# Pembelajaran Mesin
# ( Praktikum )
# TI -C4

FAKULTAS VOKASI
UNIVERSITAS AIRLANGGA

[152111283042] | [NELA ANJANI P P A] | [05 November 2023]

**TUGAS INDIVIDU DATA IRIS SVM**

1. menampilkan data iris

```
import numpy as np
import pandas as pd
from sklearn import datasets
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split,
cross_val_score
from sklearn.svm import SVC

df = pd.read_csv('IRIS.csv')
print(" Data Awal ".center(100, "="))
print(df)
print("==========================================================
====")

# Separate features (x) and target (y)
X =
df.drop(columns=['sepal_length','sepal_width','petal_length','p
etal_width'])
print(X)
y = df['species']
print(y)
```

Hasil

```
==================================== Data Awal ====================================
     sepal_length  sepal_width  petal_length  petal_width         species
0             5.1          3.5           1.4          0.2     Iris-setosa
1             4.9          3.0           1.4          0.2     Iris-setosa
2             4.7          3.2           1.3          0.2     Iris-setosa
3             4.6          3.1           1.5          0.2     Iris-setosa
4             5.0          3.6           1.4          0.2     Iris-setosa
..            ...          ...           ...          ...             ...
145           6.7          3.0           5.2          2.3  Iris-virginica
146           6.3          2.5           5.0          1.9  Iris-virginica
147           6.5          3.0           5.2          2.0  Iris-virginica
148           6.2          3.4           5.4          2.3  Iris-virginica
149           5.9          3.0           5.1          1.8  Iris-virginica

[150 rows x 5 columns]
```

2. grouping variable

```
print("GROUPING VARIABEL".center(100, "="))
X=df.iloc[:,0:4].values
y=df.iloc[:,4].values
print("data variabel".center(75,"="))
print(X)
print("data kelas".center(75,"="))
```

```
print(y)
```

Hasil

```
==========================================GROUPING VARIABEL==========================================
==============================data variabel==============================
[[5.1 3.5 1.4 0.2]
 [4.9 3.  1.4 0.2]
 [4.7 3.2 1.3 0.2]
 [4.6 3.1 1.5 0.2]
 [5.  3.6 1.4 0.2]
 [5.4 3.9 1.7 0.4]
 [4.6 3.4 1.4 0.3]
 [5.  3.4 1.5 0.2]
 [4.4 2.9 1.4 0.2]
 [4.9 3.1 1.5 0.1]
 [5.4 3.7 1.5 0.2]
 [4.8 3.4 1.6 0.2]
 [4.8 3.  1.4 0.1]
 [4.3 3.  1.1 0.1]
 [5.8 4.  1.2 0.2]
 [5.7 4.4 1.5 0.4]
 [5.4 3.9 1.3 0.4]
 [5.1 3.5 1.4 0.3]
 [5.7 3.8 1.7 0.3]
 [5.1 3.8 1.5 0.3]
 [5.4 3.4 1.7 0.2]
 [5.1 3.7 1.5 0.4]
 [4.6 3.6 1.  0.2]
 [5.1 3.3 1.7 0.5]
 [4.8 3.4 1.9 0.2]
```

3. membagi data training dan testing

```
# Split data into training and testing
x_train, x_test, y_train, y_test = train_test_split(X, y,
test_size=0.30, random_state=100)
print("Instance variabel data training".center(75, "="))
print(x_train)
print("Instance kelas data training".center(75, "="))
print(y_train)
print("Instance variabel data testing".center(75, "="))
print(x_test)
print("Instance kelas data testing".center(75, "="))
print(y_test)
print("============================================================
====")
print()
```

Hasil

a. Variable data training

```
======================Instance variabel data training======================
[[4.6 3.4 1.4 0.3]
 [5.  3.  1.6 0.2]
 [5.1 3.7 1.5 0.4]
 [5.8 2.6 4.  1.2]
 [4.9 3.1 1.5 0.1]
 [5.1 3.3 1.7 0.5]
 [5.  3.2 1.2 0.2]
 [6.5 2.8 4.6 1.5]
 [7.9 3.8 6.4 2. ]
 [6.1 3.  4.9 1.8]
 [5.4 3.  4.5 1.5]
 [6.4 2.7 5.3 1.9]
 [5.7 2.9 4.2 1.3]
 [7.7 3.8 6.7 2.2]
 [6.5 3.2 5.1 2. ]
 [5.8 2.7 3.9 1.2]
 [4.6 3.6 1.  0.2]
 [6.9 3.1 5.4 2.1]
 [6.7 3.3 5.7 2.1]
 [6.3 2.8 5.1 1.5]
 [5.5 4.2 1.4 0.2]
 [4.4 3.2 1.3 0.2]
 [5.8 2.7 5.1 1.9]
 [5.4 3.9 1.7 0.4]
```

