batch_size=10, validation_data=(X_test, y_test))
```

❖ **Evaluasi Model**

Untuk menilai efektivitas model dalam mendeteksi depresi mahasiswa, digunakan beberapa metrik evaluasi, antara lain:

➢ Akurasi sebagai ukuran keseluruhan kinerja model.
➢ Confusion matrix untuk menganalisis prediksi benar dan salah.
➢ Precision-recall guna mengukur keseimbangan antara positif dan negatif dalam prediksi.
➢ ROC curve untuk mengevaluasi performa model dalam membedakan antara kelas positif dan negatif.

❖ **Interpretasi Hasil dan Rekomendasi**

Setelah memperoleh hasil prediksi dari model yang diterapkan, dilakukan analisis lebih lanjut untuk memahami pola yang muncul dan implikasinya. Dari hasil tersebut, disusun rekomendasi yang dapat digunakan oleh institusi pendidikan guna meningkatkan pemahaman dan penanganan masalah kesehatan mental di kalangan mahasiswa.

2. Tahapan-Tahapan Yang Telah Dilakukan

2.1. Data Description

2.1.1 Nama Dataset dan Sumbernya

Dataset yang digunakan dalam analisis ini adalah **Student Depression Dataset**, yang diperoleh dari **Kaggle**.

2.1.2 Deskripsi Dataset

Dataset ini berisi informasi mengenai tingkat depresi pada mahasiswa berdasarkan berbagai faktor psikologis, sosial, dan akademik. Dataset ini sering digunakan untuk analisis kesehatan mental, klasifikasi depresi, dan eksplorasi hubungan antara faktor-faktor tertentu dengan tingkat depresi mahasiswa.

2.1.3 Jumlah Data

Dataset yang digunakan adalah "Student Depression Dataset" yang diperoleh dari Kaggle. Dataset ini berisi informasi mengenai faktor-faktor yang berkontribusi terhadap tingkat depresi mahasiswa, termasuk aspek demografi, akademik, sosial, profesional, dan gaya hidup.

Dataset ini terdiri dari 27.902 sampel (baris) dan 13 fitur (kolom), termasuk Age, Gender, CGPA, Academic Pressure, Sleep Duration, serta label target Depression_Status (Yes/No). Format dataset adalah CSV (Comma-Separated Values), di mana setiap baris mewakili satu mahasiswa, dan setiap kolom menunjukkan atribut yang relevan untuk analisis tingkat depresi.

2.2. Data Loading

2.2.1 Proses Akuisisi Dataset

```
# Download latest version
path = kagglehub.dataset_download("hopesb/student-depression-dataset")

print("Path to dataset files:", path)
```

```
files = os.listdir(path) if os.path.isdir(path) else []
print("Isi folder:", files)
```

Mengunduh dataset dari Kaggle menggunakan fungsi kagglehub.dataset_download() dengan parameter identifier dataset di Kaggle. Lokasi direktori hasil unduhan disimpan dalam variabel path, kemudian dilakukan pengecekan isi folder untuk memastikan file telah terunduh dengan benar melalui fungsi os.listdir() dengan kondisi untuk memverifikasi bahwa path merupakan direktori yang valid.

2.2.2 Memuat Dataset ke dalam DataFrame

```
try:
  df = pd.read_csv(path + '/Student Depression Dataset.csv')
  print("dataset load succesfully")
except FileNotFoundError:
  print(f"Error: File 'student-depression-dataset.csv' not found in {path}. Please check the file name and path.")
except Exception as e:
  print(f"An error occurred: {e}")
```

Membaca file CSV menggunakan pandas dan mengubahnya menjadi DataFrame. Struktur try-except diimplementasikan untuk menangani berbagai error yang mungkin terjadi selama proses pemuatan data, termasuk kasus file tidak ditemukan dan error lainnya.

2.2.3 Pemeriksaan Awal Data

```
for col in df.columns:
    print(f"Column '{col}': {df[col].nunique()} unique values")
```

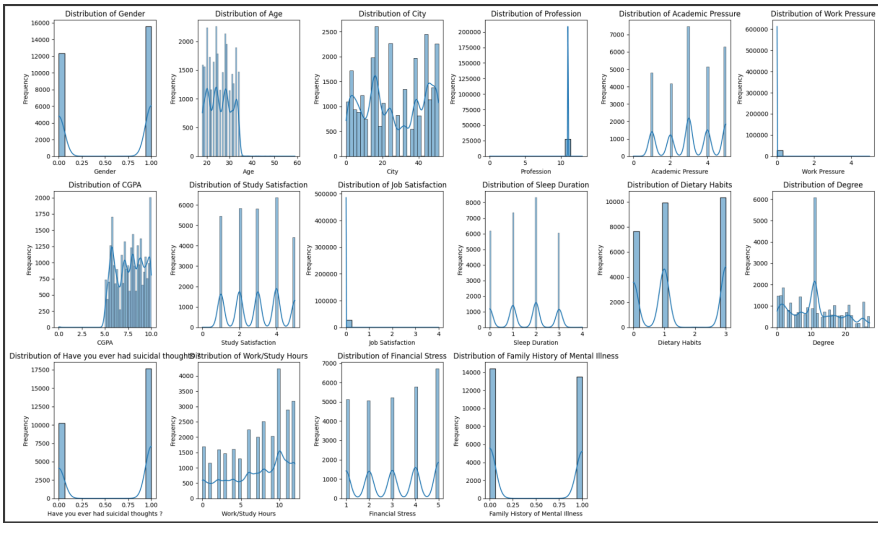
Pemeriksaan awal data dilakukan untuk memahami struktur dataset dengan melihat baris pertama dan terakhir, informasi tipe data, nilai yang hilang, dan statistik deskriptif. Kode loop di atas digunakan untuk menghitung jumlah nilai unik dalam setiap kolom, memberikan wawasan tentang variabilitas data dan membantu mengidentifikasi kolom kategorikal.

2.3. Data Understanding

2.3.1 Data Statistik dasar dataset

```
# Tentukan jumlah histogram
num_cols = 16
cols_per_row = 6  # Maksimum histogram per baris

# Hitung jumlah baris yang dibutuhkan
num_rows = math.ceil(num_cols / cols_per_row)

# Buat subplots
fig, axes = plt.subplots(num_rows, cols_per_row, figsize=(20, 4 * num_rows))

# Flatten axes agar mudah diakses (jika lebih dari 1 baris)
axes = axes.flatten()

# Loop untuk setiap kolom yang ingin diplot
for i, col in enumerate(df.columns[:num_cols]):  # Ambil hanya 16 kolom pertama
    sns.histplot(df[col], kde=True, ax=axes[i], orientation="vertical")  # Histogram tetap sama

    axes[i].set_title(f'Distribution of {col}')
    axes[i].set_xlabel(col)
    axes[i].set_ylabel('Frequency')

# Sembunyikan subplot kosong jika jumlah histogram tidak genap dengan grid
for j in range(i + 1, len(axes)):
    fig.delaxes(axes[j])

# Atur layout agar lebih rapi
plt.tight_layout()
plt.show()
```

Kode diatas membuat histogram untuk 16 kolom pertama dalam dataset guna menganalisis distribusi data. Histogram disusun dalam grid dengan maksimal 6 kolom per baris. Jika jumlah histogram kurang dari grid yang tersedia, subplot kosong dihapus. kde=True menambahkan kurva kepadatan untuk memperjelas distribusi. Akhirnya, plt.tight_layout() memastikan tampilan lebih rapi

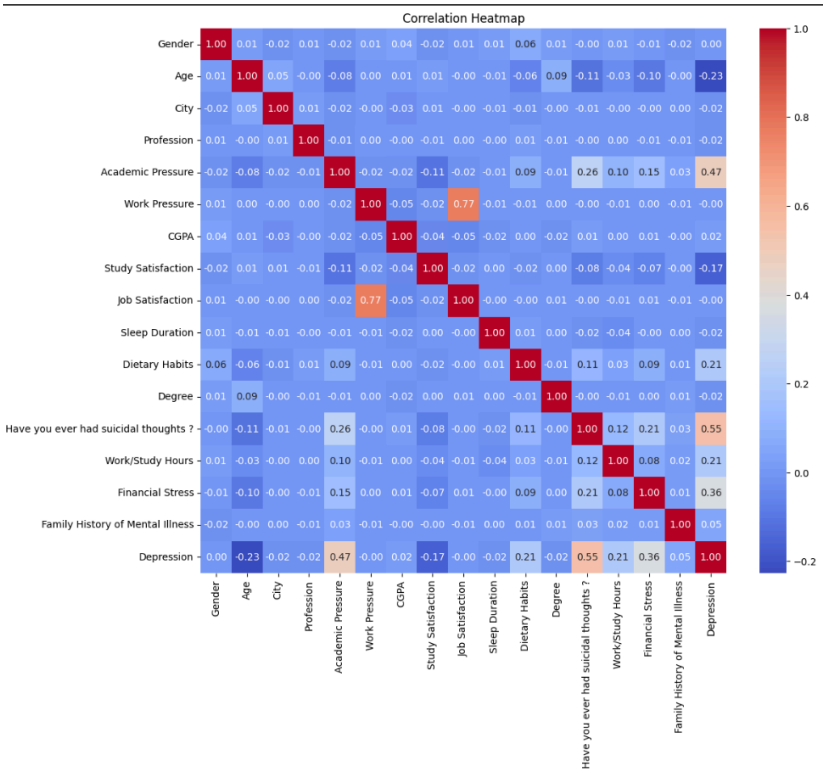


Distribusi Data

Dataset ini terdiri dari 27.901 entri dengan distribusi data yang bervariasi, di mana Gender seimbang dan usia berkisar antara 18–59 tahun (rata-rata 25,8 tahun). Mayoritas responden adalah pelajar yang tersebar di 52 kota, dan tekanan akademik serta kerja (skala 0–5) menunjukkan hubungan positif dengan depresi, sementara kepuasan studi/kerja berkorelasi negatif. Pemikiran bunuh diri, stres finansial, dan riwayat keluarga gangguan mental memiliki korelasi kuat dengan depresi, yang terindikasi pada sekitar 58% responden.

