Stock Price Prediction using open-domain news and social indicators

Tushar Gupta (tg2749)

Hitesh Agarwal (ha2598)

Kehao Guo (kg2937)

Abstract

In the age of social media, financial markets today are heavily influenced by public opinion. Digital media has exponentially increased the velocity of information flow, impact of which can be seen directly and in-directly on stock-prices and valuations of tech-giants. In the project, we aim to study the impact of signals like tweets and news articles on financial markets and build an automated system for aiding market decisions for investors.

1. Introduction

Big Data analytics plays a huge role in today's day and age due to the volume, velocity and variety of information generated each second. We aim to utilize tools specifically designed for Big Data analysis such as Pyspark, BiqQuery and Airflow to build a scalable data infrastructure & Flask for application development. For stock price prediction, we plan to experiment with various machine learning models to recommend the probable Close price for a company based on a variety of indicators. To go beyond, the traditional models readily available on the internet we aim to incorporate data from multiple sources like Tweets, Financial news and reddit as most of the systems rely solely on one of the input methods. Other improvements include, an interactive web-application which loads information about the company taking in data from multiple sources for easy visualization.

2. Background

Stock price prediction is an active research area in recent years. Better predictions over the stock prices lead to better returns in stocks. Therefore, significant efforts are put to build efficient and robust prediction models that can predict the future trend of a specific stock or the overall market. Research studies over the same topic, as well as events in the recent years provides evidences of a strong relationship between the new articles or tweets from top influencers about a company and its stock prices fluctuations. Following is discussion on previous researches on how future stock price can be predicted using historical data and sentiment on the news or twitter data about a particular company.

The interest in this topic was initiated by Bollen et al [1] in their research in 2010 where they attempted to predict the behaviour of the stock market by measuring the mood of people on Twitter. The authors considered the tweet data of all twitter users in 2008 and used the OpinionFinder and Google Profile of Mood States (GPOMS) algorithm to classify public sentiment into 6 categories, namely, Calm, Alert, Sure, Vital, Kind and Happy. They cross validated the resulting mood time series by comparing its ability to detect the public's response to the presidential elections and Thanksgiving day in 2008. Their results show a remarkable accuracy of nearly 87% in predicting the up and down changes in the closing values of Dow Jones Industrial Index (DJIA).

Mittal and Goel in their research [2] use twitter data to predict public mood and use the predicted mood and previous days' DJIA values to predict the stock market movements. They also propose a new cross validation method for financial data and obtain 75.56% accuracy using Self Organizing Fuzzy Neural Networks (SOFNN) on the Twitter feeds and DJIA values.

Nagar and Hahsler in their research [3] presented an automated text mining based approach to aggregate news stories from various sources and create a News Corpus. The Corpus is filtered down to relevant sentences and analyzed using NLP. A sentiment metric, called NewsSentiment, utilizing the count of positive and negative polarity words is proposed as a measure of the sentiment of the overall news corpus. They have used various open source packages and tools to develop the news collection and aggregation engine as well as the sentiment evaluation engine.

In [4], Joshi et al created three different classification models (RF, SVM and Naïve Bayes) which depict the polarity of news articles being positive or negative, and then predict the price of the stock based on the news sentiment score.

3. Current Progress

Our team is currently working in parallel on the following 2 tasks. In the first one, we are focusing on collecting data for multiple tech companies and storing it in a more processed format for easy model application. While in the second one, we are working to build a browser based application.

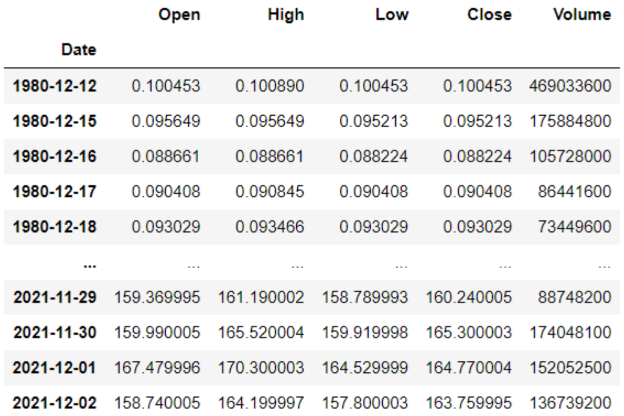
3.1. Stock Price Data Collection: We have currently collected data for the top-5 tech giants in US as these generate a huge amount of information on social media because of their popularity in the masses. Specifically, these are

- Apple (APPL)
- Google (GOOG)
- Microsoft (MSFT)
- Tesla (TSLA)
- Amazon (AMZN)

We recently changed our data source to the Yahoo Finance API same as the one mentioned in the HW4 due to the variety of options it provides. For collecting the data we use the max period parameter with a **1d** granularity to fetch all possible data. Statistics on the data and stock schema can be found below,

Company Name	Data Range	Num Rows
Microsoft	Mar 1986 – YTD	9007
Apple	Dec 1980 – YTD	10333
Google	Aug 2004 - YTD	4355
Tesla	June2010 – YTD	2880
Amazon	May 1997 – YTD	6181

Schema of the data can be found below:



Date	Open	High	Low	Close	Volume
1980-12-12	0.100453	0.100890	0.100453	0.100453	469033600
1980-12-15	0.095649	0.095649	0.095213	0.095213	175884800
1980-12-16	0.088661	0.088661	0.088224	0.088224	105728000
1980-12-17	0.090408	0.090845	0.090408	0.090408	86441600
1980-12-18	0.093029	0.093466	0.093029	0.093029	73449600
...	...	...	...	...	...
2021-11-29	159.369995	161.190002	158.789993	160.240005	88748200
2021-11-30	159.990005	165.520004	159.919998	165.300003	174048100
2021-12-01	167.479996	170.300003	164.529999	164.770004	152052500
2021-12-02	158.740005	164.199997	157.800003	163.759995	136739200

Figure 1

Keeping in mind the cost of cloud resources and the size of the data, our current model creation is being done on local computer. We plan to shift the entire data to BigQuery for the final application.

3.2 Tweet Data Collection

So far, we have collected Tweets for the above-mentioned companies from the Kaggle data set [5]. The dataset, as a part of the paper published in the 2020 IEEE International Conference on Big Data under the 6th Special Session on Intelligent Data Mining track, is created to determine possible speculators and influencers in a stock market. The dataset contains tweets from 2015 to 2020 for each company. It contains over 3 million unique tweets with their information such as tweet id, author of the tweet, postdate, the text body of the tweet, and the number of comments, likes, and retweets of tweets matched with the related company. Tweets are collected from Twitter by a parsing script that is based on Selenium.

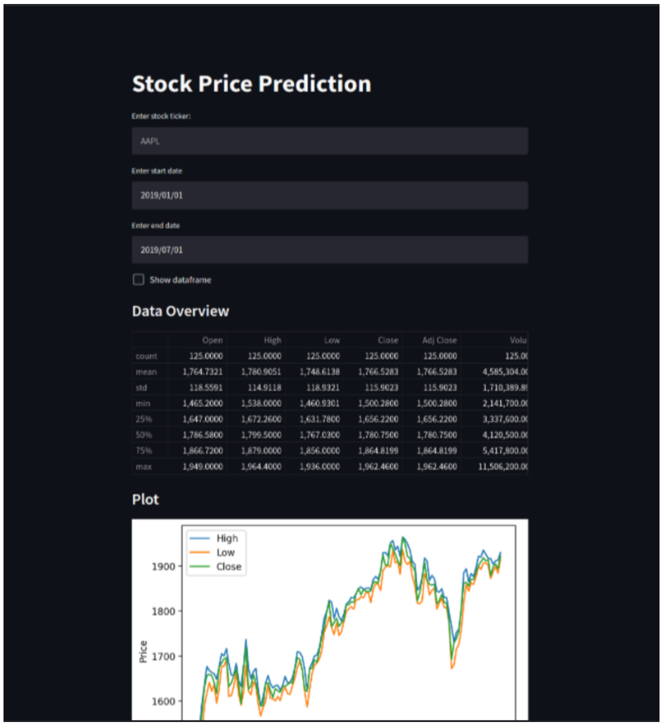
3.3 Financial New Data Collection

The financial news for the five companies are yet to be gathered from two different sources: Finhub.com and Eodhistoricaldata.com, which contain the past company news, sorted by date, type of news and certain tickers with the given parameters. Both the websites require API calls to get the data by passing the date and company ticker parameters. Since there are limitations with the number of tweets that the API response provides, we are still in the process of collecting the news data for all the companies.

3.4 Web Application:

The application is an interactive user interface implemented in Streamlit framework, which is highly integrated with python libraries and provides useful APIs for rendering dynamic web pages to perform data-related tasks and demonstrate predictions and visualization to users.

The UI would update its content whenever a user makes changes to inputs of the system. When an update takes place, the system would fetch historical price data of the given stock for the past N days and fetch information including tweets and financial news in the last **N** days. It would then use the fetched data to predict Close price. Finally, the application would provide visualizations and insight into the original data and predicted stock data.
The Figure 2 shows a tentative view into the design and features of the web-app.



