# CSE 482 FINAL PROJECT

Project Title: Twitter Sentiment Data for Predictive Modeling

Summary of Team Member Participation:

Fill out the following table for each team member of the group.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Participate in data collection** | **Participate in preprocessing** | **Participate in data analysis/ experiment** | **Participate in writing the final report** | **Participate in creating video presentation** | **Completed Assigned Tasks** |
| David Cho | X | X | X | X |  | X |
| Brandon Hu | X | X | X | X |  | X |

Team Member Roles and Contributions:

|  |  |
| --- | --- |
| **Name** | **Roles and Contributions** |
| David Cho | Responsible for collecting Twitter data and creating twitter sentiment analyzer. Co-author of final report. |
| Brandon Hu | Responsible for collection of gold, oil, corn, dow, and Microsoft datasets. Responsible for analysis of data, helped preprocess data. Co-author of final report. |

I approve the content of the final report (please add your signature below):

David Cho: ------------------------David Cho--------------------------

Brandon Hu: ------------------Brandon Hu--------------------------------

Twitter Sentiment Data for Predictive Modeling

David Cho, Brandon Hu

<https://github.com/davidcho9/CSE482-Project.git>

**ABSTRACT**

The goal of our project is to create a regression model that predicts the stock price of Microsoft based on Twitter sentiment data about Microsoft from the day before, and also use the oil price, gold price, corn price, and the Dow Jones Industrial Average.

# INTRODUCTION

This project was created with the following in mind:

1. The main goal of our project is to be able to confidently, and accurately predict the stock price of specific companies given information from Twitter and other sources that could relate to changes in stock price. This goal, if sufficiently achieved, can be monumental in one’s investment portfolio for the fact that one can use the model to decide whether to buy/sell a stock that they hold with enough confidence that their action will grant them a net positive in gains over a period of time. This project uses sentiment analysis, which is a metric used to determine whether a given statement is overall a positive or negative statement. This can be potentially useful as it is proven that individuals who participate in the stock market often times buy/sell based on emotion. If one can determine how the general populace views a particular company on a given day, then it may correlate to a change in the stock price.
2. In this project, we hypothesize that twitter sentiment is a significant factor when determining future stock prices. Microsoft is used as the target stock to predict for this project specifically.
3. The plan for this project is to create a linear regression model based on the following attributes: historical gold price, oil price, corn price, the Dow Jones Industrial Average, and Twitter sentiment data on
4. For this project, we created our own version of a sentiment analyzer to analyze individual tweet’s positivity. We collected the twitter data using a py3.6 library called GetOldTweets3 found on GitHub that allows us to collect data from more than one week in the past. We also used the Federal Bank of St. Louis website to get the datasets for gold and oil, finance.yahoo.com for Microsoft and Dow historical data, and investing.com for the corn dataset. We used the z-score of the Microsoft opening prices to check for any outliers, then removed those outliers from the dataset.
5. Some challenges revealed themselves during the data collection and preprocessing stages. One of which was the collection of historical twitter data from 2013. The twitter historical API only allows users to collect data from time periods of up to a week in the past, which is not helpful for us, as this would leave us with about only a couple of months worth of twitter data, thus bottlenecking the rest of the datasets into only a month also. We circumvented this problem by using a library written by Mottl who posted a twitter web crawler on GitHub that returns tweets from any time period. More problems occurred and will be discussed in the next section (Section 2. Data).
6. The findings of our project were unreliable and inconclusive. This result may be caused by several reasons, the main reason being model underfitting due to irrelevant attributes or lack of relevant attributes. Due to the insignificance of our findings, we can neither confirm or deny the importance of twitter sentiment data in a model based on the same attributes as our model.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*CSE881-2015*, Month 1–2, 2004, City, State, Country.

Copyright 2004 ACM 1-58113-000-0/00/0004…$5.00.

