

# MODUL

# STMIK WIDYA PRATAMA



## DATA MINING

### Supervised Learning 1



#### DESKRIPSI MATAKULIAH

#### CAPAIAN PEMBELAJARAN

#### DOSEN PENGAMPU

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN  
KOMPUTER**  
**(STMIK) WIDYA PRATAMA**

## CAPAIAN PEMBELAJARAN

## MATERI PEMBELAJARAN

## 1. Perhitungan Python Naïve Bayes Classifier

Kasus yang akan digunakan adalah dataset covid19 dengan data yang sudah diubah dari kategorik menjadi numeric. Langkah pertama adalah mengimport library yang akan digunakan, sebagai berikut :

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn import metrics
```

Keterangan kode :

- `import pandas as pd` biasa digunakan untuk mengubah dimensi data, membuat tabel, memeriksa data, membaca data dan lain sebagainya.
- `import numpy as np` berfungsi untuk memudahkan operasi perhitungan tipe data numeric seperti penjumlahan, perkalian, pengurangan, pemangkatan dan operasi aritmatika lainnya
- Sklearn merupakan library yang didalamnya terdapat banyak algoritme dan pekerjaan, seperti klasifikasi, regresi, clustering, preprocessing, dimensionality reduction, model selection dan feature selection
- `from sklearn.model_selection import train_test_split` memisahkan antara data training dan data testing
- `from sklearn import metrics` untuk memanggil berbagai metric yang akan digunakan seperti MAE, MSE, accuracy dan yang lainnya

```
df=pd.read_csv('/content/drive/MyDrive/Data Mining/studi_num.csv')
df
```

Keterangan kode :

- `df = pd.read_csv("/content/drive/MyDrive/Machine Learning/Supervised Learning/Regresion/nilai.csv")`  
membaca dataset yang sudah disediakan dalam format csv. Simpan dataset di folder yang sama dengan folder proyek yang sedang dikerjakan. Jika tidak, maka jalur data harus dijelaskan seperti "D/data/studi\_num.csv"
- Separator berfungsi untuk menjelaskan pemisah pada dataset. Jika menyimpan dataset dalam bentuk csv dan dipisahkan dengan koma, maka pilih seperti pada kode di atas. Jika bukan, maka ubah tanda koma dengan separator yang digunakan seperti ; atau tab

- `df.head()` digunakan untuk menampilkan sebanyak 5 data teratas. Hasilnya sebagai berikut :

	JURUSAN	GENDER	ASAL_SEKOLAH	RERATA_SKS	ASISTEN	LAMA_STUDI
0	1	1	1	1	1	1
1	1	1	1	1	1	1
2	1	1	1	1	1	1
3	2	2	1	1	1	1
4	2	2	2	1	1	1
5	2	1	2	1	1	1
6	2	1	1	1	2	2
7	1	1	1	2	1	1
8	1	2	1	2	1	1
9	2	1	1	1	2	2
10	3	1	1	1	1	1
11	2	1	2	2	1	1
12	2	2	1	2	1	2

```
X = df.iloc[:, :-1].values
y = df.iloc[:, 5].values
```

Keterangan kode :

- Menentukan fitur predictor yang akan diwakili oleh X
- Menentukan fitur target yang akan diwakili oleh y

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1)
```

Keterangan kode :

- `from sklearn.model_selection import train_test_split` Memanggil fungsi untuk memisahkan dataset kedalam dua bagian yaitu data training dan data testing
- `X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)` Mengatur hyperparameter untuk data training dan data testing. Ukuran data testing adalah 0.2

Mengatur hyperparameter untuk data training dan data testing. Ukuran data testing adalah 0.2

artinya 30% dari keseluruhan dataset. Random\_state = 0 artinya pemilihan data testing tidak akan berubah setiap kali mengatur nilainya dengan 1.

```
from sklearn.naive_bayes import GaussianNB
```

Keterangan kode :

- Memanggil algoritma Naïve Bayes Classifier untuk digunakan nanti

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

Keterangan kode :

- `model = GaussianNB()` Memanggil algoritma Naïve Bayes Classifier untuk digunakan oleh variable model
- `model.fit(X_train, y_train)` untuk melakukan pemodelan terhadap dataset yang sudah diatur sebelumnya

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

Keterangan kode :

- Melakukan pengujian terhadap data testing sebanyak 30%
- `y_pred` untuk menampilkan hasil prediksi. Hasilnya adalah `array([1, 2, 1, 1])`

```
np.array(y_test)
```

Keterangan kode :

- Untuk menampilkan nilai actual dari data training yaitu `array ([1, 1, 1, 1])`
- Hasil pada baris ini dan baris sebelumnya terdapat satu prediksi yang berbeda, seharunya bernilai 1, sedangkan prediksinya bernilai 2

```
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

Keterangan kode :

- Menghitung nilai akurasi berdasarkan pengujian model terhadap data testing. Hasilnya adalah  
*Accuracy : 0.75*

## TES FORMATIF