# US Stock Price Prediction

## Susan Wu, Shuoqi Zhang

**Abstraction**

Recent advances in machine learning algorithms have created new potentials for financial investors to optimize their decision-making on investment choices and strategies. Hence, the purpose of this project is to predict the stock price performance of firms in the US stock market. In this project, we developed a best-fitted linear regression model by using several big data techniques to explore selected datasets. We examined our datasets in the Quadratic, Lasso, Quantile, and Random Forest regression models. The performance of the Random Forest model on our datasets is most reasonable and effective, and this model finally gives predictions on the annual mean of the closing price of firms with MSE lower than 11. In addition to the model construction itself, model limitations and further possible improvements are stated.

## 1. Background and Introduction

The financial market is growing more and more complex with an increasing number of large companies and fluctuations. Among global financial markets, the US stock market is one of the most focused stock markets in the worldwide financial field, as it has a substantial global influence on other financial markets including bonds, commodities, and so on. From the financial aspect, changes of the stock prices of the larger companies significantly impact the stock prices of other medium and small companies, so by looking at the larger companies' stock performance, it reflects the overall trend and performance of the US stock market as a whole. Therefore, investors and stockholders need to pay close attention to the performance of large firms' stocks in the US stock market. To maximize their returns in stock tradings, accurate data-driven prediction models are essential for optimum decision-making. From this perspective, the dependent variable in this project is the next year's annual mean of the closing stock price of each S&P 500 firm.

### 1.1 Datasets Description

According to what we want to focus on, we have chosen 2 data sets, "200+ financial indicators of US stocks(2014-2018)", and "S&P 500 stock data". The first dataset includes features that reflect the financial status of each firm, and the data are from pre-covid19 period. One of the reasons to use this data set is that the individual company's financial status largely impacts the stock price of that firm from the financial aspect. When considering more recent dated data, covid19 as a pandemic is a large concern. The global economy and financial markets have been negatively and unexpectedly affected during this special period. The situation is considered to be abnormal. Then generalization of data from this period is considered to be low. Hence, we decided to focus on the 2014-2018 data. In the second dataset, columns are different types of stock prices of each S&P 500 firm from 2013-2017. The US equity markets can be represented by the S&P index. (Rodriguez-Nieto, V.Mollick, 2). Thus, the range of target firms narrowed to firms that are listed in the S&P index. Stock prices recorded in each column are daily information. Since the first

dataset we selected contains annual information for each firm, we chose the closing price columns, and took the average of the closing price over a year of each firm as our dependent variable to better match the information in the same time spans.

It's worth mentioning that, in the dataset "200+ financial indicators of US stocks(2014-2018)", there are only 4 nominal variables among all the 225 features. The rest of features are all continuous variables, however, the scales of those features vary significantly across the whole dataset, which should be paid attention in the following data processing. For the interpretation, these variables are all elements that describe the financial status of a firm, such as "Revenue", "R&D Expense", and so on.

## 2. Data Cleaning and Processing

### 2.1 S&P 500 US Company Stock dataset

As we discussed, we selected two datasets on which data cleaning techniques were applied. The first dataset includes information of the S&P 500 US company's daily stock price (open, close, high, low, and volume). Firstly, we checked the missing values for each column and found out that there were three columns containing missing values. SInce the total number of missing values in those columns are less than 10, we simply dropped those rows. After that, we checked the outliers on each company basis and removed the outliers that are excess the 99% confidence interval of the mean in each column. We didn't use the IQR method to remove the outliers because in that case, the more than 10% data points would be removed, which might lead to information deficiency. In order to have more possible features in future model fittings, we also constructed monthly average, maximum, and minimum values of each feature for each company.

### 2.2 Financial Indicators of US stocks (2014 - 2018)

We first combined the datasets from 2014 to 2018 together. Then the companies that don't have five years' data are removed, since we chose to use data from 5 years to make the prediction. After that, we checked the missing value distribution of each column. We dropped the columns that had more than 75% of data that were missing. After that, we detected the missing value locations, then replaced those spots by the average value of each column grouping by different sectors, such as financial services, healthcare, technology, etc. We didn't replace them by 0 or the overall average value of the entire column, since it gave a better model prediction accuracy. We observed and dropped one duplicated column that has a different feature name format but the same content compared to one of the columns in the dataset. This duplicated column pair is "Cash per Share" and "cashPerShare".

After all the data cleaning processes, we performed the inner join on two cleaned datasets by matching years and company names. Our final dataset has 2247 rows and 322 columns. Then we constructed our training, validation, and test sets. For the training and validation set, we used the data from 2014 to 2016 for our features, and the data from 2015 to 2017 for the predicted variable, annual mean of the close price. We randomly selected the 70% of combined datasets to be the training set and 30% to be the validation set. Then we chose the financial indicator dataset containing the information from 2017 (with 2018 average close price as prediction y) as the test dataset.

## 3. Feature Selection and Transformation

Since we have more than 300 columns in our dataset, we performed several different feature selection techniques that were chosen according to the characteristics of the data in our dataset. We checked the collinearity between features, the column properties. We also used some automatic feature selection tools such as stepwise regression. In addition to that, we used the visualizations to support our feature selection and feature transformation decisions.

### 3.1 Columns with 0 values and low variance

Before checking the correlation of the features, we first removed the features with limited information. We removed the features with over 30% of the values being 0, after we comprehended definitions of those features and ensured those 0 values were not meaningful. Then we detected the features with 0 variance, because the "0 variance" means a feature won't be informative and useful in the prediction.

### 3.2 High Collinearity

#### 3.2.1 Feature Correlation

High collinearity between features can highly affect the model performance. The grouped high correlated features keep the model complex and reduce the precision of estimated coefficients. We calculated the correlation values across all the pairs of features and made a correlation graph of some features to demonstrate. As we can see from **Figure 3.2.1,** we noticed that there existed features that are highly correlated with many other features. After examining the dataset, we found 80 columns that are highly correlated ( correlation value bigger than 0.5) with more than 10 other features. Then we dropped these 80 columns.

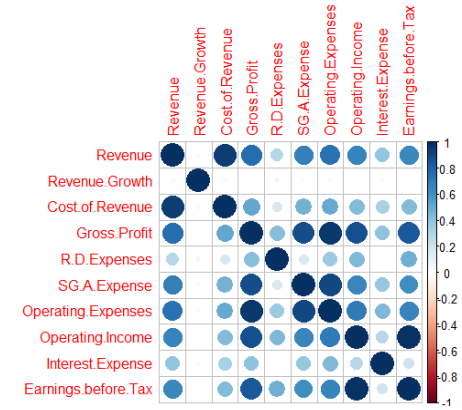


**Figure 3.2.1**, Correlation

#### 3.2.2 Correlation between Features and the Dependent Variable

Then, for the remaining columns, we examined the correlation between them and the dependent variable, yearly average closing price, to narrow down the number of features again. The remaining variables all didn't have a theoretical high correlation with the dependent variable (>0.5), so we kept the variables that are comparatively more related to y (>0.3) as the selected features.

### 3.3 Stepwise Selection

The first automatic feature selection method we used is the stepwise regression selection. We set the function to do both forward and backward selection to select the best model evaluated by the $R^2$ value (coefficient of determination). We extracted the features with the highest $R^2$ value through the stepwise selection. We examined the result by calculating the MSE and got an extreme large MSE, which is over 10000. It should be mentioned that the stepwise selection only removed 6 features from our dataset.

### 3.4 Lasso Modeling with Feature Selection

Another common way to do feature selection is to use the Lasso Regression to shrink as many coefficients to be 0 as possible and remove those features with 0 coefficients. We applied the Lasso Regression on our training data, but similar to the stepwise selection, it only removed 5 features and obtained a large MSE value.

