Mini Project 3 – Twitter Sentimental Analysis Using NLP and Python

Scenario: By analyzing text data, we can find meaningful insights from non-numeric data that can help us achieve our objective. With the help of NLP and its concepts, we can do it. Twitter is one of the biggest platforms that people use to write their messages, express their feelings about a particular topic, and share knowledge in the form of text. By analyzing text data, we can make good decisions for different use cases like judging the sentiment of the human tweets, and any product review/comments can tell us the performance of a product in the market.

NLP allows us to study and understand the colinearity of the data. So we can predict our objective.

Objective: Use Python libraries such as Pandas for data operations, Seaborn and Matplotlib for data visualization and EDA tasks, NLTK to extract and analyze the information, Sklearn for model building and performance visualization, to predict our different categories of people's mindsets.

Dataset description: The data contain information about many Tweets in the form of text and their types, as mentioned below.

Tweets: Data is in the form of a sentence written by individuals.

category: Numeric(0: Neutral, -1: Negative, 1: Positive) (It is our dependent variable)

The following tasks are to be performed:

- Read the Data from the Given excel file.
- Change our dependent variable to categorical. (0 to “Neutral,” -1 to “Negative”, 1 to “Positive”)
- Do Missing value analysis and drop all null/missing values
- Do text cleaning. (remove every symbol except alphanumeric, transform all words to lower case, and remove punctuation and stopwords)
- Create a new column and find the length of each sentence (how many words they contain)
- Split data into dependent(X) and independent(y) dataframe
- Do operations on text data

Hints:

  - Do one-hot encoding for each sentence (use TensorFlow)
  - Add padding from the front side (use Tensorflow)
  - Build an LSTM model and compile it (describe features, input length, vocabulary size, information drop-out layer, activation function for output,)
  - Do dummy variable creation for the dependent variable
  - split the data into tests and train

- Train new model
- Normalize the prediction as same as the original data(prediction might be in decimal, so whoever is nearest to 1 is predicted as yes and set other as 0)
- Measure performance metrics and accuracy
- print Classification report