

Lembar Jawaban UTS

Machine Learning

Nama : Farhan Mualif

NIM : 5210411219

Kelas : H

Pernyataan:

Dengan mengerjakan ujian ini, maka saya menyatakan bahwa semua jawaban **SAYA KERJAKAN SENDIRI** tanpa bekerjasama maupun meminta bantuan siapapun. Saya bersedia diberi **nilai E** jika pernyataan ini terbukti salah.

Tanda tangan:

1. Untuk mengambil data dalam format csv kita menggunakan perintah `read_csv('nama_file')`, selanjutnya perintah tersebut kita tampung dengan variabel `df`, lalu untuk menampilkan data tersebut kita menggunakan perintah `print(df)`. Berikut code dan output yang dihasilkan:

Code:

```
# membaca data
df = pd.read_csv('Iris.csv')
print(df)
```

Output:

| | Id | SepalLengthCm | SepalWidthCm | PetalLengthCm | PetalWidthCm | Species |
|-----|-----|---------------|--------------|---------------|--------------|----------------|
| 0 | 1 | 5.1 | 3.5 | 1.4 | 0.2 | Iris-setosa |
| 1 | 2 | 4.9 | 3.0 | 1.4 | 0.2 | Iris-setosa |
| 2 | 3 | 4.7 | 3.2 | 1.3 | 0.2 | Iris-setosa |
| 3 | 4 | 4.6 | 3.1 | 1.5 | 0.2 | Iris-setosa |
| 4 | 5 | 5.0 | 3.6 | 1.4 | 0.2 | Iris-setosa |
| .. | ... | ... | ... | ... | ... | ... |
| 145 | 146 | 6.7 | 3.0 | 5.2 | 2.3 | Iris-virginica |
| 146 | 147 | 6.3 | 2.5 | 5.0 | 1.9 | Iris-virginica |
| 147 | 148 | 6.5 | 3.0 | 5.2 | 2.0 | Iris-virginica |
| 148 | 149 | 6.2 | 3.4 | 5.4 | 2.3 | Iris-virginica |
| 149 | 150 | 5.9 | 3.0 | 5.1 | 1.8 | Iris-virginica |

2. Untuk mengambil fitur (x) kita melakukan perintah code berikut:

```
# mengambil fitur (x) dan label (y)
x = df.drop(['Id', 'Species'], axis=1)
```

Arti dari code diatas adalah menghapus kolom Id, dan Species, dan axis=1 memiliki arti bahwa yang akan kita hapus adalah kolom, setelah kita melakukan hal tersebut maka kolom yang tersisa adalah ['SepalLengthCm', 'SepalWidthCm','PetalLengthCm' ,'PetalWidthCm']. Selanjutnya untuk mengambil label (y), kita cukup menggunakan perintah code berikut:

```
y = df['Species']
```

Code diatas berarti kita hanya mengambil data pada kolom Species saja

3. Sebelum kita melakukan mambagi data, kita harus mengimport module *train_test_split* terlebih dahulu seperti code berikut

```
from sklearn.model_selection import train_test_split
```

untuk membagi data latih dengan data uji kita menggunakan perintah code berikut:

```
# Membagi dataset menjadi data latih dan data uji
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=1)
```

Variabel *x_train* dan *y_train* berguna untuk menampung data latih, sedangkan *y_test* dan *x_test* berguna untuk menampung data uji, selanjutnya untuk parameter *test_size* berguna untuk menentukan data uji sebesar 0.3 atau 30%

4. Sebelum melakukan normalisasi kita harus import module *MinMaxScaler* terlebih dahulu

```
from sklearn.preprocessing import MinMaxScaler
```

Selanjutnya kita melakukan code seperti berikut:

```
# melakukan normalisasi
scaler = MinMaxScaler()
x_train_scaled = scaler.fit_transform(x_train)
x_test_scaled = scaler.transform(x_test)
```

Variabel *scaler* berguna untuk menampung *MinMaxScaler*, variabel *x_train_scaled* berguna untuk menampung data latih yang akan di normalisasi, dan *x_test_scaled* berguna untuk menampung data uji yang akan dinormalisasi

5. Sebelum melakukan proses untuk melatih model KNN kita harus melakukan import module *KNeighborsClassifier* dengan cara berikut:

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

Setelah melakukan import selanjutnya lakukan coding berikut:

```
# proses training KNN
KNN = KNeighborsClassifier(n_neighbors=5)
KNN.fit(x_train_scaled, y_train)
```