# DATA

The datasets used for the gold and oil tables were downloaded and found on the Federal Bank of St. Louis website (fred.stlouisfed.org/). The Dow Jones and Microsoft historical stock information was found on finance.yahoo.com. The corn price data was found on investing.com. All datasets were downloaded in .csv format. The data collected for Twitter used the library published by Mottl called GetOldTweets3. GetOldTweets3 acts similar to a web crawler in that it requests related pages from the twitter webserver, and requesting too many pages too quickly can result in a temporary page request denial for overloading their servers. This problem can be solved by requesting tweets day by day (not real time days but historical days, i.e request 2013-04-01), then telling the script to sleep for 180 seconds, then requesting the next day. Each day takes around 30 seconds to collect tweets, which means collecting data for one year takes approximately 21 hours. However, since the collection of the twitter data, and writing of the twitter data into a CSV file is all automated, the script is allowed to run in the background without it stopped due to twitter blocking page requests. Due to the vast amount of tweets provided in a day, we decided to only collect the top 500 tweets for any provided day. We thought this to be fair as without narrowing down the amount of tweets collected at once, we would not be able to automate our data collection and thus not be able to have a large enough dataset, and compromised that searching by only the top tweets removes all duplicate (retweets) datapoints and is representative of the majority feeling, because top tweets include only tweets with the most likes and favorites.

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Type** | **Description** |
| Date | Ordinal | Date data occurs |
| Tweet | Nominal | Text value of tweets related to Microsoft |
| Dow Ind Avg | Ratio | Opening price of Dow Jones Industrial Average |
| Corn | Ratio | Price of corn |
| Gold | Ratio | Price of gold |
| Oil | Ratio | Price of oil |

**Table 1**: Attributes of the data acquired from yahoo.com, invsting.com, fred.stlouis.gov and twitter.com.

We initially collected only one years worth of data which, after removing missing values and weekends, summed up to ~260 days/datapoints. For our final model, we used three years worth of data, from April 4th, 2013 to March 31, 2016, which netted us exactly 742 data points to work with. The resulting data points added up to 742 instead of 1,095 because the stock market is only open during weekdays, and some datasets had missing values other than weekends. To ensure that all of the datasets had the same exact days in the same order, we had to find missing datapoints from one dataset, and remove those datapoints from all other datasets.

|  |  |
| --- | --- |
| Number of observations | 747 (~37 MB) |
| Number of attributes | 742 time steps (Apr 2013-Apr 2016) |
| % missing values | 32% |

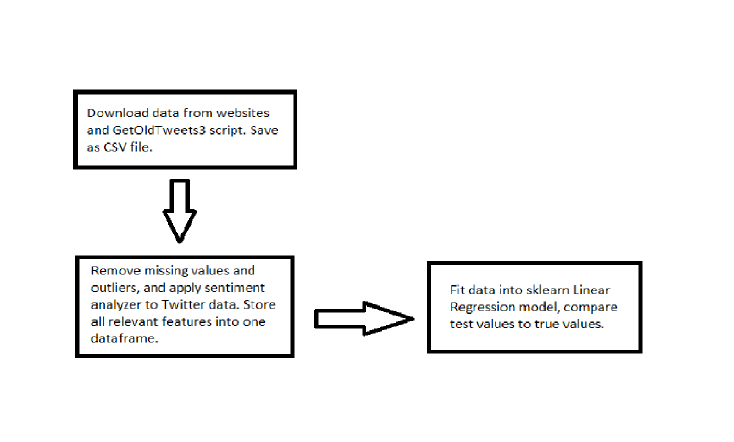
**Table 2**: Summary statistics of the raw data from yahoo.com, invsting.com, fred.stlouis.gov and twitter.com..

We then used Z-score transformation on all datasets to find outlier points and remove them from all datasets. For the twitter data collected by GetOldTweets3, each date has around 500 tweets. We created a sentiment analyzer that for each day’s worth of tweets, reads each tweet individually, sums up the total number of positive and negative words in the tweet, and adds the sum to a total for that day. We then divide the sum of positive/negative words by the total number of tweets for that day, which gives us a sentiment score for all tweets for any particular day. This score usually lies around -1 to 1, but is not limited to those bounds, and averages to around .05. The dictionary of positive and negative words were provided by the Opinion Lexicon on www.cs.uic.edu/~liub/FBS/sentiment-analysis.html.

Once all of the data was collected and all missing and outlier points were removed, and all dates were in order, we combined the datasets into one table and CSV file, with dates as the index and prices and sentiment scores as the attributes (742 rows, 5 columns). The predictor attributes of the table are the gold, oil, corn, dow, and twitter scores, and the target attribute is the closing stock price of Microsoft.

