ML_Prac2

October 18, 2023

1 Email Spam Classification

1.0.1 Aim: Classify the email using the binary classification method. Email Spam detection has two states: a) Normal State – Not Spam, b) Abnormal State – Spam. Use K-Nearest Neighbors and Support Vector Machine for classification

Name: Chinmay Gokhale

Div: BE-A

Roll No. B211047 #### import the libraries

[1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

[2]:
```
df=pd.read_csv('emails.csv')
df
```

[2]:

	Email No.	the	to	ect	and	for	of	a	you	hou	…	connevey	\
0	Email 1	0	0	1	0	0	0	2	0	0	…	0	
1	Email 2	8	13	24	6	6	2	102	1	27	…	0	
2	Email 3	0	0	1	0	0	0	8	0	0	…	0	
3	Email 4	0	5	22	0	5	1	51	2	10	…	0	
4	Email 5	7	6	17	1	5	2	57	0	9	…	0	
…	…	…	..	…	…	…	..	…	…	…	…	…	
5167	Email 5168	2	2	2	3	0	0	32	0	0	…	0	
5168	Email 5169	35	27	11	2	6	5	151	4	3	…	0	
5169	Email 5170	0	0	1	1	0	0	11	0	0	…	0	
5170	Email 5171	2	7	1	0	2	1	28	2	0	…	0	
5171	Email 5172	22	24	5	1	6	5	148	8	2	…	0	

	jay	valued	lay	infrastructure	military	allowing	ff	dry	\
0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	1	0	
2	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	

```
4       0       0    0               0          0         0   1    0
…   …     …   …             …        …       …   ..  …
5167    0       0    0               0          0         0   0    0
5168    0       0    0               0          0         0   1    0
5169    0       0    0               0          0         0   0    0
5170    0       0    0               0          0         0   1    0
5171    0       0    0               0          0         0   0    0

      Prediction
0              0
1              0
2              0
3              0
4              0
…            …
5167           0
5168           0
5169           1
5170           1
5171           0

[5172 rows x 3002 columns]
```

[3]:
```
df.shape
```

[3]:
```
(5172, 3002)
```

[4]:
```
df.size
```

[4]:
```
15526344
```

[5]:
```
df.head()
```

[5]:
```
   Email No.  the  to  ect  and  for  of    a  you  hou  …  connevey  jay  \
0   Email 1    0   0    1    0    0   0    2    0    0  …         0    0
1   Email 2    8  13   24    6    6   2  102    1   27  …         0    0
2   Email 3    0   0    1    0    0   0    8    0    0  …         0    0
3   Email 4    0   5   22    0    5   1   51    2   10  …         0    0
4   Email 5    7   6   17    1    5   2   57    0    9  …         0    0

   valued  lay  infrastructure  military  allowing  ff  dry  Prediction
0       0    0               0         0         0   0    0           0
1       0    0               0         0         0   1    0           0
2       0    0               0         0         0   0    0           0
3       0    0               0         0         0   0    0           0
4       0    0               0         0         0   1    0           0
```

```
[5 rows x 3002 columns]
```

[6]:
```
df.head
```

[6]:
```
<bound method NDFrame.head of        Email No.  the  to  ect  and  for  of    a
you  hou  …  connevey  \
0         Email 1    0   0    1    0    0   0    2    0    0  …          0
1         Email 2    8  13   24    6    6   2  102    1   27  …          0
2         Email 3    0   0    1    0    0   0    8    0    0  …          0
3         Email 4    0   5   22    0    5   1   51    2   10  …          0
4         Email 5    7   6   17    1    5   2   57    0    9  …          0
…           …  …  ..  …  …  …  ..  …  …  …  …       …
5167  Email 5168    2   2    2    3    0   0   32    0    0  …          0
5168  Email 5169   35  27   11    2    6   5  151    4    3  …          0
5169  Email 5170    0   0    1    1    0   0   11    0    0  …          0
5170  Email 5171    2   7    1    0    2   1   28    2    0  …          0
5171  Email 5172   22  24    5    1    6   5  148    8    2  …          0

      jay  valued  lay  infrastructure  military  allowing  ff  dry  \
0       0       0    0               0         0         0   0    0
1       0       0    0               0         0         0   1    0
2       0       0    0               0         0         0   0    0
3       0       0    0               0         0         0   0    0
4       0       0    0               0         0         0   1    0
…   …     …  …             …       …       …  ..  …
5167    0       0    0               0         0         0   0    0
5168    0       0    0               0         0         0   1    0
5169    0       0    0               0         0         0   0    0
5170    0       0    0               0         0         0   1    0
5171    0       0    0               0         0         0   0    0

      Prediction
0              0
1              0
2              0
3              0
4              0
…          …
5167           0
5168           0
5169           1
5170           1
5171           0

[5172 rows x 3002 columns]>
```

