Experiment No- 1.4

Student Name: Yash Kumar **UID: 20BCS9256**
Branch: CSE **Section/Group: 616- B**
Semester: 5 **Date of Performance: 27-09-22**
Subject Name: Machine Learning **Subject Code: 20CSP -317**

1. **Aim/Overview of the practical:** Implementing SVM on any dataset and analyse the accuracy.

2. Task to be done/ Which logistics used: Analysing accuracy by implementing the SVM on any dataset.

3. **Steps of experiment/Code:**

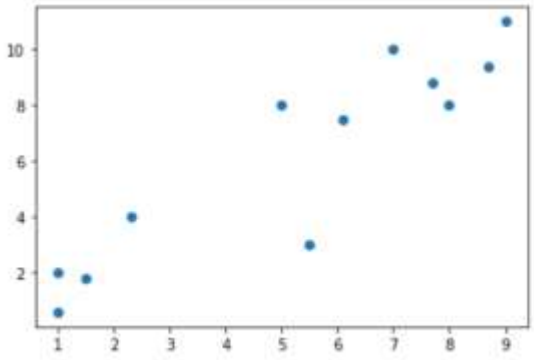
1. **Importing libraries such as matplotlib and SVM.**

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

```
#linear data
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)
```

```
<matplotlib.collections.PathCollection at 0x1bf924c6b80>
```



2. Shaping the data for training the model and defining the model:

```
#shaping data for training the model
training_x= np.vstack((x,y)).T
training_y= [0,1,0,1,0,1,1,1,0,0,1,1,]
```

```
#define the model
clf= svm.SVC(kernel='linear', C=1.0)
```

```
#train the model
clf.fit(training_x,training_y)
```

```
SVC(kernel='linear')
```

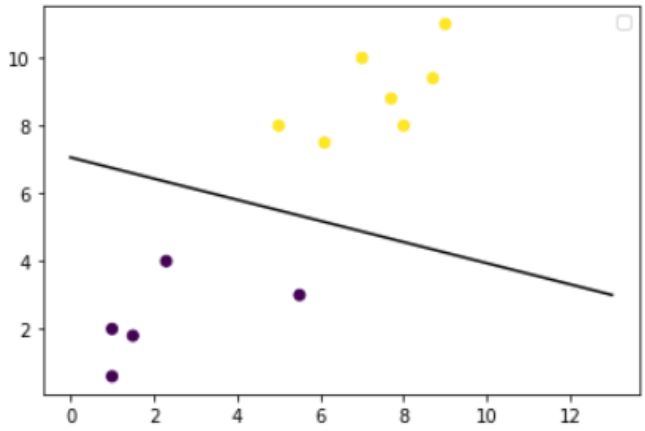
3. Get the weight values for the linear equation from the trained SVM model.

```
# 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-')
```

4. Showing the plot visually.

```
# show the plot visually

plt.scatter(training_x[:,0], training_x[:,1], c= training_y)
plt.legend()
plt.show()
```

```
No handles with labels found to put in legend.
```



5. Again importing libraries to work for non- linear data:

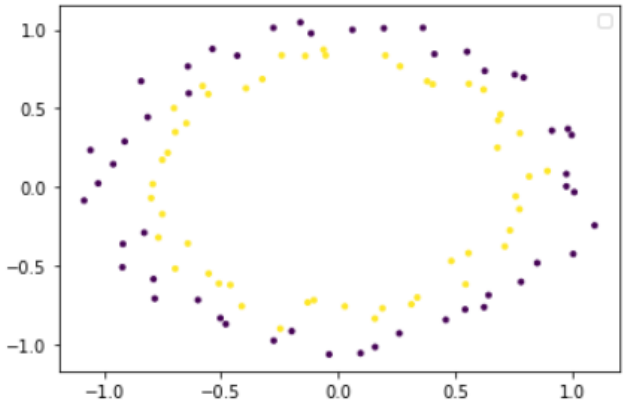
```
import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets
from sklearn import svm
```

6. Showing raw non-linear data and making non-linear algorithm for model:

```
# non- linear data

circle_x, circle_y= datasets.make_circles(n_samples=100, noise= 0.05)
```

```
# show raw non-linear data

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

```
No handles with labels found to put in legend.
```




```
# make non-linear algortihm for model

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

```
# training non-linear model

nonLinear_clf.fit(circle_x, circle_y)
```

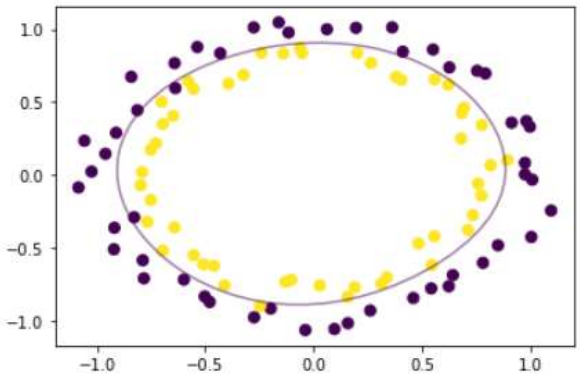
```
SVC()
```

7. Plot the decision boundary for non-linear SVM problem:

```
# Plot the decision boundary for non_linear svm problem

def plot_decision_boundary(model, ax=None):
    if ax is None:
        ax= plt.gca()

    x_lim= ax.get_xlim()
    y_lim= ax.get_ylim()

    # create grid to evaluate model

    x= np.linspace(x_lim[0], x_lim[1], 30)
    y= np.linspace(y_lim[0], y_lim[1], 30)

    Y, X= np.meshgrid(y,x)

   # show data

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

   # get the decision boundary based on the decision model

    P= model.decision_function(xy).reshape(X.shape)

    # Plot decision boundary

    ax.contour(X, Y, P,
               levels =[0], alpha= 0.5,
               linestyles= ['-'])
```

8. Plotting data and decision boundary:

```
# plot data and decsion 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()
```



9. Checking accuracy:

```
:  from sklearn.model_selection import cross_val_score
   clf = svm.SVC(kernel='linear', C=1, random_state=42)
   score=cross_val_score(clf, training_x,training_y, cv=10)
   print("avg accuracy:\t{0:, .4f)".format(np.mean(score)))
   print("avg std:\t{0:,.4f}".format(np.std(score)))
```

Learning Outcomes (What I have learnt):

1. I have learnt about implementing SVM for both linear as well as non- linear data.

2. I have learnt about plotting data and decision boundary.

3. I have learnt about various libraries which are supported by python such as sklearn, numpy, matplotlib.

4. I have learnt about the various functions provided by various libraries.

5. I have understood the experiment very well.

Evaluation Grid:

	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