

Manyam Avinash

187233

DSC LAB-8

Assignment-7

In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/spambase/spambase.data"
colnames=list(range(57))
colnames.append("spam")
# print(colnames)
dataset = pd.read_csv(url, names=colnames)
dataset.head()
```

Out[1]:

	0	1	2	3	4	5	6	7	8	9	...	48	49	50	51	52	53	54	55	56	spam
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 x 58 columns

In [2]:

```
X = dataset.drop('spam', axis=1)
y = dataset['spam']
```

In [3]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.30)
```

In [4]:

```
from sklearn.svm import SVC
svclassifier = SVC(kernel='poly', degree=2)
svclassifier.fit(X_train, y_train)
y_pred = svclassifier.predict(X_test)
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
[[787  18]
 [480  96]]

              precision    recall  f1-score   support

         0       0.62      0.98      0.76       805
         1       0.84      0.17      0.28       576

 accuracy          0.64       1381
 macro avg          0.73       1381
weighted avg          0.71       1381
```

In [5]:

```
from sklearn.svm import SVC
C = 3.0
svclassifier = SVC(kernel='poly', degree=2,C=C)
svclassifier.fit(X_train, y_train)
y_pred = svclassifier.predict(X_test)
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
[[784  21]
 [476 100]]
```

	precision	recall	f1-score	support
0	0.62	0.97	0.76	805
1	0.83	0.17	0.29	576
accuracy			0.64	1381
macro avg	0.72	0.57	0.52	1381
weighted avg	0.71	0.64	0.56	1381

In [6]:

```
from sklearn.svm import SVC
C = 1000.0
svclassifier = SVC(kernel='poly', degree=2 , C=C)
svclassifier.fit(X_train, y_train)
y_pred = svclassifier.predict(X_test)
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
[[795  10]
 [342 234]]
```

	precision	recall	f1-score	support
0	0.70	0.99	0.82	805
1	0.96	0.41	0.57	576
accuracy			0.75	1381
macro avg	0.83	0.70	0.69	1381
weighted avg	0.81	0.75	0.72	1381

In [7]:

```
from sklearn.svm import SVC
svclassifier = SVC(kernel='linear')
svclassifier.fit(X_train, y_train)
y_pred = svclassifier.predict(X_test)
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
```

```
[[762  43]
 [ 48 528]]
```

	precision	recall	f1-score	support
0	0.94	0.95	0.94	805
1	0.92	0.92	0.92	576
accuracy			0.93	1381
macro avg	0.93	0.93	0.93	1381
weighted avg	0.93	0.93	0.93	1381

In [8]:

```
from sklearn.svm import SVC
```

```

C=3.0
svclassifier = SVC(kernel='linear' , C=C)
svclassifier.fit(X_train, y_train)
y_pred = svclassifier.predict(X_test)
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))

```

```

[[763  42]
 [ 49 527]]

```

	precision	recall	f1-score	support
0	0.94	0.95	0.94	805
1	0.93	0.91	0.92	576
accuracy			0.93	1381
macro avg	0.93	0.93	0.93	1381
weighted avg	0.93	0.93	0.93	1381

In []:

```

from sklearn.svm import SVC
C=1000.0
svclassifier = SVC(kernel='linear' , C=C)
svclassifier.fit(X_train, y_train)
y_pred = svclassifier.predict(X_test)
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))

```

In []:

```


```