

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

Mounted at /content/drive

▼ Implementasi Naive Bayes

Dataset yang digunakan berisi rincian pengguna di situs jejaring sosial untuk menemukan apakah pengguna membeli produk dengan mengklik iklan di situs berdasarkan gaji, usia, dan jenis kelamin mereka.

Mari kita mulai pemrograman dengan mengimpor library penting yang diperlukan

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import sklearn
```

▼ Importing dataset

```
dataset = pd.read_csv('/content/drive/MyDrive/MATERI/Pembelajaran Mesin/Praktikum Genap 20212022/Social_Network_Ads.csv')
X = dataset.iloc[:, [1, 2, 3]].values
y = dataset.iloc[:, -1].values
```

Karena dataset berisi variabel karakter, maka harus dilakukan encode menggunakan LabelEncoder.

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
X[:,0] = le.fit_transform(X[:,0])
```

▼ Train test Split

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 0)
```

▼ Feature scaling

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

▼ Melatih model Naive Bayes di set pelatihan

```
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, y_train)
```

GaussianNB()

Mari kita prediksi hasil tes

```
y_pred = classifier.predict(X_test)
```

▼ Nilai yang diprediksi dan aktual -

y_pred

```
array([0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1,
       0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1])
```

y_test

```
array([0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1,
       0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1,
       0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1])
```

Untuk 8 nilai pertama, keduanya sama. Kita dapat mengevaluasi matriks menggunakan confusion matriks dan skor akurasi dengan membandingkan nilai tes yang diprediksi dan aktual.

```
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
ac = accuracy_score(y_test, y_pred)
print (cm)
print (ac)
```

```
[[56  2]
 [ 4 18]]
0.925
```