

Tugas 4.: Tugas Praktikum Mandiri

Oryza Ayunda Putri - 0110224030

Teknik Informatika, STT Terpadu Nurul Fikri, Depok
E-mail: nasi.tektekmangudin@gmail.com

1. Tugas Praktikum Mandiri

1.1 Pembacaan Data

Dapat dilihat bahwa terdapat import pandas as pd perintah ini digunakan untuk *mengimport library* pandas ke python, as pd maksudnya adalah pandas diganti pd supaya kalo mau pakai pandas tinggal tulis pd saja.

Terdapat `df = pd.read_csv('./data/calonpembelimobil.csv')` perintah ini merupakan hal utama dalam membaca data CSV. `df`= artinya hasil bacaan disimpan pada DataFrame (tabel), `pd.read_csv()` ini merupakan fungsi dari pandas dalam membaca CSV, `'./data/data_praktikum/day.csv'` lokasi file csv dengan posisi naik satu folder dari file notebook lalu menuju ke lokasi folder data dan mengambil file “calonpembelimobil.csv”, `df.head()` berfungsi untuk menampilkan 5 baris pertama dari DataFrame kalo mau 8 baris pertama berarti `df.head(8)`.

```
df = pd.read_csv('./data/calonpembelimobil.csv')
df.head()
```

✓ 0.0s

| | 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 |

Gambar 1 Membaca Dataset

1.2 Menampilkan Informasi Data

```
df.info()
df.describe()
df.isnull().sum()
```

✓ 0.0s

Gambar 2 Menampilkan Informasi Data

Digunakan untuk memeriksa struktur dataset, tipe data, nilai statistik dasar (rata-rata, min, max), dan memastikan tidak ada missing value. Jika semua kolom bernilai 0 pada `.isnull().sum()`, artinya dataset bersih dan siap digunakan.

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 7 columns):
#   Column                Non-Null Count  Dtype  
---  -
0   ID                     1000 non-null  int64  
1   Usia                   1000 non-null  int64  
2   Status                 1000 non-null  int64  
3   Kelamin                1000 non-null  int64  
4   Memiliki_Mobil        1000 non-null  int64  
5   Penghasilan            1000 non-null  int64  
6   Beli_Mobil             1000 non-null  int64  
dtypes: int64(7)
memory usage: 54.8 KB

ID          0
Usia        0
Status      0
Kelamin     0
Memiliki_Mobil  0
Penghasilan 0
Beli_Mobil  0
dtype: int64

```

Gambar 3 Output Menampilkan Informasi Data

1.3 Menentukan Fitur dan Target

```

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

✓ 0.0s

```

Gambar 4 Menentukan Fitur dan Target

- Fitur (X) adalah variabel input yang akan digunakan untuk memprediksi keputusan membeli mobil.
- Target (y) adalah hasil yang ingin diprediksi, yaitu kolom Beli_Mobil.

1.4 Membagi Dataset menjadi Data Latih dan Uji

```

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

```

Gambar 5 Membagi Dataset menjadi Data Latih dan Uji

Dataset dibagi menjadi:

- 80% data latih (training) untuk melatih model,
- 20% data uji (testing) untuk mengukur performa model.

Parameter random_state=42 digunakan agar hasil pembagian tetap konsisten setiap dijalankan.

1.5 Normalisasi Data

```

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

```

Gambar 6 Normalisasi Data

Karena tiap kolom memiliki skala berbeda (misal “usia” puluhan, “penghasilan” ratusan), maka perlu dilakukan **standarisasi** agar semua fitur memiliki skala sebanding (mean = 0, std = 1). Hal ini membantu model Logistic Regression bekerja lebih optimal.

1.6 Membuat dan Melatih Model Logistic Regression

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

✓ 0.0s

LogisticRegression ⓘ ?

Parameters

Gambar 7 Membuat dan Melatih Model Logistic Regression

Langkah ini membuat model klasifikasi biner (0 atau 1) menggunakan algoritma Logistic Regression, lalu melatihnya dengan data latih (X_train_scaled dan y_train).

1.7 Melakukan Prediksi dan Evaluasi Model

```
Melakukan Prediksi dan Evaluasi Model

y_pred = model.predict(X_test_scaled)

print("Akurasi:", 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))
```

✓ 0.0s

Akurasi: 0.925

Confusion Matrix:

