

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 KNN

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/Dataset/gizi.csv')
df
```

	Tinggi	Berat	L Perut	L Panggul	Lemak	Label
0	160.0	70	78.0	99.0	33.3	3
1	162.0	56	74.0	90.0	31.7	3
2	155.0	63	76.5	95.5	37.8	3
3	156.0	54	74.0	88.0	31.0	2
4	155.0	55	79.0	88.0	27.0	3
5	155.0	55	67.0	91.0	29.8	2
6	151.5	58	76.0	94.0	31.6	3
7	151.5	62	79.0	98.0	37.3	3
8	159.0	49	72.0	89.0	28.7	2
9	151.0	58	77.0	99.0	34.4	3
10	153.0	52	72.0	89.0	31.0	2
11	159.0	49	65.0	87.0	24.6	2

Keterangan kode :

```
df=pd.read_csv('/content/drive/MyDrive/Dataset/gizi.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/gizi.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 :

```
df.head()
```

	Tinggi	Berat	L Perut	L Panggul	Lemak	Label
0	160.0	70	78.0	99.0	33.3	3
1	162.0	56	74.0	90.0	31.7	3
2	155.0	63	76.5	95.5	37.8	3
3	156.0	54	74.0	88.0	31.0	2
4	155.0	55	79.0	88.0	27.0	3

```
X=df[['Tinggi', 'Berat', 'L Perut', 'L Panggul', 'Lemak']]  
y=df['Label']
```

Keterangan kode :

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

```
print(X)
```

	Tinggi	Berat	L Perut	L Panggul	Lemak
0	160.0	70	78.0	99.0	33.3
1	162.0	56	74.0	90.0	31.7
2	155.0	63	76.5	95.5	37.8
3	156.0	54	74.0	88.0	31.0
4	155.0	55	79.0	88.0	27.0
5	155.0	55	67.0	91.0	29.8
6	151.5	58	76.0	94.0	31.6
7	151.5	62	79.0	98.0	37.3
8	159.0	49	72.0	89.0	28.7
9	151.0	58	77.0	99.0	34.4
10	153.0	52	72.0	89.0	31.0
11	159.0	49	65.0	87.0	24.6

```
print(y)
```

```
0      3  
1      3  
2      3  
3      2  
4      3  
5      2  
6      3  
7      3  
8      2  
9      3  
10     2  
11     2  
Name: Label, dtype: int64
```

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

```
from sklearn.neighbors import KNeighborsClassifier
```

Keterangan kode :

- Memanggil algoritma KNN untuk digunakan nanti

```
model = KNeighborsClassifier(n_neighbors=3, weights='distance', metric='euclidean')
model.fit(X_train, y_train)
```

Keterangan kode :

- ```
model = KNeighborsClassifier(n_neighbors=3, weights='distance', metric='euclidean')
```

 Memanggil algoritma KNN Classifier untuk digunakan oleh variable model, dengan penentuan nilai K = 3
- ```
model.fit(X_train, y_train)
```

 untuk melakukan pemodelan terhadap dataset yang sudah diatur sebelumnya

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

Keterangan kode :

- Melakukan pengujian terhadap data testing sebanyak 30%

Mencoba dengan data baru (data testing)

Tinggi : 159

Berat : 49

L Perut : 65

L Panggul : 87

Lemak : 24.6

```
print(model.predict([[159, 49, 65, 87, 24.6]]))
```

[2]

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

Keterangan kode :

- Menghitung nilai akurasi berdasarkan pengujian model terhadap data testing. Hasilnya adalah *Accuracy : 1.0*



TES FORMATIF