Task to be performed

Overview:
In this project, you will perform sentiment analysis on a dataset of Twitter posts using Recurrent Neural Networks (RNN). You will preprocess the data, conduct exploratory data analysis (EDA), build an RNN model for sentiment classification, and evaluate the performance of your model. The goal is to classify the sentiment of tweets as positive, negative, or neutral based on the text data.

Dataset Overview:

It contains tweet data, including the tweet text, sentiment labels (positive, negative, or neutral), and other metadata (e.g., tweet ID, user information, and date of the tweet).

Tasks:

Part 1: Data Processing

1. **Load the Dataset:**
   - Load the CSV file into an appropriate data structure (e.g., DataFrame).

2. **Data Cleaning:**
   - Check for and handle missing values in the dataset.
   - Remove duplicates if any exist.
   - Perform text cleaning on tweet text (e.g., remove URLs, mentions, hashtags, special characters).
   - Tokenise the text and convert words to lowercase.
   - Remove stop words and apply stemming or lemmatisation.
3. **Feature Engineering:**
   - Convert the text data into numerical format (e.g., using TF-IDF, Word2Vec, or embeddings).
   - Create a sequence of tokenized words for each tweet.

Part 2: Exploratory Data Analysis (EDA)

1. **Basic Statistics:**
   - Summarise the dataset (mean, median, mode, etc.).
   - Explore the distribution of tweet sentiments (e.g., how many positive, negative, and neutral tweets are there?).
2. **Visualisations:**
   - Create visualisations to showcase:
     - The distribution of sentiments.

- The frequency of top words in positive, negative, and neutral sentiments.
- Word clouds for positive and negative tweets.
- The relationship between tweet length and sentiment.

3. **Insights:**
   - Write a brief summary of your findings from the EDA. What patterns or trends did you observe in the sentiment distribution?

Part 3: Building the RNN Model

1. **Model Architecture:**
   - Build an RNN model using LSTM (Long Short-Term Memory) or GRU (Gated Recurrent Units) for sentiment classification.
   - Use an embedding layer to represent the text data.
2. **Model Implementation:**
   - Split the dataset into training and testing sets.
   - Train the RNN model using the training set and evaluate using the test set.
   - Implement dropout and batch normalisation (if necessary) to improve model performance.
3. **Evaluation:**
   - Evaluate the performance of your RNN model using metrics such as accuracy, precision, recall, and F1-score.
   - Plot learning curves to monitor training progress and avoid overfitting.
   - Perform hyperparameter tuning (e.g., number of layers, hidden units, learning rate).
4. **Model Improvement:**
   - Implement techniques such as grid search, cross-validation, or transfer learning to improve model performance.

Part 4: Presentation

1. **Documentation:**
   - Prepare a report documenting your entire process, including data preprocessing steps, EDA findings, model architecture, and evaluation results.
   - Include visualisations and code snippets where applicable.
2. **Presentation:**
   - Create a presentation summarizing your project for your classmates. Cover the following:
     - Overview of the dataset and objectives.
     - Key findings from EDA.
     - Methodology for building the RNN model.

- Evaluation results and performance metrics.
- Challenges faced and how you improved model performance.
- Demonstration of the sentiment classification model on sample tweets.
