**Institute of Information Technology (IIT)**

Jahangirnagar University



**Lab Report: 07**

Submitted by:

Name: Md. Shakil Hossain  
Roll No: 2023

Lab Date: 28/08/2023  
Submission Date: 01/09/2023

Support Vector Machines with Python

# Import Libraries

In [1]:

**import** pandas **as** pd

**import** numpy **as** np

**import** matplotlib.pyplot **as** plt

**import** seaborn **as** sns

**%**matplotlib inline

# Get the Data

In [2]:

IRIS **=** pd.read\_csv('IRIS.csv')

In [3]:

IRIS.keys()

Out[3]:

Index(['sepal\_length', 'sepal\_width', 'petal\_length', 'petal\_width', 'species'],

dtype='object')

In [4]:

IRIS.head()

Out[4]:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 |

|  |  |
| --- | --- |
| 0 | 0.2 |
| 1 | 0.2 |
| 2 | 0.2 |
| 3 | 0.2 |
| 4 | 0.2 |

...

145 2.3

146 1.9

147 2.0

148 2.3

149 1.8

Name: petal\_width, Length: 150, dtype: float64

In [6]:

print(IRIS['species'])

1. Iris-setosa
2. Iris-setosa
3. Iris-setosa
4. Iris-setosa
5. Iris-setosa

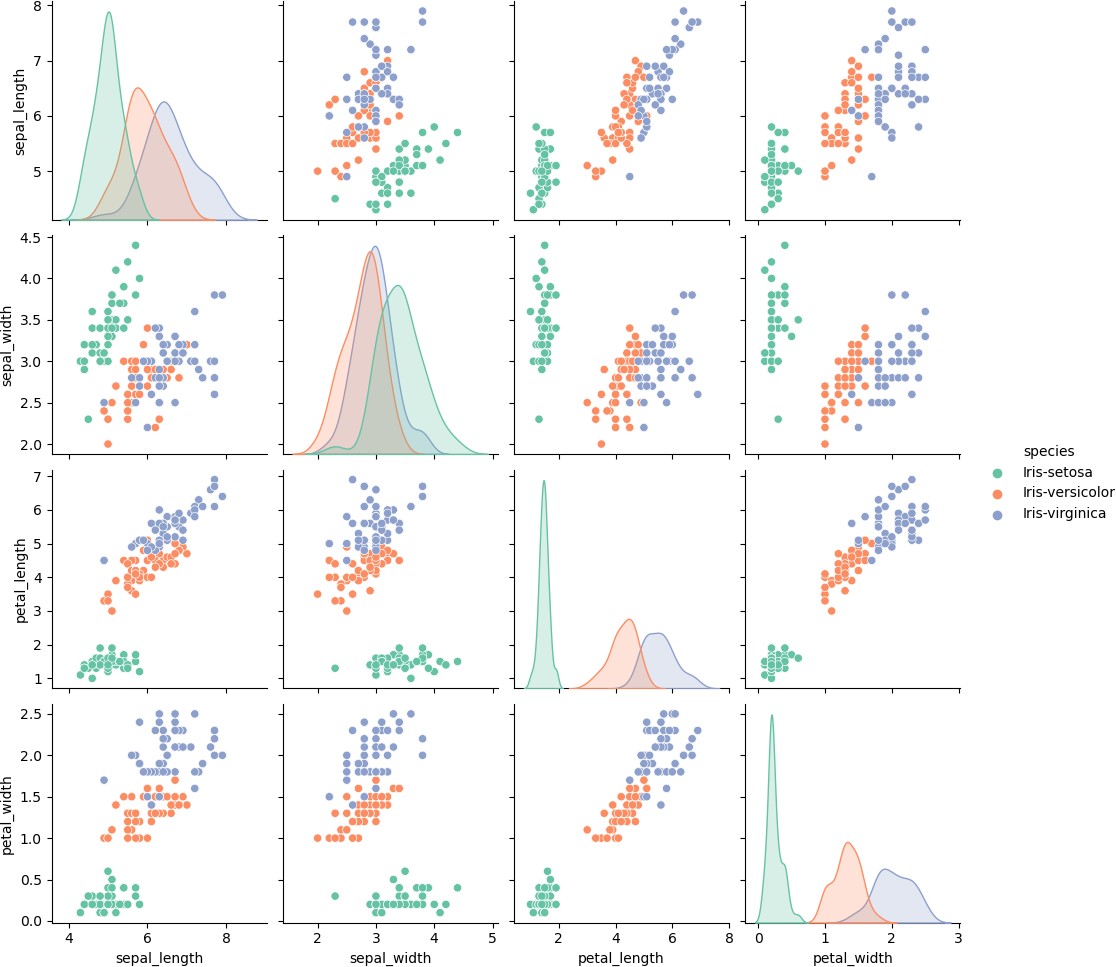
...

