assignment-2

August 19, 2024

[2]:
```
import numpy as np
import pandas as pd
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.isna().sum()
```

```
[5]: Email No.     0
     the           0
     to            0
     ect           0
     and           0
                  ..
     military      0
     allowing      0
     ff            0
     dry           0
     Prediction    0
     Length: 3002, dtype: int64
```

```
[6]: df.drop(columns='Email No.', inplace=True)
```

```
[7]: df['Prediction'] = df['Prediction'].replace({0:'Not spam', 1:'Spam'})
     df
```

```
[7]:       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    Not spam
1          0    0               0         0         0   1    0    Not spam
2          0    0               0         0         0   0    0    Not spam
3          0    0               0         0         0   0    0    Not spam
4          0    0               0         0         0   1    0    Not spam
…        …  …             …         …         …  ..  …          …
5167       0    0               0         0         0   0    0    Not spam
5168       0    0               0         0         0   1    0    Not spam
5169       0    0               0         0         0   0    0        Spam
5170       0    0               0         0         0   1    0        Spam
5171       0    0               0         0         0   0    0    Not spam

[5172 rows x 3001 columns]
```

[8]:
```
X = df.drop(columns='Prediction',axis = 1)
Y = df['Prediction']
```

[9]:
```
X.columns
```

[9]:
```
Index(['the', 'to', 'ect', 'and', 'for', 'of', 'a', 'you', 'hou', 'in',
       …
       'enhancements', 'connevey', 'jay', 'valued', 'lay', 'infrastructure',
       'military', 'allowing', 'ff', 'dry'],
      dtype='object', length=3000)
```

[10]:
```
Y.head()
```

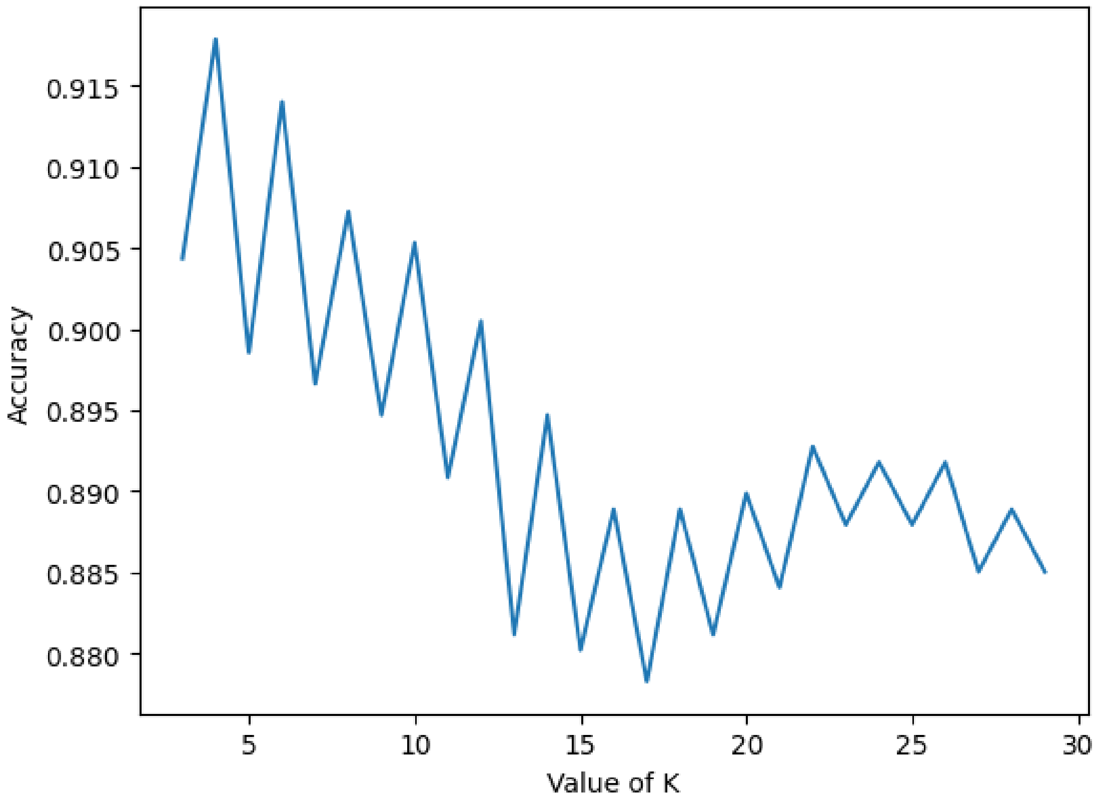
[10]:
```
0    Not spam
1    Not spam
2    Not spam
3    Not spam
4    Not spam
Name: Prediction, dtype: object
```

[11]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.2,␣
↪random_state=1)
```

[14]:
```
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics
KN = KNeighborsClassifier

scores = {}
scores_list = []
max_stats={'n':3,'M':0.00}
```

```
def calcKNN(n):
    knn = KN(n_neighbors=n,p=1)
    knn.fit(x_train, y_train)
    y_pred = knn.predict(x_test)
    M = metrics.accuracy_score(y_test,y_pred)
    M=M*100
    scores[n] = metrics.accuracy_score(y_test,y_pred)
    scores_list.append(metrics.accuracy_score(y_test,y_pred))
    if M>max_stats['M']:
        max_stats['M']=M
        max_stats['n']=n

for i in range(3,30):
    calcKNN(i)

plt.plot(range(3,30),scores_list)
plt.xlabel("Value of K")
plt.ylabel("Accuracy")
print(max_stats)
```

```
{'n': 4, 'M': 91.78743961352657}
```



[15]:
```
knn_model = KNeighborsClassifier(n_neighbors=4)
knn_model.fit(x_train, y_train)
y_pred = knn_model.predict(x_test)
print(metrics.classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

    Not spam       0.90      0.91      0.91       719
        Spam       0.80      0.77      0.78       316

    accuracy                           0.87      1035
   macro avg       0.85      0.84      0.84      1035
weighted avg       0.87      0.87      0.87      1035
```

[16]:
```
from sklearn.svm import SVC
svm_model =SVC()
svm_model.fit(x_train, y_train)
```

[16]:
```
SVC()
```

[17]:
```
y_pred = svm_model.predict(x_test)
print(metrics.classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

    Not spam       0.79      0.97      0.87       719
        Spam       0.87      0.40      0.55       316

    accuracy                           0.80      1035
   macro avg       0.83      0.69      0.71      1035
weighted avg       0.81      0.80      0.77      1035
```