# METHODOLOGY

The method we used for our predictive model was a linear regression model provided by the scikit learn library. We used the first 500 datapoints as the training set, and the last 242 datapoints as the testing set.



The following is a brief summary of the code used for this project:

* getTweets.ipynb: this is the Jupyter notebook file that I wrote to collect data from Twitter. It uses the GetOldTweets3 library to collect the tweets and stores the tweets in a csv file with the date as the filename.
* twitterSentimentAnalysis.ipynb: this is the Jupyter notebook file that takes in all of the files created by getTweets.ipynb and applies sentiment analysis on it. It then stores the results in a csv file.
* preProcess.ipynb: this is the Jupyter notebook file to remove missing rows from all datasets and creates a final dataframe with the correct attributes and dates in order.
* Model.ipynb: this is the Jupyter notebook file that takes in the dataset created by preProcess.ipynb and applies it to a linear regression model.

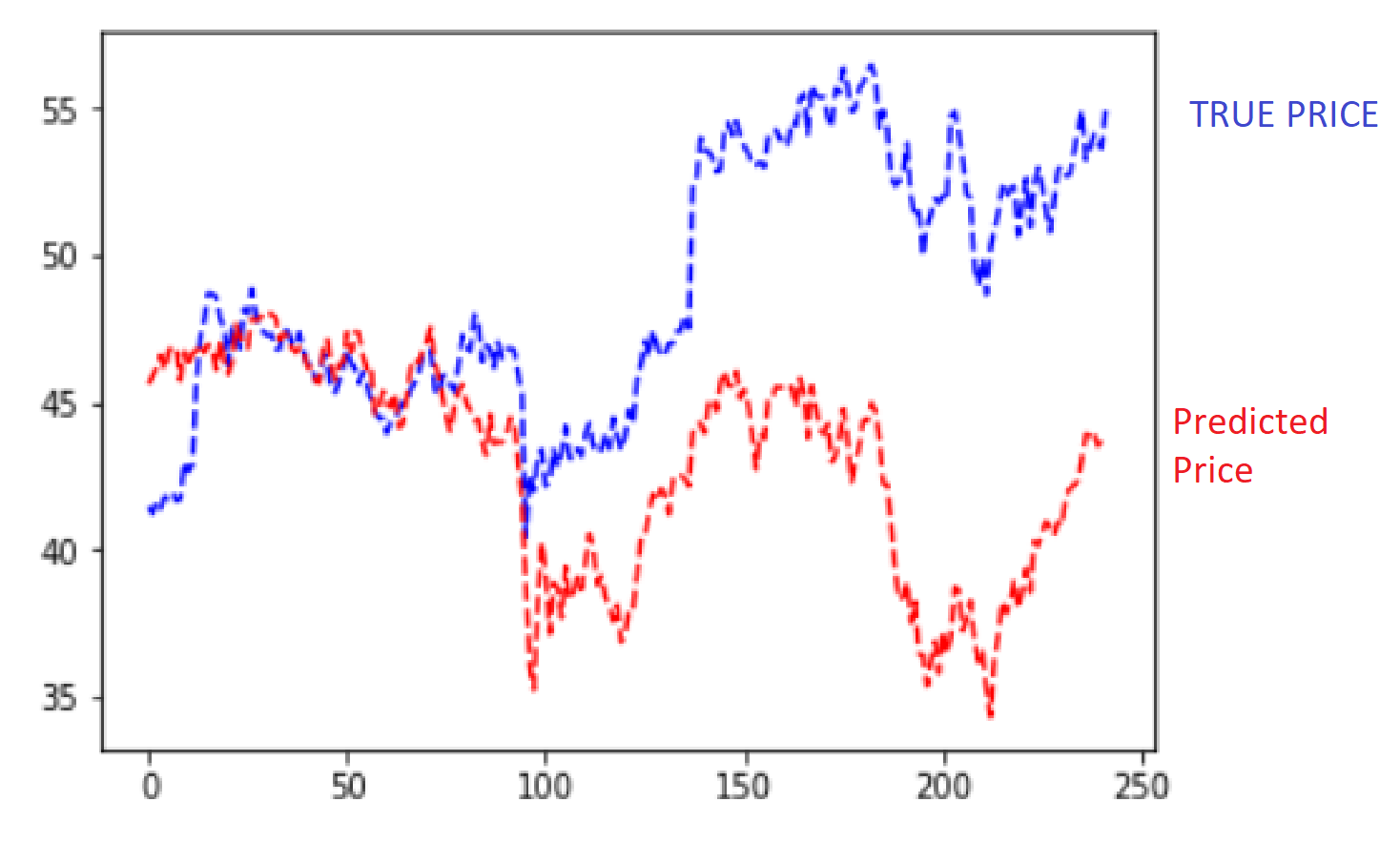
Twitter collection software can be found at <https://github.com/Mottl/GetOldTweets3>. The dictionary for positive and negative words can be found at <https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html>.

