**import** pandas **as** pd

df **=** pd**.**read\_csv('emails.csv')

df**.**shape

df**.**head()

*#input data*

x **=** df**.**drop(['Email No.', 'Prediction'], axis **=** 1)

y **=** df['Prediction']

x**.**dtypes

**import** seaborn **as** sb

sb**.**countplot(x **=** y)

y**.**value\_counts()

*# Feature Scaling*

**from** sklearn.preprocessing **import** MinMaxScaler

scaler **=** MinMaxScaler()

x\_scaled **=** scaler**.**fit\_transform(x)

x\_scaled

*# spliting data into training and testing*

*#cross-validation*

**from** sklearn.model\_selection **import** train\_test\_split

x\_train, x\_test, y\_train, y\_test **=** train\_test\_split(

x, y, random\_state **=** 0, test\_size **=** 0.25)

x\_scaled**.**shape

x\_train**.**shape

x\_test**.**shape

**from** sklearn.neighbors **import** KNeighborsClassifier

knn **=** KNeighborsClassifier(n\_neighbors**=**5)

*# train the algorithm*

knn**.**fit(x\_train,y\_train)

*# Predict on test data*

y\_pred **=** knn**.**predict(x\_test)

**from** sklearn.metrics **import** ConfusionMatrixDisplay, accuracy\_score

**from** sklearn.metrics **import** classification\_report

ConfusionMatrixDisplay**.**from\_predictions(y\_test, y\_pred)

y\_test**.**value\_counts()

accuracy\_score(y\_test,y\_pred)

print(classification\_report(y\_test,y\_pred))

**from** sklearn.svm **import** SVC

svm **=** SVC(kernel**=**'rbf')

svm**.**fit(x\_train,y\_train)

y\_pred **=** svm**.**predict(x\_test)

accuracy\_score(y\_test,y\_pred)