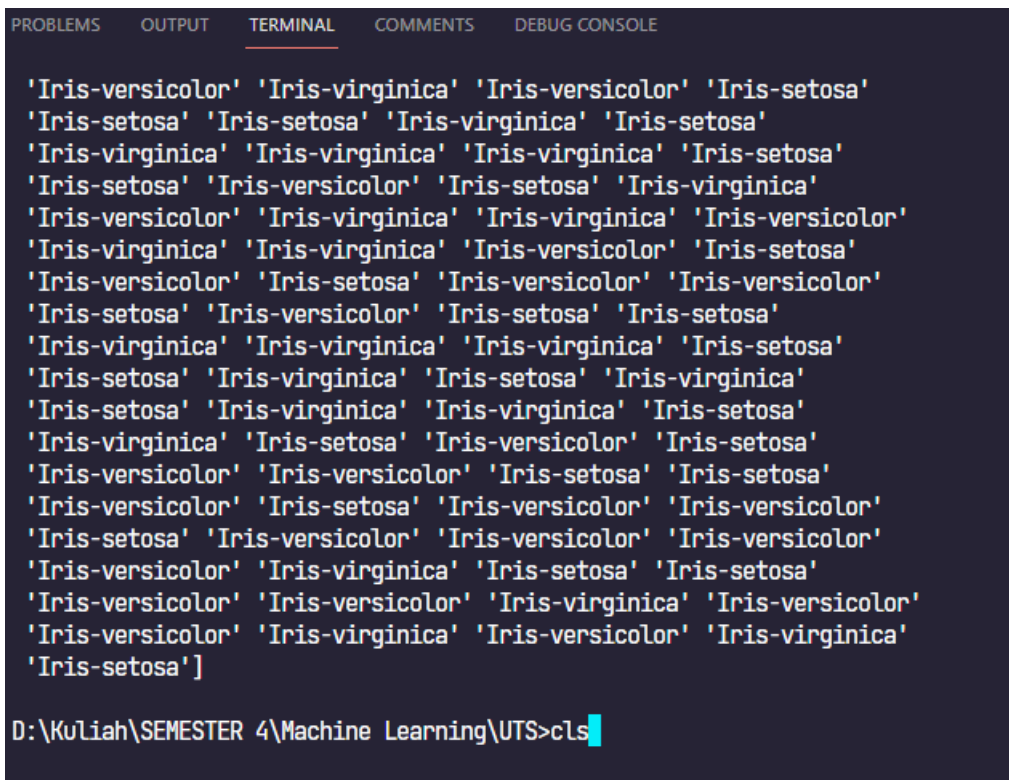
Variabel KNN diatas berguna untuk menampung fungsi *KNeighborsClassifier* dengan parameter *n_neighbors=5* yang berarti saat melakukan klasifikasi, model akan mencari 5 tetangga terdekat

untuk digunakan sebagai acuan dalam menentukan label kelas untuk data yang sedang diuji. Selanjutnya `KNN.fit(x_train_scaled, y_train)` digunakan untuk melatih model KNN dengan parameter pertama yaitu `x_train_scaled` yang merupakan data latih yang telah dinormalisasi dan parameter kedua yaitu `y_train` yang merupakan data kelas dari data latih.

6. Untuk melakukan prediksi label kelas, lakukan code seperti berikut:

```
y_prediction = KNN.predict(x_train_scaled)
print('hasil prediksi ', y_prediction)
```

