

Tugas 2: Machine Learning – Regresi Logistik

Muhammad Zaidan Ramdhan - 0110222040

Teknik Informatika, STT Terpadu Nurul Fikri, Depok

*E-mail: muha22040ti@student.nurulfikri.ac.id

Laporan ini bertujuan membahas adalah Regresi Logistik (*Logistic Regression*), sebuah algoritma klasifikasi yang sangat fundamental. Regresi Logistik digunakan untuk memodelkan probabilitas suatu hasil biner (dua kategori), yang dalam kasus ini adalah keputusan Beli Mobil (1) atau Tidak Beli Mobil (0). Dengan menggunakan fitur-fitur calon pembeli seperti Usia, Status, Kelamin, Kepemilikan Mobil, dan Penghasilan, model ini mempelajari hubungan antara variabel-variabel tersebut dan probabilitas pembelian. Secara matematis, ia menggunakan fungsi *sigmoid* untuk memetakan hasil kombinasi linier fitur ke dalam rentang 0 hingga 1. Setelah dilatih menggunakan data historis (data latih), model Regresi Logistik ini kemudian dapat digunakan untuk membuat prediksi pada calon pembeli baru, menentukan siapa yang paling mungkin menjadi konsumen mobil di masa depan.

1. Tugas mandiri – membuat model Logistic Regression

```
1. Melakukan mounting G-Drive

from google.colab import drive
drive.mount("/content/gdrive")

Mounted at /content/gdrive
```

Gambar 1. 1

Pada *Gambar 1.1* di atas, merupakan sebuah code untuk mounted atau menghubungkan google colab dengan google drive.

```
2. Membuat path untuk dataset

path = "/content/gdrive/MyDrive/machine_learning/pertemuan04"
```

Gambar 1. 2

Pada *Gambar 1.2* di atas, kita membuat sebuah path untuk mengatur folder yang akan digunakan, dalam case ini kita menggunakan folder pertemuan04.

3. Load dataset

```
import pandas as pd

df = pd.read_csv(path + "/praktikum_mandiri/data/calonpembelimobil.csv")
df.head()
```

	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 1.3

Pada *Gambar 1.3* di atas, merupakan code untuk menampilkan dataset pada sebuah tabel menggunakan dataset calonpembelimobil.csv dan hanya menampilkan 5 data.

- **df = pd.read_csv(path + '/praktikum_mandiri/data/calonpembelimobil.csv)**
merupakan sebuah code untuk membaca file csv yang terdapat pada folder data dengan diikuti file calonpembelimobil.csv.

4. Import library yang dibutuhkan

```
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
```

Gambar 1.4

Pada *Gambar 1.4* di atas, merupakan sebuah code untuk import beberapa library yang akan kita gunakan.

5. Memisahkan variabel X dan Y

```
X = df.drop(['ID', 'Beli_Mobil'], axis=1)
y = df['Beli_Mobil']
```

Gambar 1. 5

Pada Gambar 1.5 di atas, merupakan sebuah code untuk memisahkan variabel X dan Y.

6. Membagi dataset menjadi data training dan testing

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

Gambar 1. 6

Pada Gambar 1.5 di atas, merupakan sebuah code untuk membagi dataset menjadi data training dan data testing.

7. Buat model LogisticRegression

```
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)
```

▼

LogisticRegression ⓘ ?

LogisticRegression(max_iter=1000)

Gambar 1. 7

Pada *Gambar 1.7* di atas, membuat model LogisticRegression.

```
8. Uji model dengan data testing

y_pred = model.predict(X_test)
```

Gambar 1. 8

Pada *Gambar 1.8* di atas, merupakan sebuah code untuk menguji model dengan data testing.

```
10. Evaluasi Model dengan Data Uji

print(f"Akurasi Model: {akurasi:.4f}")
print("Confusion Matrix:")
print(conf_mat)

Akurasi Model: 0.9300
Confusion Matrix:
[[ 55   9]
 [  5 131]]
```

Gambar 1. 9

Pada *Gambar 1.9* di atas, merupakan sebuah code untuk evaluasi model ddengan data uji atau testing.

11. Membuat data calon pembeli baru

```
data_baru = pd.DataFrame({  
    'Usia': [30, 60, 25],  
    'Status': [1, 2, 3], # Misalnya 1=Menikah, 2=Cerai, 3=Lajang  
    'Kelamin': [1, 0, 1], # 1=Pria, 0=Wanita  
    'Memiliki_Mobil': [0, 1, 0], # 0=Tidak, 1=Ya  
    'Penghasilan': [150, 400, 80]  
})
```

Gambar 1. 10

Pada *Gambar 1.10* di atas, merupakan sebuah kode membuat data calon pembeli baru.

13. Gabungkan hasil prediksi dengan data baru untuk tampilan

```
hasil_prediksi = data_baru.copy()
hasil_prediksi['Prediksi_Beli_Mobil'] = prediksi_baru
hasil_prediksi['Probabilitas_Beli'] = prediksi_proba_baru

print("\n--- Prediksi pada Calon Pembeli Baru ---")
print(hasil_prediksi)
```

```
--- Prediksi pada Calon Pembeli Baru ---
   Usia  Status  Kelamin  Memiliki_Mobil  Penghasilan  Prediksi_Beli_Mobil  \
0    30      1        1           0         150             0
1    60      2        0           1         400             1
2    25      3        1           0          80             0

   Probabilitas_Beli
0          0.007904
1          0.999991
2          0.000078
```

Gambar 1. 11

Pada Gambar 1.11 di atas, merupakan code untuk menampilkan prediksi pada calon pembeli baru.

Link github praktikum:

https://github.com/MuhZaidanRamdhan/machine-learning/blob/main/pertemuan03/praktikum_04/notebooks/praktikum_04.ipynb

Link github tugas:

https://github.com/MuhZaidanRamdhan/machine-learning/blob/main/pertemuan04/praktikum_mandiri/notebooks/latihan_04.ipynb