Manyam Avinash

187233

DSC LAB-8

Assignment-7

SVM

In [1]:

```
from sklearn import svm
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import pandas as pd
```

In [2]:

```
data = pd.read_csv('spambase.data',header=None)
data.head()
```

Out[2]:

	0	1	2	3	4	5	6	7	8	9	...	48	49	50	51	52	53	54	55	56	57
0	0.00	0.64	0.64	0.0	0.32	0.00	0.00	0.00	0.00	0.00	...	0.00	0.000	0.0	0.778	0.000	0.000	3.756	61	278	1
1	0.21	0.28	0.50	0.0	0.14	0.28	0.21	0.07	0.00	0.94	...	0.00	0.132	0.0	0.372	0.180	0.048	5.114	101	1028	1
2	0.06	0.00	0.71	0.0	1.23	0.19	0.19	0.12	0.64	0.25	...	0.01	0.143	0.0	0.276	0.184	0.010	9.821	485	2259	1
3	0.00	0.00	0.00	0.0	0.63	0.00	0.31	0.63	0.31	0.63	...	0.00	0.137	0.0	0.137	0.000	0.000	3.537	40	191	1
4	0.00	0.00	0.00	0.0	0.63	0.00	0.31	0.63	0.31	0.63	...	0.00	0.135	0.0	0.135	0.000	0.000	3.537	40	191	1

5 rows × 58 columns

In [3]:

```
X = data.iloc[:, :57].values
print(X)
```

```
[[0.000e+00 6.400e-01 6.400e-01 ... 3.756e+00 6.100e+01 2.780e+02]
 [2.100e-01 2.800e-01 5.000e-01 ... 5.114e+00 1.010e+02 1.028e+03]
 [6.000e-02 0.000e+00 7.100e-01 ... 9.821e+00 4.850e+02 2.259e+03]
 ...
 [3.000e-01 0.000e+00 3.000e-01 ... 1.404e+00 6.000e+00 1.180e+02]
 [9.600e-01 0.000e+00 0.000e+00 ... 1.147e+00 5.000e+00 7.800e+01]
 [0.000e+00 0.000e+00 6.500e-01 ... 1.250e+00 5.000e+00 4.000e+01]]
```

In [4]:

```
Y = data[57].values
print(Y)
```

```
[1 1 1 ... 0 0 0]
```

In [5]:

```
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,train_size=0.7)
print("Training_data size :" , X_train.shape)
print("Test_data size :" , X_test.shape)
```

```
Training_data size : (3220, 57)
Test_data size : (1381, 57)
```

LInear

In [6]:

```
clf = svm.SVC(kernel='linear')
clf.fit(X_train,Y_train)
prediction = clf.predict(X_test)
accuracy = accuracy_score(Y_test,prediction)
print(prediction)
print("accuracy =",accuracy*100)
```

```
[0 0 0 ... 0 0 0]
accuracy = 92.39681390296887
```

Quadratic

In [7]:

```
clf2 = svm.SVC(kernel='poly',degree=2)
clf2.fit(X_train,Y_train)
clf2
prediction = clf.predict(X_test)
accuracy = accuracy_score(Y_test,prediction)
print(prediction)
print("accuracy =",accuracy*100)
```

```
[0 0 0 ... 0 0 0]
accuracy = 92.39681390296887
```

In [ ]:

```

```