Stock Price Prediction

Enter stock ticker:

AAPL

Enter start date

2019/01/01

Enter end date

2019/07/01

□ Show dataframe

Data Overview

	Open	High	Low	Close	Adj Close	Volu
count	125.0000	125.0000	125.0000	125.0000	125.0000	125.0
mean	1,764.7321	1,780.9051	1,748.6138	1,766.5283	1,766.5283	4,585,304.0
std	118.5591	114.9118	118.9321	115.9023	115.9023	1,710,389.8
min	1,465.2000	1,538.0000	1,460.9301	1,500.2800	1,500.2800	2,141,700.0
25%	1,647.0000	1,672.2600	1,631.7800	1,656.2200	1,656.2200	3,337,600.0
50%	1,786.5800	1,799.5000	1,767.0300	1,780.7500	1,780.7500	4,120,500.0
75%	1,866.7200	1,879.0000	1,856.0000	1,864.8199	1,864.8199	5,417,800.0
max	1,949.0000	1,964.4000	1,936.0000	1,962.4600	1,962.4600	11,506,200.0

Plot

Figure 2

3.5 Machine learning models

Using the collected data on stock prices, we wanted to see if it is possible to predict the Close price of the stocks based on simply stock historical information like High, Low, Open, Volume, Close etc., similar to what we do in the Assignment 4. Using the notion, we have performed the following analysis.

3.5.1 Close Price Prediction based on stock information only

3.5.1.1 Data Pre-Processing

- The below analysis has been performed solely for Tesla at the moment
- Pre-processing includes removing duplicates, dropping NULL values
- We experimented with multiple lag values of 5, 10,20 days from which 10 days give the best possible results.
- Using the past 10 days information a rolling mean is calculated for each column in the data
- The current Close price is taken into a new data frame and shifted by 1 day to concatenate with the rolling mean information. By this, we have current Close price and the corresponding past 10 days rolling mean.
- Data is split into train and test in a 70/30 ratio
- For the analysis, we have only taken past 2-year information into account to remove any ambiguous financial events

3.5.1.2 Statistical Models

a. Linear Regression: Figure 3 below depicts a graph between the real and predicted values. We can see they indeed follow a relationship.

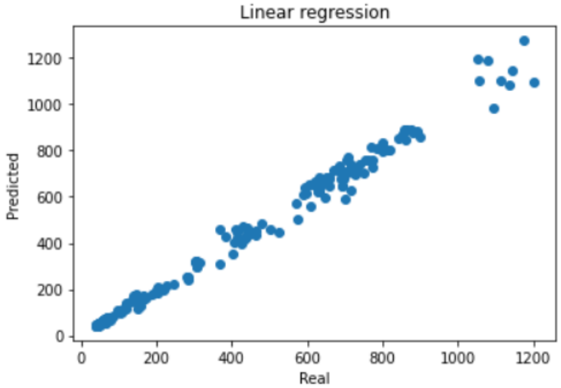


Figure 3

Mean Absolute Error: 10.91

b. Random Forest: We now move to a more advanced algorithm which is also able to select the most optimum features from the dataset thus helping to remove any kind of overfitting if any. We currently do the analysis in python's sklearn as it contains the library RandomizedSearchCV for parameter search. We vary the number of features from 1, 5 and do a 5-fold cross validation. As a result, the model contains 4 features. The relationship between real and predicted can be seen below:

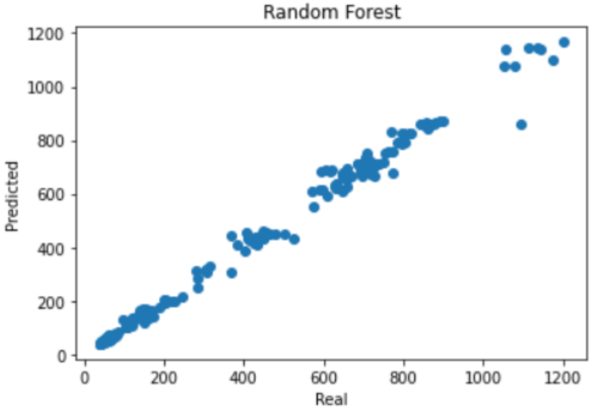


Figure 4

Mean Absolute Error: 9.42

The model shows an improvement over Linear Regression which was expected.

c. Gradient Boosted Trees: Boosted trees are a type of ensemble methods which use of a combination of weak-learners: Decision trees to predict the target variable. They are generally known to perform better than Random forests. The resulting predictions can be visualized below:

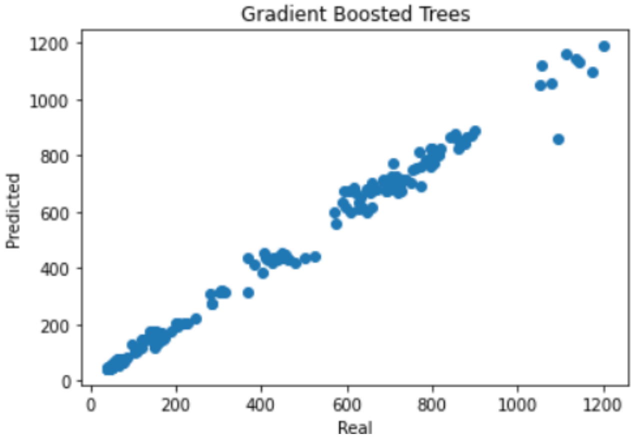


Figure 5

We again use the RandomizedSearchCV to determine the best hyperparameters to the model. The best model has 2 features and performs worse that Random Forest models.
Mean absolute error: 10.26

