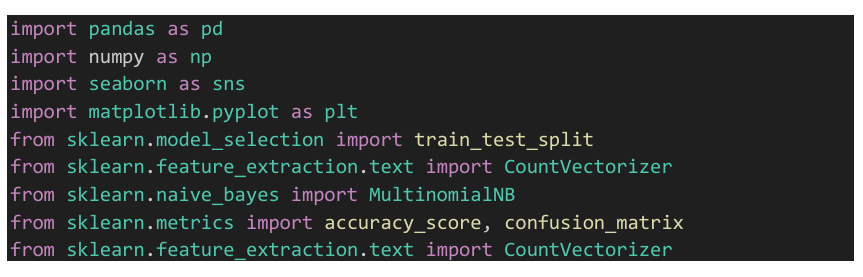
Text Classification using Naive Bayes

# 1. Importing Libraries

This section imports all necessary libraries for data handling, preprocessing, model training, and evaluation.



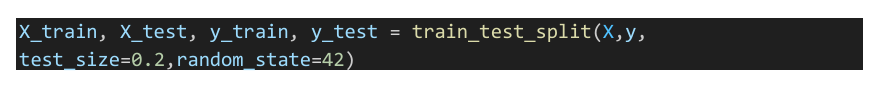
# 2. Load the Dataset

The dataset 'synthetic\_text\_data.csv' is loaded using pandas. It should contain two columns: 'text' and 'label'.



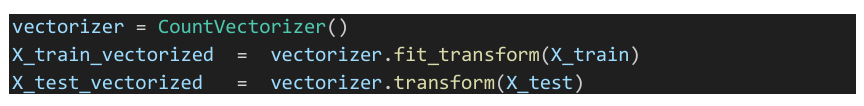
# 3. Split the Dataset

The dataset is split into training and testing sets using an 80-20 split. This is done to evaluate the model on unseen data.



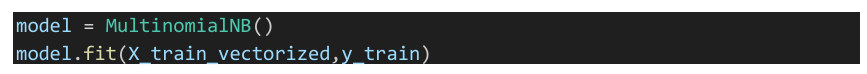
# 4. Vectorize the Text Data

Text data is converted into numerical vectors using CountVectorizer. The training set is used to fit and transform the data, while the test set is only transformed.



# 5. Train the Naive Bayes Model

A Multinomial Naive Bayes model is used to train on the vectorized training data. This algorithm is suitable for text classification tasks.



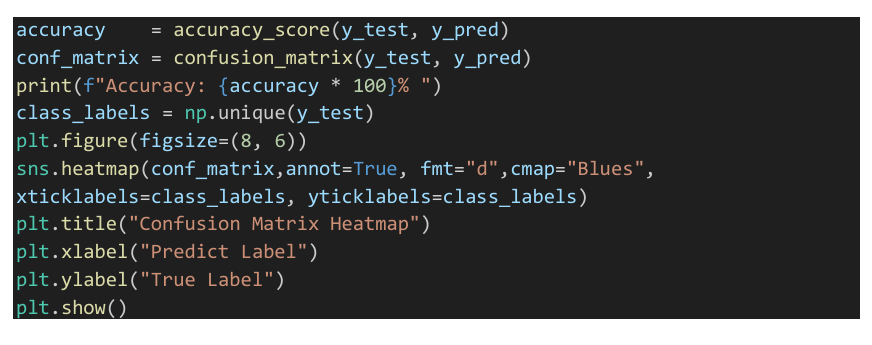
# 6. Make Predictions and Evaluate

The model predicts the labels of the test set. Accuracy and a confusion matrix are generated to evaluate the model's performance.



# 7. Visualize the Confusion Matrix

A heatmap of the confusion matrix is plotted using seaborn and matplotlib to better understand the model's classification performance.



# 8. Summary:

This code performs text classification using the Multinomial Naive Bayes algorithm. It reads a CSV file containing text and labels, vectorizes the text using CountVectorizer, and trains the model on the training data. The model's performance is evaluated using accuracy and a confusion matrix, which is visualized using a heatmap.

