Experiment:- 4

Student Name:Anshuman Singh **UID: 20BCS2665**
Branch: CSE **Section/Group: 20BCS_WM_902/A**
Semester: 5th **Subject Code: 20CSP-317**
Subject Name: MACHINE LEARNING LAB

Aim/Overview of the practical: Implement SVM on any data set and analyze the accuracy with Logistic regression.

Task to be done:

Implement SVM on any data set.

Apparatus/Simulator used:

- Jupyter Notebook/Google Collab
- Python
- pandas Library
- seaborn Library
- Standard Dataset

Code and Output:

exp 4

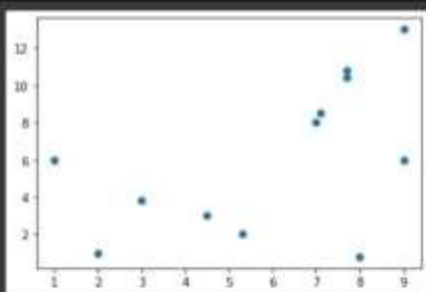
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[1]
```
import matplotlib.pyplot as plt
import numpy as np
from sklearn import svm
```

[15]
```
# linear data
X = np.array([2, 9, 3, 8, 1, 9, 7, 7.7, 5.3, 4.5, 7.7, 7.1])
y = np.array([1, 6, 3.8, 0.8, 6, 13, 8, 10.4, 2, 3, 10.8, 8.5])
```

[16]
```
# show unclassified data
plt.scatter(X, y)
plt.show()
```



[17]
```
# 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)
```

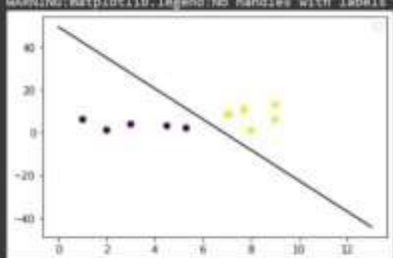
[19]
```
# train the model
clf.fit(training_X, training_y)
```

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

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

```python
# 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.
```



✓ 0s completed at 3:43 PM

exp 4 ☆

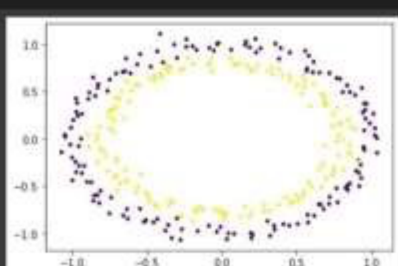
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

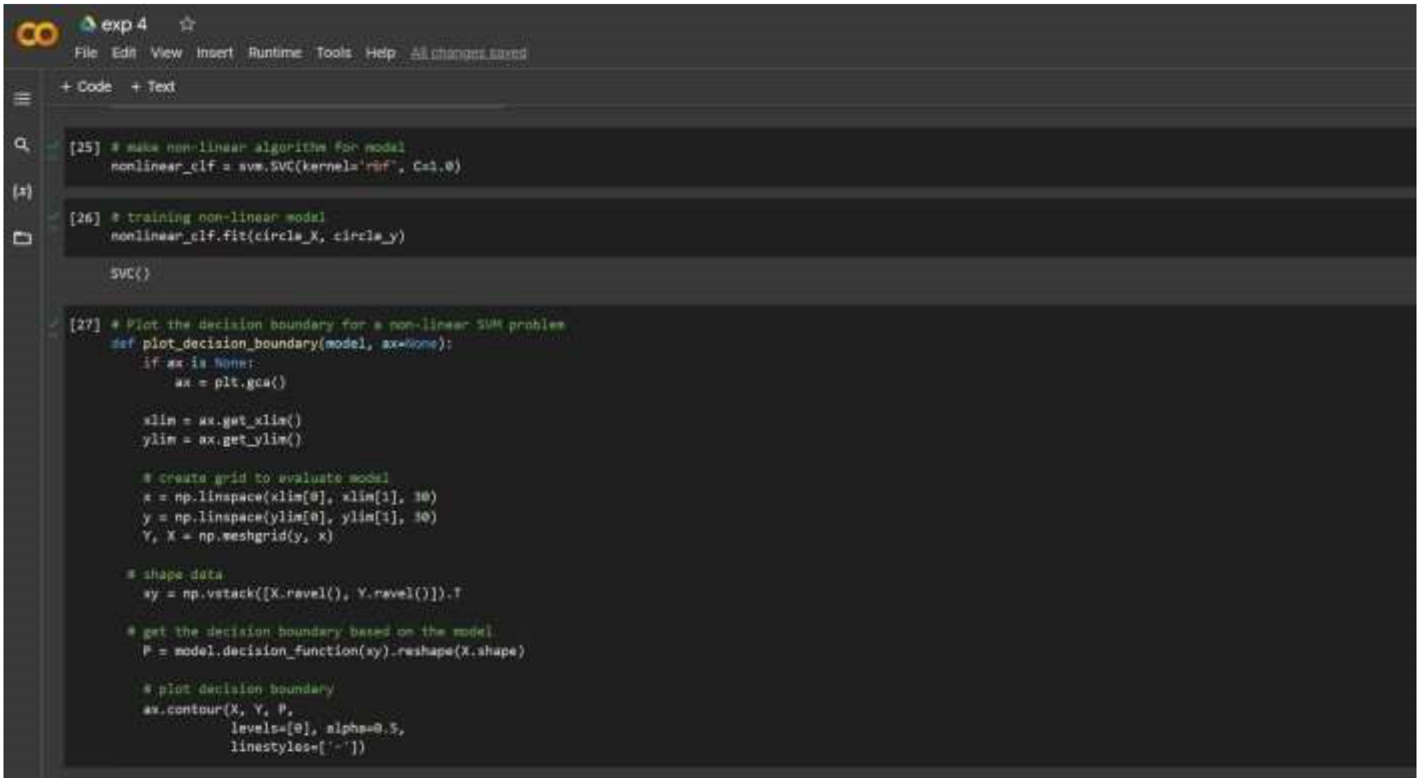
```python
[21] import matplotlib.pyplot as plt
     import numpy as np
     from sklearn import datasets
     from sklearn import svm
```

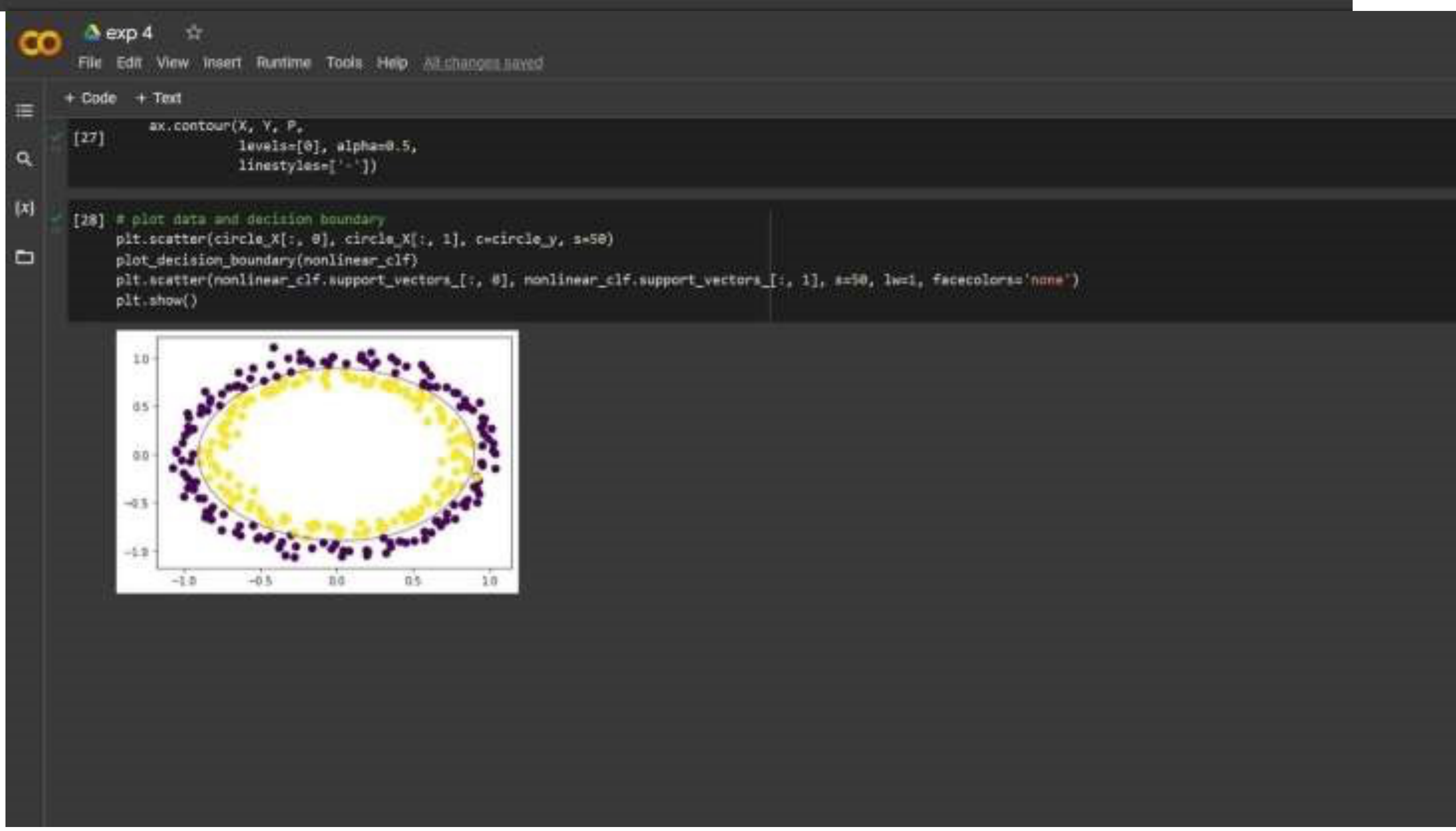
```python
[23] # non-linear data
     circle_X, circle_y = datasets.make_circles(n_samples=300, noise=0.05)
```

