



```
In [2]: import pandas as pd
import numpy as np
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
from sklearn.svm import SVC
```

```
In [3]: df=pd.read_csv('emails.csv')
```

```
In [4]: df.head()
```

```
Out[4]:
```

	Email No.	the	to	ect	and	for	of	a	you	hou	...	connevey	jay	valued	I
0	Email 1	0	0	1	0	0	0	2	0	0	...	0	0	0	
1	Email 2	8	13	24	6	6	2	102	1	27	...	0	0	0	
2	Email 3	0	0	1	0	0	0	8	0	0	...	0	0	0	
3	Email 4	0	5	22	0	5	1	51	2	10	...	0	0	0	
4	Email 5	7	6	17	1	5	2	57	0	9	...	0	0	0	

5 rows × 3002 columns

```
In [6]: df.columns
```

```
Out[6]: Index(['Email No.', 'the', 'to', 'ect', 'and', 'for', 'of', 'a', 'you', 'ho  
u',  
...,  
            'connevey', 'jay', 'valued', 'lay', 'infrastructure', 'military',  
            'allowing', 'ff', 'dry', 'Prediction'],  
              dtype='object', length=3002)
```

```
In [7]: df.details
```

```
Out[7]: 0      0
1      0
2      0
3      0
4      0
..
5167   0
5168   0
5169   0
5170   0
5171   0
Name: details, Length: 5172, dtype: int64
```

```
In [9]: df.isnull().sum()
```

```
Out[9]: Email No.      0
the      0
to      0
ect      0
and      0
..
military  0
allowing  0
ff      0
dry      0
Prediction 0
Length: 3002, dtype: int64
```

```
In [14]: X= df.drop(columns=['Prediction','Email No.'])
Y=df['Prediction']
```

```
In [15]: from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.2,random_stat
```

```
In [16]: knn= KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train,Y_train)
knn_pred=knn.predict(X_test)
```

```
In [18]: print("Accuracy:",accuracy_score(Y_test,knn_pred))
print("Confusion_metrix:",confusion_matrix(Y_test,knn_pred))
print("Classification_report:",classification_report(Y_test,knn_pred))
```

Accuracy: 0.8628019323671497

Confusion_metrix: [[646 93]
[49 247]]

Classification_report:		precision	recall	f1-score	support
0	0.93	0.87	0.90	739	
1	0.73	0.83	0.78	296	
accuracy		0.86		1035	
macro avg	0.83	0.85	0.84	1035	
weighted avg	0.87	0.86	0.87	1035	

```
In [22]: svm=SVC(kernel='linear')
svm.fit(X_train,Y_train)
svm_pred=svm.predict(X_test)
```

```
In [23]: print("Accuracy:",accuracy_score(Y_test,svm_pred))
print("Confusion_metrix:",confusion_matrix(Y_test,svm_pred))
print("Classification_report:",classification_report(Y_test,svm_pred))
```

Accuracy: 0.9594202898550724

Confusion_metrix: [[715 24]
[18 278]]

Classification_report:			precision	recall	f1-score	support
0	0.98	0.97	0.97	739		
1	0.92	0.94	0.93	296		
accuracy			0.96	1035		
macro avg			0.95	0.95	0.95	1035
weighted avg			0.96	0.96	0.96	1035

In []: