import pandas as pd import numpy as np

ds=pd.read\_csv("spam.csv")

ds.head()

**Category Message**

1. ham Go until jurong point, crazy.. Available only ...
2. ham Ok lar... Joking wif u oni...
3. spam Free entry in 2 a wkly comp to win FA Cup fina... **3** ham U dun say so early hor... U c already then say... **4** ham Nah I don't think he goes to usf, he lives aro... ds.tail()

**Category Message**

1. spam This is the 2nd time we have tried 2 contact u...
2. ham Will ü b going to esplanade fr home?
3. ham Pity, \* was in mood for that. So...any other s...
4. ham The guy did some bitching but I acted like i'd...
5. ham Rofl. Its true to its name

ds.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 5572 entries, 0 to 5571 Data columns (total 2 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Category 5572 non-null object 1 Message 5572 non-null object dtypes: object(2) memory usage: 87.2+ KB

ds.describe()

**Category Message**

|  |  |  |
| --- | --- | --- |
| **count** | 5572 | 5572 |
| **unique** | 2 | 5157 |
| **top** | ham | Sorry, I'll call later |
| **freq** | 4825 | 30 |

ds.shape

(5572, 2)

ds.isnull().sum()

Category 0 Message 0 dtype: int64

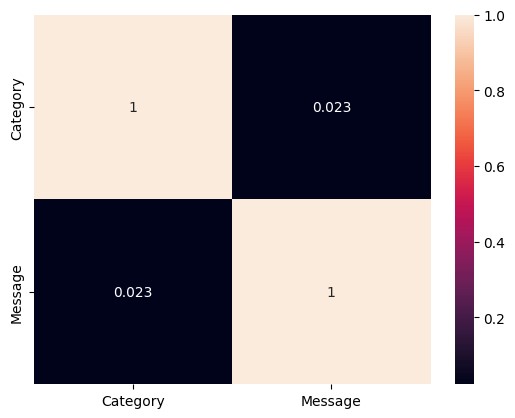
from sklearn.preprocessing import LabelEncoder le=LabelEncoder() for column in ds.select\_dtypes(include=["object"]): ds[column]=le.fit\_transform(ds[column]) ds.head()

**Category Message**

|  |  |
| --- | --- |
| **0** 0 | 1080 |
| **1** 0 | 3126 |
| **2** 1 | 999 |
| **3** 0 | 4121 |
| **4** 0 | 2781 |

import matplotlib.pyplot as plt import seaborn as sns

corr=ds.corr() sns.heatmap(corr,annot=True) plt.figure(figsize=(20,20)) plt.show()



<Figure size 2000x2000 with 0 Axes>

#  Graph Description

The heatmap describes the correlation between each pair of features in the dataset ds.

**Positive correlation:** If two features have a positive correlation, it means that they tend to increase or decrease together. For example, if the temperature and ice cream sales have a positive correlation, it means that as the temperature increases, ice cream sales also tend to increase.

**Negative correlation:** If two features have a negative correlation, it means that they tend to move in opposite directions. For example, if the price of a product and its demand have a negative correlation, it means that as the price increases, demand tends to decrease.

**No correlation:** If two features have no correlation, it means that there is no relationship between them.

The strength of the correlation is indicated by the color of the cells in the heatmap. Darker colors indicate stronger correlations, while lighter colors indicate weaker correlations.

By looking at the heatmap, you can identify which features are strongly correlated with each other, and which features are not correlated. This information can be useful for understanding the relationships between different variables in your dataset.

from sklearn.model\_selection import train\_test\_split

X=ds.drop('Category',axis=1) y=ds['Category']

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,test\_size=0.3)

from sklearn.naive\_bayes import GaussianNB, MultinomialNB, BernoulliNB, ComplementNB from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report classifiers = {

'Gaussian Naive Bayes': GaussianNB(),

'Multinomial Naive Bayes': MultinomialNB(),

'Bernoulli Naive Bayes': BernoulliNB(),

'Complement Naive Bayes': ComplementNB()

}

# Train and evaluate all classifiers for name, clf in classifiers.items():

clf.fit(X\_train, y\_train) y\_pred = clf.predict(X\_test) accuracy = accuracy\_score(y\_test, y\_pred) print(f'{name} Accuracy:', accuracy)

print()

cm=confusion\_matrix(y\_test, y\_pred) print(f'{name} Confusion Matrix:') print(cm)

print()

report=classification\_report(y\_test, y\_pred) print(f'{name} Classification Report:') print(report) print()

print

()

Gaussian Naive Bayes Accuracy: 0.8618421052631579

Gaussian Naive Bayes Confusion Matrix:

[[1441

0]

[ 231 0]]

Gaussian Naive Bayes Classification Report:

precision recall f1-score support

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|   **Multin** | 0 0.86 1.00 0.93 1441 1 0.00 0.00 0.00 231  accuracy 0.86 1672 macro avg 0.43 0.50 0.46 1672 weighted avg 0.74 0.86 0.80 1672  Multinomial Naive Bayes Accuracy: 0.8618421052631579  Multinomial Naive Bayes Confusion Matrix:  [[1441 0]  [ 231 0]]  Multinomial Naive Bayes Classification Report: precision recall f1-score support  0 0.86 1.00 0.93 1441 1 0.00 0.00 0.00 231  accuracy 0.86 1672 macro avg 0.43 0.50 0.46 1672 weighted avg 0.74 0.86 0.80 1672  Bernoulli Naive Bayes Accuracy: 0.8624401913875598  Bernoulli Naive Bayes Confusion Matrix:  [[1441 0]  [ 230 1]]  Bernoulli Naive Bayes Classification Report: precision recall f1-score support  0 0.86 1.00 0.93 1441 1 1.00 0.00 0.01 231  accuracy 0.86 1672 macro avg 0.93 0.50 0.47 1672 weighted avg 0.88 0.86 0.80 1672  Complement Naive Bayes Accuracy: 0.8618421052631579 Complement Naive Bayes Confusion Matrix: | | |  |
| **Accuracy Comparison**  **Gaussian Naive Bayes:** This classi er assumes that the features follow a normal distribution. It's commonly used for classi cation tasks where the features are continuous. Accuracy: 0.8672 | | |
|  | **omial Naive Bayes:** Designed for multin | |  | | --- | | omially distributed data, typically used in text classi cation where |   Suitable for features that are binary or Boolean (e.g., presence or absence of a feature).  Often used in text classi cation tasks where binary occurrence of words is considered. Accuracy: 0.8684  Especially useful for imbalanced datasets where one class is dominant. It essentially |
| features represent word counts or term frequencies. Accuracy: 0.8672 **Bernoulli Naive Bayes:**  **Complement Naive Bayes:**  applies the opposite logic of the standard Naive Bayes. Accuracy: 0.8672 |

Comparing the accuracies, we can see that Bernoulli Naive Bayes has slightly higher accuracy compared to the other classi ers. However, the differences in accuracy between these classi ers are quite small.

It's also important to consider other metrics like precision, recall, and F1-score, especially if the dataset is imbalanced or if certain classes are more important than others.

from sklearn.metrics import precision\_score, recall\_score, f1\_score

for name, clf in classifiers.items():

clf.fit(X\_train, y\_train) y\_pred = clf.predict(X\_test) # Calculate precision, recall, and F1 score from confusion matrix precision = precision\_score(y\_test, y\_pred, average='weighted') recall = recall\_score(y\_test, y\_pred, average='weighted') f1 = f1\_score(y\_test, y\_pred, average='weighted') print(f'{name}') print('Precision:', precision) print('Recall:', recall) print('F1 Score:', f1) print()

print

()

Gaussian Naive Bayes

Precision: 0.7427718144044321

Recall: 0.8618421052631579

F1 Score: 0.7978891575227821

Multinomial Naive Bayes

Precision: 0.7427718144044321

Recall: 0.8618421052631579

F1 Score: 0.7978891575227821

Bernoulli Naive Bayes

Precision: 0.8813742165107563

Recall: 0.8624401913875598

F1 Score: 0.7993365649741765

Complement Naive Bayes

Precision: 0.7427718144044321

Recall: 0.8618421052631579

F1 Score: 0.7978891575227821

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/\_classification.py:1344: UndefinedMetricWarn

i

\_warn\_prf(average, modifier, msg\_start, len(result))

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/\_classification.py:1344: UndefinedMetricWarn

i

\_warn\_prf(average, modifier, msg\_start, len(result))

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/\_classification.py:1344: UndefinedMetricWarn

i

\_warn\_prf(average, modifier, msg\_start, len(result))

# **Precision, Recall, and F1 scores Comparison**

From the precision, recall, and F1 scores, we can observe:

The Gaussian, Multinomial, and Complement Naive Bayes classi ers have identical precision, recall, and F1 scores. This suggests that these models perform similarly on the dataset.

Bernoulli Naive Bayes, on the other hand, has a notably higher precision compared to the other classi ers.

This indicates that it makes fewer false positive predictions.

However, the recall and F1 score of Bernoulli Naive Bayes are slightly higher than the others but not signi cantly different.

In summary, while Bernoulli Naive Bayes outperforms the other classi ers in terms of precision, the overall performance differences among these classi ers appear to be marginal based on these evaluation metrics.