In [1]:
```
from numpy import loadtxt
from xgboost import XGBClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

In [2]:
```
# load data
df = loadtxt('../data/clean_emails.csv', delimiter=",")
df = df.astype('int')
print(df)
```

```
[[    0     0     0]
 [    0     1     0]
 [    0     2     0]
 ...
 [ 7114  9428     0]
 [ 7114   414     0]
 [ 7114 73483     0]]
```

In [3]:
```
# split data into X and y
X = df[:,0:1]
Y = df[:,2]
```

In [4]:
```
print(Y)
print(len(Y))
print(max(Y))
```

```
[0 0 0 ... 0 0 0]
517372
33700
```

In [5]:
```
# split data into train and test sets
seed = 7
test_size = 0.33
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=test_siz
```

In [6]:
```
# Standard Decision Tree Classifier
from sklearn.metrics  import f1_score,accuracy_score
from sklearn.tree import DecisionTreeClassifier
DT= DecisionTreeClassifier()
DT.fit(X_train,y_train)
pred=DT.predict(X_test)
print(accuracy_score(y_test,pred))
```

```
0.7673911897524205
```

In [7]:

```
#Support Vector Machine Classifier
#from sklearn.svm import SVC
#svm_model_linear = SVC(kernel = 'linear', C = 1).fit(X_train, y_train)
#svm_predictions = svm_model_linear.predict(X_test)

# model accuracy for X_test
#accuracy = svm_model_linear.score(X_test, y_test)

# creating a confusion matrix
#cm = confusion_matrix(y_test, svm_predictions)
```

In [10]:

```
#KNN classifier
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors = 7).fit(X_train, y_train)

# accuracy on X_test
accuracy = knn.score(X_test, y_test)
print(accuracy)
```

```
0.7470787721178682
```

In [ ]: