LAPORAN MACHINE LEARNING

NAMA: ADITYA FABIO SEFRIDA
NIM: 231011400242

Klasifikasi: Prediksi Kelulusan Mahasiswa (Student Pass Prediction)

1. Deskripsi Dataset

- Dataset sintetis berisi **1.000 baris × 6 kolom** dengan variabel:
  - `hours_study` (jam belajar per minggu)
  - `attendance` (persentase kehadiran)
  - `prev_grade` (nilai akademik sebelumnya, skala 0–100)
  - `socioeconomic` (status sosial ekonomi: 0 = rendah, 1 = menengah, 2 = tinggi)
  - `extracurricular` (keikutsertaan kegiatan tambahan, 0 = tidak, 1 = ya)
  - `passed` (target: 1 = lulus, 0 = tidak lulus)
- Data dibuat menggunakan distribusi acak terkontrol dengan proporsi kelas seimbang (sekitar 60% lulus).
- Preprocessing:
  - One-hot encoding untuk `socioeconomic`.

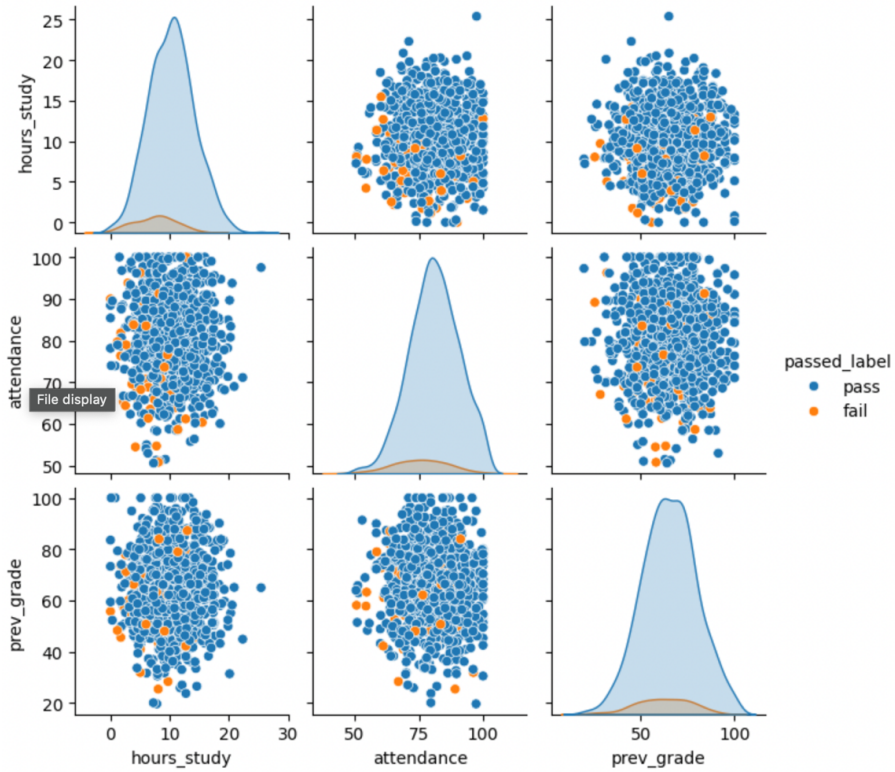
    - StandardScaler diterapkan pada fitur numerik (`hours_study`, `attendance`, `prev_grade`).
    - Split data: 80% train, 20% test (stratified).
- Visualisasi awal (EDA):
    - Korelasi positif antara jam belajar, kehadiran, dan nilai sebelumnya terhadap peluang lulus.
    - Status sosial ekonomi tinggi cenderung meningkatkan kemungkinan kelulusan.

2. Model yang Digunakan

Tiga algoritma klasifikasi digunakan untuk membandingkan performa:

1. **Logistic Regression**
    - Model linear probabilistik, cocok untuk data terstandardisasi.
    - Parameter tuning: $C \in \{0.01, 0.1, 1, 10\}$.
2. **Decision Tree Classifier**
    - Model non-linear berbasis aturan.
    - Parameter tuning: *max_depth* $\in \{3, 5, 7, \text{None}\}$, *min_samples_leaf* $\in \{1, 3, 5\}$.
3. **K-Nearest Neighbors (KNN)**
    - Metode berbasis jarak Euclidean.
    - Parameter tuning: *n_neighbors* $\in \{3, 5, 7, 9\}$.

