```
import pandas as pd


df = pd.read_csv('D:/Training Datasets/emails.csv')


df.shape


df.head()

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

y = df['Prediction']

x.shape


x.dtypes


set(x.dtypes)


import seaborn as sns
sns.countplot(x = y);


# In[10]:


y.value_counts()


# In[11]:


# Feature Scaling
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
x_scaled = scaler.fit_transform(x)


# In[12]:


x_scaled


# In[13]:


# Cross Validation
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x_scaled, y,
random_state = 0 , test_size = 0.25)
```

```
# In[14]:


x_scaled.shape


# In[15]:


x_train.shape


# In[16]:


x_test.shape


# In[17]:


# Import the class
from sklearn.neighbors import KNeighborsClassifier


# In[18]:


# Create the object
knn = KNeighborsClassifier(n_neighbors=5)


# In[19]:


# Train the algorithm
knn.fit(x_train, y_train)


# In[20]:


# predict on test data
y_pred = knn.predict(x_test)


# In[21]:


# import the evaluation metrics
from sklearn.metrics import ConfusionMatrixDisplay, accuracy_score
from sklearn.metrics import classification_report


# In[22]:
```

```
ConfusionMatrixDisplay.from_predictions(y_test, y_pred)


# In[23]:


y_test.value_counts()


# In[24]:


accuracy_score(y_test, y_pred)


# In[25]:


print(classification_report(y_test, y_pred))


# In[26]:


import numpy as np
import matplotlib as plt


# In[27]:


error = []
for k in range(1,41):
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(x_train, y_train)
    pred = knn.predict(x_test)
    error.append(np.mean(pred != y_test))


# In[28]:


error


# In[29]:


knn = KNeighborsClassifier(n_neighbors=1)
knn.fit(x_train, y_train)


# In[30]:


y_pred = knn.predict(x_test)
```

```
# In[31]:


accuracy_score(y_test, y_pred)


# In[32]:


from sklearn.svm import SVC
svm = SVC(kernel='poly')
svm.fit(x_train, y_train)


# In[33]:


y_pred = svm.predict(x_test)


# In[34]:


accuracy_score(y_test, y_pred)


# In[35]:


# Linear:0.9767981438515081
# RBF:0.9450889404485692
# Poly:0.7548337200309359


# In[ ]:




# In[ ]:




# In[ ]:




# In[36]:


import pandas as pd
import numpy as np
import matplotlib as plb
```

```
import seaborn as sns

# In[37]:

df = pd.read_csv('D:/Training Datasets/emails.csv')

# In[38]:

df

# In[39]:

df.head()

# In[40]:

df.shape

# In[41]:

df.size

# In[42]:

df.dtypes

# In[43]:

set(df.dtypes)

# In[44]:

df.isnull().sum()

# In[45]:

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

```
# In[46]:


set(x.dtypes)


# In[47]:


y = df['Prediction']
y


# In[48]:


sns.countplot(x = y)


# In[49]:


y.value_counts()


# In[50]:


from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
x_scaled = scaler.fit_transform(x)
x_scaled


# In[51]:


from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x_scaled,y,random_state
= 0, test_size = 0.25)


# In[52]:


x_train.shape


# In[53]:


y_train.shape


# In[54]:


x_test.shape
```

```
# In[55]:


y_test.shape


# In[56]:


from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors = 5)
knn.fit(x_train,y_train)


# In[57]:


y_pred = knn.predict(x_test)
y_pred


# In[58]:


from sklearn.metrics import accuracy_score
accuracy_score(y_test,y_pred)


# In[59]:


from sklearn.svm import SVC
svm = SVC(kernel = "poly")
svm.fit(x_train, y_train)


# In[60]:


y_pred = svm.predict(x_test)
accuracy_score(y_test,y_pred)
```