pip install numpy

pip install pandas

pip install seaborn

pip install matplotlib

pip install matplotlib.pyplot as plt

pip install sklearn

pip install sklearn.model

pip install sklearn.preprocessing

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

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

from sklearn.metrics import confusion\_matrix, accuracy\_score

from sklearn.neighbors import KNeighborsClassifier

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import confusion\_matrix, accuracy\_score

from sklearn.metrics import confusion\_matrix, accuracy\_score

from sklearn.metrics import classification\_report

df=pd.read\_csv("emails.csv")

print(df.head() )

print(df.info())

df.isnull().sum()

X = df.iloc[:, 1:-1].values

y = df.iloc[:, -1].values

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.30, random\_state=101)

sc\_X = StandardScaler()

X\_train = sc\_X.fit\_transform(X\_train)

X\_test = sc\_X.transform(X\_test)

classifier = KNeighborsClassifier(n\_neighbors=5)

classifier.fit(X\_train, y\_train)

y\_pred = classifier.predict(X\_test)

cm = confusion\_matrix(y\_test, y\_pred)

cm

cl\_report=classification\_report(y\_test,y\_pred)

print(cl\_report)

print("Accuracy Score for KNN : ", accuracy\_score(y\_pred,y\_test))

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

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

from sklearn.svm import SVC

from sklearn.metrics import accuracy\_score

from sklearn.metrics import confusion\_matrix, accuracy\_score

from sklearn.metrics import classification\_report

df=pd.read\_csv("emails.csv")

print(df.head() )

print(df.info())

print(df.isnull().sum() )

X = df.iloc[:, 1:-1].values

y = df.iloc[:, -1].values

X.shape

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.30, random\_state=101)

svc = SVC(C=1.0,kernel='rbf',gamma='auto')

svc.fit(X\_train,y\_train)

y\_pred2 = svc.predict(X\_test)

cm = confusion\_matrix(y\_test, y\_pred2)

cm

print("Accuracy Score for SVC : ", accuracy\_score(y\_pred2,y\_test))

cl\_report=classification\_report(y\_test,y\_pred2)

print(cl\_report)