Department of Computer Science and Engineering

29th Batch

Lab Report 7

Course title : Artificial Intelligence Lab
Course Code : CSE - 414

Submitted By	Submitted To
Name : Md. Nahid Hasan ID : 221311131 Section : D Semester : 8th Batch : 29th	Name : Md. Mahfujur Rahman Designation : Lecturer, Varendra University, Rajshahi. Name : D.M. Asadujjaman Designation : Lecturer, Varendra University, Rajshahi.

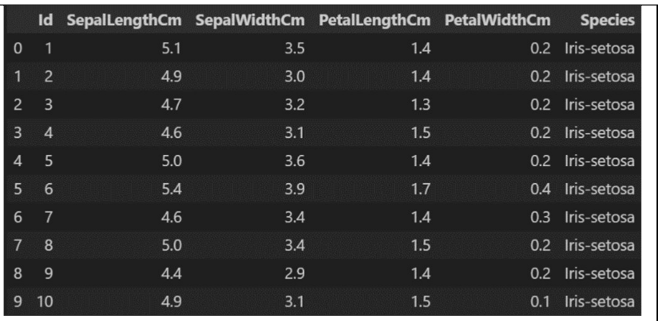
Signature

Signature

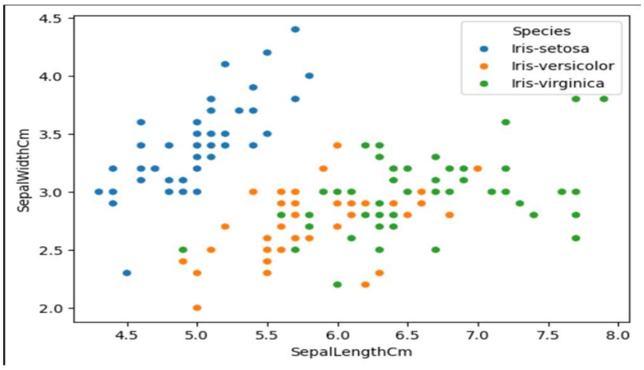
➢ Question: Making Predictions Using Iris Dataset

❖ Solution(Code & Output):

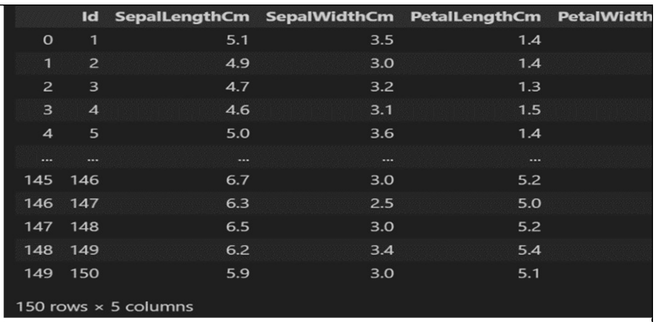
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv('iris.csv')
df.head()

df.describe()
df.info()
df.columns

sns.scatterplot(data=df, x =
'SepalLengthCm', y =
'SepalWidthCm', hue='Species')
```



	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
5	6	5.4	3.9	1.7	0.4	Iris-setosa
6	7	4.6	3.4	1.4	0.3	Iris-setosa
7	8	5.0	3.4	1.5	0.2	Iris-setosa
8	9	4.4	2.9	1.4	0.2	Iris-setosa
9	10	4.9	3.1	1.5	0.1	Iris-setosa

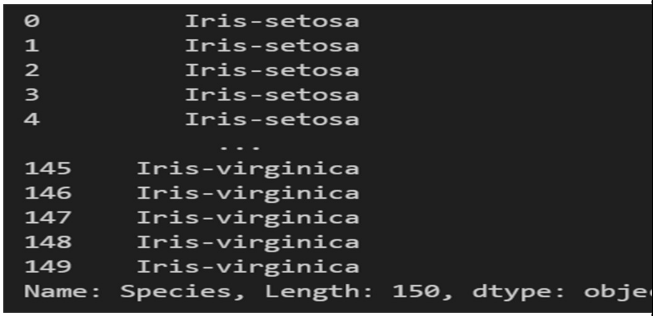


