Experiment – 4

Student Name: Aniket Kumar **UID:20BCS5306**
Branch: CSE **Section/Group: 703/ B**
Semester: 5th **Date of Performance:15-09-22**
Subject Name: Machine Learning Lab **Subject Code: CSP-317**

1. Aim/Overview of the practical:

Implement Exploratory Data Analysis on any data set.

2. Source Code:

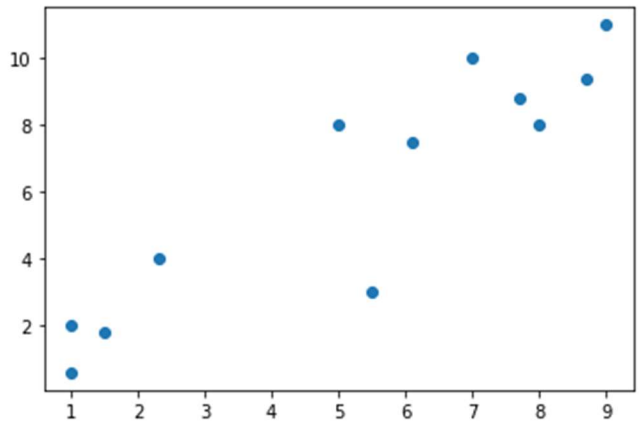
```
# ANKIT
#20BCS9637
import matplotlib.pyplot as plt
import numpy as np
from sklearn import svm

X = np.array([1, 5, 1.5, 8, 1, 9, 7, 8.7, 2.3, 5.5, 7.7, 6.1])
y = np.array([2, 8, 1.8, 8, 0.6, 11, 10, 9.4, 4, 3, 8.8, 7.5])

plt.scatter(X, y)
plt.show()

training_X = np.vstack((X, y)).T
training_y = [0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1]

clf = svm.SVC(kernel='linear', C=1.0)
SVC(kernel='linear')

# get the weight values for the linear equation from the trained SVM model
w = clf.coef_[0]
```

```
# get the y-offset for the linear equation
a = -w[0] / w[1]

# make the x-axis space for the data points
XX = np.linspace(0, 13)

# get the y-values to plot the decision boundary
yy = a * XX - clf.intercept_[0] / w[1]

# plot the decision boundary
plt.plot(XX, yy, 'k-')

# show the plot visually
plt.scatter(training_X[:, 0], training_X[:, 1], c=training_y)
plt.legend()
plt.show()

import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets
from sklearn import svm

circle_X, circle_y = datasets.make_circles(n_samples=300, noise=0.05)

plt.scatter(circle_X[:, 0], circle_X[:, 1], c=circle_y, marker='.')
plt.show()

nonlinear_clf = svm.SVC(kernel='rbf', C=1.0)

nonlinear_clf.fit(circle_X, circle_y)
SVC()

# Plot the decision boundary for a non-linear SVM problem
def plot_decision_boundary(model, ax=None):
    if ax is None:
        ax = plt.gca()

    xlim = ax.get_xlim()
```

```
    ylim = ax.get_ylim()

    # create grid to evaluate model
    x = np.linspace(xlim[0], xlim[1], 30)
    y = np.linspace(ylim[0], ylim[1], 30)
    Y, X = np.meshgrid(y, x)

  # shape data
    xy = np.vstack([X.ravel(), Y.ravel()]).T

  # get the decision boundary based on the model
    P = model.decision_function(xy).reshape(X.shape)

    # plot decision boundary
    ax.contour(X, Y, P,
               levels=[0], alpha=0.5,
               linestyles=['-'])

plt.scatter(circle_X[:, 0], circle_X[:, 1], c=circle_y, s=50)
plot_decision_boundary(nonlinear_clf)
plt.scatter(nonlinear_clf.support_vectors_[:, 0], nonlinear_clf.support_vectors_[:, 1], s=50,
 lw=1, facecolors='none')

plt.show()
```

