Synapse – Sentiment Analysis

Sentiment Analysis is the process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic, product, etc., is positive, negative, or neutral.
For this project, I have streamed live twitter data into a database and performed sentiment tagging using an NLP library - TextBlob.

1. Data:
   I used the Twitter API to search for tweets containing specific keywords related to "Avengers" and streamed this into the database using the resources listed below.
   - A Twitter account and API credentials.
   - A MySQL database.
   - The Tweepy and mysql-connector Python Libraries.

   The database has around 3000 rows, where each row contains the following fields:
   - Primary Key
   - Username
   - Created_at
   - Tweet
   - Retweet_count
   - Location

   Sample tweets from the database:

Primary Key	Username	Created_at	Tweet	Retweet_count	Location
1	'UrKllinMeSmalls'	'2019-07-15 23:49:02'	'@Medic968 Yup. Thor with their coaching/analytics will turn it around big time'	0	'The Sandlot, Where Else?
2	'scottlangstaco'	'2019-07-15 23:49:08'	@classicparker_: \"He\'s like Captain America and Thor rolled into one.	0	'Queens'

2. Data Cleaning and Pre-processing:
   Text data generally requires some pre-processing before we can feed it to a machine learning algorithm. We need to put it into a format that an algorithm can understand. Hence, the following Natural Language Processing tasks were performed on the dataset using the NLTK library.
   - Removal of punctuations and special characters.
   - Encoding ASCII to utf-8 - this step had to be performed since the tweets that were streamed from the twitter API were of ASCII characters and the sentiment analysis libraries supports only UTF-8 encoding.
   - Removal of stop words- removal of stop words was a necessary step because these words can be distracting, noninformative (or non-discriminative) and are additional memory overhead.

- Stemming- helps reducing a word to its word stem that affixes to suffixes and prefixes or to the roots of words known as a lemma.

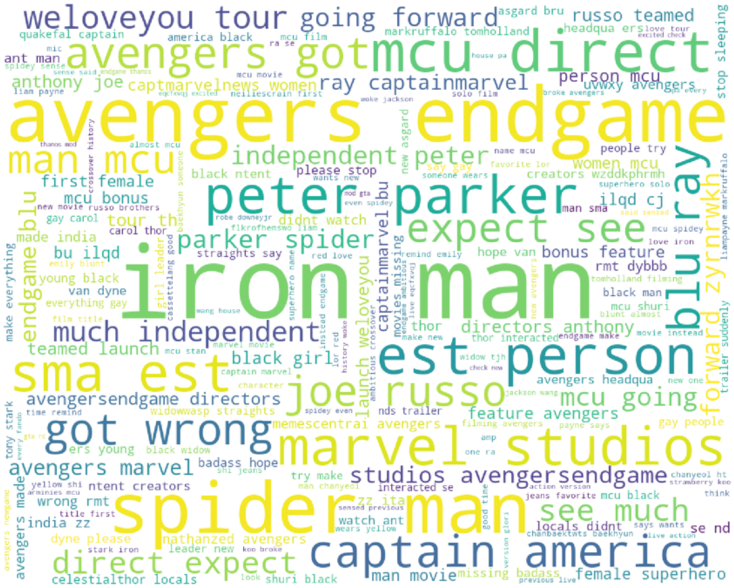
3. Sentiment analysis.

   I have used TextBlob to perform Sentiment Analysis. It is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as noun phrase extraction, sentiment analysis, classification, translation, and more.

   - The input for TextBlob is the cleaned twitter dataset which is a CSV file containing a cleaned tweet in every row.
   - The sentiment function of textblob returns two properties, polarity and subjectivity in the form of a named tuple Sentiment (polarity, subjectivity). The polarity score is a float which lies within the range [-1.0, 1.0] where 1 means a positive statement and -1 means a negative statement.
   - Subjective sentences generally refer to personal opinion, emotion or judgment whereas objective sentences refer to factual information. The subjectivity score is also a float which lies within the range [0.0, 1.0] where 0.0 is very objective and 1.0 is very subjective.

4. Results.

   The database consists of tweets related to “Avengers”. As seen from the word cloud, words like “avengers” “endgame” , “spider man” seem to be the some of the most frequent ones.



Sentiment scores:

```
Textblob results:
percentage of positive tweets: 38.70545930042498%
percentage of negative tweets: 20.398823144818568%
percentage of neutral tweets: 40.89571755475646%
```

The results from the sentiment scores indicate that majority of the tweets are neutral at around 40%.

I tried to go with a simplistic approach that shows various aspects like retrieving tweets, cleaning up the data and sentiment analysis.
If I had to do this again, I would try to implement a machine learning algorithm from scratch(maybe use unsupervised techniques like clustering) and use a larger dataset to train and test the model.