K-Nearest Neighbors and Support Vector Machine Algorithms

Name : Satbhai Anket Shantaram

Roll No. : 4286, Batch : B7

In [1]:
```
import pandas as pd
df=pd.read_csv('C:\\Users\\DELL\\Downloads\\emails.csv')
df.head(2)
```

Out[1]:

	Email No.	the	to	ect	and	for	of	a	you	hou	...	connevey	jay	valued	lay	infrastructure
0	Email 1	0	0	1	0	0	0	2	0	0	...	0	0	0	0	0
1	Email 2	8	13	24	6	6	2	102	1	27	...	0	0	0	0	0

2 rows × 3002 columns

In [2]:
```
df.tail(2)
```

Out[2]:

	Email No.	the	to	ect	and	for	of	a	you	hou	...	connevey	jay	valued	lay	infrastruc
5170	Email 5171	2	7	1	0	2	1	28	2	0	...	0	0	0	0	
5171	Email 5172	22	24	5	1	6	5	148	8	2	...	0	0	0	0	

2 rows × 3002 columns

In [3]:
```
df.shape
```

Out[3]:
```
(5172, 3002)
```

In [4]:
```
x=df.drop(['Email No.','Prediction'],axis=1)
y=df['Prediction']
```

In [5]:
```
x.shape
```

Out[5]:
```
(5172, 3000)
```

In [6]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25,random_state=50
```

In [7]:
```
x_train.shape
```

Out[7]:
```
(3879, 3000)
```

In [8]:
```
x_test.shape
```

Out[8]:
```
(1293, 3000)
```

K-Nearest Neighbors Algorithm

In [9]:
```
from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier()
knn.fit(x_train , y_train)
```

Out[9]:
```
KNeighborsClassifier()
```

In [10]:
```
y_pred=knn.predict(x_test)
```

In [11]:
```
from sklearn import metrics as mt
# for calculate accuracy and error
```

In [12]:
```
# Accuracy our model
mt.accuracy_score(y_pred,y_test)
```

Out[12]:
```
0.8739365815931941
```

In [13]:
```
# Error our model
mt.mean_absolute_error(y_pred,y_test)
```

Out[13]:
```
0.12606341840680588
```

In [14]:
```
# mean square error
a=mt.mean_squared_error(y_pred,y_test)
a
```

Out[14]:
```
0.12606341840680588
```

In [15]:
```
# root mean squared error
a**(1/2)
```

Out[15]:
```
0.35505410630889184
```

Support Vector Machine Algorithm

In [16]:
```
from sklearn.svm import SVC
svm=SVC(C=1)
svm.fit(x_train,y_train)
```

Out[16]:
```
SVC(C=1)
```

In [17]:
```
pred_y=svm.predict(x_test)
```

In [18]:
```
# accuracy our model
mt.accuracy_score(pred_y,y_test)
```

Out[18]:
```
0.7950502706883217
```

In [19]:
```
# error our model
mt.mean_absolute_error(pred_y,y_test)
```

Out[19]:
```
0.20494972931167826
```

In [20]:
```
# mean square error
b=mt.mean_squared_error(pred_y,y_test)
b
```

Out[20]:
```
0.20494972931167826
```

In [21]:
```
# root mean square error
b**(1/2)
```

Out[21]:
```
0.45271373881480365
```