```
from sklearn.preprocessing import
LabelEncoder,StandardScaler

x = df.drop("Species", axis = 1)
y = df["Species"]

x
y


le = LabelEncoder()
y_encoded = le.fit_transform(y)

y_encoded
```

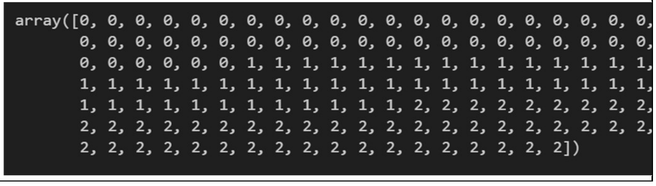


	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidth
0	1	5.1	3.5	1.4	
1	2	4.9	3.0	1.4	
2	3	4.7	3.2	1.3	
3	4	4.6	3.1	1.5	
4	5	5.0	3.6	1.4	
...	...	...	...	...	
145	146	6.7	3.0	5.2	
146	147	6.3	2.5	5.0	
147	148	6.5	3.0	5.2	
148	149	6.2	3.4	5.4	
149	150	5.9	3.0	5.1	

150 rows × 5 columns

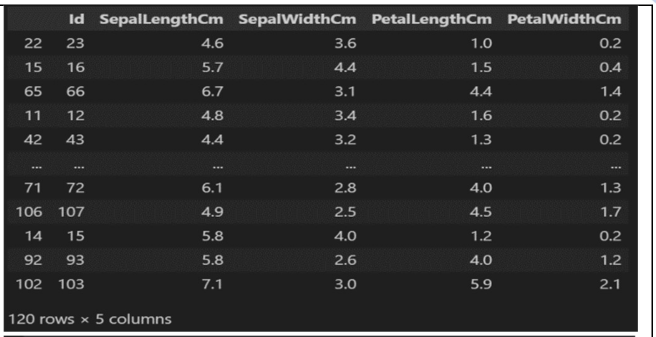


```
0          Iris-setosa
1          Iris-setosa
2          Iris-setosa
3          Iris-setosa
4          Iris-setosa
               ...
145     Iris-virginica
146     Iris-virginica
147     Iris-virginica
148     Iris-virginica
149     Iris-virginica
Name: Species, Length: 150, dtype: obje
```



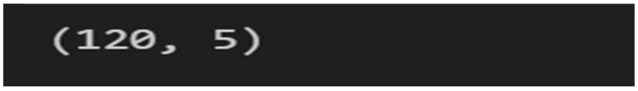
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 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])
```

```
from sklearn.model_selection import
train_test_split
from sklearn.metrics import
classification_report, confusion_matrix

X_train, X_test, y_train, y_test =
train_test_split(x,y_encoded,
test_size= 0.2, random_state=42)

X_train
X_train.shape
X_test.shape
X_test

scaler = StandardScaler()
X_train_scaled =
scaler.fit_transform(X_train)
X_train_scaled

X_test_scaled =
scaler.transform(X_test)
X_test_scaled
```

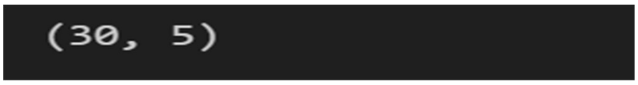


	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
22	23	4.6	3.6	1.0	0.2
15	16	5.7	4.4	1.5	0.4
65	66	6.7	3.1	4.4	1.4
11	12	4.8	3.4	1.6	0.2
42	43	4.4	3.2	1.3	0.2
...	...	...	...	...	...
71	72	6.1	2.8	4.0	1.3
106	107	4.9	2.5	4.5	1.7
14	15	5.8	4.0	1.2	0.2
92	93	5.8	2.6	4.0	1.2
102	103	7.1	3.0	5.9	2.1

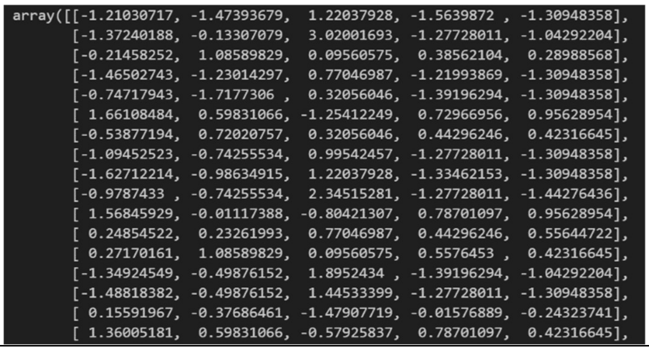
120 rows × 5 columns



```
(120, 5)
```

