```
import pandas as pd

data=pd.read_csv("Downloads/emails.csv")

data
```

```
       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]
```

```
data.head()
```

```
  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]
```

```
from sklearn.model_selection import train_test_split

X=data.drop('Email No.',axis=1)
y=data['Prediction']
```

```
X
```

```
      the  to  ect  and  for  of    a  you  hou  in  ...  connevey
jay  \
0       0   0    1    0    0   0    2    0    0   0  ...          0
0
1       8  13   24    6    6   2  102    1   27  18  ...          0
```

```
0
2       0   0    1    0    0   0    8    0    0   4  ...          0
0
3       0   5   22    0    5   1   51    2   10   1  ...          0
0
4       7   6   17    1    5   2   57    0    9   3  ...          0
0
...   ...  ..  ...  ...  ...  ..  ...  ...  ...  ..  ...        ...  ..
.
5167    2   2    2    3    0   0   32    0    0   5  ...          0
0
5168   35  27   11    2    6   5  151    4    3  23  ...          0
0
5169    0   0    1    1    0   0   11    0    0   1  ...          0
0
5170    2   7    1    0    2   1   28    2    0   8  ...          0
0
5171   22  24    5    1    6   5  148    8    2  23  ...          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
...      ...  ...             ...       ...       ...  ..  ...
...
5167       0    0               0         0         0   0    0
0
5168       0    0               0         0         0   1    0
0
5169       0    0               0         0         0   0    0
1
5170       0    0               0         0         0   1    0
1
5171       0    0               0         0         0   0    0
0

[5172 rows x 3001 columns]


y
```

```
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
```

```
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size =
0.25, random_state = 0)
```

```
x_train
```

```
      the  to  ect  and  for  of    a  you  hou  in  ...  connevey
jay  \
2382   32  25   14   16   15   2  193    7    4  41  ...         0
0
3848   42  19    1    8    7   7  213    1    3  51  ...         0
0
667     1   3    1    0    1   0   10    1    0   4  ...         0
0
4641   14  11    1    5    4   6   72    8    3  17  ...         0
0
3650    6   5    1    3    2   3   38    3    0   6  ...         0
0
...   ...  ..  ...  ...  ...  ..  ...  ...  ...  ..  ...       ...  ..
.
4931   17  26    1    5    4   5  100   18    1  17  ...         0
0
3264    0   3    1    1    1   0    9    2    0   5  ...         0
0
1653   12  18    8   11   12   5  146    9    2  29  ...         0
0
2607   10   6   96    8    5   2  123    0   49  22  ...         0
0
2732    1   8    2    1    1   0   71    5    0  16  ...         0
0
```

```
      valued  lay  infrastructure  military  allowing  ff  dry
Prediction
2382       0    0               0         0         0   1    0
0
3848       0    1               0         4         0   3    0
1
667        0    0               0         0         0   0    0
```

```
1
4641       0    0                0          0          0   3    0
0
3650       0    0                0          0          0   1    0
1
...      ...  ...              ...        ...        ...  ..  ...
...
4931       0    0                0          0          0   0    0
0
3264       0    0                0          0          0   0    0
1
1653       0    0                0          0          0   2    0
1
2607       0    0                0          0          0   1    0
0
2732       0    0                0          0          0   0    0
1

[3879 rows x 3001 columns]
```

y_train

```
2382    0
3848    1
667     1
4641    0
3650    1
       ..
4931    0
3264    1
1653    1
2607    0
2732    1
Name: Prediction, Length: 3879, dtype: int64
```

x_test

```
      the  to  ect  and  for  of    a  you  hou  in  ...  connevey
jay  \
3324   14   7    2    2    2   4   37    0    1   9  ...          0
0
15      6   2    1    0    2   0   36    3    1   8  ...          0
0
4950   40   6    4    6    5  11  130    3    0  31  ...          0
0
3964   42  19    1    8    7   7  214    1    3  52  ...          0
0
2315    8  14    8    3    3   0   87    0    4  10  ...          0
0
...   ...  ..  ...  ...  ...  ..  ...  ...  ...  ..  ...        ...  ..
```

```
.
3551    0   0    1    0    1   0    3    0    0   0  ...          0
0
1118    2   3    4    3    2   0   41    2    0   4  ...          0
0
2239    3   1    1    1    2   0   22    2    0   0  ...          0
0
1511    8   3   12    4    2   1   47    0    7   9  ...          0
0
1240    1   6    2    4    6   3   53    9    4   6  ...          0
0

      valued  lay  infrastructure  military  allowing  ff  dry
Prediction
3324       0    0               0         0         0   0    0
0
15         0    0               0         0         0   0    0
0
4950       0    0               0         0         0   1    0
0
3964       0    1               0         4         0   3    0
1
2315       0    0               0         0         0   5    0
0
...      ...  ...             ...       ...       ...  ..  ...
...
3551       0    0               0         0         0   0    0
0
1118       0    0               0         0         0   0    0
1
2239       0    0               0         0         0   0    0
0
1511       0    0               0         0         0   0    0
0
1240       0    0               0         0         0   2    0
1

[1293 rows x 3001 columns]

y

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
```

```
y_test
```

```
3324    0
15      0
4950    0
3964    1
2315    0
       ..
3551    0
1118    1
2239    0
1511    0
1240    1
Name: Prediction, Length: 1293, dtype: int64
```

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import
confusion_matrix,accuracy_score,precision_score, recall_score
```

```
knn=KNeighborsClassifier(n_neighbors=10)
```

```
knn.fit(x_train,y_train)
```

```
KNeighborsClassifier(n_neighbors=10)
```

```
y_pred=knn.predict(x_test)
```

```
y_pred
```

```
array([0, 0, 0, ..., 0, 0, 1])
```

```
from sklearn import metrics
acc=metrics.accuracy_score(y_pred,y_test)
```

```
acc
```

```
0.8646558391337974
```

```
err=(1-acc)
err
```

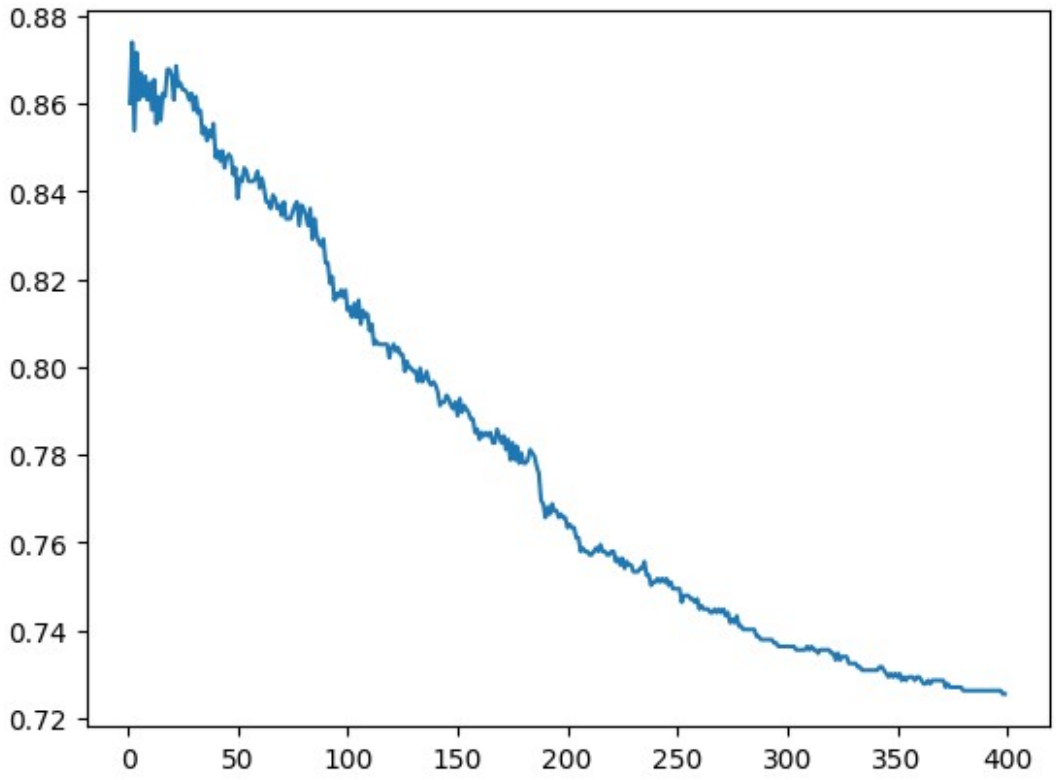
```
0.1353441608662026
```

```
def knn(x_train,y_train,x_test,y_test,n):
    n_range=range(1,n)
    results=[]
    for n in n_range:
        knn=KNeighborsClassifier(n_neighbors=n)
        knn.fit(x_train,y_train)
```

```
        y_pred=knn.predict(x_test)
        acc=metrics.accuracy_score(y_pred,y_test)
        results.append(acc)
    return results

import matplotlib.pyplot as plt
n=400
output=knn(x_train,y_train,x_test,y_test,n)
n_range=range(1,n)
plt.plot(n_range,output)
```

```
[<matplotlib.lines.Line2D at 0x70ea875b5570>]
```




```
#email classification using SVC

from sklearn.svm import SVC
from sklearn.svm import LinearSVC
from sklearn.metrics import accuracy_score
import math
import time

start=time.time()
model=SVC(kernel='poly',C=2)
model.fit(x_train,y_train)
pred=model.predict(x_test)
```

```
acc=accuracy_score(y_test,pred)
print(round(acc*100,1),'%')
end=time.time()

print(f"{end-start:.5f}sec")
```

```
75.6 %
21.91650sec
```

```
start=time.time()
model=LinearSVC(C=3)
model.fit(x_train,y_train)
pred=model.predict(x_test)
acc=accuracy_score(y_test,pred)
print(round(acc*100,1),'%')
end=time.time()

print(f"{end-start:.5f}sec")
```

```
/home/student/.local/lib/python3.10/site-packages/sklearn/svm/
_classes.py:31: FutureWarning: The default value of `dual` will change
from `True` to `'auto'` in 1.5. Set the value of `dual` explicitly to
suppress the warning.
  warnings.warn(

99.6 %
1.67155sec

/home/student/.local/lib/python3.10/site-packages/sklearn/svm/
_base.py:1237: ConvergenceWarning: Liblinear failed to converge,
increase the number of iterations.
  warnings.warn(
```

```
start=time.time()
model=SVC(kernel='sigmoid',C=2)
model.fit(x_train,y_train)
pred=model.predict(x_test)
acc=accuracy_score(y_test,pred)
print(round(acc*100,1),'%')
end=time.time()

print(f"{end-start:.5f}sec")
```

```
59.6 %
16.24308sec
```

```
start=time.time()
model=SVC(kernel='rbf',C=2)
model.fit(x_train,y_train)
pred=model.predict(x_test)
acc=accuracy_score(y_test,pred)
```

```
print(round(acc*100,1),'%')
end=time.time()

print(f"{end-start:.5f}sec")
```

```
83.7 %
22.57503sec
```