Pada fungsi predict diatas terdapat paramater yang merupakan data latih yang sebelumnya sudah di normalisasi. Berikut merupakan output hasil prediksi yang berupa label yang telah ditentukan berdasarkan algoritma KNN:



```
PROBLEMS OUTPUT TERMINAL COMMENTS DEBUG CONSOLE

'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor' 'Iris-setosa'
'Iris-setosa' 'Iris-setosa' 'Iris-virginica' 'Iris-setosa'
'Iris-virginica' 'Iris-virginica' 'Iris-virginica' 'Iris-setosa'
'Iris-setosa' 'Iris-versicolor' 'Iris-setosa' 'Iris-virginica'
'Iris-versicolor' 'Iris-virginica' 'Iris-virginica' 'Iris-versicolor'
'Iris-virginica' 'Iris-virginica' 'Iris-versicolor' 'Iris-setosa'
'Iris-versicolor' 'Iris-setosa' 'Iris-versicolor' 'Iris-versicolor'
'Iris-setosa' 'Iris-versicolor' 'Iris-setosa' 'Iris-setosa'
'Iris-virginica' 'Iris-virginica' 'Iris-virginica' 'Iris-setosa'
'Iris-setosa' 'Iris-virginica' 'Iris-setosa' 'Iris-virginica'
'Iris-setosa' 'Iris-virginica' 'Iris-virginica' 'Iris-setosa'
'Iris-virginica' 'Iris-setosa' 'Iris-versicolor' 'Iris-setosa'
'Iris-versicolor' 'Iris-versicolor' 'Iris-setosa' 'Iris-setosa'
'Iris-versicolor' 'Iris-setosa' 'Iris-versicolor' 'Iris-versicolor'
'Iris-setosa' 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor'
'Iris-versicolor' 'Iris-virginica' 'Iris-setosa' 'Iris-setosa'
'Iris-versicolor' 'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor'
'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor' 'Iris-virginica'
'Iris-setosa']

D:\Kuliah\SEMESTER 4\Machine Learning\UTS>cls
```

7. Sebelum melakukan penghitungan akurasi model, kita diharuskan untuk melakukan import modul `accuracy_score` terlebih dahulu seperti pada code berikut:

```
from sklearn.metrics import accuracy_score
```

Setelah melakukan import lakukan code seperti berikut:

```
# akurasi model
accuracy = accuracy_score(y_test, y_prediction)
print('akurasi model: ', accuracy)
```

fungsi `accuracy_score` menerima dua parameter, parameter pertama adalah variabel `y_test` yang mana variabel tersebut adalah data uji, parameter kedua merupakan variabel `y_prediction` yang mana variabel tersebut merupakan data hasil prediksi dari algoritma KNN.

Berikut merupakan code secara lengkap:

```
Welcome  main.py x
main.py > ...
1  import pandas as pd
2  from sklearn.model_selection import train_test_split
3  from sklearn.preprocessing import MinMaxScaler
4  from sklearn.neighbors import KNeighborsClassifier
5  from sklearn.metrics import accuracy_score
6
7  # membaca data
8  df = pd.read_csv('Iris.csv')
9
10 # Memisahkan antara fitur (x) dan label (y)
11 x = df.drop(['Id', 'Species'], axis=1)
12 y = df['Species']
13
14 # Membagi dataset menjadi data latih dan data uji
15 x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=1)
16
17 # melakukan normalisasi
18 scaler = MinMaxScaler()
19 x_train_scaled = scaler.fit_transform(x_train)
20 x_test_scaled = scaler.transform(x_test)
21
22 # proses training KNN
23 KNN = KNeighborsClassifier(n_neighbors=5)
24 KNN.fit(x_train_scaled, y_train)
25
26 # prediksi
27 y_prediction = KNN.predict(x_test_scaled)
28 print('hasil prediksi: ',y_prediction)
29
30 # akurasi model
31 accuracy = accuracy_score(y_test, y_prediction)
32 print('akurasi model: ',accuracy)
```

Dan berikut output dari program diatas:

```
D:\Kuliah\SEMESTER 4\Machine Learning\UTS>C:/Python311/python.exe "d:/Kuliah/SEMESTER 4,
hasil prediksi:
['Iris-setosa' 'Iris-versicolor' 'Iris-versicolor' 'Iris-setosa'
'Iris-virginica' 'Iris-versicolor' 'Iris-virginica' 'Iris-setosa'
'Iris-setosa' 'Iris-virginica' 'Iris-versicolor' 'Iris-setosa'
'Iris-virginica' 'Iris-versicolor' 'Iris-versicolor' 'Iris-setosa'
'Iris-versicolor' 'Iris-versicolor' 'Iris-setosa' 'Iris-setosa'
'Iris-versicolor' 'Iris-versicolor' 'Iris-virginica' 'Iris-setosa'
'Iris-virginica' 'Iris-versicolor' 'Iris-setosa' 'Iris-setosa'
'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor' 'Iris-virginica'
'Iris-versicolor' 'Iris-virginica' 'Iris-virginica' 'Iris-setosa'
'Iris-versicolor' 'Iris-setosa' 'Iris-versicolor' 'Iris-virginica'
'Iris-virginica' 'Iris-setosa' 'Iris-versicolor' 'Iris-virginica'
'Iris-versicolor']
akurasi model:  0.9555555555555556

D:\Kuliah\SEMESTER 4\Machine Learning\UTS>
```