TUGAS LAB IF540 MACHINE LEARNING

WEEK [01] : [Basic Introduction of Machine Learning]

Semester Ganjil 2022/2023

In [1]:
```
import datetime
import uuid

# Fill in your name and NIM
myName = "Christopher Darren"
myNIM = "00000054804"

myDate = datetime.datetime.now()
myDevice = str(uuid.uuid1())

# Header
print("Name: \t\t{}".format(myName))
print("NIM: \t\t{}".format(myNIM))
print("Start: \t\t{}".format(myDate))
print("Device ID: \t{}".format(myDevice))
```

```
Name:           Christopher Darren
NIM:            00000054804
Start:          2023-02-09 10:48:15.747298
Device ID:      956f30d9-a82c-11ed-a37a-f02f74a116e8
```

Dataset yang dipakai:

1. [iris dataset] – sumber : [https://www.kaggle.com/datasets/uciml/iris]
2. [Nama dataset2] – sumber : [cantumkan link dataset]

Hasil kerja

In [2]:
```
# Your codes are here (replace the following codes)
#import time
#time.sleep(10)

import sys
print("Python version: {}".format(sys.version))
```

```
Python version: 3.9.12 (main, Apr  4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)]
```

Importing library

In [3]:
```
import pandas as pd
print("Pandas version: {}". format(pd.__version__))
```

```
Pandas version: 1.4.2
```

In [4]:
```
import matplotlib as mp
print("Matplotlib version: {}". format(mp.__version__))
```

```
Matplotlib version: 3.5.1
```

In [5]:
```
import numpy as np
print("Numpy version: {}".format(np.__version__))
```

```
Numpy version: 1.21.5
```

In [6]:
```
import scipy as sp
print("Scipy version: {}".format(sp.__version__))
```

```
Scipy version: 1.7.3
```

In [7]:
```
import sklearn
print("Scikit-learn version: {}".format(sklearn.__version__))
```

```
Scikit-learn version: 1.0.2
```

In [8]:
```
from sklearn.datasets import load_iris
iris_dataset = load_iris()
```

In [9]:
```
print(iris_dataset['DESCR'][:193]+"\n...")
```

```
.. _iris_dataset:

Iris plants dataset
--------------------

**Data Set Characteristics:**

    :Number of Instances: 150 (50 in each of three classes)
    :Number of Attributes: 4 numeric, pre
...
```

In [10]:
```
print("Target names: {}".format(iris_dataset['target_names']))
```

```
Target names: ['setosa' 'versicolor' 'virginica']
```

In [11]:
```
print("Feature names: {}".format(iris_dataset['feature_names']))
```

```
Feature names: ['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']
```

In [12]:
```
print("Shape of data: {}".format(iris_dataset['data'].shape))
```

```
Shape of data: (150, 4)
```

In [13]:
```
print("Type of data: {}".format(type(iris_dataset['data'])))
```

```
Type of data: <class 'numpy.ndarray'>
```

In [14]:
```
print("First five columns of data:\n{}".format(iris_dataset['data'][:5]))
```

```
First five columns of data:
[[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]]
```

In [15]:
```
print("Type of target:{}".format(type(iris_dataset['target'])))
```

```
Type of target:<class 'numpy.ndarray'>
```

In [16]:
```
print("Shape of target: {}".format(iris_dataset['target'].shape))
```

```
Shape of target: (150,)
```

In [17]:
```
print("Target:\n{}".format(iris_dataset['target']))
```

```
Target:
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2]
```

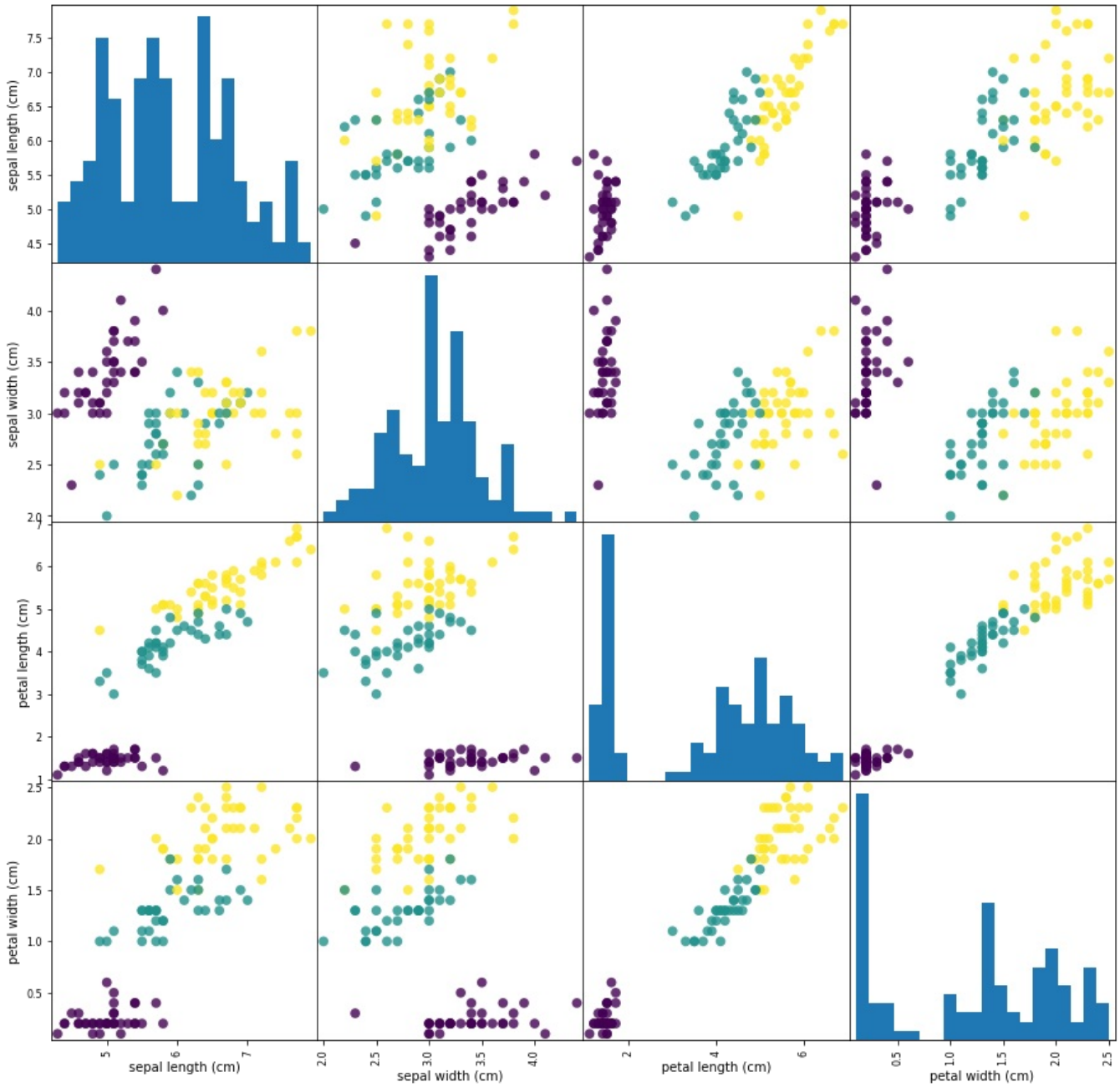
In [18]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(iris_dataset['data'], iris_dataset['target'], random_state=
```

In [19]:
```
print("X_train shape: {}".format(X_train.shape))
print("y_train shape: {}".format(y_train.shape))
print("X_test shape: {}".format(X_test.shape))
print("y_test shape: {}".format(y_test.shape))
```

