

# Praktikum 4 : Laporan Praktikum Mandiri Pertemuan 4

Sintia Sari - 0110222135 <sup>1\*</sup>

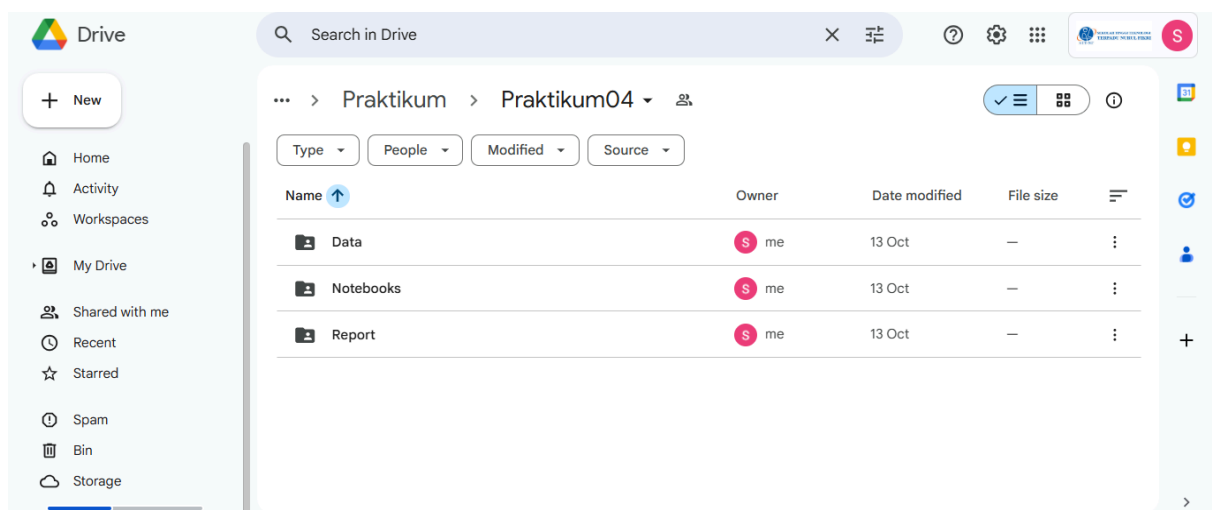
<sup>1</sup> Teknik Informatika, STT Terpadu Nurul Fikri, Depok

\*E-mail: [sint22135ti@student.nurulfikri.ac.id](mailto:sint22135ti@student.nurulfikri.ac.id)

**Abstract.** Praktikum ini menggunakan model Logistic Regression dan berhasil dibangun dan diuji dengan baik untuk memprediksi apakah seseorang akan membeli mobil atau tidak berdasarkan faktor usia, status, jenis kelamin, kepemilikan mobil, dan penghasilan. Hasil pengujian menunjukkan tingkat akurasi yang tinggi, yaitu sekitar 93% pada data uji dan 92,9% pada data baru, yang berarti model mampu mengenali pola dengan baik dan memberikan prediksi yang konsisten. Secara keseluruhan, model ini dapat digunakan sebagai alat bantu dalam menganalisis peluang pembelian mobil berdasarkan karakteristik calon pembeli.

## Praktikum Mandiri Pertemuan 3

### 1. Direktori program



Gambar 1. Direktori Program

Dalam pengerjaan tugas maupun praktikum, semua coding dikerjakan dalam folder notebooks dengan menggunakan file Python bernama PraktikumMandiri4\_Sintia Sari\_0110222135\_ML-Pagi.ipynb & Tugas04\_SintiaSari\_0110222135\_ML-Pagi.ipynb.

Dataset mentah ditempatkan pada folder data, adapun laporan maupun hasil visualisasi disimpan pada folder reports.

## 2. Import Library Python

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

*Gambar 2. Import Library Python*

Kode tersebut mengimpor library yang dibutuhkan untuk analisis data dan pembuatan model. `pandas` digunakan untuk mengolah data, `train_test_split` untuk membagi data, `LogisticRegression` untuk membuat model prediksi, dan fungsi dari `metrics` digunakan untuk mengevaluasi hasil model.

