**STEP-BY-STEP APPROACH FOR SOLVING THE NLP ASSIGNMENT**

**OBJECTIVE 1:**

1. The input tweet data was shared as a JSON file. So, at first, this was converted to a Pandas DataFrame for handling tweet data using python.
2. The tweet data can be considered unstructured data. So, the next step includes cleaning the tweet DataFrame.
3. The cleaning process includes,

* Removing the # and @ texts:
* The # symbol is removed from texts and @ contained text says about the user id. So, these two are removed from the texts.
* Removing URLs:
* The URLs are all removed as they are not useful in our current analysis
* Converting other language texts to English
* The issue faced in converting the other language data is it takes more time for translation (around 90mins).
* At first non-English texts are identified. The time for identifying the language took around 5mins and then they are passed to a user-defined function for translation
* So, now all the non-English texts are translated into English for further cleaning steps
* The output till Step 3 is stored as a CSV file named “translated\_tweet.csv” for code purpose
* Removing Punctuations
* All the punctuations and emojis that are available in texts are removed in this step
* Removing Stop words
* The English stop words that are available in the NLTK corpus are loaded and removed from the texts
* Lemmatization
* In this step the words in the text are converted to their root words. For example, see, saw, and seen mean “see”.
* This is done using the lemmatizer method in NLTK

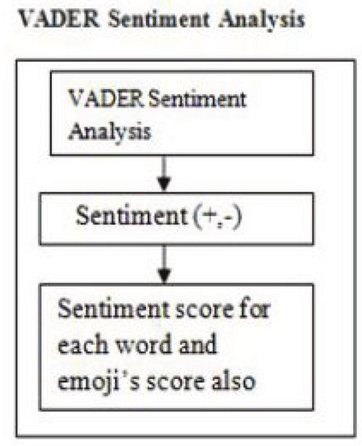
1. Now we have cleaned tweet data, which can we can use for entity recognition. There are two ways identified to do this,
   * Text Blob Noun phrase detection
   * Spacy’s NER model
2. Both the models are tried in finding entities, but Text Blob comparatively is performing very well compared to Spacy’s. It is able to identify more entities compared to Spacy’s.
3. The Text Blob method finds the noun phrases in a text and assigns them as an entity.
4. Once the entities for each text are collected, we will find the frequency of entities across all tweets and are stored as “Objective1.csv”

**OBJECTIVE 2:**

1. From Objective 1 we collected entities for each tweet, using these entities sentiment will be predicted.
2. We are using an already pre-trained model for sentiment analysis. Here we are using the NLTKS VADER module
3. VADER (Valence Aware Dictionary for Sentiment Reasoning) is a lexicon-based sentiment analysis tool that can be used to determine the sentiment expressed in a text. It is a rule-based approach that uses a dictionary of words annotated with sentiment polarity scores to analyze the sentiment of a piece of text.
4. Vader is used because it is designed to work well with social media text, which is often informal and written in a unique way.
5. The scores range from -1 (very negative) to 1 (very positive).
6. To analyze the sentiment of an entity it needs to be analyzed w.r.t to tweet. So, the entity along with a portion of the tweet will be passed to the SentimentIntensityAnalyzer function of vaderSentiment which will produce the sentiment scores for each entity.
7. All the entities having scores> 1 are encoded as positive; scores < 1 as negative and

scores =1 as neutral.

1. The final result with encoded scores will be stored as “Objective2.csv”.



The reference code for Vader sentiment analysis is available on the web. Links are shared below,

<https://plainenglish.io/blog/twitter-sentiment-analysis-using-vader-tweepy-b2a62fba151e>

<https://towardsdatascience.com/social-media-sentiment-analysis-with-vader-c29d0c96fa90>