b. Instance kelas data training

```
======================Instance kelas data training======================
['Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-versicolor' 'Iris-setosa'
 'Iris-setosa' 'Iris-setosa' 'Iris-versicolor' 'Iris-virginica'
 'Iris-virginica' 'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor'
 'Iris-virginica' 'Iris-virginica' 'Iris-versicolor' 'Iris-setosa'
 'Iris-virginica' 'Iris-virginica' 'Iris-virginica' 'Iris-setosa'
 'Iris-setosa' 'Iris-virginica' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa'
 'Iris-versicolor' 'Iris-virginica' 'Iris-virginica' 'Iris-virginica'
 'Iris-setosa' 'Iris-versicolor' 'Iris-versicolor' 'Iris-setosa'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-virginica' 'Iris-setosa'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-virginica' 'Iris-virginica'
 'Iris-setosa' 'Iris-versicolor' 'Iris-setosa' 'Iris-versicolor'
 'Iris-virginica' 'Iris-virginica' 'Iris-virginica' 'Iris-versicolor'
 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa' 'Iris-virginica'
 'Iris-setosa' 'Iris-setosa' 'Iris-versicolor' 'Iris-setosa'
 'Iris-virginica' 'Iris-versicolor' 'Iris-virginica' 'Iris-setosa'
 'Iris-virginica' 'Iris-virginica' 'Iris-setosa' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-setosa' 'Iris-setosa' 'Iris-setosa'
 'Iris-virginica' 'Iris-versicolor' 'Iris-versicolor' 'Iris-setosa'
 'Iris-virginica' 'Iris-virginica' 'Iris-virginica' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor' 'Iris-virginica'
 'Iris-virginica' 'Iris-setosa' 'Iris-setosa' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor' 'Iris-virginica'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor'
```

c. Variable data testing

```
======================Instance variabel data testing====================
[[6.4 2.8 5.6 2.1]
 [4.8 3.4 1.6 0.2]
 [7.7 2.6 6.9 2.3]
 [5.7 4.4 1.5 0.4]
 [6.3 2.7 4.9 1.8]
 [7.7 3.  6.1 2.3]
 [5.2 4.1 1.5 0.1]
 [4.9 3.  1.4 0.2]
 [6.5 3.  5.5 1.8]
 [4.8 3.  1.4 0.3]
 [5.  3.5 1.3 0.3]
 [6.4 3.2 5.3 2.3]
 [5.  3.4 1.6 0.4]
 [5.2 3.4 1.4 0.2]
 [6.7 3.  5.2 2.3]
 [6.2 2.9 4.3 1.3]
 [6.  2.2 4.  1. ]
 [6.7 3.  5.  1.7]
 [7.7 2.8 6.7 2. ]
 [6.8 3.  5.5 2.1]
 [7.2 3.2 6.  1.8]
 [5.4 3.4 1.5 0.4]
 [6.3 2.5 5.  1.9]
 [4.7 3.2 1.6 0.2]
 [5.6 2.5 3.9 1.1]
 [5.9 3.  5.1 1.8]
```

d. Instance kelas data testing

```
======================Instance kelas data testing======================
['Iris-virginica' 'Iris-setosa' 'Iris-virginica' 'Iris-setosa'
 'Iris-virginica' 'Iris-virginica' 'Iris-setosa' 'Iris-setosa'
 'Iris-virginica' 'Iris-setosa' 'Iris-setosa' 'Iris-virginica'
 'Iris-setosa' 'Iris-setosa' 'Iris-virginica' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-versicolor' 'Iris-virginica' 'Iris-virginica'
 'Iris-virginica' 'Iris-setosa' 'Iris-virginica' 'Iris-setosa'
 'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor' 'Iris-setosa'
 'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor' 'Iris-versicolor'
 'Iris-virginica' 'Iris-setosa' 'Iris-setosa' 'Iris-versicolor'
 'Iris-setosa' 'Iris-versicolor' 'Iris-virginica' 'Iris-virginica'
 'Iris-setosa' 'Iris-versicolor' 'Iris-virginica' 'Iris-virginica'
 'Iris-setosa']
nica' 'Iris-virginica'
 'Iris-virginica' 'Iris-setosa' 'Iris-virginica' 'Iris-setosa'
 'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor' 'Iris-setosa'
 'Iris-versicolor' 'Iris-virginica' 'Iris-versicolor' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-setosa' 'Iris-setosa' 'Iris-versicolor'
 'Iris-setosa' 'Iris-versicolor' 'Iris-virginica' 'Iris-virginica'
 'Iris-setosa' 'Iris-versicolor' 'Iris-virginica' 'Iris-virginica'
 'Iris-setosa']
```