## 3. Menghubungkan dengan Google Drive

```
# Menghubungkan colab dengan google drive
from google.colab import drive
drive.mount('/content/gdrive')
import os

Mounted at /content/gdrive

# Memanggil dataset melalui gdrive
path = "gdrive/MyDrive/Machine Learning/Praktikum/Praktikum04/Data/"
```

*Gambar 3. Menghubungkan colab dengan Gdrive*

Penjelasan kode :

- `from google.colab import drive & drive.mount('/content/gdrive')`  
Menghubungkan Google Colab dengan Google Drive supaya file dataset bisa diakses langsung dari Drive.
- `import os`  
Digunakan untuk mengatur direktori/file di sistem (opsional, untuk navigasi folder).
- `path = "gdrive/MyDrive/Machine Learning/Praktikum/Praktikum04/Data/"`  
Menyimpan alamat folder tempat file dataset berada.

Setelah dijalankan, Colab akan menampilkan pesan `"Mounted at /content/gdrive"`, artinya Google Drive berhasil terhubung ke Colab dan bisa digunakan untuk membaca atau menyimpan file. Selain itu, variabel `path` menyimpan lokasi dataset sehingga bisa digunakan nanti.

## 4. Membaca file CSV

```
# Membaca file csv menggunakan pandas
import pandas as pd

df = pd.read_csv(path + 'calonpembelimobil.csv')
df.head()
```

*Gambar 4. Membaca dan Menampilkan Dataset*

Penjelasan kode :

- `import pandas as pd`  
Mengimpor library pandas dengan alias pd, digunakan untuk membaca dan mengolah data dalam bentuk tabel (DataFrame).
- `df = pd.read_csv(path + 'calonpembelimobil.csv')`  
Membaca file CSV bernama calonpembelimobil.csv yang ada di folder path, lalu menyimpannya ke dalam DataFrame dengan nama variabel df.
- `df.head()`

Menampilkan isi DataFrame lima baris teratas.

	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 5. Isi dari Dataset calonpembelimobil.csv*

Dataset berhasil dimuat dengan total data sebanyak 1000 baris dan 7 kolom, yaitu:

*Tabel 1. Nama dan Deskripsi Dataset*

Kolom	Deskripsi	Tipe Data
ID	Nomor unik calon pembeli	Integer
Usia	Umur calon pembeli	Integer
Status	0=single, 1=menikah, 2=menikah mempunyai anak, 3=duda/janda	Integer
Kelamin	1 = laki-laki, 0 = perempuan	Integer

Memiliki_Mobil	Jumlah mobil yang dimiliki	Integer
Penghasilan	Pendapatan calon pembeli dalam setahun	Integer
Beli_Mobil	1 = beli mobil, 0 = tidak	Integer

##### 5. Menentukan variabel dependen dan independen serta membagi dataset

```
# Pisahkan fitur (X) dan target (y)
x = df[['Usia', 'Status', 'Kelamin', 'Memiliki_Mobil', 'Penghasilan']]
y = df['Beli_Mobil']

# Bagi data menjadi data latih dan data uji
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
```

*Gambar 6. Menentukan variabel dependen dan independen serta membagi dataset*

Kode pada gambar diatas digunakan untuk menyiapkan data sebelum membuat model. Bagian pertama memisahkan data menjadi fitur (X) yaitu kolom **Usia**, **Status**, **Kelamin**, **Memiliki\_Mobil**, dan **Penghasilan**, serta target (y) yaitu kolom **Beli\_Mobil** yang ingin diprediksi. Selanjutnya, fungsi **train\_test\_split()** membagi data menjadi dua bagian, yaitu data latih (80%) untuk melatih model dan data uji (20%) untuk menguji hasil prediksi, dengan **random\_state=42** agar hasil pembagiannya selalu sama.

##### 6. Melatih Model Logistic Regression

```
# Buat model Logistic Regression
model = LogisticRegression()

# Latih model menggunakan data latih
model.fit(x_train, y_train)
```

LogisticRegression

LogisticRegression()

*Gambar 7. Melatih Model Logistic Regression*

Kode tersebut digunakan untuk membuat dan melatih model Logistic Regression. Baris pertama **model = LogisticRegression()** berfungsi untuk membuat objek model regresi logistik dari library scikit-learn. Model ini nantinya digunakan untuk memprediksi kemungkinan seseorang membeli mobil atau tidak berdasarkan data yang dimiliki. Baris kedua **model.fit(X\_train, y\_train)** berfungsi untuk melatih model menggunakan data latih, yaitu variabel **X\_train** (fitur) dan **y\_train** (target).

Dengan proses ini, model belajar mengenali pola hubungan antara fitur dan hasil agar bisa digunakan untuk membuat prediksi pada data baru.

## 7. Mengevaluasi Model dengan Data Uji

```
# Prediksi hasil menggunakan data uji
y_pred = model.predict(X_test)

# Evaluasi performa model
print("Akurasi Model :", accuracy_score(y_test, y_pred))
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("\nLaporan Klasifikasi:\n", classification_report(y_test, y_pred))
```

*Gambar 8. Mengevaluasi Model dengan Data Uji*

Kode tersebut digunakan untuk menguji dan mengevaluasi kinerja model. Baris `y_pred = model.predict(X_test)` berfungsi untuk memprediksi hasil berdasarkan data uji. Selanjutnya, fungsi `accuracy_score`, `confusion_matrix`, dan `classification_report` digunakan untuk menilai seberapa akurat model dalam memprediksi data.

```
Akurasi Model : 0.93
Confusion Matrix:
[[ 55   9]
 [   5 131]]
Laporan Klasifikasi:
              precision    recall  f1-score   support

     0       0.92       0.86       0.89        64
     1       0.94       0.96       0.95       136

 accuracy          0.93
 macro avg         0.93
 weighted avg      0.93
```

*Gambar 9. Hasil Evaluasi Model dengan Data Uji*

Dari hasilnya, akurasi model sebesar 0.93 atau 93%, yang berarti model cukup baik. Confusion matrix menunjukkan jumlah prediksi benar dan salah, sedangkan laporan klasifikasi menampilkan nilai precision, recall, dan f1-score yang juga menunjukkan performa model tergolong tinggi.

## 8. Membuat dataframe baru dengan dataset yang sama seperti yang sebelumnya

```
# Baca dataset baru
df_baru = pd.read_csv(path + 'calonpembelimobil.csv')

# Lihat 5 baris pertama dataset baru
df_baru.head()
```

*Gambar 10. Membaca dan menampilkan data*

Kode tersebut digunakan untuk membaca dan menampilkan isi dari dataset yang sama dengan dataframe sebelumnya. Baris `df_baru = pd.read_csv(path + 'calonpembelimobil.csv')` berfungsi untuk memuat file CSV bernama `calonpembelimobil.csv` ke dalam variabel `df_baru` menggunakan library pandas, sehingga data bisa diolah dalam bentuk tabel. Selanjutnya, perintah `df_baru.head()` digunakan untuk menampilkan 5 baris pertama dari dataset agar pengguna dapat melihat isi datanya.

	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 11. Isi Data dari Dataframe Baru*

Hasil yang muncul menunjukkan beberapa kolom seperti ID, Usia, Status, Kelamin, Memiliki\_Mobil, Penghasilan, dan Beli\_Mobil, yang berisi informasi calon pembeli mobil dan apakah mereka membeli mobil atau tidak. `df_baru` digunakan setelah model selesai dilatih, meskipun isi datanya sama dengan dataframe yang sebelumnya, tapi tujuan `df_baru` ini hanya untuk menguji model atau membuat prediksi baru dengan data yang baru atau dataset yang belum dibagi.

#### 9. Menggunakan Data Baru untuk Menguji Model

```
# Pastikan hanya kolom fitur yang digunakan untuk prediksi
X_baru = df_baru[['Usia', 'Status', 'Kelamin', 'Memiliki_Mobil', 'Penghasilan']]

# Prediksi apakah mereka akan membeli mobil atau tidak
prediksi_baru = model.predict(X_baru)

# Gabungkan hasil prediksi ke dataset baru
df_baru['Prediksi_Beli_Mobil'] = prediksi_baru

# Tampilkan hasil
df_baru.head()
```

*Gambar 12. Dataset Baru untuk Menguji Model*

Kode tersebut digunakan untuk melakukan prediksi pembelian mobil menggunakan model yang sudah dilatih sebelumnya. Pada baris pertama, `X_baru = df_baru[['Usia', 'Status', 'Kelamin', 'Memiliki_Mobil', 'Penghasilan']]` berfungsi untuk mengambil hanya kolom fitur yang digunakan dalam pelatihan model, agar

sesuai dengan data yang dikenali oleh model. Kemudian, baris `prediksi_baru = model.predict(X_baru)` digunakan untuk memprediksi apakah setiap calon pembeli akan membeli mobil atau tidak berdasarkan fitur tersebut. Selanjutnya, hasil prediksi disimpan ke kolom baru bernama `'Prediksi_Beli_Mobil'` menggunakan `df_baru['Prediksi_Beli_Mobil'] = prediksi_baru`. Terakhir, `df_baru.head()` menampilkan 5 baris pertama dari data.

