# Using Social Media Sentiment Analysis to Predict Stock Prices

Final Project for CSE 6240 - Web Search & Text Mining

## Project Goal

The goal of this project is to explore the concept of emotional theory in the stock market by understanding if social media sentiment analysis can be used to predict fluctuations in the stock price of a given company. To do so, we extracted, cleaned and generated sentiment for more than 3 million tweets on 54 companies and 4 million reddits on 30 companies traded in the two main U.S. stock exchanges (NYSE and Nasdaq).

https://github.com/phpinto/social_media_sentiment_analysis_to_predict_stock_prices

## Folder Structure

The folders are organized the following way:

- baselines:
    - This folder contains the Jupyter Notebooks to run the baseline models for both the twitter (twitter_baselines.ipynb) and reddit (reddit_baselines.ipynb) datasets.
- data:
    - Contains a text file with the link to the google drive containing our datasets.
- data_collection:
    - reddit - The RedditData notebook executes all of the API calls and writes all the data to a csv, reddit_data.csv The CleanReddit notebook loads the reddit_data.csv and performs perfunctory cleaning operations for easy use and writes the data to reddit_data_clean.csv The Sentiment notebook loads the reddit_data_clean.csv and performs the sentiment analysis, writing out the reddit_text_sentiment.csv **It takes a long time to run these notebooks. For the convenience of the user, the reddit_data.csv, reddit_data_clean.csv, and reddit_text_sentiment files are provided in zipped format.**
    - stocks - iex_cloud.php contains the code used to connect to the IEX Cloud Api. To run the code, you must first install the composer PHP dependency manager in your machine and run the command "composer install" in the data_collection folder. You also need an API key and registered account with IEX Cloud. After running this code and writing the stock data to a MySQL database, the phpMyAdmin GUI was used to convert it to a csv file. **All the data acquired in this folder is available in the google drive link shown at /data/link_to_download_data.txt**
    - twitter - TwitterData.py -> contains the code to extract tweets using GetOldTweets3 python library. The parameters passed include: start and end date (in this case it was fixed from "2017-01-01" to "2020-04-02"), number of tweets to be extracted per hasthag per day (in this case it was fixed at 500) and hasthags to be extracted. The hasthags to be extracted come from a CSV file - "brands.csv" - which is included in the Google drive folder shared below. Because it takes a long time to extract all tweets, we extracted 10 tweets a time and we define the rows to be used as a parameter in the terminal. The command to run in the terminal includes "TwitterData.py brands.csv $first row - last row$ $output name". The twitter.php file writes the twitter data in the csv file into a MySQL database. **All the data acquired in this folder is available in the google drive link shown at /data/link_to_download_data.txt**
    - fortune_500 - This is a legacy folder for scraping revenue data from companies in the Fortune 500 lists. Because we pivoted from predicting revenuse to predicting stock prices, this folder is no longer relevant.
- twitter_code:
    - Creating_cleaned_database.ipynb -> Reads in the final raw dataset with all the tweets extracted. We have provided the CSV file "TwitterRaw.csv" in the Google drive folder share below. This jupyter notebook combines the raw database with 3 additional auxiliary tables "companies.csv", "index_names.csv" and "companies_industry.csv" also provided in the Google drive. These 3 datasets are used to link each hasthag and brand to a company and to an industry. After joining the 3 datasets, this notebook provides code to clean the data and exports a new csv file "clean_tweet.csv" which we will also provide in the Google drive. This csv file is then used in the next jupyter notebook file to generate sentiment for each tweet. Finally, this notebook provides all the raw data statistics that we present on the final report.
    - Generating_sentiment.ipynb -> Reads in the "clean_tweet.csv" dataset with all the clean tweets. We have provided

the CSV file "clean_tweet.csv" in the Google drive folder share below. This jupyter notebook uses nlkt.sentiment package to generate sentiment for each cleaned tweet. It generates 4 scores - compounded, positive, negative and neutral. It saves it as a csv file "twitter_sentiment.csv" provided in the Google drive folder. In the jupyter notebook we also do a explanatory data analysis of the sentiment generated, by date, industry and company. We use word cloud to better visualize our results. Finally, we use CountVectorizer() to generate term frequency matrix, which we export in this notebook as "term_freq_df.csv", available on Google Drive.

- Term_frequency.ipynb -> Reads in the "term_freq_df.csv" available on Google Drive and performs an analysis of the top tokes on the dataset, top positive and top negative tokens (excluding StopWords). We plot frequencies and we visualize how many times each token appears on positive vs negative comments. Some of the visualizations are used in the final report.
- Correlation_Stock_Twitter.ipynb -> Reads in "twitter_sentiment.csv" available on Google Drive and stock prices data "stock_prices.csv" available on Google Drive as well. It removes all data for stocks that is not available on Twitter dataset and combines both datasets, aggreagting tweets on a daily basis per company. A mean compounded, positive, negative and neutral score is calculates, as well as a weigthed compounded, positive, negative and neutral score, which is calculated by adjusting to the number of retweets. We then compared the standard mean with the weighted mean and computed correlations for industries and companies. We exported a csv file "stocks_data_twitter.csv" that we will use in the "Twitter_stockPrices_prediction.ipynb".
- Twitter_stockPrices_Prediction.ipynb -> Reads in "stocks_data_twitter.csv" available on Google Drive. Performs K-means clustering using all mean sentiment scores as features. Imports the libraries for modeling, adds the time lag variables and creates functions for the models, for performance evaluation of the models and for the threshold analysis. Tests different models on different segmentation strategies.

# Running the code

To run the code you will have to download the datasets available on:
https://drive.google.com/drive/folders/1JFBayDIIdu5C6wPM9jueHmPk6dd48ajC?usp=sharing

**Save all data sets to the /data/ folder. Don't keep the reddit_datasets, twitter_datasets and stocks_datasets folders seen on google drive. Open each of those folders on google drive and save all the individual data files into the /data/ folder**

## Running code - "baselines"

After downloading all the datasets into the /data/ folder, run the "jupyter notbook" command in this folder. After that, you can use your web browser to run all the code contained in the two notebooks.

## Running code - "reditt_code"

The models notebook contains the procedure for testing out Random Forest and Boosting as well as a procedure for seeing results by company for the baselines and for the Random Forest and Boosting. Run the "jupyter notbook" command in this folder and use your web browser to run the code contained in the notebooks.

## Running code - "twitter_code"

Download the twitter datasets separately and add them to the "/data/", folder as instructed earlier. The jupyter notebooks should be able to run this way. The notebooks are in order, which means you need to run the previous notebooks to get to the following ones. Alternatively, you can just import the datasets on one of the jupyternotebook files, since we are providing all the datasets on Google drive. So for instances, if you are just interested in modeling, you can just go to "Twitter_stockPrices_Prediction.ipynb" and read the "stocks_data_twitter.csv" available in the folder.