### 3.5 Gini Impurity Score Selection

We also did the selection of features based on the importance of each variable. We processed this using the RandomForestRegressor in python, which provided us with the importance of the features by calculating the Gini impurity value. When we applied the RandomForestRegressor, we used 5-fold cross validation to choose the best model. We extracted the part of the variables according to the importance of each variable. We can see from **Figure 3.5.1**.

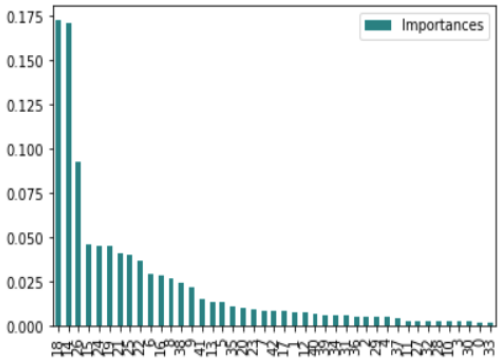


**Figure 3.5.1**, Feature Importance

We selected several different proportions of the features ordered by decreasing importance. Those selected data were fitted in different regression models, then compared them using the training MSE. MSE decreased largely to around 8000, but the MSE was still significantly high.

### 3.6 Feature Normalization

After viewing the performance of selected features from feature selections in different models, we realized that we need to look deeper into variable properties to decide if we need to do some feature transformations and select features instead of only depending on techniques.

As it's shown in **Figure 3.6.1** and **Figure 3.6.2**, some of the features have a very wide range of distribution which means that the data varies a lot. Besides, the scales of data differ significantly across columns. For example, the feature, "Revenue", has a minimum value of over 1000000, but the feature, "Operating Cash Flow", has the largest value less than 2000.

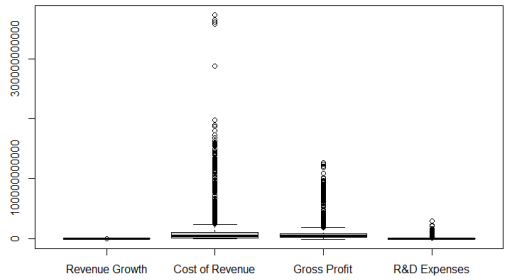


**Figure 3.6.1**, Boxplot of Four Variables

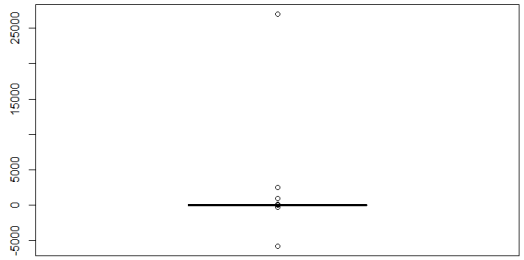


**Figure 3.6.2**, Boxplot of Operation Cash Flow Per Share

In the Gini Impurity Score selection, the feature that has an original large value will automatically have higher importance and impact in the prediction. The features that were selected from the Gini Impurity score selection, stepwise selection, and Lasso Regression all contained columns that had a certain amount of

extreme large values. That's why we decided to do the data transformation that normalizes parts of our data. Rescaling each feature having a very large range, and standardizing different features aimed to reduce the huge value variation impact from the features that hold extremely large values. We mainly used two different normalization techniques, $l_2$ normalization, and standardization with a formula: $\frac{x-\mu}{\sigma}$. We then applied these different normalization methods in models below.

After the normalization, we applied the Lasso Regression selection and Gini Impurity importance selection again. With different normalization methods, the results of the feature selection are different. Lasso Regression encountered all-0 coefficient situations, because the values are all too small after a certain normalization process. Then we selected the features mostly depending on the correlation, Gini impurity value, and the theoretical definition of the variables.

After the feature selection, we tried three different combinations of normalization methods to check the accuracy of model fittings. The first combination is only normalizing the columns with huge values to reduce the effect among different columns; the second combination is to normalize the columns with huge values using the $l_2$ norm and normalize other variables containing smaller values using the standardization; the third combination is to normalize all the data using the $l_2$ normalization. Along with this process, we tried the prediction with both normalized y values and original y values respectively. We applied the three types of normalized data into the well-performed models to make a comparison among them. More details will be discussed in the model section.