[7]:
```
df.tail
```

[7]:
```
<bound method NDFrame.tail of        Email No.  the  to  ect  and  for  of    a  you  hou  …  connevey  \
0         Email 1    0   0    1    0    0   0    2    0    0  …         0
1         Email 2    8  13   24    6    6   2  102    1   27  …         0
2         Email 3    0   0    1    0    0   0    8    0    0  …         0
3         Email 4    0   5   22    0    5   1   51    2   10  …         0
4         Email 5    7   6   17    1    5   2   57    0    9  …         0
…            …  …  ..  …  …  …  ..  …  …  …  …       …
5167  Email 5168    2   2    2    3    0   0   32    0    0  …         0
5168  Email 5169   35  27   11    2    6   5  151    4    3  …         0
5169  Email 5170    0   0    1    1    0   0   11    0    0  …         0
5170  Email 5171    2   7    1    0    2   1   28    2    0  …         0
5171  Email 5172   22  24    5    1    6   5  148    8    2  …         0

      jay  valued  lay  infrastructure  military  allowing  ff  dry  \
0       0       0    0               0         0         0   0    0
1       0       0    0               0         0         0   1    0
2       0       0    0               0         0         0   0    0
3       0       0    0               0         0         0   0    0
4       0       0    0               0         0         0   1    0
…   …      …  …             …        …       …  ..  …
5167    0       0    0               0         0         0   0    0
5168    0       0    0               0         0         0   1    0
5169    0       0    0               0         0         0   0    0
5170    0       0    0               0         0         0   1    0
5171    0       0    0               0         0         0   0    0

      Prediction
0              0
1              0
2              0
3              0
4              0
…          …
5167           0
5168           0
5169           1
5170           1
5171           0

[5172 rows x 3002 columns]>
```

[8]:
```
df.tail()
```

[8]:
```
       Email No.  the  to  ect  and  for  of    a  you  hou  …  connevey  \
5167  Email 5168    2   2    2    3    0   0   32    0    0  …         0
5168  Email 5169   35  27   11    2    6   5  151    4    3  …         0
```

```
5169  Email 5170    0   0    1    1    0   0   11    0    0  …         0
5170  Email 5171    2   7    1    0    2   1   28    2    0  …         0
5171  Email 5172   22  24    5    1    6   5  148    8    2  …         0

      jay  valued  lay  infrastructure  military  allowing  ff  dry  \
5167    0       0    0               0         0         0   0    0
5168    0       0    0               0         0         0   1    0
5169    0       0    0               0         0         0   0    0
5170    0       0    0               0         0         0   1    0
5171    0       0    0               0         0         0   0    0

      Prediction
5167           0
5168           0
5169           1
5170           1
5171           0

[5 rows x 3002 columns]
```

[9]:
```
df.describe
```

[9]:
```
<bound method NDFrame.describe of        Email No.  the  to  ect  and  for  of
a  you  hou  …  connevey  \
0        Email 1    0   0    1    0    0   0    2    0    0  …         0
1        Email 2    8  13   24    6    6   2  102    1   27  …         0
2        Email 3    0   0    1    0    0   0    8    0    0  …         0
3        Email 4    0   5   22    0    5   1   51    2   10  …         0
4        Email 5    7   6   17    1    5   2   57    0    9  …         0
…          …  …  ..  …  …  …  ..  …  …  …  …       …
5167  Email 5168    2   2    2    3    0   0   32    0    0  …         0
5168  Email 5169   35  27   11    2    6   5  151    4    3  …         0
5169  Email 5170    0   0    1    1    0   0   11    0    0  …         0
5170  Email 5171    2   7    1    0    2   1   28    2    0  …         0
5171  Email 5172   22  24    5    1    6   5  148    8    2  …         0

      jay  valued  lay  infrastructure  military  allowing  ff  dry  \
0       0       0    0               0         0         0   0    0
1       0       0    0               0         0         0   1    0
2       0       0    0               0         0         0   0    0
3       0       0    0               0         0         0   0    0
4       0       0    0               0         0         0   1    0
…   …     …  …             …       …       …  ..  …
5167    0       0    0               0         0         0   0    0
5168    0       0    0               0         0         0   1    0
5169    0       0    0               0         0         0   0    0
5170    0       0    0               0         0         0   1    0
```

```
5171    0       0    0               0         0         0   0    0

      Prediction
0              0
1              0
2              0
3              0
4              0
…            …
5167           0
5168           0
5169           1
5170           1
5171           0

[5172 rows x 3002 columns]>
```

[10]:
```
df.describe()
```

[10]:
```
                the           to          ect          and          for  \
count  5172.000000  5172.000000  5172.000000  5172.000000  5172.000000
mean      6.640565     6.188128     5.143852     3.075599     3.124710
std      11.745009     9.534576    14.101142     6.045970     4.680522
min       0.000000     0.000000     1.000000     0.000000     0.000000
25%       0.000000     1.000000     1.000000     0.000000     1.000000
50%       3.000000     3.000000     1.000000     1.000000     2.000000
75%       8.000000     7.000000     4.000000     3.000000     4.000000
max     210.000000   132.000000   344.000000    89.000000    47.000000

                of           a          you          hou           in  …  \
count  5172.000000  5172.000000  5172.000000  5172.000000  5172.000000  …
mean      2.627030    55.517401     2.466551     2.024362    10.600155  …
std       6.229845    87.574172     4.314444     6.967878    19.281892  …
min       0.000000     0.000000     0.000000     0.000000     0.000000  …
25%       0.000000    12.000000     0.000000     0.000000     1.000000  …
50%       1.000000    28.000000     1.000000     0.000000     5.000000  …
75%       2.000000    62.250000     3.000000     1.000000    12.000000  …
max      77.000000  1898.000000    70.000000   167.000000   223.000000  …

          connevey          jay       valued          lay  infrastructure  \
count  5172.000000  5172.000000  5172.000000  5172.000000     5172.000000
mean      0.005027     0.012568     0.010634     0.098028        0.004254
std       0.105788     0.199682     0.116693     0.569532        0.096252
min       0.000000     0.000000     0.000000     0.000000        0.000000
25%       0.000000     0.000000     0.000000     0.000000        0.000000
50%       0.000000     0.000000     0.000000     0.000000        0.000000
75%       0.000000     0.000000     0.000000     0.000000        0.000000
```

```
max       4.000000     7.000000     2.000000    12.000000        3.000000

          military     allowing           ff          dry   Prediction
count  5172.000000  5172.000000  5172.000000  5172.000000  5172.000000
mean      0.006574     0.004060     0.914733     0.006961     0.290023
std       0.138908     0.072145     2.780203     0.098086     0.453817
min       0.000000     0.000000     0.000000     0.000000     0.000000
25%       0.000000     0.000000     0.000000     0.000000     0.000000
50%       0.000000     0.000000     0.000000     0.000000     0.000000
75%       0.000000     0.000000     1.000000     0.000000     1.000000
max       4.000000     3.000000   114.000000     4.000000     1.000000

[8 rows x 3001 columns]
```

[11]:
```
df.dtypes
```

[11]:
```
Email No.     object
the            int64
to             int64
ect            int64
and            int64
               …
military       int64
allowing       int64
ff             int64
dry            int64
Prediction     int64
Length: 3002, dtype: object
```

[12]:
```
df
```

[12]:
```
        Email No.  the  to  ect  and  for  of    a  you  hou  …  connevey  \
0        Email 1    0   0    1    0    0   0    2    0    0  …         0
1        Email 2    8  13   24    6    6   2  102    1   27  …         0
2        Email 3    0   0    1    0    0   0    8    0    0  …         0
3        Email 4    0   5   22    0    5   1   51    2   10  …         0
4        Email 5    7   6   17    1    5   2   57    0    9  …         0
…          …   …  ..  …   …   …  ..  …   …  …  …       …
5167  Email 5168    2   2    2    3    0   0   32    0    0  …         0
5168  Email 5169   35  27   11    2    6   5  151    4    3  …         0
5169  Email 5170    0   0    1    1    0   0   11    0    0  …         0
5170  Email 5171    2   7    1    0    2   1   28    2    0  …         0
5171  Email 5172   22  24    5    1    6   5  148    8    2  …         0

      jay  valued  lay  infrastructure  military  allowing  ff  dry  \
0       0       0    0               0         0         0   0    0
1       0       0    0               0         0         0   1    0
```

```
2       0       0    0               0          0         0   0    0
3       0       0    0               0          0         0   0    0
4       0       0    0               0          0         0   1    0
…   …       …   …              …        …        …   ..   …
5167    0       0    0               0          0         0   0    0
5168    0       0    0               0          0         0   1    0
5169    0       0    0               0          0         0   0    0
5170    0       0    0               0          0         0   1    0
5171    0       0    0               0          0         0   0    0

      Prediction
0              0
1              0
2              0
3              0
4              0
…            …
5167           0
5168           0
5169           1
5170           1
5171           0

[5172 rows x 3002 columns]
```

2 Splitting the data into input and output data

[13]:
```
X=df.drop(["Email No.", "Prediction"],axis=1)
```

[14]:
```
X
```

[14]:
```
      the  to  ect  and  for  of    a  you  hou  in  …  enhancements  \
0       0   0    1    0    0   0    2    0    0   0  …              0
1       8  13   24    6    6   2  102    1   27  18  …              0
2       0   0    1    0    0   0    8    0    0   4  …              0
3       0   5   22    0    5   1   51    2   10   1  …              0
4       7   6   17    1    5   2   57    0    9   3  …              0
…   …   ..  …   …   …   ..  …   …   …   ..  …           …
5167    2   2    2    3    0   0   32    0    0   5  …              0
5168   35  27   11    2    6   5  151    4    3  23  …              0
5169    0   0    1    1    0   0   11    0    0   1  …              0
5170    2   7    1    0    2   1   28    2    0   8  …              0
5171   22  24    5    1    6   5  148    8    2  23  …              0

      connevey  jay  valued  lay  infrastructure  military  allowing  ff  dry
0            0    0       0    0               0         0         0   0    0
```

```
1            0    0       0    0                0          0          0   1    0
2            0    0       0    0                0          0          0   0    0
3            0    0       0    0                0          0          0   0    0
4            0    0       0    0                0          0          0   1    0
…          …  …    …  …                …        …      …  ..  …
5167         0    0       0    0                0          0          0   0    0
5168         0    0       0    0                0          0          0   1    0
5169         0    0       0    0                0          0          0   0    0
5170         0    0       0    0                0          0          0   1    0
5171         0    0       0    0                0          0          0   0    0

[5172 rows x 3000 columns]
```

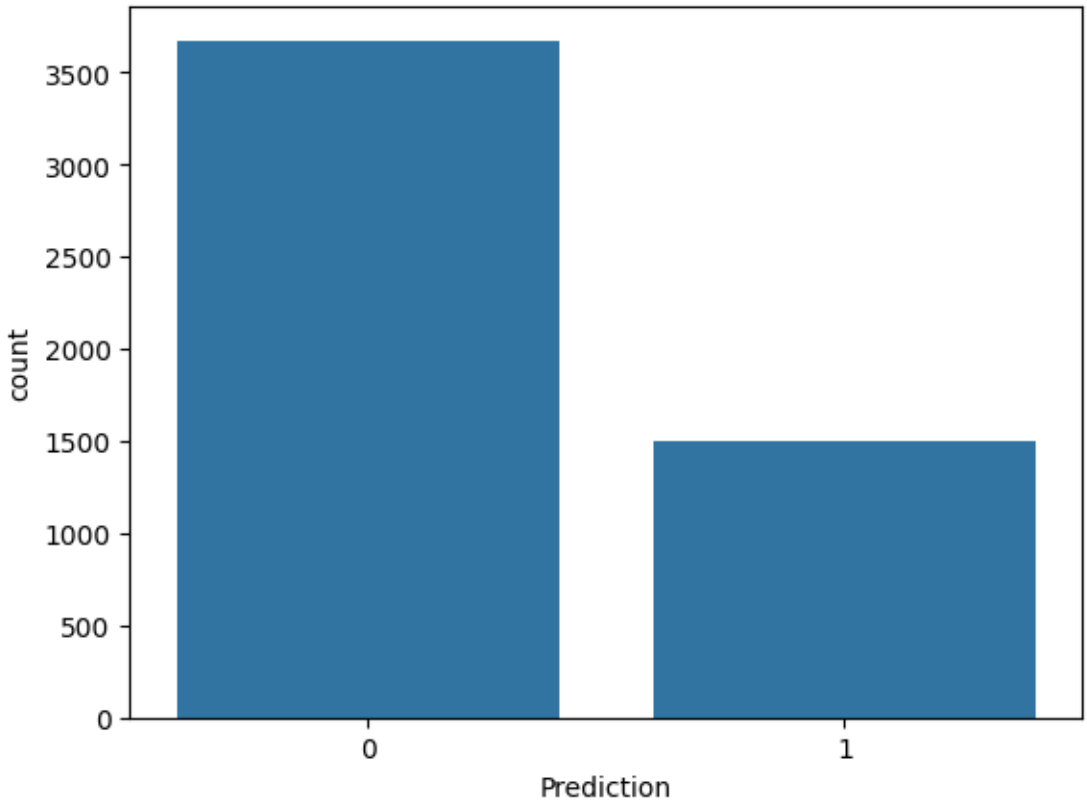
[15]:
```
Y=df["Prediction"]
```

[16]:
```
Y
```

[16]:
```
0       0
1       0
2       0
3       0
4       0
       ..
5167    0
5168    0
5169    1
5170    1
5171    0
Name: Prediction, Length: 5172, dtype: int64
```

[17]:
```
sns.countplot(x=Y)
```

[17]:
```
<Axes: xlabel='Prediction', ylabel='count'>
```



[18]:
```
Y.value_counts
```

[18]:
```
<bound method IndexOpsMixin.value_counts of 0       0
1       0
2       0
3       0
4       0
       ..
5167    0
5168    0
5169    1
5170    1
5171    0
Name: Prediction, Length: 5172, dtype: int64>
```

3 Feature Scaling

[19]:
```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
X_Scale=scaler.fit_transform(X)
```

[20]:
```
X_Scale
```

[20]:
```
array([[0.        , 0.        , 0.        , …, 0.        , 0.        ,
        0.        ],
       [0.03809524, 0.09848485, 0.06705539, …, 0.        , 0.00877193,
        0.        ],
       [0.        , 0.        , 0.        , …, 0.        , 0.        ,
        0.        ],
       …,
       [0.        , 0.        , 0.        , …, 0.        , 0.        ,
        0.        ],
       [0.00952381, 0.0530303 , 0.        , …, 0.        , 0.00877193,
        0.        ],
       [0.1047619 , 0.18181818, 0.01166181, …, 0.        , 0.        ,
        0.        ]])
```

3.0.1 Cross Validation

[21]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_Test=train_test_split(X_Scale,Y,test_size=0.
↪47,random_state47)
X_train.shape
```

[21]:
```
(3879, 3000)
```

[22]:
```
X_Scale.shape
```

[22]:
```
(5172, 3000)
```

[23]:
```
X_test.shape
```

[23]:
```
(1293, 3000)
```

[24]:
```
Y_train.shape
```

[24]:
```
(3879,)
```

4 calling the models

[25]:
```
from sklearn.neighbors import KNeighborsClassifier
knn= KNeighborsClassifier()
knn.fit(X_train,Y_train)
```

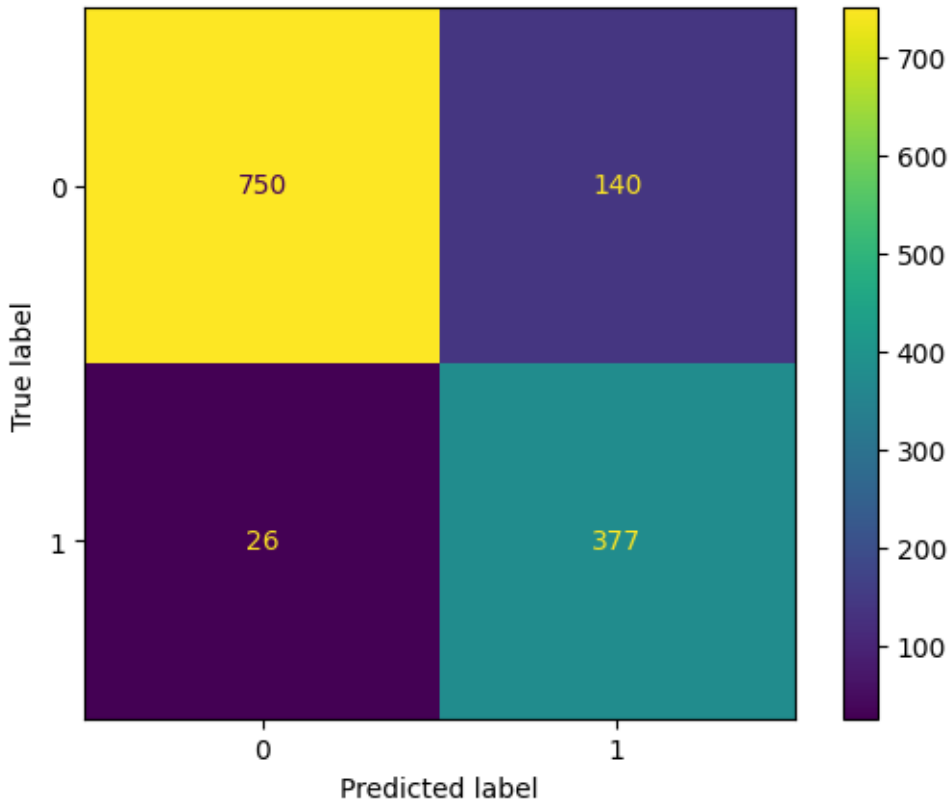
[25]:
```
KNeighborsClassifier()
```

[26]:
```
Y_pred=knn.predict(X_test)
Y_pred
```

[26]:
```
array([0, 0, 1, …, 0, 1, 0])
```

[27]:
```
from sklearn.metrics import
  ↪accuracy_score,classification_report,ConfusionMatrixDisplay
ConfusionMatrixDisplay.from_predictions(Y_Test,Y_pred)
```

[27]:
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at
0x7fed8c985660>
```



accuracy_score(Y_train,Y_Test)

[28]:
```
accuracy_score(Y_Test,Y_pred)
```

[28]:
```
0.871616395978345
```

[29]:
```
print(classification_report(Y_Test,Y_pred))
```

```
              precision    recall  f1-score   support

           0       0.97      0.84      0.90       890
           1       0.73      0.94      0.82       403

    accuracy                           0.87      1293
   macro avg       0.85      0.89      0.86      1293
weighted avg       0.89      0.87      0.88      1293
```

[ ]:
```

```