In [27]:
```
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score, f1_score,recall_scc
```

In [5]:
```
df=pd.read_csv('C:/Users/NITISH BOKKA/Downloads/archive (12)/emails.csv')
```

In [7]:
```
df.isnull().sum()
```

Out[7]:
```
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 [11]:
```
df=df.drop('Email No.', axis=True)
```

In [13]:
```
X=df.drop('Prediction',axis=1)
y=df['Prediction']
```

In [17]:
```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=42
```

In [19]:
```
svm=SVC(gamma='auto')
svm.fit(X_train,y_train)
y_pred=svm.predict(X_test)
```

In [35]:
```
print(accuracy_score(y_test,y_pred))
print(precision_score(y_test,y_pred))
print(recall_score(y_test,y_pred))
print(1-accuracy_score(y_test,y_pred))
```

```
0.9120772946859903
0.89272030651341
0.7871621621621622
0.0879227053140097
```

In [41]:
```
n_neighbors=range(3,14)
for i in n_neighbors:
    knn=KNeighborsClassifier(n_neighbors=i)
    knn.fit(X_train,y_train)
    y_pred=knn.predict(X_test)
    print(i, accuracy_score(y_test,y_pred))
```

```
3 0.8685990338164251
4 0.8676328502415459
5 0.8628019323671497
6 0.8714975845410629
7 0.8685990338164251
8 0.8830917874396135
9 0.8782608695652174
10 0.885024154589372
11 0.8830917874396135
12 0.8879227053140096
13 0.8830917874396135
```

In [45]:
```
knn=KNeighborsClassifier(n_neighbors=12)
knn.fit(X_train,y_train)
y_pred=knn.predict(X_test)
```

In [46]:
```
print(accuracy_score(y_test,y_pred))
print(precision_score(y_test,y_pred))
print(recall_score(y_test,y_pred))
print(1-accuracy_score(y_test,y_pred))
```

```
0.8879227053140096
0.7903225806451613
0.8277027027027027
0.11207729468599037
```

In [49]:
```
n_neighbors=range(3,14)
for i in n_neighbors:
    knn=KNeighborsClassifier(n_neighbors=i,metric='manhattan')
    knn.fit(X_train,y_train)
    y_pred=knn.predict(X_test)
    print(i, accuracy_score(y_test,y_pred))
```

```
3 0.9043478260869565
4 0.9246376811594202
5 0.9014492753623189
6 0.9188405797101449
7 0.9014492753623189
8 0.9091787439613527
9 0.8946859903381642
10 0.9062801932367149
11 0.9033816425120773
12 0.9101449275362319
13 0.9004830917874396
```

In [51]:
```
knn=KNeighborsClassifier(n_neighbors=4, metric='manhattan')
knn.fit(X_train,y_train)
y_pred=knn.predict(X_test)
```

In [53]:
```
print(accuracy_score(y_test,y_pred))
print(precision_score(y_test,y_pred))
print(recall_score(y_test,y_pred))
print(1-accuracy_score(y_test,y_pred))
```

```
0.9246376811594202
0.8303030303030303
0.9256756756756757
0.07536231884057976
```

In [ ]: