

**import** pandas **as** pd **import** numpy **as** np **import** seaborn **as** sns

**import** matplotlib.pyplot **as** plt



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



**1**

Email

2

8 13 24 6 6 2 102 1 27 ...

0 0

0 0

0

0

0 1 0

**3**

Email

4

0 5 22 0 5 1 51 2 10 ...

0 0

0 0

0

0

0 0 0

**Email No.**

**the**

**to ect and for of a you hou ... connevey jay valued lay infrastructure military allowing ff dry Pre**

**0**

Email

1

0 0 1 0 0 0 2 0 0 ...

0 0

0 0

0

0

0 0 0

**2**

Email

3

0 0 1 0 0 0 8 0 0 ...

0 0

0 0

0

0

0 0 0

**4**

Email

5

7 6 17 1 5 2 57 0 9 ...

0 0

0 0

0

0

0 1 0

5 rows × 3002 columns

df**.**head()



df**.**info()

<class 'pandas.core.frame.DataFrame'> RangeIndex: 5172 entries, 0 to 5171

Columns: 3002 entries, Email No. to Prediction dtypes: int64(3001), object(1)

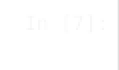
memory usage: 118.5+ MB



Length: 3002, dtype: int64

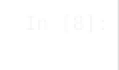
df**.**isnull()**.**sum()

| Email No. | 0 |
| --- | --- |
| the | 0 |
| to | 0 |
| ect | 0 |
| and | 0 |
|  | .. |
| military | 0 |
| allowing | 0 |
| ff | 0 |
| dry | 0 |
| Prediction | 0 |



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

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



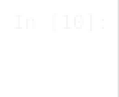
**from** sklearn.model\_selection **import** train\_test\_split

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



**from** sklearn.preprocessing **import** StandardScaler sc\_X **=** StandardScaler()

X\_train **=** sc\_X**.**fit\_transform(X\_train) X\_test **=** sc\_X**.**transform(X\_test)

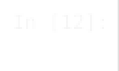


KNeighborsClassifier()

**from** sklearn.neighbors **import** KNeighborsClassifier classifier **=** KNeighborsClassifier(n\_neighbors**=**5) classifier**.**fit(X\_train, y\_train)



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



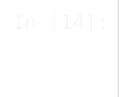
**from** sklearn.metrics **import** confusion\_matrix, accuracy\_score cm **=** confusion\_matrix(y\_test, y\_pred)



array([[866, 248],

[ 16, 422]], dtype=int64)

cm



**from** sklearn.metrics **import** classification\_report cl\_report**=**classification\_report(y\_test,y\_pred) print(cl\_report)

|  | precision | recall | f1-score | support |
| --- | --- | --- | --- | --- |
| 0 | 0.98 | 0.78 | 0.87 | 1114 |
| 1 | 0.63 | 0.96 | 0.76 | 438 |
| accuracy |  |  | 0.83 | 1552 |
| macro avg | 0.81 | 0.87 | 0.81 | 1552 |
| weighted avg | 0.88 | 0.83 | 0.84 | 1552 |



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

Accuracy Score for KNN : 0.8298969072164949

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js