3.5.1.3 Deep Learning model (LSTM)
LSTM (Long-Short term memory networks) are recent advancements in Deep learning which are adept at dealing with sequential information and have been used in variety of use cases. We use the same here to check if Close can be predicted using its past values.
Data Preprocessing:

- For the task, we first scaled the Close Price using sklearn's MinMaxScaler in the range of (0,1)
- The data used is APPL stock data for it's entire period.
- Here, we experimented with a feeding in the past 60 days of information to the current day as LSTM are able to remember information for a longer period of time
- The first 95% information is used for training the LSTM model in keras and the rest for validation

Model:
We use a 2 layer LSTM with 128 and 64 size of hidden units. After that we add a full-connected layer of size 25 which finally gives a single output. Mean Squared error is used as a metric to improve and adam optimizer to optimize.

In 1 epoch the loss decreases to 9*(10^-5)

Figure 6 in Appendix shows the predictions along with the validation set.

4. Planned Experiments

The next experiments that we have planned for us are the following:

4.1 Train sentiment analysis model on the tweet and financial data information

4.2 Aggregate the sentiments as per date and insert it into the above statistical models for improving predictions

4.3 Increase the number of features in the LSTM feature vector for improving predictions

4.4 Analyze the expected returns and risk value for each company to give a more holistic view to investors.

4.5 Correlation analysis of stock prices between companies

4.6 Convert the final model pipeline into Airflow for daily updates to predictions and other statistics

4.6 Integrate model with the web-application

5. References

[1] J. Bollen and H. Mao. Twitter mood as a stock market predictor. IEEE Computer, 44(10):91–94
https://doi.org/10.1016/j.jocs.2010.12.007

[2] A. Mittal and A. Goel Stock Prediction Using Twitter Sentiment Analysis -
https://cs229.stanford.edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis.pdf

[3] Anurag Nagar, Michael Hahsler, Using Text and Data Mining Techniques to extract Stock Market Sentiment from Live News Streams, IPCSIT vol. XX (2012) IACSIT Press, Singapore

[4] Stock Trend Prediction using news sentiment analysis – Joshi et al.
https://arxiv.org/pdf/1607.01958.pdf

[h
[5] Tweets about the Top Companies from 2015 to 2020
https://www.kaggle.com/omermetinn/tweets-about-the-top-companies-from-2015-to-2020?select=Company_Tweet.csv

6. Appendix

1. Predictions using LSTM for Apple

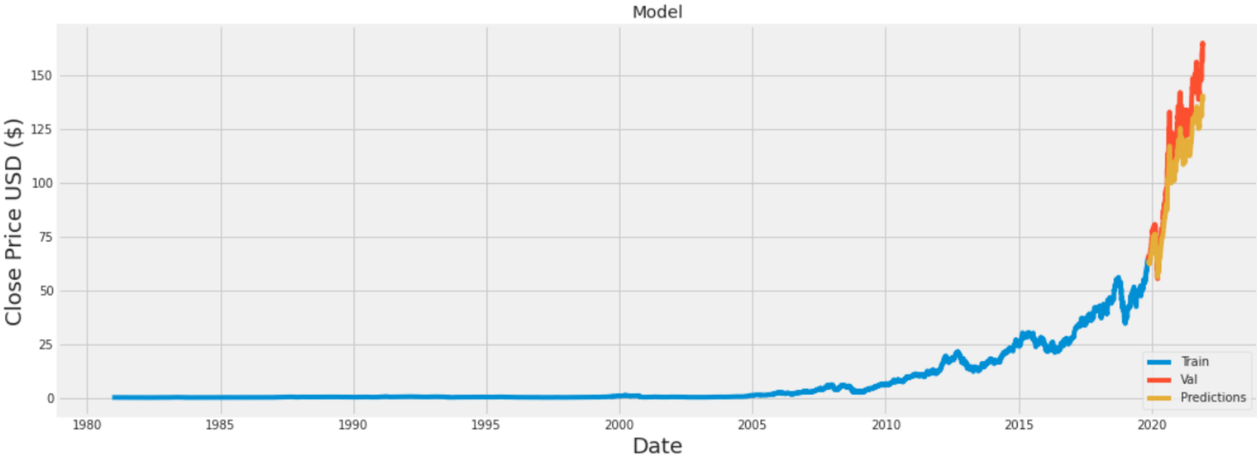


Figure 6