```
X_train shape: (112, 4)
y_train shape: (112,)
X_test shape: (38, 4)
y_test shape: (38,)
```

Look at Your Data

In [20]:
```
iris_dataframe = pd.DataFrame(X_train, columns=iris_dataset.feature_names)
grr = pd.plotting.scatter_matrix(iris_dataframe, c=y_train , figsize=(15,15), marker='o', hist_kwds={'bins':20}
```



Building the First Model: KNN

In [21]:
```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=1)
```

In [22]:
```
knn.fit(X_train, y_train)
```

Out[22]:
```
KNeighborsClassifier(n_neighbors=1)
```

Making Predictions

In [23]:
```
X_new = np.array([[5, 2.9, 1, 0.2]])
print("X_new.shape: {}".format(X_new.shape))
```

```
X_new.shape: (1, 4)
```

In [24]:
```
prediction = knn.predict(X_new)
print("Prediction: {}".format(prediction))
print("Predicted target name: {}".format(iris_dataset['target_names'][prediction]))
```

```
Prediction: [0]
Predicted target name: ['setosa']
```

Evaluating the Model

In [25]:
```
y_pred = knn.predict(X_test)
print("Test set predictions:\n {}".format(y_pred))
```

```
Test set predictions:
 [2 1 0 2 0 2 0 1 1 1 2 1 1 1 1 0 1 1 0 0 2 1 0 0 2 0 0 1 1 0 2 1 0 2 2 1 0
 2]
```

In [26]:
```
#print("Test set score: {:.2f}".format(np.mean(y_pred == y_test)))
print("Test set score: {:.2f}".format(knn.score(X_test, y_test)))
```

```
Test set score: 0.97
```

Kesimpulan

Berikan simpulan yang dilakukan dari hasil kerja menggunakan algoritma dan 2 dataset yang dipilih. Simpulan bisa berkisar antara (bisa di modifikasi):

```
- Simpulan perbandingan dataset
- Hasil akurasi yang diberikan (jika ada dalam modul)
- Hasil perbandingan akurasi antara algoritma (jika ada dalam modul)
- Hasil pemikiran dan observasi akhir dari kerja menurut mahasiswa.
```

kesimpulan pada week ini adalah mengingatkan kembali saya terhadap penggunaan python di jupyter notebook

In [27]:
```
# Footer
myDate = datetime.datetime.now()
print("I certify that this is my own work.")
print("Signed by:")
print("Name: \t{}".format(myName))
print("NIM: \t{}".format(myNIM))
print("Time-stamp:\t{}".format(myDate))
```

```
I certify that this is my own work.
Signed by:
Name:   Christopher Darren
NIM:    00000054804
Time-stamp:     2023-02-09 10:48:18.865210
```

Save the notebook, then convert the notebook to html (by running the next code).

In [28]:
```
!jupyter nbconvert --to html "./Template Laporan Tugas Mingguan Lab IF540.ipynb" --output-dir="./"
```

```
[NbConvertApp] Converting notebook ./Template Laporan Tugas Mingguan Lab IF540.ipynb to html
[NbConvertApp] Writing 833498 bytes to Template Laporan Tugas Mingguan Lab IF540.html
```

Next step:

- convert the generated html file to PDF using the online tool: https://www.sejda.com/html-to-pdf
- choose the following settings:
  - Page size: One long page
  - Page Orientation: auto
  - Use print stylesheet
- Submit your ipython notebook and PDF files

Markdown basics https://markdown-guide.readthedocs.io/en/latest/basics.html#

In [ ]: