TEXT ANALYTICS
Group Assignment

Group 6

Student Name	PG ID
Akshay Ramdev	12220032
Charanjeet Singh	12220064
Pooja Nilesh Doshi	12220028
Snigdha Debashis Bhattacharjee	12220067
Vinayak Dave	12220047

README FILE

Requirements

- Python 3.x
- Pandas library (install with pip install pandas)

Steps to be followed in each of the 6 questions to run the code in Python Notebook:

1. **Create a New Python Notebook**
    - Open Jupyter Notebook or any other Python notebook environment.
    - Create a new Python notebook by clicking on "New" and selecting "Python 3" (or any other appropriate kernel).
2. **Import Common Libraries**
    - In the first code cell of the notebook, import the necessary libraries by adding the following code:
        1. import json
        2. import pandas as pd
3. **Load the JSON Data**
    - In the next code cell, copy and paste the provided code snippet.
    - Make sure the dataset file (**dataset.txt**) is in the same directory as the notebook or specify the correct file path in the code.
    - Run the code cell to load the JSON data.
4. **Review the Output**
    - After running the code cell, the output will be displayed below the cell.
    - You will see the pandas DataFrame containing the extracted 'reviewText' and 'overall' fields from the JSON data.

5. **Execute the Notebook**
    - Continue running the subsequent code cells to perform any further analysis or data manipulations, if desired.
    - You can add new code cells to the notebook and execute them as needed.

The following steps must be followed question specific:

Q0.py:

Steps to Run the Code in a Python Notebook

1. Run the “Below steps to be followed in each of the 6 questions:” instructed steps.

Q1.py:

Steps to Run the Code in a Python Notebook

1. Run the “Below steps to be followed in each of the 6 questions:” instructed steps .
2. **Execute Further Data Preprocessing**
    - Continue running the subsequent code cells to perform further data preprocessing steps.
    - The code provided in the subsequent cells performs the following tasks:
        - Lowercase the review text using spaCy library.
        - Remove punctuation from the review text.
        - Calculate IDF scores for the words in the review text.
        - Display the top 20 and bottom 20 words based on IDF scores.
3. **Save the Notebook**
    - Once you have executed the code and reviewed the output, save the notebook to retain your work.

By following these steps, you can execute the provided code in a Python notebook environment, load and analyse JSON data, and perform additional data pre-processing tasks.

Q2.py:

Steps to Run the Code in a Python Notebook

1. Run the “Below steps to be followed in each of the 6 questions:” instructed steps.
2. **Process the Data**
    - Continue running the subsequent code cells to perform further data processing steps.
    - The code provided in the subsequent cells performs the following tasks:
        - Selects a subset of the DataFrame for demonstration purposes.
        - Loads the spaCy English language model.
        - Performs sentence detection on the review text using spaCy.
        - Constructs a new DataFrame with the reviewer ID and individual sentences.
3. **Save the Notebook**
    - Once you have executed the code and reviewed the output, save the notebook to retain your work.

By following these steps, you can execute the provided code in a Python notebook environment, load, and process JSON data, and perform sentence detection on the review text.

Q3.py

Steps to Run the Code in a Python Notebook

1. Run the “Below steps to be followed in each of the 6 questions:” instructed steps.
2. Import the additional libraries by adding the following code:
   import spacy
3. **Review the Output**
    - After running the code cell, the output will be displayed below the cell.
    - You will see the pandas DataFrame containing the extracted fields 'reviewerID', 'reviewText', and 'overall' from the JSON data.
4. **Perform Tokenization**
    - Continue running the subsequent code cells to perform tokenization using spaCy.
    - The code provided in the subsequent cells performs the following tasks:
    - Selects a subset of the DataFrame for demonstration purposes.
    - Loads the spaCy English language model.
    - Performs tokenization, lemma extraction, and part-of-speech tagging on the review text using spaCy.
    - Constructs a new DataFrame with the reviewer ID, tokens, lemmas, and part-of-speech tags.
5. **Save the Notebook**

- Once you have executed the code and reviewed the output, save the notebook to retain your work.

By following these steps, you can execute the provided code in a Python notebook environment, load, and process JSON data, and perform sentence detection on the review text.

Q4.py

Steps to Run the Code in a Python Notebook

1. Run the "Below steps to be followed in each of the 6 questions:" instructed steps.
2. Import the additional libraries by adding the following code:
    - import spacy
    - from sklearn.model_selection import train_test_split
    - from sklearn.feature_extraction.text import TfidfVectorizer
    - from sklearn.naive_bayes import MultinomialNB
    - from sklearn.metrics import classification_report
3. **Load and Split the Data**
    - In the next code cell, load and split the data into training and testing sets.
    - Make sure the **df** variable contains the appropriate DataFrame that contains the necessary fields for training and testing.
    - Adjust the **test_size** and **random_state** parameters of the **train_test_split** function as needed.
    - Run the code cell to perform the data splitting.
4. **Create TF-IDF Features**
    - Continue running the subsequent code cells to create TF-IDF features using the **TfidfVectorizer** class.
    - The code provided in the subsequent cells performs the following tasks:
        - Creates an instance of the **TfidfVectorizer** class with the desired configuration.
        - Converts the training data into TF-IDF features using the **fit_transform** method of the vectorizer.
        - Initializes a Naive Bayes classifier (**MultinomialNB**).
        - Trains the classifier using the TF-IDF features and the corresponding target labels from the training set.
        - Converts the test data into TF-IDF features using the **transform** method of the vectorizer.
        - Predicts the target labels for the test data using the trained Naive Bayes classifier.
        - Computes the classification report to evaluate the performance of the classifier.
5. **Review the Output**
    - After running the code cells, the output will be displayed below each cell.
    - You will see the TF-IDF features, test predictions, and the classification report containing metrics such as precision, recall, and F1-score for each class.
6. **Save the Notebook**
    - Once you have executed the code and reviewed the output, save the notebook to retain your work.

By following these steps, you can execute the provided code in a Python notebook environment, perform text classification using Naive Bayes with TF-IDF features, and evaluate the model's performance using classification metrics.

Q5.py

Steps to Run the Code

1. **Import Libraries**
    - from summa import summarizer
2. **Filter and Preprocess the Data**
    - The subsequent code cells filter the DataFrame to select a subset of reviews with specific ratings.
    - Adjust the conditions and number of reviews (head()) as needed.
    - The filtered reviews are reindexed and displayed in the output.
3. **Perform Text Summarization**
    - Install the Summa library by running !pip install summa in a code cell (if not already installed).
    - Use the provided code to perform text summarization on the selected review texts.
    - The code creates a single string by joining the review texts and then uses the Summa library to generate summaries.
    - Adjust the parameters such as ratio or words to customize the length of the summary.
    - The original review text and the generated summary are displayed in the output.
4. **Review the Output**
    - After running the code cells, the output will be displayed below each cell.
    - You will see the loaded data, filtered reviews, and the original review texts along with their respective summaries.
5. **Save the Notebook**
    - Once you have executed the code and reviewed the output, save the notebook to retain your work.

By following these steps, you can execute the provided code in a Python notebook environment, perform text summarization using the Summa library, and generate summaries for specific subsets of reviews based on their ratings.