Understanding Text Representation and Classification with Bag of Words Model

Dataset:

As a text dataset, you can use the 20 Newsgroups dataset which is a collection of approximately 18,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups.

Task Part 1: Introduction to CountVectorizer and Bag of Words Model

In this task, you will learn the basics of text representation using the Bag of Words model with CountVectorizer. The objective is to convert the text data into a format that can be used in machine learning algorithms.

1. Import the necessary libraries.
2. Load the dataset.
3. Use CountVectorizer from sklearn to transform the text data into numerical vectors. To do that, you have to learn how the parameters of CountVectorizer such as max_features, min_df, max_df, and stop_words work, and how they affect the resulting vectors.
4. Display the vocabulary (features) that were extracted from the text data.

Task Part 2: Exploratory Data Analysis of Text Data with Bag of Words Model

In this task, you will perform exploratory data analysis on the vectors created from the text data.

1. Investigate the distribution of word counts.
2. Identify and discuss any unusual frequent words. For example, in the 20 Newsgroups dataset, the term 'ax' is often found as a high-frequency term, which is actually a part of a sequence of characters ("axaxaxaxax...") that was often included in the quotes and is considered as noise. Besides 'ax' and any other unusual word you might find, exclude the following words: a, an, the, and, it, for, or, but, in, my, your, our, and their.
3. Define a list of custom stop words to be ignored by CountVectorizer. This can include the standard English stop words as well as dataset-specific noise (like 'ax' in the 20 Newsgroups dataset).
4. Re-run the CountVectorizer with the custom stop words and create a new Bag of Words model.

5. Identify the most and least frequent words in the vocabulary.
6. Visualize the findings using two appropriate plots. If you are not familiar with data visualization in Python, you may find this matplotlib tutorial helpful.

Task Part 3: Text Classification with Bag of Words Model

In this task, you will use the vectors created from the text data to perform text classification.

1. Split the dataset into training and test sets.
2. Train a simple classifier such as Naive Bayes or Logistic Regression on the training set.
3. Evaluate the classifier on the test set using appropriate metrics such as accuracy, precision, recall, and F1-score.
4. Interpret the results. Note that the labels in the classification report, such as 'sci.electronics', 'talk.politics.misc', and 'rec.autos', are the names of the newsgroups from which the data in the 20 Newsgroups dataset was collected. The 'sci.', 'talk.', and 'rec.' prefixes are part of the newsgroup names and indicate the broader category of the newsgroup.
5. Discuss how different parameters of CountVectorizer can affect the model performance.