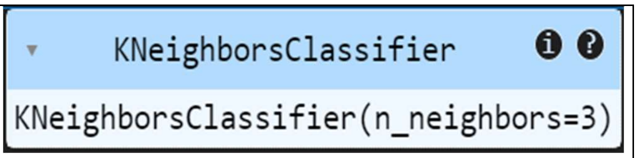


```
(30, 5)
```

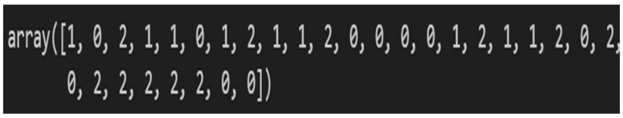


```
array([[-1.21030717, -1.47393679,  1.22037928, -1.5639872 , -1.30948358],
       [-1.37240188, -0.13307079,  3.02001693, -1.27728011, -1.04292204],
       [-0.21458252,  1.08589829,  0.09560575,  0.38562104,  0.28988568],
       [-1.46502743, -1.23014297,  0.77046987, -1.21993869, -1.30948358],
       [-0.74717943, -1.7177306 ,  0.32056046, -1.39196294, -1.30948358],
       [ 1.66108484,  0.59831066, -1.25412249,  0.72966956,  0.95628954],
       [-0.53877194,  0.72020757,  0.32056046,  0.44296246,  0.42316645],
       [-1.09452523, -0.74255534,  0.99542457, -1.27728011, -1.30948358],
       [-1.62712214, -0.98634915,  1.22037928, -1.33462153, -1.30948358],
       [-0.9787433 , -0.74255534,  2.34515281, -1.27728011, -1.44276436],
       [ 1.56845929, -0.01117388, -0.80421307,  0.78701097,  0.95628954],
       [ 0.24854522,  0.23261993,  0.77046987,  0.44296246,  0.55644722],
       [ 0.27170161,  1.08589829,  0.09560575,  0.5576453 ,  0.42316645],
       [-1.34924549, -0.49876152,  1.8952434 , -1.39196294, -1.04292204],
       [-1.48818382, -0.49876152,  1.44533399, -1.27728011, -1.30948358],
       [ 0.15591967, -0.37686461, -1.47907719, -0.01576889, -0.24323741],
       [ 1.36005181,  0.59831066, -0.57925837,  0.78701097,  0.42316645],
```

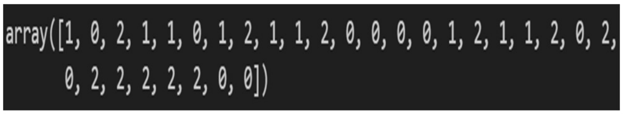
```
from sklearn.neighbors import
KNeighborsClassifier

model=KNeighborsClassifier(n_neighbors=
3)
model.fit(X_train_scaled, y_train)

pred = model.predict(X_test_scaled)
pred

y_test
cmatri = confusion_matrix(p,y_test)
cmatri
```



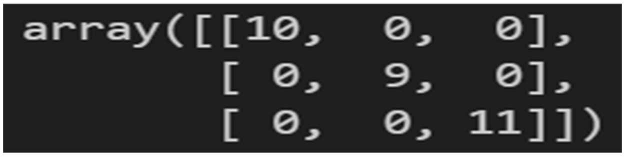
```
KNeighborsClassifier(n_neighbors=3)
```



```
array([1, 0, 2, 1, 1, 0, 1, 2, 1, 1, 2, 0, 0, 0, 0, 1, 2, 1, 1, 2, 0, 2,
       0, 2, 2, 2, 2, 2, 0, 0])
```



```
array([1, 0, 2, 1, 1, 0, 1, 2, 1, 1, 2, 0, 0, 0, 0, 1, 2, 1, 1, 2, 0, 2,
       0, 2, 2, 2, 2, 2, 0, 0])
```



```
array([[10,  0,  0],
       [ 0,  9,  0],
       [ 0,  0, 11]])
```

❖ Conclusion:

In this lab, I learned how to use Python to train a model and make predictions. I used pandas for data, seaborn and matplotlib for graphs, and scikit-learn for machine learning. It was helped me understand how prediction works with real data. I feel more confident now about building simple ML projects.