*#Anuprita Girme #COBA56*

0

Prediction

0

import pandas as pd

from sklearn.model\_selection import train\_test\_split from sklearn.neighbors import KNeighborsClassifier from sklearn.svm import SVC

from sklearn.metrics import accuracy\_score, classification\_report data = pd.read\_csv("emails.csv")

data

Email No. connevey \

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| the | to | ect | and | for | of | a | you | hou | ... |
| 0 | 0 | 1 | 0 | 0 | 0 | 2 | 0 | 0 | ... |
| 8 | 13 | 24 | 6 | 6 | 2 | 102 | 1 | 27 | ... |
| 0 | 0 | 1 | 0 | 0 | 0 | 8 | 0 | 0 | ... |
| 0 | 5 | 22 | 0 | 5 | 1 | 51 | 2 | 10 | ... |
| 7 | 6 | 17 | 1 | 5 | 2 | 57 | 0 | 9 | ... |
| ... | .. | ... | ... | ... | .. | ... | ... | ... | ... |
| 2 | 2 | 2 | 3 | 0 | 0 | 32 | 0 | 0 | ... |
| 35 | 27 | 11 | 2 | 6 | 5 | 151 | 4 | 3 | ... |
| 0 | 0 | 1 | 1 | 0 | 0 | 11 | 0 | 0 | ... |
| 2 | 7 | 1 | 0 | 2 | 1 | 28 | 2 | 0 | ... |
| 22 | 24 | 5 | 1 | 6 | 5 | 148 | 8 | 2 | ... |

1. Email 1

0

1. Email 2

0

1. Email 3

0

1. Email 4

0

1. Email 5

0

... ...

...

5167 Email 5168

0

5168 Email 5169

0

5169 Email 5170

0

5170 Email 5171

0

5171 Email 5172

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

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 1 | 0 |
| 2 | 0 |
| 3 | 0 |
| 4 | 0 |
| ... | ... |
| 5167 | 0 |
| 5168 | 0 |
| 5169 | 1 |
| 5170 | 1 |
|  | 5171 | 0 |

[5172 rows x 3002 columns]

data.drop(['Email No.'],axis=1, inplace=True)

X = data.drop("Prediction", axis=1) y = data["Prediction"] print("Features: ",X) print("Target: ",y)

Features: 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

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ff | connevey dry | jay | valued | lay | infrastructure | military | allowing |
| 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] Target: 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=0.3, random\_state=42)

knn\_model = KNeighborsClassifier(n\_neighbors=5) knn\_model.fit(X\_train, y\_train)

svm\_model = SVC() svm\_model.fit(X\_train, y\_train)

SVC()

knn\_predictions = knn\_model.predict(X\_test) knn\_accuracy = accuracy\_score(y\_test, knn\_predictions)

knn\_report = classification\_report(y\_test, knn\_predictions)

C:\Users\rohit\anaconda3\lib\site-packages\sklearn\neighbors\

\_classification.py:228: FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode`

typically preserves the axis it acts along. In SciPy 1.11.0, this behavior will change: the default value of `keepdims` will become False, the `axis` over which the statistic is taken will be eliminated, and the value None will no longer be accepted. Set

`keepdims` to True or False to avoid this warning.

mode, \_ = stats.mode(\_y[neigh\_ind, k], axis=1) print(knn\_predictions)

[0 0 1 ... 0 0 0]

print("K-Nearest Neighbors Accuracy:") print(knn\_accuracy)

print("K-Nearest Neighbors Classification Report:") print(knn\_report)

K-Nearest Neighbors Accuracy: 0.8608247422680413

K-Nearest Neighbors Classification Report:

precision recall f1-score support

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 0.93 | 0.87 | 0.90 | 1097 |
| 1 | 0.73 | 0.83 | 0.78 | 455 |
|  |  |  |  |  |
| accuracy |  |  | 0.86 | 1552 |
| macro avg | 0.83 | 0.85 | 0.84 | 1552 |
| weighted avg | 0.87 | 0.86 | 0.86 | 1552 |

svm\_predictions = svm\_model.predict(X\_test) svm\_accuracy = accuracy\_score(y\_test, svm\_predictions)

svm\_report = classification\_report(y\_test, svm\_predictions) print(svm\_predictions)

[0 0 1 ... 0 0 0]

print("Support Vector Machine Accuracy:") print(svm\_accuracy)

print("Support Vector Machine Classification Report:") print(svm\_report)

Support Vector Machine Accuracy: 0.803479381443299

Support Vector Machine Classification Report:

precision recall f1-score support

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 0.79 | 0.99 | 0.88 | 1097 |
| 1 | 0.92 | 0.36 | 0.52 | 455 |
|  |  |  |  |  |
| accuracy |  |  | 0.80 | 1552 |
| macro avg | 0.85 | 0.67 | 0.70 | 1552 |

weighted avg 0.83 0.80 0.77 1552