Lab_2

October 31, 2023

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: Yash Ghorpade

Div: BE-A

Roll No: B211046

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

[3]:
```
df=pd.read_csv("emails.csv")
df
```

[3]:

	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]
```

[4]:
```
df.shape
```

[4]: (5172, 3002)

[5]:
```
df.size
```

[5]: 15526344

[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.describe
```

[8] :
```
<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]>
```

[9]:
```
df.dtypes
```

[9] :
```
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
```

[10]:
```
df
```

[10] :

	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]
```

[12]:
```
X=df.drop(["Email No.", "Prediction"],axis=1)
```

[13]:
```
X
```

[13]:
```
      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]
```

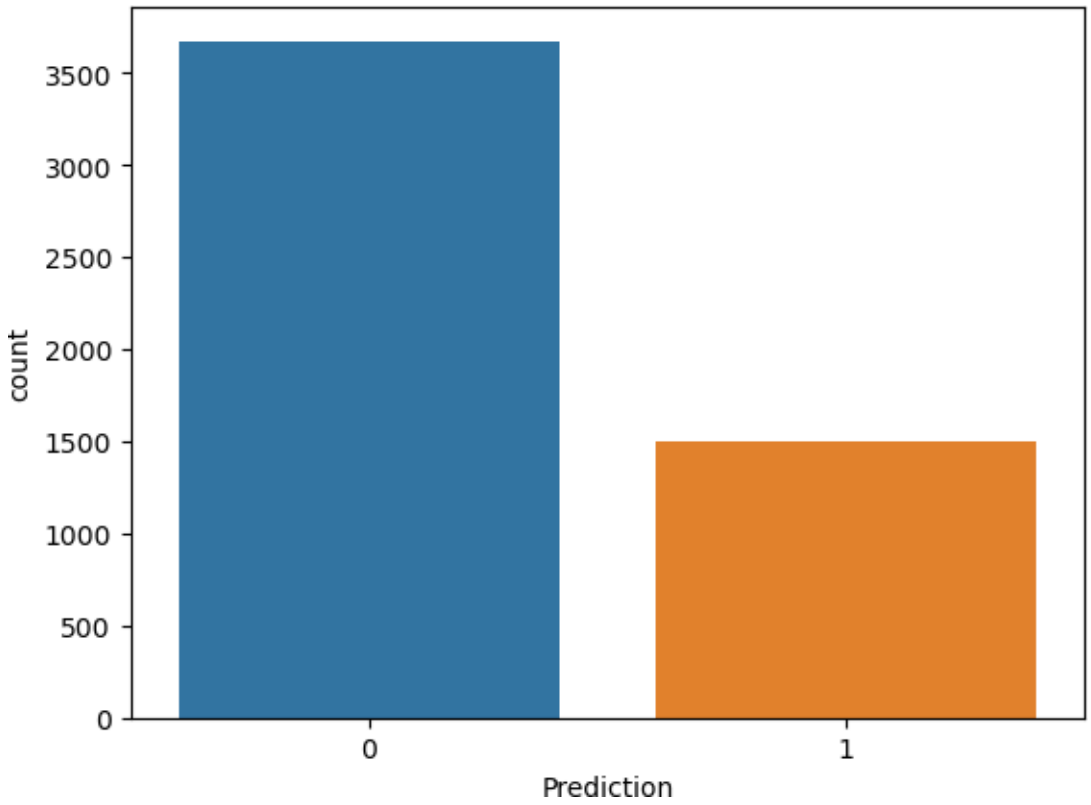
[14]:
```
Y=df["Prediction"]
```

[15]:
```
Y
```

[15]:
```
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
```

[16]:
```
sns.countplot(x=Y)
```

[16]:
```
<Axes: xlabel='Prediction', ylabel='count'>
```



[17]:
```
Y.value_counts
```

[17]:
```
<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>
```

[18]:
```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()
X_Scale=scaler.fit_transform(X)
```

[19]:
```
X_Scale
```

[19]:
```
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.        ]])
```

[20]:
```
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.
↪46,random_state = 46)
X_train.shape
```

[20]: (2792, 3000)

[21]:
```
X_Scale.shape
```

[21]: (5172, 3000)

[22]:
```
X_test.shape
```

[22]: (2380, 3000)

[23]:
```
Y_train.shape
```

[23]: (2792,)

[24]:
```
from sklearn.neighbors import KNeighborsClassifier
knn=  KNeighborsClassifier()
knn.fit(X_train,Y_train)
```

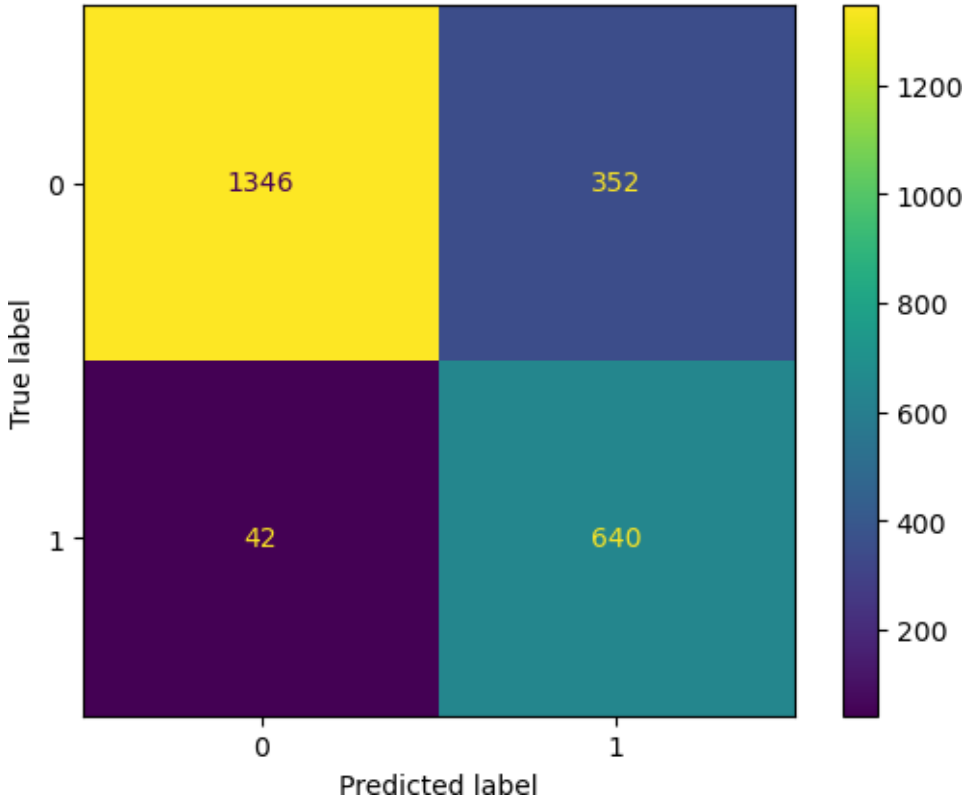
[24]: KNeighborsClassifier()

[25]:
```
Y_pred=knn.predict(X_test)
Y_pred
```

[25]: array([0, 0, 1, ..., 0, 1, 0], dtype=int64)

[26]:
```
 from sklearn.metrics import␣
 ↪accuracy_score,classification_report,ConfusionMatrixDisplay
ConfusionMatrixDisplay.from_predictions(Y_Test,Y_pred)
```

[26]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x208d3e813c0>



[27]:
```
accuracy_score(Y_Test,Y_pred)
```

[27]: 0.8344537815126051

[28]:
```
print(classification_report(Y_Test,Y_pred))
```

	precision	recall	f1-score	support
0	0.97	0.79	0.87	1698
1	0.65	0.94	0.76	682
accuracy			0.83	2380
macro avg	0.81	0.87	0.82	2380
weighted avg	0.88	0.83	0.84	2380

[ ]: