Stock Predictions Using Data from News Sources, Financial Resources, and Historical Stock Prices

By: Dao Vang

dao-v.github.io

github.com/dao-v/Stock_Predictions

Purpose & Problem

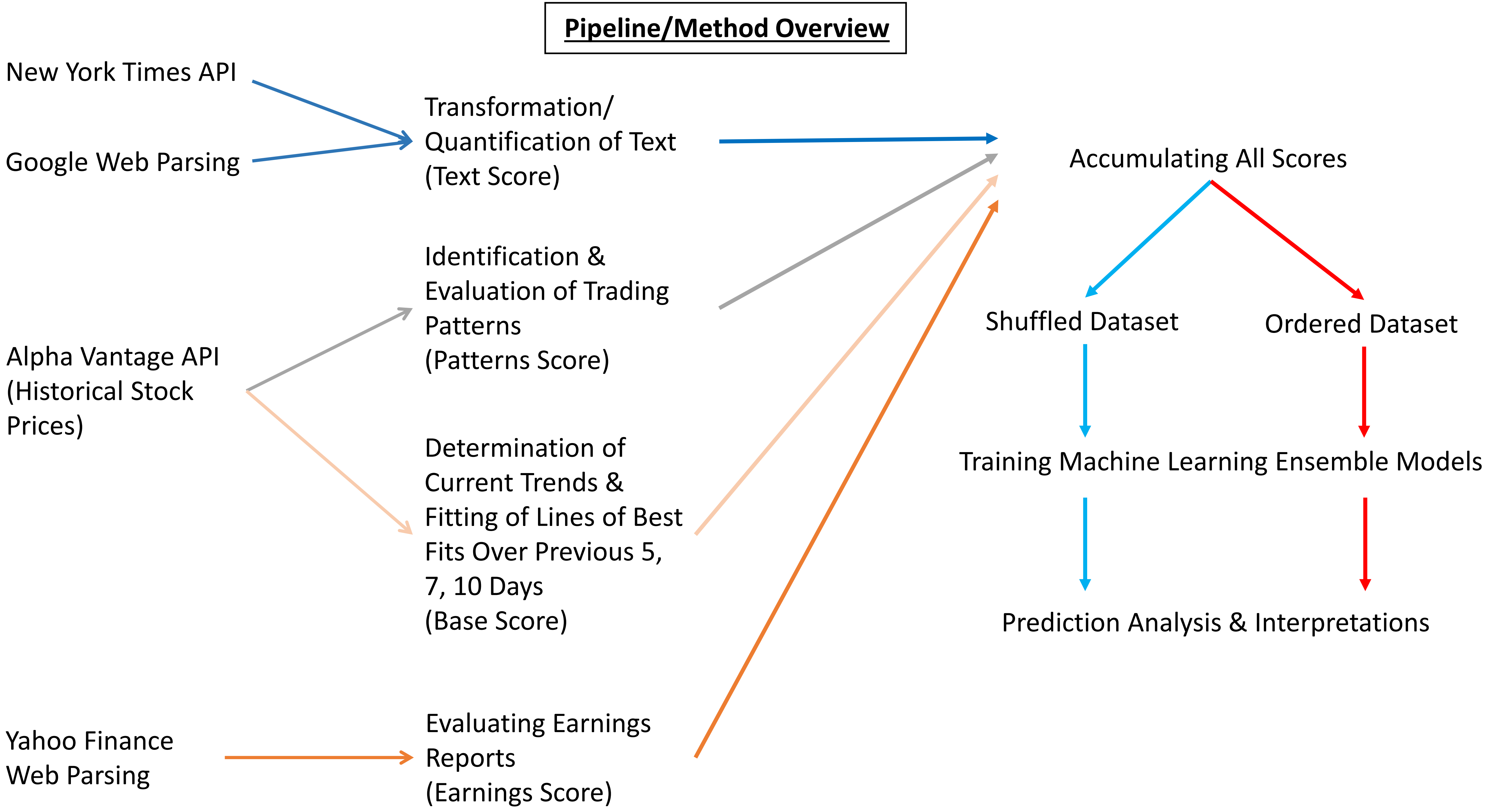
- To create machine learning models or methods that will predict the price movement of individual stocks
- These models will incorporate data from news outlets, historical stock price data, earnings reports, and trading patterns.
- The problem with creating a predictive model is the inability to accurately quantify quantitative data, such as the words in a news article

Features/Variables

- The four previously mentioned variables attempts to cover different areas that could affect the movement of stocks:
  - Media
  - Historical price data
  - Company’s performance
  - Traders’ mentality & behaviors

Pipeline/Method Overview

1. Data Extraction
   - Extract all data related to the variables
2. Data Manipulation & Data Wrangling
   - Manipulate and transform the data to a usable form
3. Statistical Analysis
   - Interpret the data to gather insight on performance of some variables
4. Machine Learning with Ensemble Models (Sci-Kit Learn)
   - Train the machine learning models using Random Forest, AdaBoost, and Gradient Boosting
5. Prediction Analysis
   - Analyze of results and performance of all the models


Pipeline/Method Overview
New York Times API
Google Web Parsing
Transformation/ Quantification of Text (Text Score)
Alpha Vantage API (Historical Stock Prices)
Identification & Evaluation of Trading Patterns (Patterns Score)
Determination of Current Trends & Fitting of Lines of Best Fits Over Previous 5, 7, 10 Days (Base Score)
Yahoo Finance Web Parsing
Evaluating Earnings Reports (Earnings Score)
Accumulating All Scores
Shuffled Dataset
Ordered Dataset
Training Machine Learning Ensemble Models
Prediction Analysis & Interpretations

Step 1: Data Extraction

<u>Media:</u>

- New York Times API
- Google Search web parsing

<u>Historical Prices:</u>

- Alpha Vantage API
    - 20 years of stock prices

<u>Earnings Reports</u>

- Yahoo Finance web parsing

<u>Trading Patterns:</u>

- Found within historical prices:
    - The interception between the 30-day and 60-day moving averages

Step 2: Data Manipulation & Data Wrangling

Media:
- Extract every word from all articles
- Give two different values to each word:
  - One for if the word is verb-like
  - The second if the word is relevant importance
- Evaluate articles using valuated vocabulary to provide the **Text Score**

Historical Prices:
- Recognize current trend based on previous 1-day moving averages through the previous 5, 7, and 10 days using linear equations (lines of best fit)
- Calculate an estimated prediction value based on the average slopes of the linear equations
- Provide a **Base Score** according to trend

Earnings Reports
- Obtain dates for each quarter
- Extract estimated earnings per share (EPS) values provided by Yahoo Finance
- Provide an **Earnings Score** based on the direction (+/-) of the estimated EPS

Trading Patterns:
- Analyze frequency of trading pattern being followed by stock movements
- Evaluate the dates where the trading pattern is found with a **Patterns Score** if the price moves in accordance to the expected pattern movement

Step 3: Statistical Analysis of Trading Pattern

- According to the obtained statistics, the trading pattern was followed ~50-65% of the time it appeared
- The trading pattern does not have any logical/real-world significance and is simply a pattern sought out by day and swing traders
- However, even with low statistics, the Trading Score was used in training the machine learning models

Step 4: Combining and Splitting the Dataset

- Many columns of information and calculations were all accumulated into one final dataset that contained all the 4 scores from the previous steps
- The final dataset was labeled by using the next day closing price
- Splitting the dataset for training and testing dataset were done in 2 ways:
  - Shuffled
    - This is the default setting for Sci-Kit Learn
  - Ordered
    - This was done to include the newest data in the training set (to train on the effects of the coronavirus)

Step 5: Machine Learning with Ensemble

Shuffled Dataset	Ordered Dataset
Random Forest with Shuffled Dataset	Random Forest with Ordered Dataset Random Forest with Ordered Dataset & No Bootstrapping
AdaBoost with Shuffled Dataset	AdaBoost with Ordered Dataset
Gradient Boosting with Shuffled Dataset	Gradient Boosting with Ordered Dataset

Step 5: Machine Learning with Ensemble

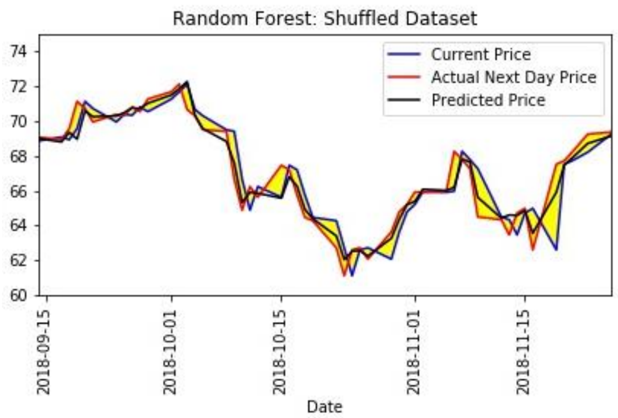
- All ensemble methods were trained with a tree range of 64-128 to find the best number of trees to use as a model parameter:
  - To be considered the best, the model had to have the highest R^2 value with a cross-validation method of 5-folds
- All models were then used to predict on the entire dataset after the best trees were found to be used for further analysis

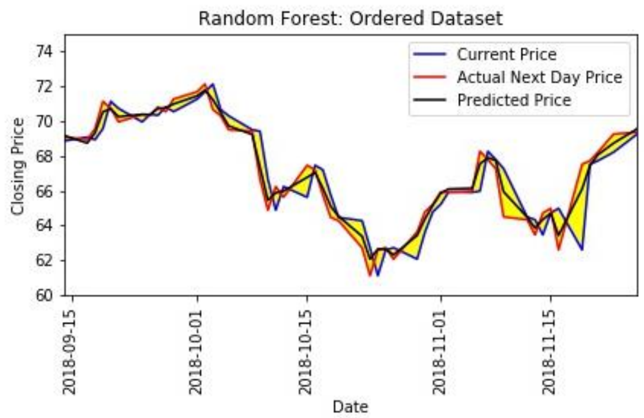
Step 6: Prediction Analysis

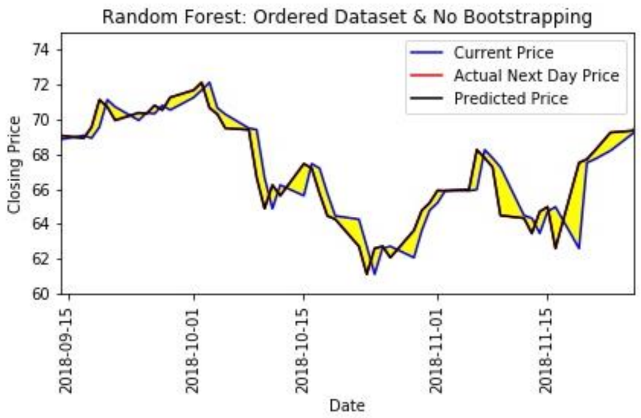
- Analysis on the performance of the machine learning models was done to examine:
  - The distance between the labels and predicted values
  - The direction of the predicted values against the direction of the labels from the current closing price (frequency)

Step 6: Random Forest

- The goal is for the predictions (black) to lie inside the yellow regions or near the red line
- The best results (visually) was by the ordered dataset and no bootstrapping
    - This could be due most of the dataset being seen during training of the model
    - The price predictions were almost exactly as actual price as there was no visible red line in the graph
- The second-best model was with the shuffled dataset

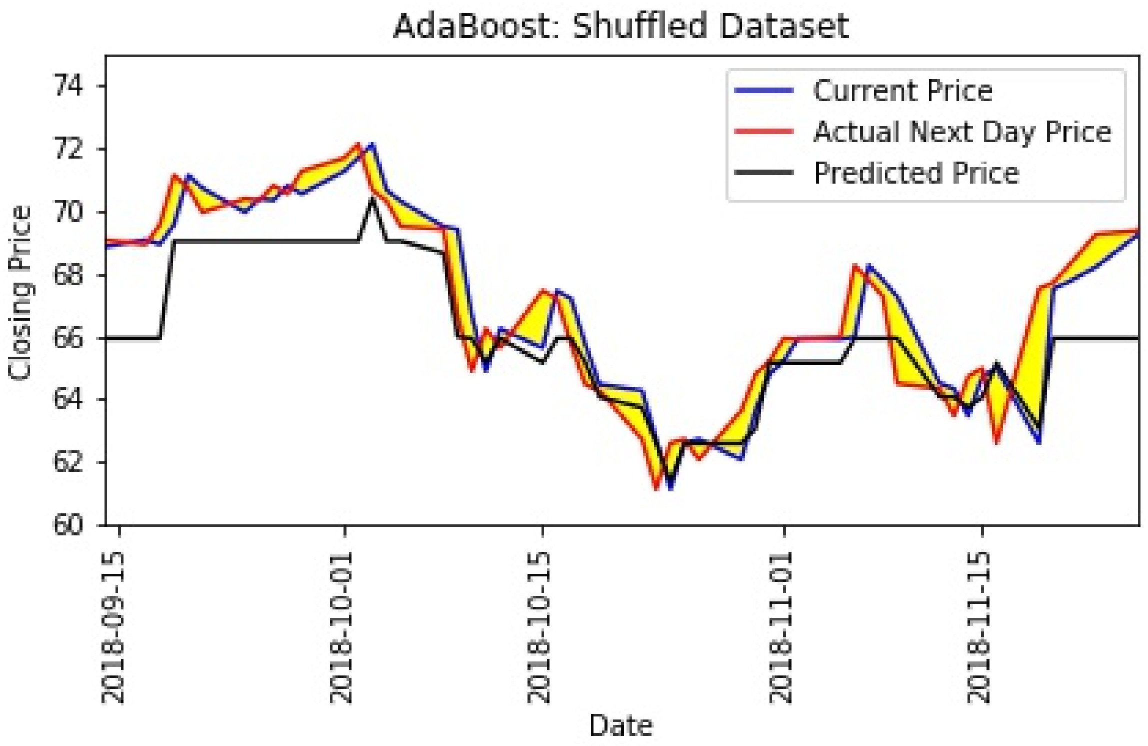


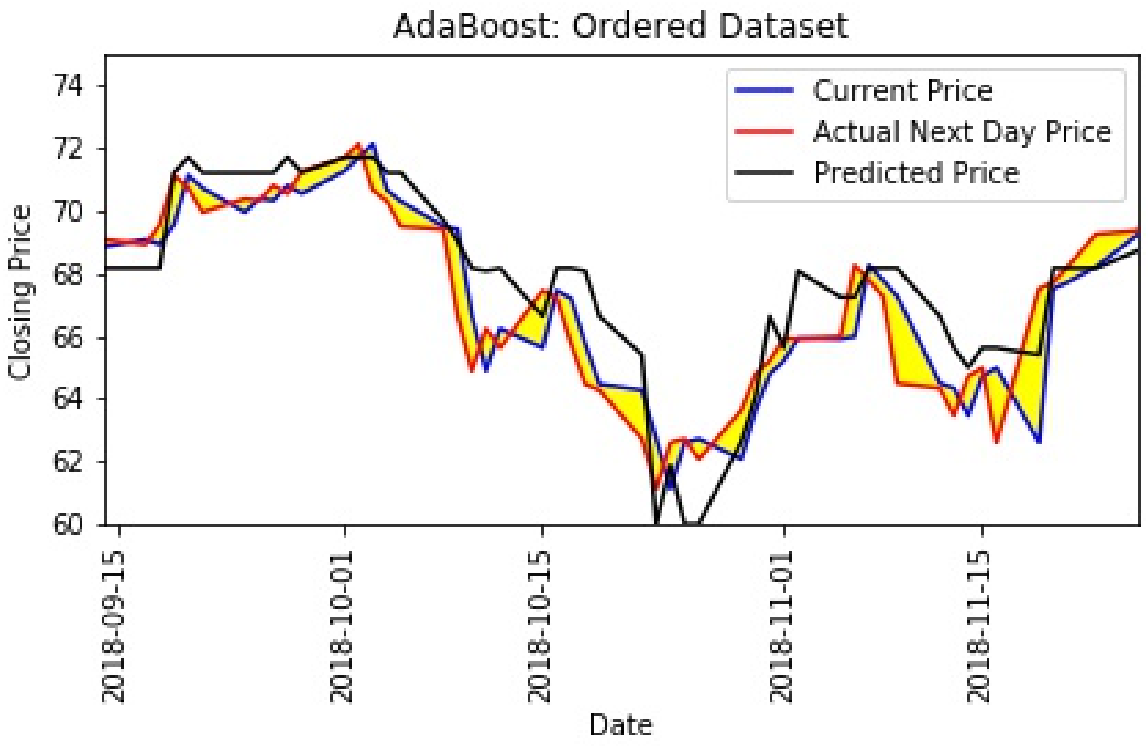


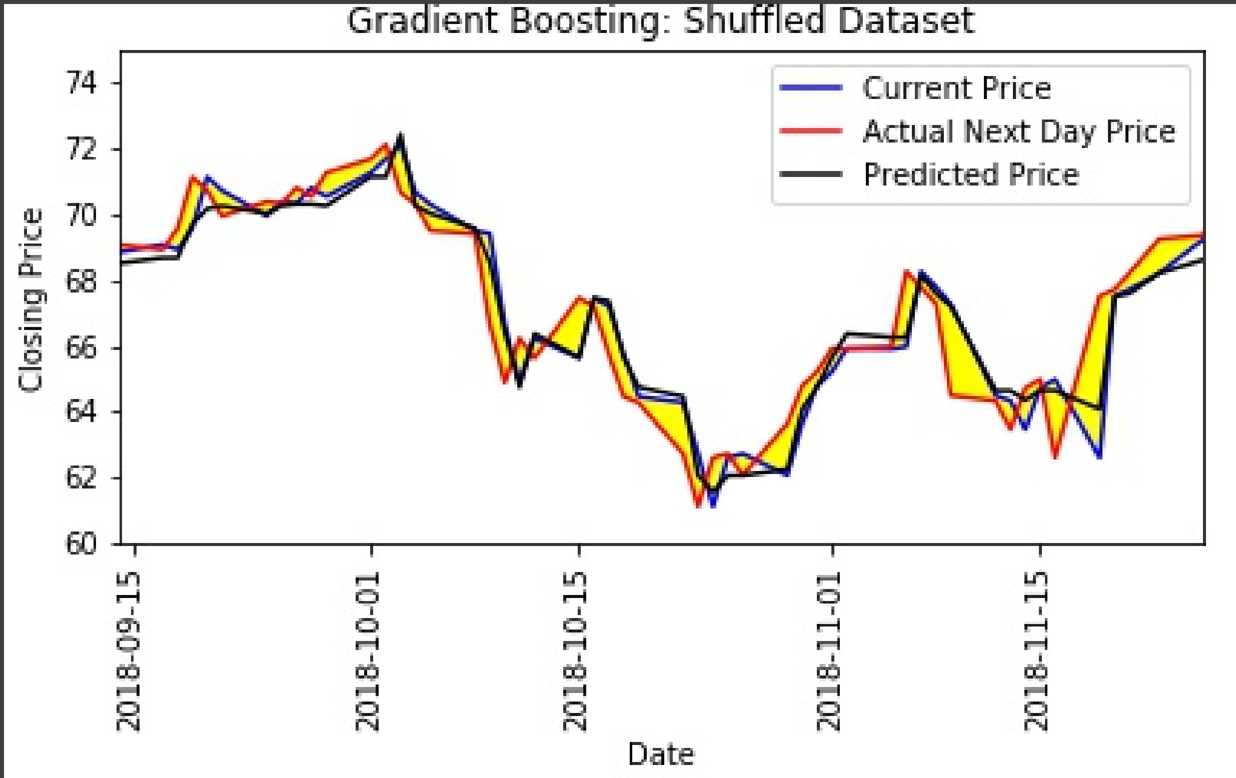


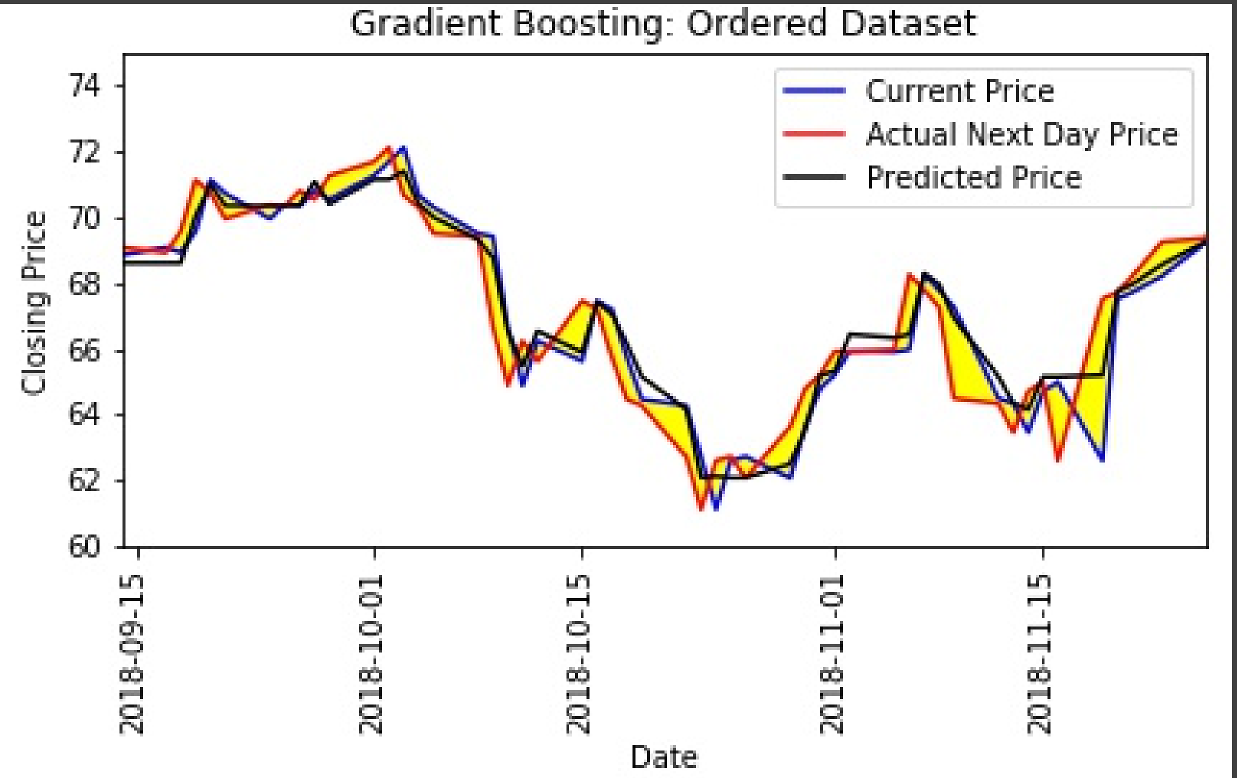
Step 6: AdaBoost

- Both models did not perform well as the predictions were over-exaggerated on in both directions









Step 6: Gradient Boosting

- These two models performed better than the AdaBoost models but does not beat the random forest models

Step 7: Interpretations

- Although the R^2 was used to determine the best parameters for the models, after examining the statistics of the prediction results, the R^2 did not provide useful information on the performance of the models
- The frequency of the model moving in the same direction as the label was the better determinant on the performance of the models:
  - If: Label – Current Price = +
    - Want: Prediction – Current Price = +
  - Using this statistic to determine the models' performance resulted in the Random Forest models providing the most reliable predictions with **high accuracy of predicting the direction of the label**

Conclusion

Media:

- Many of the models did not feature select for any of the New York Times and Google Text Scores
- Due to the oddity of the New York Times API, relevant information to companies was rare to obtain

Historical Prices:

- Most of the feature selection was highest for the closing and high prices features (50%+)
- The estimated predicted score using the averaged slopes from linear equations was given some importance (10-15%)

Earnings Reports

- Because there weren't many earnings dates as there were other variables, the Earnings Score was given a very small weight

Trading Patterns:

- The Patterns Score was given a small weight as well but since the pattern itself had a nearly ~50% change of it following the pattern, it was not too important in the end
- The moving averages of 30- and 60-days were selected