TASK: Sentiment Analysis

Description: Perform Sentiment Analysis on tweets that you can fetch from Twitter API or use any dataset for the same.

This given task can be divided into two main sub-tasks, i.e. fetching tweets and performing sentiment analysis.

Objective: To stream n tweets containing a particular keyword using twitter API

**Resources needed**:

* The twitter API [access keys and access tokens]
* tweepy python module

Methodology: Register for the twitter developer program and obtain API keys and tokens and store them in a credentials file. Then import the credentials and tweepy module. Call the tweepy authenticator function to authenticate keys and tokens and return the auth object. Passing this object to tweepy.API method generates an API stream from which we can filter in tweets with the required keyword/words.

**Limitations**:

* the twitter API v1 only allows streaming upto 150 tweets per hour for the free developer version per project. After this limit is exceeded, you will get an error and multiple violations result in exponentioal increment of “ban” time.
* The access tokens and keys are sensitive information that can be potentially used to tweet from the users account, send direct messages and so on.

**Other methods**: no other viable and or convenient methods of extracting data from twitter were found.

Objective: To clean the tweets and perform sentiment analysis on them.

**Resources needed**:

* pandas python module
* nltk python module
* SentimentIntensityAnalyzer class from nltk.sentiment.vader
* numpy python module
* re python module [aka regualr expression]
* credentials file [created for twitter API keys and tokens]

Methodology: define a method that takes in tweet strings and uses regular expression library to filter out usernames (@....), removes hashtag symbols, URLs, and other symbols from the given tweets.

nltk- Vader sentiment analyzer is a Rule-based Model for Sentiment Analysis of Social Media Text. It stores words that increase or decrease the sentiment of the given text in a dictionary called BOOSTER\_DICT (increases or decreases the sentiment of given text word by word). It stores keywords that negate the sentiment of given text like not, isn’t, aren’t, etc.. and special case idioms that have better or worse sentiment due to the order of words eg. “Badass” is comprised of “bad” and “ass” both indicating negative sentiment but together they have an unexpected positive sentiment.

The polarity\_scores method in SENTIMENT\_ANALYZER class then calculates positive, negative, neutral scores based on data in booster dict, negations dict, and idioms.

**Limitations**: slower than SpaCy but doesnt require any training datasets or models

**Other methods** : SpaCy , TextBlob,TensorFlow ,etc.