	ID	Usia	Status	Kelamin	Memiliki_Mobil	Penghasilan	Beli_Mobil	Prediksi_Beli_Mobil
0	1	32	1	0	0	240	1	1
1	2	49	2	1	1	100	0	0
2	3	52	1	0	2	250	1	1
3	4	26	2	1	1	130	0	0
4	5	45	3	0	2	237	1	1

*Gambar 13. Dataset Baru untuk Menguji Model*

Hasil kode tersebut menampilkan tabel data awal dengan tambahan kolom `"Prediksi_Beli_Mobil"` yang menunjukkan hasil prediksi model, apakah seseorang akan membeli mobil (1) atau tidak (0) berdasarkan faktor seperti usia, status, jenis kelamin, kepemilikan mobil, dan penghasilan. Jika dibandingkan dengan kolom `"Beli_Mobil"`, sebagian besar hasil prediksi sesuai dengan data sebenarnya, menandakan model sudah bekerja dengan baik. Namun, jika terdapat perbedaan nilai, berarti model masih memiliki sedikit kesalahan dalam memprediksi beberapa data.

## 10. Evaluasi Performa Model

```
# Evaluasi performa model pada df_baru
print("Akurasi Model pada df_baru :", accuracy_score(df_baru['Beli_Mobil'], df_baru['Prediksi_Beli_Mobil']))
print("\nConfusion Matrix pada df_baru:\n", confusion_matrix(df_baru['Beli_Mobil'], df_baru['Prediksi_Beli_Mobil']))
print("\nLaporan Klasifikasi pada df_baru:\n", classification_report(df_baru['Beli_Mobil'], df_baru['Prediksi_Beli_Mobil']))
```

*Gambar 14. Evaluasi Performa Model*

Kode di atas digunakan untuk mengevaluasi kinerja model pada dataset baru (`df_baru`). Baris pertama menghitung akurasi menggunakan fungsi `accuracy_score`, yang menunjukkan seberapa banyak prediksi model sesuai dengan data sebenarnya. Baris kedua menampilkan `confusion matrix`, yaitu tabel yang memperlihatkan jumlah prediksi benar dan salah untuk setiap kelas (membeli atau tidak membeli mobil). Baris ketiga menampilkan laporan klasifikasi yang berisi nilai precision, recall, dan f1-score untuk setiap kelas.

```

Akurasi Model pada df_baru : 0.929

Confusion Matrix pada df_baru:
[[334  33]
 [ 38 595]]

Laporan Klasifikasi pada df_baru:

```

	precision	recall	f1-score	support
0	0.90	0.91	0.90	367
1	0.95	0.94	0.94	633
accuracy			0.93	1000
macro avg	0.92	0.93	0.92	1000
weighted avg	0.93	0.93	0.93	1000

*Gambar 15. Hasil Evaluasi Performa Model*

Hasilnya menunjukkan akurasi model sebesar 0.929 atau sekitar 92,9%, artinya model mampu memprediksi dengan sangat baik. Confusion matrix menunjukkan sebagian besar data diprediksi dengan benar (334 + 595), hanya ada sedikit kesalahan (33 dan 38). Nilai precision, recall, dan f1-score yang tinggi pada kedua kelas juga menandakan bahwa model bekerja dengan baik dalam mengenali siapa yang akan membeli mobil dan siapa yang tidak.

Perbedaan utama antara hasil evaluasi model pada `df_baru` dan data uji sebelumnya terletak pada tujuan pengujian dan akurasinya. Data uji sebelumnya digunakan untuk menilai kemampuan model terhadap data hasil pembagian dataset, dengan akurasi sekitar 93%. Sementara `df_baru` digunakan untuk menguji model pada data baru, dengan akurasi 92,9%. Hasil yang hampir sama menunjukkan bahwa model konsisten dan mampu menggeneralisasi dengan baik terhadap data baru.

## References

Link Github :

- 1) Praktikum dikelas :

[https://github.com/ssintyaaa/MachineLearning/blob/main/praktikum%2004/notebooks/Praktikum04\\_Sintia\\_Sari\\_0110222135\\_ML\\_Pagi%20\(1\).ipynb](https://github.com/ssintyaaa/MachineLearning/blob/main/praktikum%2004/notebooks/Praktikum04_Sintia_Sari_0110222135_ML_Pagi%20(1).ipynb)

- 2) Praktikum mandiri :

[https://github.com/ssintyaaa/MachineLearning/blob/main/praktikum%2004/notebooks/Tugas04\\_Sintia\\_Sari\\_0110222135\\_ML\\_Pagi%20\(1\).ipynb](https://github.com/ssintyaaa/MachineLearning/blob/main/praktikum%2004/notebooks/Tugas04_Sintia_Sari_0110222135_ML_Pagi%20(1).ipynb)

Link Gdrive :

 Praktikum04