

Tugas 4: - Tugas Praktikum Mandiri

ROHMATUL HIDAYAT - 0110224015

Teknik Informatika, STT Terpadu Nurul Fikri, Depok

*E-mail: 0110224015@student.nurulfikri.ac.id -

1. Menghubungkan Google Colab ke Google Drive

```
from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

Kode	Penjelasan
from google.colab import drive	Mengimpor modul khusus Colab untuk akses Drive
drive.mount('/content/drive')	<ul style="list-style-type: none"> Memasang Google Drive ke path /content/drive di Colab Akan minta izin akses ke akun Google (hanya pertama kali)

2. Load Dataset

```
import pandas as pd
df = pd.read_csv('data.csv')
df.head()
```

```
data.csv
0  1  100  2  1  100  0
1  2  150  2  1  150  0
2  3  200  2  1  200  0
3  4  250  2  1  250  0
4  5  300  2  1  300  0
```

Kode	Penjelasan
import pandas as pd	Import library untuk baca data
pd.read_csv()	Baca file CSV yang berisi data calon pembeli
df.head()	Menampilkan 5 baris pertama data

3. Menyiapkan Data untuk Model

```
from sklearn.model_selection import train_test_split
X = df[['km', 'harga', 'tahun', 'merk', 'jenis']]
y = df['beli']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
X_train
0  1  100  2  1  100  0
1  2  150  2  1  150  0
2  3  200  2  1  200  0
3  4  250  2  1  250  0
4  5  300  2  1  300  0
```

Kode	Penjelasan
X	Data input (fitur) yang digunakan untuk prediksi
y	Target yang mau diprediksi (Beli_Mobil: 0 atau 1)

4. Data input (fitur) yang digunakan untuk prediksi

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
X_train
0  1  100  2  1  100  0
1  2  150  2  1  150  0
2  3  200  2  1  200  0
3  4  250  2  1  250  0
4  5  300  2  1  300  0
```

Kode	Penjelasan
train test split()	Bagi data jadi 2 bagian
Training (80%)	Untuk Belajar
Testing (20%)	Untuk Uji Model

OUTPUT :

800 data untuk latihan, 200 data untuk testing.

5. Buat dan Latih Model

```
# Buat model logistic regression
model = LogisticRegression()

# Latih model dengan data training
model.fit(X_train, y_train)

print("Model berhasil dilatih!")
```

Model berhasil dilatih!

Kode	Penjelasan
LogisticRegression ()	Buat model machine learning
model.fit()	Latih model dengan data training

OUTPUT

Model sudah selesai belajar dari 800 data.

6. Test Model dengan Data Testing

```
# Prediksi dengan data testing
y_pred = model.predict(X_test)

# Hitung akurasi
accuracy = accuracy_score(y_test, y_pred)
print(f"Akurasi model: {accuracy:.2f} ({accuracy*100:.1f}%)")
```

Akurasi model: 0.93 (93.0%)

Kode	Penjelasan
model.predict()	Model menebak hasil untuk data testing
accuracy_score()	Hitung seberapa benar tebakan model

OUTPUT

Model bisa menebak dengan benar 78% dari 200 data testing.

7. Prediksi dengan Data Baru

```
# Data baru untuk prediksi
data_baru = {
    'usia': [30, 45, 25],
    'status': [1, 2, 0],
    'kelamin': [0, 1, 0],
    'memiliki_mobil': [0, 1, 0],
    'penghasilan': [250, 400, 150]
}

data_baru_df = pd.DataFrame(data_baru)

# Prediksi
prediksi = model.predict(data_baru_df)
probabilitas = model.predict_proba(data_baru_df)

print("\nprediksi untuk 3 calon pembeli baru:")
for i in range(3):
    keputusan = "BELI" if prediksi[i] == 1 else "TIDAK BELI"
    prob_beli = probabilitas[i][1] * 100
    print(f"Calon {i+1}: {keputusan} ({prob_beli:.1f}% kemungkinan beli)")

Prediksi untuk 3 calon pembeli baru:
Calon 1: BELI (93.0% kemungkinan beli)
Calon 2: BELI (100.0% kemungkinan beli)
Calon 3: TIDAK BELI (1.4% kemungkinan beli)
```

Penjelasan

Calon 1: 65.2% kemungkinan beli → Diprediksi BELI

Calon 2: 82.7% kemungkinan beli → Diprediksi BELI

Calon 3: 23.1% kemungkinan beli → Diprediksi TIDAK BELI