## Tugas Mandiri Pertemuan 12\_Andrean Yonathan\_Institut Teknologi Sepuluh Nopember

# Latihan (1)

#### In [1]: # import library pandas import pandas as pd

Melakukan import library yang dibutuhkan

# Import library numpy import numpy as np # Import library matplotlib dan seaborn untuk visualisasi import matplotlib.pyplot as plt import seaborn as sns # me-non aktifkan peringatan pada python import warnings warnings.filterwarnings('ignore') **Load Dataset** In [2]: #Panggil file (load file bernama CarPrice Assignment.csv) dan simpan dalam dataframe dataset = 'Iris AfterClean.csv'

## iris.head()

0

1

2

3

iris = pd.read csv(dataset)

In [3]: # tampilkan 5 baris data SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm Out[3]: Species

0.2 Iris-setosa

0.2 Iris-setosa

0.4 Iris-setosa

0.1 Iris-setosa

0.2 Iris-setosa

# Latihan (2)

4.6

5.0

5.4

4.9

5.4

definisi variabel X(feature kolom) dan y(species/label): In [4]: X=iris.iloc[:, 0:3].values y=iris.iloc[:, 4].values

3.1

3.6

3.9

3.1

3.7

from sklearn.preprocessing import LabelEncoder

1.5

1.4

1.7

1.5

1.5

### Latihan (3) transform label data species dengan menggunakan library LabelEncoder

le = LabelEncoder() y = le.fit transform(y)

In [5]:

In [6]:

3. Building Machine Learning Models Latihan (4)

import library dalam kebutuhan membangun model

from sklearn.metrics import classification report

# Import Library Random Forest Classifier

# from sklearn import linear model from sklearn import linear model

from sklearn.ensemble import RandomForestClassifier

from sklearn.model\_selection import train test split

from sklearn.metrics import make\_scorer, accuracy\_score,precision\_score

from sklearn.model selection import KFold, train test split, cross val score

#### # Import Library Confussion Matrix from sklearn.metrics import confusion matrix from sklearn.metrics import accuracy\_score ,precision\_score,recall\_score,f1\_score

#Model Select

Training Set Test Set

Dataset

Splitting The Data into Training And Testing Dataset

#### Bangun model Random Forest dan akurasi nya In [8]: random forest = RandomForestClassifier(n estimators=100) random\_forest.fit(X\_train,y\_train)

Latihan (6)

dan random\_state=0

#Train and Test split

Y\_prediction = random\_forest.predict(X\_test)

f1 = f1\_score(y\_test,Y\_prediction,average='micro') print('Confusion matrix for Random Forest\n', cm) print('accuracy\_random\_Forest : %.3f' %accuracy) print('precision\_random\_Forest : %.3f' %precision) print('recall\_random\_Forest : %.3f' %recall) print('f1-score random Forest : %.3f' %f1) Confusion matrix for Random Forest [[12 0 0] [ 0 13 2] [ 0 0 15]] accuracy random Forest : 0.952 precision random Forest: 0.952 recall random Forest: 0.952 f1-score random Forest : 0.952 Latihan (7) Visualisasikan Nilai Confusion Matrix dari Model Random Forest In [9]: from sklearn import metrics

- 12

- 10

split data train dan test dengan function train\_test\_split() dengan train\_size=0.7, test\_size=0.3

X train, X test, y train, y test=train test split(X,y,test size=0.3,random state=0)

# Actual labe

Latihan (8)

12

plt.show()

plt.xlabel('Predicted label') plt.ylabel('Actual label')

Confussion Matrix for Random Forest

13

0

Predicted label

from sklearn.tree import DecisionTreeClassifier

decision tree = DecisionTreeClassifier()

decision\_tree.fit(X\_train,y\_train) Y pred = decision tree.predict(X test)

cm = confusion matrix(y\_test, Y\_pred) accuracy = accuracy\_score(y\_test,Y\_pred)

f1 = f1\_score(y\_test,Y\_pred,average='micro') print('Confusion matrix for DecisionTree\n', cm)

15

Bangun model DecisionTreeClassifier dan akurasi nya

accuracy\_dt=round(accuracy\_score(y\_test,Y\_pred)\* 100, 2)

precision = precision\_score(y\_test, Y\_pred,average='micro') recall = recall\_score(y\_test, Y\_pred, average='micro')

acc\_decision\_tree = round(decision\_tree.score(X\_train, y\_train) \* 100, 2)

Visualisasikan Nilai Confusion Matrix dari Model Decision Tree

p = sns.heatmap(pd.DataFrame(cdt), annot=True, cmap="YlGnBu",fmt='g')

#### print('accuracy\_random\_Forest : %.3f' %accuracy) print('precision\_random\_Forest : %.3f' %precision) print('recall random Forest : %.3f' %recall) print('f1-score random Forest : %.3f' %f1)

plt.ylabel('Actual label') plt.show() Confussion matrix for Decision Tree 12 0 10 Actual labe 12 3 13 Predicted label

X[2] <= 2.5gini = 0.0gini = 0.528samples = 14samples = 28value = [0, 0, 14]value = [12, 15, 1]X[2] <= 4.85gini = 0.0gini = 0.117samples = 12samples = 16value = [12, 0, 0]value = [0, 15, 1]Z X[1] <= 2.95gini = 0.0gini = 0.5samples = 14samples = 2value = [0, 14, 0]value = [0, 1, 1]qini = 0.0gini = 0.0samples = 1samples = 1value = [0, 0, 1]value = [0, 1, 0]

X[2] <= 4.95

# Latihan (5)

In [7]:

 $\verb|accuracy_rf=| round(accuracy_score(y_test, Y_prediction) * 100, 2)|$ acc\_random\_forest = round(random\_forest.score(X\_train, y\_train) \* 100, 2) cm = confusion\_matrix(y\_test, Y\_prediction) accuracy = accuracy\_score(y\_test,Y\_prediction) precision = precision\_score(y\_test, Y\_prediction,average='micro') recall = recall\_score(y\_test, Y\_prediction, average='micro')

cm = metrics.confusion matrix(y test, Y prediction) p = sns.heatmap(pd.DataFrame(cm), annot=True, cmap="YlGnBu", fmt='g') plt.title('Confussion Matrix for Random Forest', pad=30)

In [10]:

In [11]:

In [12]:

Confusion matrix for DecisionTree [[12 0 0] [ 0 12 3] [ 0 2 13]] accuracy\_random\_Forest : 0.881

Latihan (9)

Latihan (10)

plt.show()

from sklearn.tree import plot\_tree plt.figure(figsize = (15,10))

precision\_random\_Forest : 0.881 recall\_random\_Forest : 0.881 f1-score\_random\_Forest : 0.881

from sklearn import metrics

plt.xlabel('Predicted label')

cdt = metrics.confusion\_matrix(y\_test, Y\_pred)

plt.title('Confussion matrix for Decision Tree', pad=30)

gini = 0.663samples = 42

plot\_tree(decision\_tree.fit(X\_test,y\_test) ,filled=True)

Plot hasil model DecisionTreeClassifier dengan library plot\_tree

value = [12, 15, 15]