1. Iris-virginica
2. Iris-virginica
3. Iris-virginica
4. Iris-virginica
5. Iris-virginica

Name: species, Length: 150, dtype: object

Out[7]:

<seaborn.axisgrid.PairGrid at 0x1ef17977df0>



# Set up DataFrame

In [8]:

df\_IRIS **=** pd.DataFrame(IRIS['sepal\_length'],columns**=**IRIS['species']) df\_IRIS.info()

<class 'pandas.core.frame.DataFrame'> Index: 0 entries

Columns: 150 entries, Iris-setosa to Iris-virginica dtypes: object(150)

memory usage: 0.0+ bytes

IRIS['petal\_width']

Out[9]:

|  |  |
| --- | --- |
| 0 | 0.2 |
| 1 | 0.2 |
| 2 | 0.2 |
| 3 | 0.2 |
| 4 | 0.2 |

...

145 2.3

146 1.9

147 2.0

148 2.3

149 1.8

Name: petal\_width, Length: 150, dtype: float64

In [10]:

df\_petal\_width **=** pd.DataFrame(IRIS['petal\_width'],columns**=**['IRIS'])

# Train Test Split

In [11]:

**from** sklearn.model\_selection **import** train\_test\_split

In [12]:

x**=**IRIS.iloc[:,:**-**1]

y**=**IRIS.iloc[:,4]

x\_train,x\_test, y\_train, y\_test**=**train\_test\_split(x,y,test\_size**=**0.30)

In [13]:

**from** sklearn.svm **import** SVC model**=**SVC()

In [14]:

model.fit(x\_train, y\_train)

Out[14]:

▾ SVC

SVC()

pred**=**model.predict(x\_test)

# Model Evaluation

In [16]:

**from** sklearn.metrics **import** classification\_report, confusion\_matrix

In [17]:

print(confusion\_matrix(y\_test,pred))

|  |  |  |
| --- | --- | --- |
| [[20 | 0 | 0] |
| [ 0 | 12 | 0] |
| [ 0 | 2 | 11]] |

In [18]:

print(classification\_report(y\_test, pred))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| precision | recall | f1-score | support |  |
| Iris-setosa 1.00 | 1.00 | 1.00 | 20 |
| Iris-versicolor 0.86 | 1.00 | 0.92 | 12 |
| Iris-virginica 1.00 | 0.85 | 0.92 | 13 |
| accuracy |  | 0.96 | 45 |
| macro avg 0.95 | 0.95 | 0.95 | 45 |
| weighted avg 0.96 | 0.96 | 0.96 | 45 |
| **Gridsearch** |  |  |  |
| In [19]: |  |  |  |
| param\_grid **=** {'C': [0.1,1,5, | 10, 50, | 100, 1000], | 'gamma': | [10,1,0.1,0.01,0.001,0.0001], |
| In [20]: |  |  |  |  |

**from** sklearn.model\_selection **import** GridSearchCV

In [21]:

grid **=** GridSearchCV(SVC(),param\_grid,refit**=True**,verbose**=**5)



me= 0.0s

[CV 1/5] END ....C=10, gamma=0.1, kernel=linear;, score=1.000 total ti me= 0.0s

[CV 2/5] END ....C=10, gamma=0.1, kernel=linear;, score=0.952 total ti me= 0.0s

[CV 3/5] END ....C=10, gamma=0.1, kernel=linear;, score=1.000 total ti me= 0.0s

[CV 4/5] END ....C=10, gamma=0.1, kernel=linear;, score=0.905 total ti me= 0.0s

[CV 5/5] END ....C=10, gamma=0.1, kernel=linear;, score=1.000 total ti me= 0.0s

[CV 1/5] END ......C=10, gamma=0.01, kernel=rbf;, score=0.952 total ti me= 0.0s

[CV 2/5] END ......C=10, gamma=0.01, kernel=rbf;, score=0.952 total ti me= 0.0s

[CV 3/5] END ......C=10, gamma=0.01, kernel=rbf;, score=1.000 total ti me= 0.0s

[CV 4/5] END ......C=10, gamma=0.01, kernel=rbf;, score=0.905 total ti me= 0.0s

grid.fit(x\_train,y\_train)

In [23]:

grid.best\_params\_

Out[23]:

{'C': 5, 'gamma': 10, 'kernel': 'linear'}

In [24]:

grid.best\_estimator\_

Out[24]:

▾

SVC

SVC(C=5, gamma=10, kernel='linear')

In [25]:

grid\_predictions **=** grid.predict(x\_test)

In [26]:

print(confusion\_matrix(y\_test,grid\_predictions))

|  |  |  |
| --- | --- | --- |
| [[20 | 0 | 0] |
| [ 0 | 11 | 1] |
| [ 0 | 0 | 13]] |

print(classification\_report(y\_test,grid\_predictions))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| Iris-setosa | 1.00 | 1.00 | 1.00 | 20 |
| Iris-versicolor | 1.00 | 0.92 | 0.96 | 12 |
| Iris-virginica | 0.93 | 1.00 | 0.96 | 13 |
| accuracy |  |  | 0.98 | 45 |
| macro avg | 0.98 | 0.97 | 0.97 | 45 |
| weighted avg | 0.98 | 0.98 | 0.98 | 45 |

In [ ]: