

## IT350 - ASSIGNMENT 3

Annam Indhu Lekha

191IT207

### Problem statement:

### Sentiment Analysis of live stream twitter data

### Link to Colab:

<https://colab.research.google.com/drive/1LDSPtQQHciflumvt4TUR1KdM4NpJJcSk?usp=sharing>

1. Utility function that fetches sample tweets (15 tweets at once)

```
def get_sample_tweet():
    access_token = get_auth_token()
    search_headers = {
        'Authorization' : 'Bearer {}'.format(access_token)
    }
    search_params = {
        'q': "india",
        'lang': 'en',
    }
    search_url = str(TWI_SEARCH_URL).format(base_url)
    search_resp = requests.get(search_url, headers = search_headers, params =
search_params)

    print("Search Query Status :", str(search_resp.status_code))

    return search_resp
```

We can change search parameters and we'll get results accordingly.

```
def create_json():
    search_resp = get_sample_tweet()
    tweet_data = search_resp.json()['statuses']
    with open("./tweet_geo_located.json", "w+", encoding=ENC_UTF8) as tfile:
        json.dump(tweet_data, tfile, indent=2)
```

```

        tfile.write("\n")

import time
for i in range(2):

    time.sleep(60)

    create_json()

    positive, negative, neutral = get_count()

    visualise(positive, negative, neutral)

```

I have created a json file (that contains 15 tweets) every 60 sec for 2 times. This time gap makes sure that the tweets we fetch everytime are not the same as the tweets that were previously fetched.

## 2. Sentiment Analysis using TextBlob

This is a function that takes a tweet and returns which category it belongs to(Positive, Negative, Neutral)

```

def sentiment_analysis(tweet):

    def getSubjectivity(text):

        return TextBlob(text).sentiment.subjectivity

    #Create a function to get the polarity

    def getPolarity(text):

        return TextBlob(text).sentiment.polarity

    def getAnalysis(score):

        if score < 0:

            return 'Negative'

        elif score == 0:

            return 'Neutral'

        else:

            return 'Positive'

    subjectivity = getSubjectivity(tweet['text'])

    polarity = getPolarity(tweet['text'])

```

```
sentiment = getAnalysis(polarity)

return sentiment
```

- The output of TextBlob is polarity and subjectivity.
- Polarity score lies between (-1 to 1) where -1 identifies the most negative words such as 'disgusting', 'awful', 'pathetic', and 1 identifies the most positive words like 'excellent', 'best'.
- Subjectivity score lies between (0 and 1), It shows the amount of personal opinion. If a sentence has high subjectivity i.e. close to 1, It resembles that the text contains more personal opinion than factual information.

I was more concerned about the Polarity score as my objective was not to identify factual information.

The below function parses through all the tweets in the json file and counts the number of positive, negative and neutral tweets.

```
def get_count():

    count_pos = 0

    count_neg = 0

    count_neutral = 0

    with open('tweet_geo_located.json') as data_file:

        data = json.load(data_file)

        for tweet in data:

            sentiment = sentiment_analysis(tweet)

            if sentiment == 'Positive':

                count_pos += 1

            elif sentiment == 'Negative':

                count_neg += 1

            else:

                count_neutral += 1

    return count_pos, count_neg, count_neutral
```

### 3. Visualization

This function takes the count of positive, negative, and neutral tweets and plots a bar graph and pie chart.

```
def visualise(count_pos, count_neg, count_neutral):  
    df = pd.DataFrame({'Sentiment' : ['positive', 'negative', 'neutral'],  
                       'Count': [count_pos, count_neg, count_neutral]})  
    df.plot(x = 'Sentiment', y = 'Count', kind = 'bar')  
    plt.title("Sentiment analysis of tweets - bar graph")  
    plt.xlabel('Sentiment')  
    plt.ylabel('Tweet Count')  
    plt.show()  
  
    df.plot(x = 'Sentiment', y = 'Count', kind = 'pie', labels = ['positive',  
'negative', 'neutral'])  
    plt.title("Sentiment analysis of tweets - pie chart")  
    plt.show()
```

Representation of sentiment analysis of tweets in the span of 60 sec:

