In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
import matplotlib.pyplot as plt
```

In [3]:
```
from sklearn.model_selection import GridSearchCV
from sklearn import svm
```

In [4]:
```
df = pd.read_csv('emails.csv')
```

In [5]:
```
df.head()
```

Out[5]:

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

5 rows × 3002 columns

In [6]:
```
df.tail()
```

Out[6]:

	Email No.	the	to	ect	and	for	of	a	you	hou	...	connevey	jay	valued	lay	infrastructure	militar
5167	Email 5168	2	2	2	3	0	0	32	0	0	...	0	0	0	0	0	
5168	Email 5169	35	27	11	2	6	5	151	4	3	...	0	0	0	0	0	
5169	Email 5170	0	0	1	1	0	0	11	0	0	...	0	0	0	0	0	
5170	Email 5171	2	7	1	0	2	1	28	2	0	...	0	0	0	0	0	
5171	Email 5172	22	24	5	1	6	5	148	8	2	...	0	0	0	0	0	

5 rows × 3002 columns

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5172 entries, 0 to 5171
Columns: 3002 entries, Email No. to Prediction
dtypes: int64(3001), object(1)
memory usage: 118.5+ MB
```

In [8]:
```
df.isnull().sum()
```

Out[8]:
```
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 [9]:
```
df.dropna()
```

Out[9]:

	Email No.	the	to	ect	and	for	of	a	you	hou	...	connevey	jay	valued	lay	infrastructure	militar
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	Email 3	0	0	1	0	0	0	8	0	0	...	0	0	0	0	0	
3	Email 4	0	5	22	0	5	1	51	2	10	...	0	0	0	0	0	
4	Email 5	7	6	17	1	5	2	57	0	9	...	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	.
5167	Email 5168	2	2	2	3	0	0	32	0	0	...	0	0	0	0	0	
5168	Email 5169	35	27	11	2	6	5	151	4	3	...	0	0	0	0	0	
5169	Email 5170	0	0	1	1	0	0	11	0	0	...	0	0	0	0	0	
5170	Email 5171	2	7	1	0	2	1	28	2	0	...	0	0	0	0	0	
5171	Email 5172	22	24	5	1	6	5	148	8	2	...	0	0	0	0	0	

5172 rows × 3002 columns

In [10]:
```
df.describe()
```

Out[10]:

	the	to	ect	and	for	of	a	y
count	5172.000000	5172.000000	5172.000000	5172.000000	5172.000000	5172.000000	5172.000000	5172.0000
mean	6.640565	6.188128	5.143852	3.075599	3.124710	2.627030	55.517401	2.4665
std	11.745009	9.534576	14.101142	6.045970	4.680522	6.229845	87.574172	4.3144
min	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.0000
25%	0.000000	1.000000	1.000000	0.000000	1.000000	0.000000	12.000000	0.0000
50%	3.000000	3.000000	1.000000	1.000000	2.000000	1.000000	28.000000	1.0000
75%	8.000000	7.000000	4.000000	3.000000	4.000000	2.000000	62.250000	3.0000
max	210.000000	132.000000	344.000000	89.000000	47.000000	77.000000	1898.000000	70.0000

8 rows × 3001 columns

In [11]:
```
df.shape
```

Out[11]:
```
(5172, 3002)
```

In [39]:
```
X = df.drop(columns=['Email No.', 'Prediction'])
y = df['Prediction']
```

In [40]:
```
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_state=42
```

In [41]:
```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=5)
```

In [42]:
```
knn.fit(X_train, y_train)

# Predict on the test data
y_pred = knn.predict(X_test)
```

In [43]:
```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_
accuracy = accuracy_score(y_test, y_pred)
accuracy
```

Out[43]:
```
0.8628019323671497
```

In [44]:
```
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
roc_auc = roc_auc_score(y_test, y_pred)
print(precision)
print(recall)
print(f1)
print(roc_auc)
```

```
0.7251461988304093
0.8378378378378378
0.7774294670846394
0.855319460190908
```

In [ ]: