

PRAKTIKUM MANDIRI PEKAN 4

SYAHRI GHIFARI MAULIDI - 0110222217

¹Teknik Informatika, STT Terpadu Nurul Fikri, Depok

*E-mail: syah22217ti@student.nurulfikri.ac.id

1. import

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
import seaborn as sns
import matplotlib.pyplot as plt
```

Tujuan dari praktikum ini adalah untuk menerapkan metode *Logistic Regression* dalam memprediksi keputusan pembelian mobil berdasarkan beberapa variabel demografis dan ekonomi calon pembeli. Melalui analisis ini, diharapkan mahasiswa mampu memahami konsep dasar regresi logistik, proses pembangunan model prediksi, serta interpretasi hasil yang diperoleh.

1.1 Calling Data

```
df = pd.read_csv("../Data/calonpembelimobil.csv")
print(df.head)
print(df.info)
```

<bound method NDFrame.head of					ID	Usia	Status	Kelamin	Memiliki_Mobil	Penghasilan	Beli_Mobil
0	1	32	1	0			0	240	1		
1	2	49	2	1			1	100	0		
2	3	52	1	0			2	250	1		
3	4	26	2	1			1	130	0		
4	5	45	3	0			2	237	1		
..		
995	996	51	0	1			2	293	1		
996	997	61	1	0			2	275	1		
997	998	45	1	0			0	406	1		
998	999	48	0	1			2	432	1		
999	1000	40	3	1			1	380	1		

```
[1000 rows x 7 columns]>
```

<bound method DataFrame.info of					ID	Usia	Status	Kelamin	Memiliki_Mobil	Penghasilan	Beli_Mobil
0	1	32	1	0			0	240	1		
1	2	49	2	1			1	100	0		
2	3	52	1	0			2	250	1		
3	4	26	2	1			1	130	0		
4	5	45	3	0			2	237	1		
..		
995	996	51	0	1			2	293	1		
996	997	61	1	0			2	275	1		
997	998	45	1	0			0	406	1		
998	999	48	0	1			2	432	1		
999	1000	40	3	1			1	380	1		

```
[1000 rows x 7 columns]>
```

Dataset dimuat menggunakan `pandas.read_csv()`. Selanjutnya dilakukan pemeriksaan tipe data dan deteksi nilai kosong (missing values). Hasil pemeriksaan menunjukkan bahwa seluruh data dalam kondisi baik dan dapat digunakan untuk pelatihan model.

2. isnull

```
print(df.isnull().sum())
```

ID	0
Usia	0
Status	0
Kelamin	0
Memiliki_Mobil	0
Penghasilan	0
Beli_Mobil	0

```
dtype: int64
```

```
X = df[['Usia', 'Status', 'Kelamin', 'Memiliki_Mobil', 'Penghasilan']]
y = df['Beli_Mobil']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

```
model = LogisticRegression()
model.fit(X_train, y_train)
```

LogisticRegression

Parameters

Kode tersebut mengawali proses *data preprocessing* dan pemodelan klasifikasi prediktif. Pertama-tama, dilakukan analisis kualitas data dengan memverifikasi keberadaan nilai yang hilang (*missing values*) menggunakan `df.isnull().sum()`. Hasil yang menunjukkan nilai nol (0) pada semua kolom data (ID hingga `Beli_Mobil`) mengonfirmasi bahwa *dataset* sudah bersih dan siap tanpa perlu intervensi imputasi. Selanjutnya, data didefinisikan menjadi variabel independen yang mencakup lima fitur prediktor ('Usia', 'Status', 'Kelamin', 'Memiliki_Mobil', 'Penghasilan'), dan variabel dependen, yaitu target klasifikasi ('Beli_Mobil'). Pemisahan fitur dan target ini merupakan langkah esensial sebelum memasuki tahap pemodelan.

Data kemudian dibagi menjadi *data training* dan *data testing* menggunakan fungsi `train_test_split`, dengan alokasi 70% untuk pelatihan dan 30% untuk pengujian (`test_size=0.3`). Penggunaan `random_state=42` memastikan konsistensi dan reproduksibilitas pembagian data. Tahap krusial berikutnya adalah inisiasi dan pelatihan model Regresi Logistik (`LogisticRegression`), sebuah algoritma klasifikasi yang sesuai untuk memprediksi hasil biner. Model tersebut kemudian dilatih (`model.fit`) menggunakan *data training* (`X_train` dan `y_train`).

2.1 Predic Y

```
y_pred = model.predict(X_test)

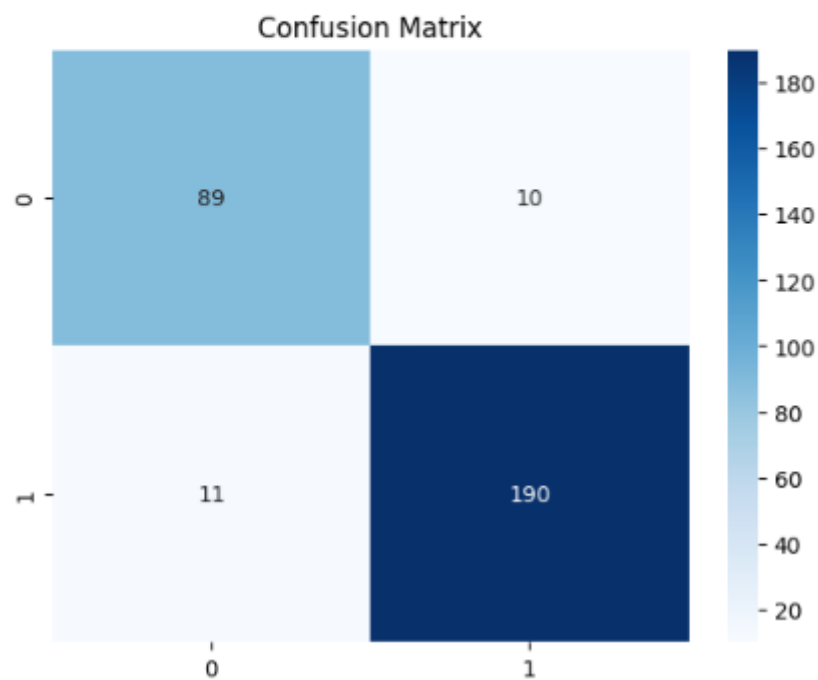
print("Akurasi :", accuracy_score(y_test, y_pred))
print("Confusion Matix", confusion_matrix(y_test, y_pred))
print("Laporan Klasifikasi", classification_report(y_test, y_pred))

sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d', cmap='Blues')
plt.title("Confusion Matrix")
plt.show()
```

```
Akurasi : 0.93
Confusion Matix [[ 89 10]
 [ 11 190]]
Laporan Klasifikasi
```

		precision	recall	f1-score	support
0	0.89	0.90	0.89	99	
1	0.95	0.95	0.95	201	
accuracy			0.93	300	
macro avg	0.92	0.92	0.92	300	
weighted avg	0.93	0.93	0.93	300	

		precision	recall	f1-score	support
0	0.89	0.90	0.89	99	
1	0.95	0.95	0.95	201	
accuracy			0.93	300	
macro avg	0.92	0.92	0.92	300	
weighted avg	0.93	0.93	0.93	300	



prediksi label kelas (y_{pred}) dari model terhadap fitur-fitur pada data uji (X_{test}). Kinerja model kemudian dievaluasi menggunakan tiga metrik kunci. Pertama, Akurasi dihitung menggunakan `accuracy_score`, yang mengukur proporsi total prediksi yang benar; hasilnya adalah 0.93 atau 93%

menunjukkan bahwa terdapat 89 *True Negatives* (TN) dan 198 *True Positives* (TP), serta hanya 10 *False Positives* (FP) dan 11 *False Negatives* (FN). Terakhir, *classification_report* digunakan untuk menyediakan metrik evaluasi yang lebih komprehensif seperti Presisi, Recall, dan F1-Score per kelas. Secara visual, matriks kebingungan divisualisasikan menggunakan *heatmap* dari pustaka Seaborn (*sns.heatmap*) untuk mempermudah interpretasi distribusi kesalahan klasifikasi

2.2 Dataframe

```
kolom = ['Usia', 'Status', 'Kelamin', 'Memiliki_Mobil', 'Penghasilan']
sample = pd.DataFrame([[30, 1, 0, 0, 150]], columns=kolom)

prediksi = model.predict(sample)
print("Prediksi:", "Membeli Mobil" if prediksi[0] == 1 else "Tidak Membeli Mobil")

Prediksi: Tidak Membeli Mobil
```

Berdasarkan nilai fitur yang diberikan dalam *DataFrame* sampel tersebut, model Regresi Logistik yang dikembangkan mengklasifikasikan subjek tersebut sebagai "Tidak Membeli Mobil". Hal ini mengindikasikan bahwa kombinasi antara usia 30, status (1), jenis kelamin (0), tidak memiliki mobil sebelumnya (0), dan tingkat penghasilan (150) berada di bawah ambang batas probabilitas yang ditetapkan oleh model untuk memicu keputusan klasifikasi positif (Membeli Mobil). Dengan kata lain, model memproyeksikan probabilitas pembelian pada sampel ini lebih mendekati kelas 0 (Tidak Membeli Mobil).

2.3 Estimasi probabilitas

```
kolom = ['Usia', 'Status', 'Kelamin', 'Memiliki_Mobil', 'Penghasilan']
sample = pd.DataFrame([[30, 1, 0, 0, 150]], columns=kolom)

prediksi = model.predict(sample)
print("Prediksi:", "Membeli Mobil" if prediksi[0] == 1 else "Tidak Membeli Mobil")
```

Prosedur ini melibatkan definisi variabel fitur (*kolom*) yang harus konsisten dengan urutan saat pelatihan model. Selanjutnya, sebuah observasi tunggal direpresentasikan dalam *DataFrame* (*sample*) dengan nilai Usia=30, Status=1, Kelamin=0, Memiliki_Mobil=0, dan Penghasilan=150.

DataFrame sampel ini kemudian diinputkan ke dalam model Regresi Logistik (*model.predict(sample)*) untuk menghasilkan inferensi, yaitu label kelas prediksi ($\{y\}$) yang disimpan dalam variabel *prediksi*. Berdasarkan hasil numerik yang dikembalikan, model mengklasifikasikan subjek ini ke dalam label kategorikal ("Membeli Mobil" jika $\{y\}=1$ atau "Tidak Membeli Mobil" jika $\{y\}=0$).

2.4 Model prediktif

```
prob = model.predict_proba(sample)
print(f"Peluang Tidak Membeli Mobil: {prob[0][0]:.2f}")
print(f"Peluang Membeli Mobil: {prob[0][1]:.2f}")

Peluang Tidak Membeli Mobil: 0.97
Peluang Membeli Mobil: 0.03
```

Prosedur ini menggunakan fungsi `model.predict_proba(sample)` untuk mendapatkan probabilitas posterior bahwa observasi sampel tersebut termasuk dalam masing-masing kelas target. Output dari fungsi ini adalah sebuah larik dua kolom: kolom pertama (`prob[0][0]`) merepresentasikan probabilitas milik kelas negatif (Tidak Membeli Mobil), dan kolom kedua (`prob[0][1]`) merepresentasikan probabilitas milik kelas positif (Membeli Mobil). Berdasarkan hasil yang dicetak, probabilitas Tidak Membeli Mobil adalah tinggi, yaitu 0.97, sementara probabilitas Membeli Mobil adalah rendah, yaitu 0.03. Angka-angka ini secara kuantitatif mendukung keputusan klasifikasi sebelumnya (Tidak Membeli Mobil), karena probabilitas untuk kelas 0 jauh melebihi ambang batas 0.5.