4. Hasil confusion Matrix

```
#Pemodelan SVM
model = SVC()
model.fit(x_train, y_train)
y_pred = model.predict(x_test)
print("hasil prediksi SVM")
print(y_pred)

#Evaluasi Confusion Matrix dan Evaluasi Akurasi Python
print("Hasil Confusion Matrix")
print(confusion_matrix(y_test, y_pred))
print("Hasil Akurasi Pemodelan SVM : ", accuracy_score(y_test,
y_pred))
```

Hasil

```
Hasil Confusion Matrix
[[16  0  0]
 [ 0 11  0]
 [ 0  1 17]]
```

5. Hasil Akurasi pemodelan SVM

```
Hasil Akurasi Pemodelan SVM :  0.9777777777777777
PS C:\ML\Tugasmg8>
```

**TUGAS KELOMPOK LANJUTAN UTS SVM DAN NN**

1. menampilkan data awal

```
import pandas as pd
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.neural_network import MLPClassifier
# Import MLPClassifier from scikit-learn

# Load data from Excel file
df = pd.read_csv('lung_cancer_examples.csv')
print(" Data Awal ".center(100, "="))
print(df)
print("============================================================"
)

# Separate features (x) and target (y)
x =
df.drop(columns=['Name','Surname','Age','Smokes','AreaQ','Alkhol',])
y = df['Result']
```

Hasil

```
==================================== Data Awal =====================================
          Name      Surname  Age  Smokes  AreaQ  Alkhol  Result
0         John         Wick   35       3      5       4       1
1         John  Constantine   27      20      2       5       1
2       Camela     Anderson   30       0      5       2       0
3         Alex       Telles   28       0      8       1       0
4        Diego     Maradona   68       4      5       6       1
5    Cristiano      Ronaldo   34       0     10       0       0
6       Mihail          Tal   58      15     10       0       0
7        Kathy        Bates   22      12      5       2       0
8        Nicole       Kidman   45       2      6       0       0
9          Ray      Milland   52      18      4       5       1
10     Fredric        March   33       4      8       0       0
11         Yul      Brynner   18      10      6       3       0
12        Joan      Crawford   25       2      5       1       0
13        Jane        Wyman   28      20      2       8       1
14        Anna      Magnani   34      25      4       8       1
15  Katharine      Hepburn   39      18      8       1       0
16  Katharine      Hepburn   42      22      3       5       1
17      Barbra    Streisand   19      12      8       0       0
18     Maggie        Smith   62       5      4       3       1
19     Glenda      Jackson   73      10      7       6       1
20       Jane        Fonda   55      15      1       3       1
21  Maximilian       Schell   33       8      8       1       0
22    Gregory         Peck   22      20      6       2       0
23     Sidney      Poitier   44       5      8       1       0
24        Rex     Harrison   77       3      2       6       1
25        Lee       Marvin   21      20      5       3       0
```

2. Grouping 2 variable

```
# Grouping yang dibagi menjadi dua
print("GROUPING VARIABEL".center(100, "="))
X=df.iloc[:,0:6].values
y=df.iloc[:,6].values
print("data variabel".center(75,"="))
print(X)
print("data kelas".center(75,"="))
print(y)
```

Hasil

```
========================================GROUPING VARIABEL========================================
==============================data variabel==============================
[['John' 'Wick' 35 3 5 4]
 ['John' 'Constantine' 27 20 2 5]
 ['Camela' 'Anderson' 30 0 5 2]
 ['Alex' 'Telles' 28 0 8 1]
 ['Diego' 'Maradona' 68 4 5 6]
 ['Cristiano' 'Ronaldo' 34 0 10 0]
 ['Mihail' 'Tal' 58 15 10 0]
 ['Kathy' 'Bates' 22 12 5 2]
 ['Nicole' 'Kidman' 45 2 6 0]
 ['Ray' 'Milland' 52 18 4 5]
 ['Fredric' 'March' 33 4 8 0]
 ['Yul' 'Brynner' 18 10 6 3]
 ['Joan' 'Crawford' 25 2 5 1]
 ['Jane' 'Wyman' 28 20 2 8]
 ['Anna' 'Magnani' 34 25 4 8]
 ['Katharine ' 'Hepburn' 39 18 8 1]
 ['Katharine ' 'Hepburn' 42 22 3 5]
 ['Barbra' 'Streisand' 19 12 8 0]
 ['Maggie ' 'Smith' 62 5 4 3]
 ['Glenda ' 'Jackson' 73 10 7 6]
 ['Jane ' 'Fonda' 55 15 1 3]
 ['Maximilian ' 'Schell' 33 8 8 1]
 ['Gregory ' 'Peck' 22 20 6 2]
 ['Sidney ' 'Poitier' 44 5 8 1]
 ['Rex ' 'Harrison' 77 3 2 6]
 ['Lee ' 'Marvin' 21 20 5 3]
```