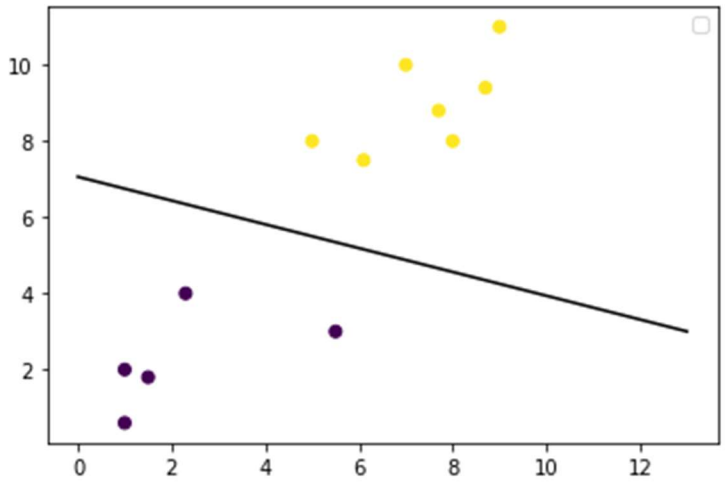
3. Result/Output

```
[ ]  # show unclassified data
     plt.scatter(X, y)
     plt.show()
```

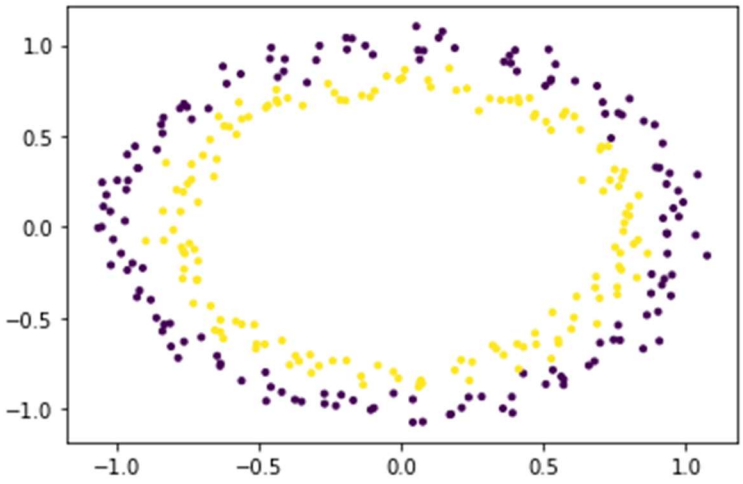


```
# get the weight values for the linear equation from the trained SVM model
w = clf.coef_[0]

# get the y-offset for the linear equation
a = -w[0] / w[1]

# make the x-axis space for the data points
XX = np.linspace(0, 13)

# get the y-values to plot the decision boundary
yy = a * XX - clf.intercept_[0] / w[1]

# plot the decision boundary
plt.plot(XX, yy, 'k-')

# show the plot visually
plt.scatter(training_X[:, 0], training_X[:, 1], c=training_y)
plt.legend()
plt.show()
```

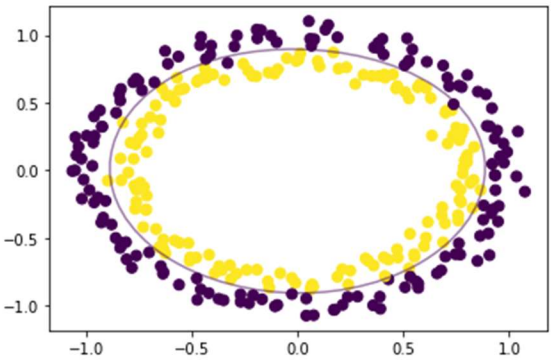
```
WARNING:matplotlib.legend:No handles with labels found to put in legend.
```



```
[ ]  # show raw non-linear data
     plt.scatter(circle_X[:, 0], circle_X[:, 1], c=circle_y, marker='.')
     plt.show()
```




```
# plot data and decision boundary
plt.scatter(circle_X[:, 0], circle_X[:, 1], c=circle_y, s=50)
plot_decision_boundary(nonlinear_clf)
plt.scatter(nonlinear_clf.support_vectors_[:, 0], nonlinear_clf.support_vectors_[:, 1], s=50, lw=1, facecolors='none')
plt.show()
```



Learning outcomes (What I have learnt):

1. Understanding of Exploratory Data analysis.

2. Able to analyze different datasets with the help of python and pandas library.
3. Learning about different library/packages of python.
4. Learning about the different methods, that are needed to analyze the given dataset.
5. Learning of different Machine Learning Functions

Evaluation Grid :

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.	Student Performance (Conduct of experiment) objectives/Outcomes.		12
2.	Viva Voce		10
3.	Submission of Work Sheet (Record)		8
	Total		30