Task documentation

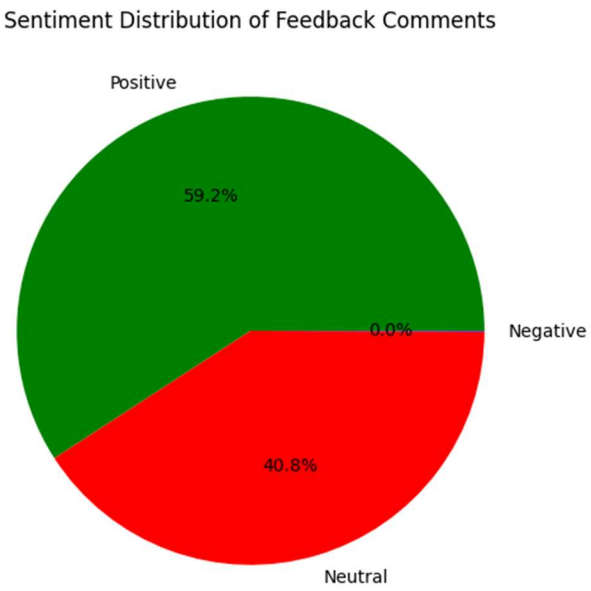
Code documentation:

The code loads a dataset from an Excel file named "AI_Engineer_Dataset_Task_1.xlsx" using the Pandas library. It assumes the dataset is present in the file and focuses on text data located within a specific column called "ParticipantResponse." To process the text data, it defines a function called "preprocess_text_nltk," utilizing the NLTK (Natural Language Toolkit) library. Within this function; It converts text to lowercase for uniformity. Tokenizes the text into individual words using NLTK's "word_tokenize" function. Removes punctuation, special characters, and digits, retaining only alphabetic characters. Eliminates common stopwords like "the," "is," and "in" using NLTK's predefined English stopwords list. Lemmatizes words to their base forms using NLTK's WordNetLemmatizer. Reconstructs the preprocessed text by joining cleaned words. The "preprocess_text_nltk" function is applied to each text entry in the "ParticipantResponse" column of the dataset. The result is stored in a new column called "text_data_preprocessed_nltk." Finally, the code prints the preprocessed text data, displaying the cleaned and transformed text entries. Additionally, the code defines a function called "get_sentiment_polarity," which assigns sentiment labels ("Positive," "Negative," or "Neutral") to text using the TextBlob library. It performs sentiment analysis, considering sentiment polarity:

- If the sentiment polarity is greater than 0, it's labeled as "Positive."
- If the sentiment polarity is less than 0, it's labeled as "Negative."
- If the sentiment polarity is exactly 0, it's labeled as "Neutral."

The sentiment analysis is applied to each comment in the dataset, assuming the preprocessed text data is in the "text_data_preprocessed_nltk" variable. Sentiment labels are added to a new column called "Sentiment" in the "text_data" DataFrame. The code calculates the distribution of sentiments in the dataset and visualizes it using a pie chart. The chart shows the distribution of sentiments (Positive, Negative, Neutral) with different colors (green, red, blue). To summarize sentiment distribution statistics, the code counts the occurrences of each sentiment category and prints a summary. Additionally, the code performs topic modeling. The preprocessed text data in "text_data_preprocessed_nltk" is tokenized, splitting text into individual words. The tokenized data is used to create a dictionary, mapping words to unique integer IDs. A matrix representing word frequencies in each document is created. The code specifies the number of topics to identify (e.g., 5). Latent Dirichlet Allocation (LDA) model is created for topic modeling. The code prints the identified topics and their representative keywords.

Result and analysis:




```
Sentiment Distribution Summary:
ParticipantResponse
Positive    107065
Neutral      73819
Negative        85

Recommendations:
Based on the sentiment analysis, there are more positive comments, indicat
ing areas of strength.
```