# EXPERIMENTAL EVALUATION

This section describes the experimental setup and results obtained.

## Experimental Setup

1. We used a Dell laptop with Windows 10 operating system to run all programs included in this project.
2. We used root mean square and R-Squared scores to evaluate the accuracy of our model. We also graphed the model’s resulting values and the true values.



## Experimental Results

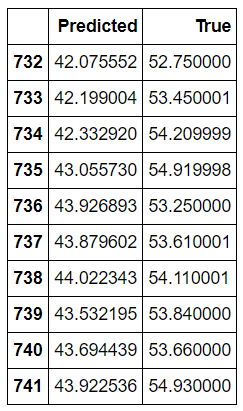


Figure 3: Table of last 10 values of predicted values and true values.

The results of our model were as follows:

* The R-squared value between the predicted results and the true results is -2.68
* The root mean squared error between the predicted results and the true results is 8.22.
* The coefficients for each attribute is as follows:

1. Corn: -0.0049
2. Dow: 0.0047
3. Gold: -0.0039
4. Oil: 0.0376
5. Twitter: -3.317
6. Intercept: -33.408

Looking at the coefficients for each attribute and considering the magnitude of the numbers in the dataset, it can be seen that the model is reliant most on the value of the Dow attribute, and the least reliant on the corn attribute.

With the results given by the model, it can be seen that the model is not useful for predicting the future price of stock, and is not even reliable enough to predict just an increase or decrease in future stock prices. This bad modeling is due to model underfitting, due to data that is not related to the target attribute. For future models, we plan to use better known stock indexing metrics as predictor attributes.

# CONCLUSIONS

Overall, no definitive conclusion can be made about whether or not tweets affect future stock prices due to the poor quality of our model.

# REFERENCES (at least 3 references)

1. “Yahoo Finance - Stock Market Live, Quotes, Business & Finance News.” *Yahoo! Finance*, Yahoo!, finance.yahoo.com/.
2. “Stock Market Quotes & Financial News.” *Investing.com*, www.investing.com/.
3. “Federal Reserve Economic Data: FRED: St. Louis Fed.” *FRED*, Federal Reserve Bank of St. Louis, fred.stlouisfed.org/.
4. <https://github.com/Mottl/GetOldTweets3>
5. “Twitter.” *Twitter*, Twitter, https://twitter.com/home.
6. Liu, Bing. *Opinion Mining, Sentiment Analysis, Opinion Extraction*, www.cs.uic.edu/~liub/FBS/sentiment-analysis.html.
7. Nitin Jindal and Bing Liu. "Identifying Comparative Sentences in Text Documents" Proceedings of the 29th Annual International ACM SIGIR Conference on Research & Development on Information Retrieval (SIGIR-06), Seattle 2006.
8. Bing Liu. Sentiment Analysis: Mining Opinions, Sentiments, and Emotions. Cambridge University Press, 2015.

Grading criteria

Note that the project accounts for 10% of your final grade. The project will be graded based on the following criteria:

1. Presentation - structure/organization and clarity of writing (including tables and figures).
2. Technical - Correctness and thoroughness of the analysis performed. What are the challenges faced and how well did you address them? How do you evaluate the performance of the method you'd applied to the data? How much detailed discussion you provide to explain the results you'd obtained (e.g., discussion about why the method works or didn't work on the data)?
3. Difficulty level - How large is the dataset used? How much effort you had to spend to collect, integrate, preprocess, and analyze the data? Are you implementing the project on a cluster or a single machine? What tools did you use (do you have to implement them or are you simply using existing libraries)?
4. Participation in the group project. How much did a team member contributes to the project.