3. Pembagian data training dan testing

```
# Split data into training and testing
x_train, x_test, y_train, y_test = train_test_split(x, y,
test_size=0.30, random_state=100)
print("Instance variabel data training".center(75, "="))
print(x_train)
print("Instance kelas data training".center(75, "="))
print(y_train)
print("Instance variabel data testing".center(75, "="))
print(x_test)
print("Instance kelas data testing".center(75, "="))
print(y_test)
print("=========================================================")
print()
```

Hasil

a. Variable data training

```
====================Instance variabel data training====================
    Result
37       1
41       0
25       0
46       0
33       1
43       1
17       0
7        0
56       1
42       1
55       1
35       0
19       1
21       0
13       1
1        1
31       1
4        1
27       0
52       1
22       0
29       1
```

b. Kelas data training

```
=======================Instance kelas data training=======================
[1 0 0 0 1 1 0 0 1 1 1 0 1 0 1 1 1 1 0 1 0 1 1 1 0 1 0 1 1 0 0 0 0 1 0 0
 0 0 1 0]
```

c. Variable data testing

```
=======================Instance variabel data testing=====================
    Result
26       0
44       0
32       1
12       0
45       0
51       1
50       0
28       1
6        0
20       1
0        1
38       0
58       0
11       0
5        0
54       0
18       1
40       1
```

d. Kelas data testing

```
=======================Instance kelas data testing=======================
[0 0 1 0 0 1 0 1 0 1 1 0 0 0 0 0 1 1]
=========================================================================
```

4. Cofusion Matrix SVM

```
# Evaluasi confusion matrix dan evaluasi akurasi SVM
print("Hasil confusion matrix SVM".center(75, "="))
print(confusion_matrix(y_test, y_pred_svm))
print("Hasil akurasi pemodelan SVM:", accuracy_score(y_test,
y_pred_svm))
```

```
print("=====================================================")
print()
```

Hasil

```
=======================Hasil confusion matrix SVM=======================
[[11  0]
 [ 0  7]]
```

5. Hasil prediksi SVM dan Akurasi

```
# Pemodelan SVM
svm_model = SVC()
svm_model.fit(x_train, y_train)
y_pred_svm = svm_model.predict(x_test)
print("Hasil prediksi SVM".center(75, "="))
print(y_pred_svm)
print("=====================================================")
print()
```

Hasil

```
============================Hasil prediksi SVM============================
[0 0 1 0 0 1 0 1 0 1 1 0 0 0 0 0 1 1]
==========================================================
Hasil akurasi pemodelan SVM: 1.0
==========================================================
```

6. Cofusion Matrix NN

```
# Evaluasi confusion matrix dan evaluasi akurasi Neural Network
print("Hasil confusion matrix Neural Network".center(75, "="))
print(confusion_matrix(y_test, y_pred_nn))
print("Hasil akurasi pemodelan Neural Network:",
accuracy_score(y_test, y_pred_nn))
print("=====================================================")
print()
```

Hasil

```
==================Hasil confusion matrix Neural Network==================
[[11  0]
 [ 0  7]]
```

7. Hasil prediksi NN dan Akurasi

```
# Pemodelan Neural Network (Multilayer Perceptron)
nn_model = MLPClassifier(hidden_layer_sizes=(100, 50),
 max_iter=1000, random_state=100)  # Adjust the parameters as
 needed
nn_model.fit(x_train, y_train)
```

```
y_pred_nn = nn_model.predict(x_test)
print("Hasil prediksi Neural Network".center(75, "="))
print(y_pred_nn)
print("===========================================================")
print()
```

Hasil

```
=======================Hasil prediksi Neural Network=======================
[0 0 1 0 0 1 0 1 0 1 1 0 0 0 0 0 1 1]
===========================================================
```

```
Hasil akurasi pemodelan Neural Network: 1.0
===========================================================
```