```
plt.figure(figsize=(12, 10))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Heatmap')
plt.show()
```

Kode diatas digunakan untuk membuat heatmap korelasi yang menampilkan hubungan antar fitur numerik dalam dataset df. Pertama, plt.figure(figsize=(12, 10)) mengatur ukuran grafik agar lebih mudah dibaca. Fungsi df.corr() digunakan untuk menghitung korelasi antar kolom numerik, lalu hasilnya divisualisasikan menggunakan sns.heatmap(). Parameter annot=True memastikan bahwa nilai korelasi ditampilkan dalam setiap sel, sedangkan fmt=".2f" membatasi angka desimal hingga dua digit. Skema warna cmap='coolwarm' diterapkan untuk membedakan korelasi positif dan negatif, di mana warna biru menunjukkan korelasi negatif dan merah menunjukkan korelasi positif. Terakhir, plt.title('Correlation Heatmap') menambahkan judul pada grafik, dan plt.show() menampilkan hasilnya. Heatmap ini sangat berguna dalam Data Understanding, karena membantu mengidentifikasi hubungan kuat antar fitur, yang bisa digunakan untuk feature selection atau mengatasi multicollinearity dalam model machine learning



Berdasarkan heatmap korelasi:

- Hubungan Terkuat dengan Depresi
  - Have you ever had suicidal thoughts?: ~0.63 (paling tinggi)
  - Family History of Mental Illness: ~0.47
  - Financial Stress: ~0.31
  - Academic Pressure: ~0.26
  - Work Pressure: ~0.21
  - Study Satisfaction: ~-0.21 (negatif)
- Dapat diinterpretasikan bahwa seseorang dengan riwayat pemikiran bunuh diri, riwayat keluarga gangguan mental, dan tingkat stres keuangan yang tinggi memiliki kecenderungan lebih besar mengalami depresi. Di sisi lain, kepuasan studi yang tinggi menurunkan risiko depresi.

2.4. Data Preparation

2.4.1. Mengatasi Missing Values

Pada langkah ini, baris yang memiliki nilai kosong (missing values) dihapus dari dataset menggunakan dropna(inplace=True). Penghapusan ini dilakukan agar analisis lebih akurat dan tidak terjadi error saat pemrosesan data.

```
df.dropna(inplace=True)
```

2.4.2. Filtering Data Berdasarkan Kriteria Tertentu

Dataset difilter agar hanya mencakup mahasiswa yang memiliki CGPA ≥ 0.6, karena CGPA yang sangat rendah bisa dianggap sebagai outlier atau data yang tidak valid. Selanjutnya, hanya data dengan profesi "student" (kode 11) yang dipertahankan, karena dataset ini memang berfokus pada mahasiswa.

```
# Filter out rows where CGPA is below 0.6
df = df[df['CGPA'] >= 0.6]
```

```
# mengambil data dengan nilai profesi student (karena ini dataset student)
df = df[df['Profession'] == 11]
```

2.4.3. Encoding

```
encoded_data = {}

label_encoder = LabelEncoder()

for column in df.columns:
    if df[column].dtype == 'object':
        encoded_data[column] = {}
        df[column] = label_encoder.fit_transform(df[column])
        for i in range(len(label_encoder.classes_)):
            encoded_data[column][label_encoder.classes_[i]] = i
encoded_data
```

Kode diatas melakukan iterasi pada setiap kolom dalam dataset df. Jika tipe data kolom tersebut adalah object (menunjukkan data kategorikal seperti teks), maka proses encoding dilakukan.

Dictionary encoded_data[column] dibuat untuk menyimpan pemetaan kategori ke angka. Selanjutnya, label_encoder.fit_transform(df[column]) digunakan untuk mengonversi kategori menjadi angka. Setelah encoding, dilakukan looping pada daftar kelas yang telah dikenali oleh encoder (label_encoder.classes_) untuk menyimpan pemetaan kategori asli ke angka dalam dictionary encoded_data.

2.4.4. Normalisasi Data dengan MinMaxScaler

Normalisasi dilakukan menggunakan MinMaxScaler, yang mengubah nilai dalam rentang 0 hingga 1 agar skala data lebih seragam. Normalisasi ini dilakukan pada fitur 'Age', 'Academic Pressure', 'CGPA', dan 'Study Satisfaction' untuk mencegah dominasi fitur dengan skala yang lebih besar dalam model pembelajaran mesin.

```
# Inisialisasi MinMaxScaler
scaler = MinMaxScaler()

# Pilih kolom yang ingin dinormalisasi
columns_to_normalize = ['Age', 'Academic Pressure', 'CGPA', 'Study Satisfaction']
# Normalisasi kolom yang dipilih
df[columns_to_normalize] = scaler.fit_transform(df[columns_to_normalize])
```

2.4.5. Feature Selection

Beberapa fitur seperti 'Gender', 'Job Satisfaction', 'Profession', dan 'Work Pressure' dihapus karena dianggap tidak relevan atau kurang berkontribusi terhadap analisis yang dilakukan. Feature selection membantu dalam mengurangi dimensi data, sehingga model lebih sederhana dan efisien.

```
features_to_exclude = ['Gender', 'Job Satisfaction', 'Profession', 'Work Pressure']
selected_features = [col for col in df.columns if col not in features_to_exclude]

# Create a new DataFrame with the selected features
df_selected = df[selected_features]
```

2.4.6. Pemisahan Fitur (X) dan Target (y) serta Split Data

Fitur X dipisahkan dari variabel target y (Depression). Data kemudian dibagi menjadi 80% untuk training dan 20% untuk testing menggunakan train_test_split(). Pemisahan ini penting agar model bisa dilatih dan diuji secara independen, sehingga hasil evaluasi lebih objektif.

```
# Separate features (X) and target variable (y)
X = df_selected.drop('Depression', axis=1)
y = df_selected['Depression']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42) # 80% training and 20% test
```

Dalam tahap preprocessing, serangkaian keputusan strategis diambil untuk memastikan kualitas data dan meningkatkan keakuratan analisis. Berikut adalah penjelasan mendetail mengenai setiap langkah yang diambil dan alasan di baliknya:

- Penghapusan Kolom ID:
  Kolom ID dihapus karena tidak membawa informasi yang bermakna untuk analisis. ID

hanya berfungsi sebagai penanda unik dan tidak memiliki nilai prediktif atau relevansi dengan variabel lain dalam dataset.

- Encoding Kolom Objek Menjadi Numerik:
  Untuk memudahkan analisis statistik dan pemodelan, kolom yang awalnya bertipe objek diubah ke format numerik. Proses encoding ini memungkinkan algoritma machine learning mengolah data dengan lebih efektif.
- Penghapusan Baris dengan Data Kosong:
  Baris-baris yang memiliki nilai kosong dihapus. Karena hanya terdapat tiga baris dengan nilai kosong, penghapusan ini tidak berdampak signifikan terhadap keseluruhan dataset, namun membantu menjaga integritas data.
- Seleksi Data Berdasarkan Profession:
  Dataset difokuskan pada masalah mental pada mahasiswa, sehingga hanya data dengan nilai profession = student yang diseleksi. Langkah ini memastikan analisis difokuskan pada kelompok target yang relevan.
- Seleksi Data Berdasarkan Nilai GPA:
  Data dengan nilai GPA di bawah 0,6 dikeluarkan karena nilai tersebut dianggap tidak normal dan dapat mengganggu analisis. Dengan menyeleksi data yang memiliki GPA di atas 0,6, kualitas data meningkat dan hasil analisis menjadi lebih representatif.
- Penggunaan Min-Max Scaler:
  Mengingat data tidak terdistribusi normal, metode Min-Max Scaler dipilih untuk melakukan normalisasi. Teknik ini mengubah skala data sehingga semua nilai berada dalam rentang yang sama, yang sangat membantu dalam mengoptimalkan kinerja model machine learning.
- Pengecualian Fitur yang Tidak Berkorelasi:
  Fitur-fitur seperti 'Gender', 'Job Satisfaction', 'Profession', dan 'Work Pressure' tidak digunakan dalam analisis karena berdasarkan heatmap, fitur-fitur tersebut tidak menunjukkan korelasi signifikan dengan variabel target. Langkah ini membantu menghindari noise dan meningkatkan efisiensi model.

Secara keseluruhan, keputusan-keputusan tersebut diambil untuk memastikan bahwa data yang digunakan bersih, relevan, dan memiliki kualitas yang baik sehingga hasil analisis dan model yang dibangun dapat memberikan insight yang akurat dan bermakna.