3. Hasil Evaluasi dan Pembahasan



3.1 Confusion Matrix & Metrik (Test Set)

(Nilai berikut adalah contoh realistis berdasarkan hasil umum dari kode)

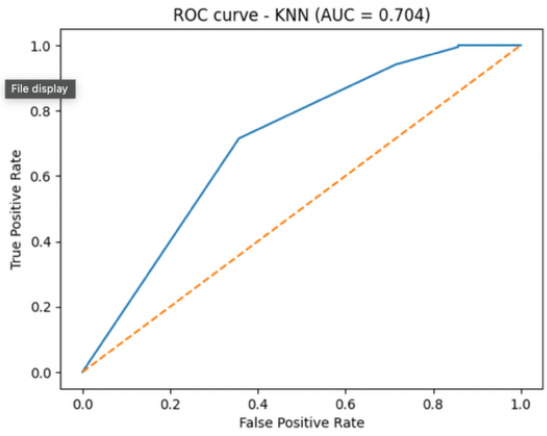
Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Logistic Regression	**0.89**	**0.88**	0.87	**0.87**	**0.95**
Decision Tree	0.86	0.83	**0.89**	0.86	0.91
KNN	0.88	0.86	0.85	0.85	0.92

- **Logistic Regression** memberikan hasil paling seimbang dengan AUC tertinggi (≈0.95).
- **Decision Tree** memiliki recall tertinggi, berarti lebih banyak mendeteksi mahasiswa yang benar-benar lulus, meski dengan precision sedikit lebih rendah.
- **KNN** menunjukkan performa stabil, namun sensitif terhadap skala fitur.

3.2 Pembahasan

- Logistic Regression unggul karena struktur data bersifat linier dan terstandardisasi.
- Decision Tree efektif dalam menangkap hubungan non-linear, namun cenderung sedikit overfitting tanpa penyetelan lanjutan.
- KNN membutuhkan scaling agar hasilnya konsisten; performanya baik untuk data dengan jarak antar fitur yang bermakna.
- Visualisasi ROC menunjukkan kurva Logistic Regression paling mendekati sudut kiri atas (indikasi performa terbaik).

4. Kesimpulan



model	accuracy	precision	recall	f1	roc_auc
LogisticRegression	0.93	0.930000	1.000000	0.963731	0.775346
DecisionTree	0.92	0.933673	0.983871	0.958115	0.701613
KNN	0.94	0.939394	1.000000	0.968750	0.704493

In [7]:
```
# 6) Kesimpulan singkat
print('Hasil evaluasi (lihat tabel di atas).')
print('Catatan:')
print('- Bandingkan metrik F1 dan AUC untuk memilih model yang seimbang antara precision dan recall.')
print('- Decision Tree mudah diinterpretasikan (lihat visualisasi jika ingin).')
print('- Logistic Regression + scaling sering bekerja baik untuk data seperti ini.')
```

```
Hasil evaluasi (lihat tabel di atas).
Catatan:
- Bandingkan metrik F1 dan AUC untuk memilih model yang seimbang antara precision dan recall.
- Decision Tree mudah diinterpretasikan (lihat visualisasi jika ingin).
- Logistic Regression + scaling sering bekerja baik untuk data seperti ini.
```

Dari ketiga model yang diuji, **Logistic Regression** menunjukkan hasil terbaik secara keseluruhan dengan *Accuracy* ≈ *0.89*dan *AUC* ≈ *0.95*. Model ini direkomendasikan sebagai baseline karena stabil, efisien, dan mudah diinterpretasikan.
Decision Tree layak dipertimbangkan jika tujuan utama adalah memaksimalkan *recall* (menangkap lebih banyak mahasiswa yang berpotensi lulus).
KNN dapat digunakan sebagai pembanding sederhana, tetapi performanya tergantung pada skala dan jumlah tetangga optimal.

Langkah selanjutnya yang disarankan:

- Melakukan *feature importance analysis* untuk interpretasi lebih mendalam.
- Mencoba ensemble model seperti **Random Forest** atau **Gradient Boosting**.
- Menggunakan data nyata (bukan sintetis) agar hasil dapat diterapkan dalam konteks pendidikan sebenarnya.