Project 2: Naive Bayes algorithm for learning to classify text

Steps followed in Implementation:

Data Loading: The script loads text data from a directory structure, where each subdirectory represents a category and contains documents belonging to that category.

Preprocessing: It filters out common English stopwords, punctuation, and some specific terms that are likely to be irrelevant for classification purposes.
Texts are then processed to build a word frequency bank, ignoring the filtered terms and focusing on words with a length of at least 5 characters.

Feature Selection: From the word frequency bank, the top 2000 words are selected as features for the classification model.

Data Preparation: A label array (Y) is created, mapping documents to their categories.
A feature matrix (X) is constructed where each row represents a document, and each column represents one of the top 2000 words. The values are the frequencies of the corresponding word in the document.

Model Training and Evaluation: The dataset is split into training and test sets.
A Multinomial Naive Bayes model is trained on the training set.
The model's accuracy is evaluated on the test set, followed by generating a confusion matrix and a classification report for a more detailed performance analysis.

Project 2: Naive Bayes algorithm for learning to classify text

Output:

```
Model accuracy on test set: 0.8394839483948395
Confusion Matrix:
 [[  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    1   0   0]
 [  0 405   0   0   0   2   1   0   3   2   0   0   2   1   2   0   9   2
    6   5  43]
 [  0   3 401  24  12  16  13   5   3   7   2   1   3   5   2   3   0   1
    1   0   1]
 [  0   0  18 430  22   3  11   2   1   0   0   1   1   7   3   2   1   0
    0   0   1]
 [  0   1   6  18 407  15   2   8   3   1   0   0   1  11   0   1   0   0
    0   0   0]
 [  0   0   5  11  18 414   3  11   2   2   0   0   1   9   0   0   1   1
    0   0   0]
 [  0   1  21  18   5   5 442   4   2   1   2   3   1   2   1   1   0   0
    0   0   0]
 [  0   0  15   6   9   7   2 417  10   9   0   3   3   8   3   5   0   2
    0   2   0]
 [  0   1   4   1   2   5   1   9 453  18   0   0   2   8   0   5   1  10
    3   6   0]
 [  0   1   1   0   0   2   2  12  10 455   1   1   2   6   1   2   1   1
    0   2   0]
 [  0   2   0   0   2   0   1   3   1   2 484   6   0   0   0   1   0   0
    0   0   0]
 [  0   0   0   0   1   2   0   3   1   0   3 471   0   2   0   0   1   1
    0   1   0]
 [  0   0   5   2   1   3   3   1   2   2   2   0 424   6   1   2   0   9
    1  14   0]
 [  0   0  10  10  12   8   2  16   8   6   1   0   5 444   3   1   0   0
    0   1   1]
 [  0  12   8   1   2   2   2   3   3   9   6   0   1   7 430   6   1   5
    1  11   1]
 [  0   5  15   2   0   3   1   2   1   3   6   1   0   4   3 421   1   3
    0   9   2]
 [  0   2   0   1   0   0   0   0   0   0   0   0   0   0   2   0 508   0
    0   0   0]
 [  0   1   2   0   2   0   1   3   5   3   1   1   7   0   5   0   0 435
    2  26  11]
 [  0  14   0   0   0   1   0   1   2   1   1   0   5   0   3   2   2   8
  466  18   4]
 [  0   6   1   0   2   0   0   4   6   2   4   1   9   2  12   3   4  78
   25 297  31]
 [  0 123   2   0   3   0   0   0   2   1   2   1   1   5   3   2  27  53
    8  48 217]]
```

Project 2: Naive Bayes algorithm for learning to classify text

Classification Report:

	precision	recall	f1-score	support
.vscode	0.00	0.00	0.00	1
alt.atheism	0.70	0.84	0.76	483
comp.graphics	0.78	0.80	0.79	503
comp.os.ms-windows.misc	0.82	0.85	0.84	503
comp.sys.ibm.pc.hardware	0.81	0.86	0.84	474
comp.sys.mac.hardware	0.85	0.87	0.86	478
comp.windows.x	0.91	0.87	0.89	509
misc.forsale	0.83	0.83	0.83	501
rec.autos	0.87	0.86	0.87	529
rec.motorcycles	0.87	0.91	0.89	500
rec.sport.baseball	0.94	0.96	0.95	502
rec.sport.hockey	0.96	0.97	0.97	486
sci.crypt	0.91	0.89	0.90	478
sci.electronics	0.84	0.84	0.84	528
sci.med	0.91	0.84	0.87	511
sci.space	0.92	0.87	0.90	482
soc.religion.christian	0.91	0.99	0.95	513
talk.politics.guns	0.71	0.86	0.78	505
talk.politics.mideast	0.91	0.88	0.89	528
talk.politics.misc	0.68	0.61	0.64	487
talk.religion.misc	0.70	0.44	0.54	498
accuracy			0.84	9999
macro avg	0.80	0.80	0.80	9999
weighted avg	0.84	0.84	0.84	9999

NOTE : a '.vscode' directory is automatically created to store editor settings. It has been inadvertently included in the dataset processing. This category does not reflect the actual content of the newsgroups and should be ignored in the analysis of the model's performance.

Model Accuracy: The model achieved an accuracy of approximately 83.95% on the test set. This is a strong performance, indicating that the model is capable of correctly classifying the majority of the documents into the correct newsgroup.

Confusion Matrix Insights: The confusion matrix provides detailed insights into the performance of the classifier across different categories. Most categories show strong diagonal elements, indicating correct classifications. However, there are notable exceptions, particularly in categories with fewer samples or those that are more challenging to differentiate due to similar content.

Project 2: Naive Bayes algorithm for learning to classify text

- talk.religion.misc shows a considerable number of misclassifications, with many documents predicted as alt.atheism and talk.politics.mideast. This indicates potential overlap in the content of these groups or insufficient differentiation capability of the model for closely related topics.
- High misclassification rates from talk.politics.misc into talk.politics.guns and talk.politics.mideast suggest topical overlap or confusion between political discussions.

Precision, Recall, and F1-Score:
The classification report provides a breakdown of precision, recall, and F1-score for each category:

- Categories like rec.sport.hockey, soc.religion.christian, and rec.sport.baseball have high precision and recall, indicating the model's strong capability in correctly identifying and classifying documents in these categories.
- talk.religion.misc and talk.politics.misc have lower precision and recall compared to other categories. This highlights challenges in accurately classifying documents with nuanced differences in content or that may share vocabulary across categories.
- The weighted averages of precision (0.84), recall (0.84), and F1-score (0.84) align with the overall model accuracy, showcasing a balanced performance across the dataset.