Sentiment Analysis of Social Media Data

This assignment is due on Oct.4th, there will be no extensions so please start early.

Dataset Description: Data has been gathered from Twitter related to the stock market crash of 2022, using the hashtag #stockmarketcrash. The dataset contains tweets categorized into three sentiment labels:

- Positive: 12,542 tweets
- Neutral: 11,498 tweets
- Negative: 9,906 tweets

dataset.csv Download dataset.csvclick to download

Objective: This assignment focuses on performing sentiment analysis on the provided dataset to classify tweets into one of the three categories: positive, neutral, or negative. You are required to use and compare different methods of sentiment analysis.

Tasks:

1. **Data Preprocessing:**
    - Preprocess the text as needed
    - Split the dataset into training and testing sets. For each sentiment label, randomly select 80% of the data for training; the remaining 20% will be used for testing. You need to report accuracy on testing data.
2. **Techniques for Sentiment Analysis:**
    - **Frequency-Based Embeddings:** Use Bag of Words and TF-IDF to vectorize the text (NLTK library). Select the top 100 features (words) using the chi-square test to refine your feature set.
    - **Word Vectors:** Employ pre-trained word vectors from Word2Vec and GloVe for feature representation. You may use vectors pretrained on Wikipedia or other corpora.
    - **Sentence Vectors [optional bonus part]:** Utilize advanced sentence pretrained embedding techniques such as the Universal Sentence Encoder (USE).
    - Implement the above methods of embeddings. Given these embeddings as features, train a Support Vector Machine (SVM) to predict the sentiment. Compare how the model's accuracy changes with each embedding type.

Deliverables:

- **Code:** Submit a Jupyter Notebook or a Python script containing all the preprocessing, model implementation, and evaluation code.
- **Report:** Provide a comprehensive report that includes:
    - **Methodology:** Detailed description of your preprocessing steps, choice of vectorization methods, and model implementation.
    - **Findings:** Summary of model performance, including accuracy comparisons and discussions on why certain embeddings performed better than others. You can show examples from data to support your discussion.

Submission Guidelines:

- Ensure that all code is well-commented and organized.
- Include all necessary libraries and dependencies required to run your code.
- No cheating allowed. For plagiarism, we will be comparing your code with your peers and with the code generated by LLMs such as Chat-GPT.
- Submit your report in PDF format, ensuring it is clearly structured and well-written.

PS: Total points for assignment will be 100, and additional 10 for bonus part.