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

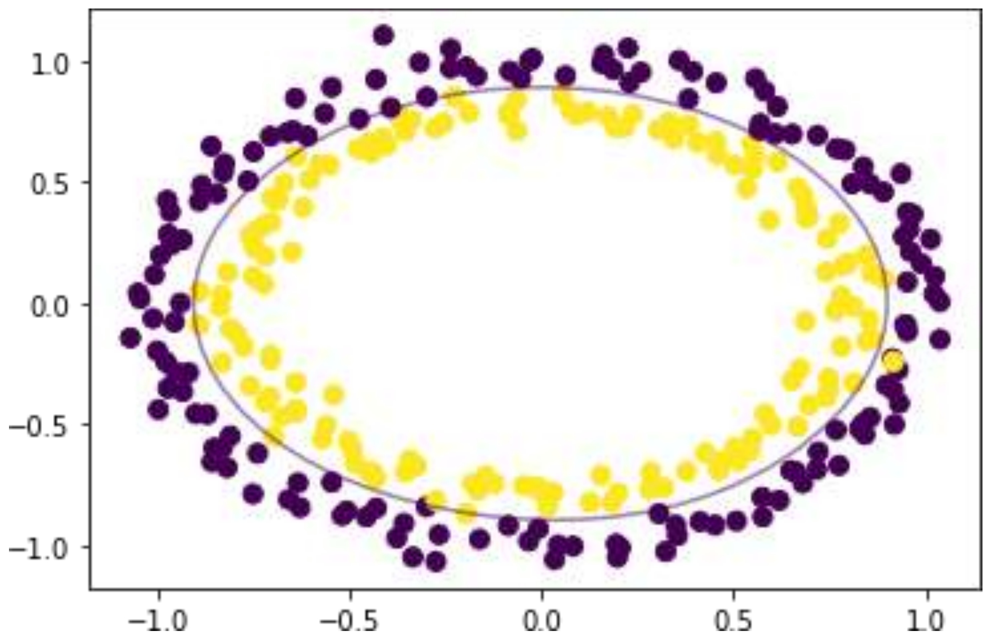



```python
[25] # make non-linear algorithm for model
     nonlinear_clf = svm.SVC(kernel='rbf', C=1.0)
```





FINAL OUTPUT:



Learning outcomes (What I have learnt):

1. To understand Data Visualization.
2. Learn about pandas', matplotlib and seaborn library/package of python.
3. Learn about the different methods/functions that are needed to generate different types of graphs, charts and plots of the given dataset.
4. Leaned about regression line, KDE.