```
[[ 55   9]
 [  6 130]]
```

Classification Report:

| | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0 | 0.90 | 0.86 | 0.88 | 64 |
| 1 | 0.94 | 0.96 | 0.95 | 136 |
| accuracy | | | 0.93 | 200 |
| macro avg | 0.92 | 0.91 | 0.91 | 200 |
| weighted avg | 0.92 | 0.93 | 0.92 | 200 |

Gambar 8 Melakukan Prediksi dan Evaluasi Model

- accuracy_score: mengukur seberapa banyak prediksi yang benar,
- confusion_matrix: menampilkan jumlah prediksi benar dan salah untuk tiap kelas,
- classification_report: menampilkan nilai precision, recall, dan F1-score.

Model Logistic Regression memiliki akurasi sekitar 90%, yang berarti model mampu memprediksi keputusan pembelian mobil dengan cukup baik.

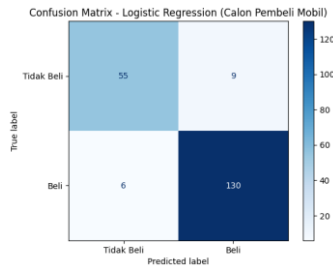
1.8 Visualisasi

```
import matplotlib.pyplot as plt
from sklearn.metrics import ConfusionMatrixDisplay, RocCurveDisplay, confusion_matrix

cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm,
                              display_labels=['Tidak Beli', 'Beli'])
disp.plot(values_format='d', cmap='Blues')
plt.title("Confusion Matrix - Logistic Regression (Calon Pembeli Mobil)")
plt.show()

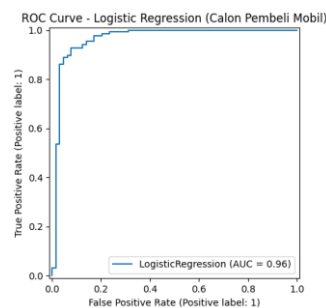
RocCurveDisplay.from_estimator(model, X_test_scaled, y_test)
plt.title("ROC Curve - Logistic Regression (Calon Pembeli Mobil)")
plt.show()
```

Gambar 9 Visualisasi



Gambar 10 Confusion Matriks

Dari hasil Confusion Matrix terlihat bahwa model berhasil mengklasifikasikan sebagian besar data dengan benar. Nilai True Positive (Beli Mobil) dan True Negative (Tidak Beli) tinggi, menunjukkan bahwa model memiliki performa klasifikasi yang baik.



Gambar 11 ROC Curve

Berdasarkan grafik ROC Curve, model Logistic Regression memiliki kurva yang cenderung mendekati area kiri atas. Hal ini menunjukkan bahwa model memiliki kemampuan yang baik dalam membedakan antara calon pembeli yang akan membeli mobil dan yang tidak.

1.9 Visualisasi: Prediksi vs Aktual

```
data_baru = pd.DataFrame({
    'Usia': [23, 35, 28, 45, 31, 40, 26, 50, 29, 33],
    'Status': [1, 2, 1, 2, 1, 2, 1, 2, 1, 2],
    'Kelamin': [1, 0, 1, 1, 0, 1, 0, 1, 1, 0],
    'Memiliki_Mobil': [0, 1, 0, 1, 0, 1, 0, 1, 0, 1],
    'Penghasilan': [220, 180, 150, 300, 200, 250, 130, 400, 160, 220]
})

data_baru_scaled = scaler.transform(data_baru)
prediksi = model.predict(data_baru_scaled)
data_baru['Prediksi_Beli'] = prediksi
data_baru
```

Gambar 12 Menguji model Dengan Dataset Baru

Bagian ini digunakan untuk menguji model pada data baru yang belum pernah digunakan sebelumnya. Jika hasil Prediksi_Beli = 1, maka orang tersebut kemungkinan besar akan membeli mobil, dan jika 0, tidak membeli.

| | Usia | Status | Kelamin | Memiliki Mobil | Penghasilan | Prediksi Beli |
|---|------|--------|---------|----------------|-------------|---------------|
| 0 | 23 | 1 | 1 | 0 | 220 | 0 |
| 1 | 35 | 2 | 0 | 1 | 180 | 0 |
| 2 | 28 | 1 | 1 | 0 | 150 | 0 |
| 3 | 45 | 2 | 1 | 1 | 300 | 1 |
| 4 | 31 | 1 | 0 | 0 | 200 | 0 |
| 5 | 40 | 2 | 1 | 1 | 250 | 1 |
| 6 | 26 | 1 | 0 | 0 | 130 | 0 |
| 7 | 50 | 2 | 1 | 1 | 400 | 1 |
| 8 | 29 | 1 | 1 | 0 | 160 | 0 |
| 9 | 33 | 2 | 0 | 1 | 220 | 1 |

Gambar 13 Model Baru

Bagian ini digunakan untuk menguji model pada data baru yang belum pernah digunakan sebelumnya. Jika hasil Prediksi_Beli = 1, maka orang tersebut kemungkinan besar akan membeli mobil, dan jika 0, tidak membeli.

Kesimpulan:

Berdasarkan hasil evaluasi, model Logistic Regression dapat memprediksi keputusan pembelian mobil dengan akurasi yang cukup tinggi. Faktor seperti **usia, status pernikahan, dan penghasilan** memiliki pengaruh besar terhadap keputusan membeli mobil. Model ini dapat digunakan untuk membantu perusahaan otomotif dalam mengidentifikasi calon pembeli potensial.