### 3.7 Polynomial Transformation

We visualized the relations between features and the dependent variable, the significant linear relationships are not shown. There is an example graph, **Figure 3.7.1**, showing a sample feature that is not significantly linearly related to the dependent variable.


**Figure 3.7.1,** Relationship Visualization

Considering the results of using some selected features to fit in different models were not good, we decided to try the polynomial transformation on some of the features to be better fitted in linear regression models. We wrote a function to perform the polynomial transformation. The function takes selected training data and the degree of the polynomial transformation. It returns the MSE corresponding to different possible training sets and different linear regression models. After exploring the results of different combinations of inputs, the best degree of the polynomial transformation for each scenario is found. However, the MSE was not improved significantly. Therefore, We abandoned this method.

## 4. Model Selection and Fitting

### 4.1 Quadratic Loss

Initially, we constructed a simple linear regression model using only Quadratic Loss. This is the basic model to make comparisons with further developed models. The reason to choose this model is explicit. Selected features

are likely to have linear relations with the dependent variable, annual mean of closing stock prices. Also, this model is comparatively simple not only in the model construction itself but also in the interpretation of the results. The objective for this linear model is defined as following:

$$min \sum_{i=1}^{n} (y_i - w^T x_i)^2$$

After data cleaning and feature selections, the MSE of this model-fitted more than 100000. Hence, we continued to try more complex models to avoid the potential collinearity and non-linear relation issues.

### 4.2 Lasso Regression( $l_1$ Regularizer)

As we tried to use the Lasso Regression to make a feature selection, we then examined the training sets in the Lasso Regression. It's a modification of linear regression. It minimized the complexity of the model by restricting the sum of the absolute value of the coefficients. The $l_1$ penalty term constrains some weights to zero so that other coefficients are able to take non-zero values. The objective function is defined as following:

$$min \frac{1}{2n} \sum_{i=1}^{n} (y_i - w^T x_i)^2 + \lambda|w|$$

The dataset that was after applied by feature selection of Lasso Regression was fitted into the Lasso Regression Model. The MSE was not improved compared to the Quadratic Loss Model, since there was no feature abandoned after this feature selection process.

### 4.3 Quantile Loss

#### 4.3.1 Quantile Loss with no Regularizer

After fitting the Quadratic Regression and Lasso Regression, we found out that our datasets are not well-fitted in the model. We were curious about our prediction uncertainty. Therefore, the quantile loss is chosen. The objective function is defined as following:

$$min \frac{1}{n} \sum_{i=1}^{n} \alpha(y_i + w)_+ + (1 - \alpha)(y_i - w)_-$$

After several more feature selections were applied to the dataset after the Lasso Regression, we fitted the updated dataset into this Quantile Regression Model. The MSE dropped to approximately 2000, which was still too large.

#### 4.3.2 Quantile Loss with $l_1$ Regularizer

After the 1st round of feature selection processes, the fitted models didn't perform well. We generally tried to add a regularizer to see if the model will be fitted better with our datasets. Also, the scenario of "over-fitting" was discovered after our 1st attempt on the model fitting. Hence, we considered the $l_1$ regularizer to be attached in order to shrink the size of coefficients by putting the penalty on large coefficients. The objective function is defined as following:

$$min \frac{1}{n} \sum_{i=1}^{n} \alpha(y_i + w)_+ + (1 - \alpha)(y_i - w)_-$$

$$+ \lambda|w|$$

Using the same dataset as in the quantile loss with no regularizer model, the MSE only improved a bit compared to the previous models.

### 4.3 Random Forest Regression

The Random Forest Regression contains trees that are created from a different random sample of rows. And at each node, it selects different

samples of features for splitting. Besides the randomness, each tree makes its own prediction. This provides abilities of preventing overfitting and well-fitting the non-linear related features in a linear model to the Random Forest Regression that has higher accuracy compared to other linear regression models. From our datasets, after visualizing the relations between features and the dependent variable, we found out that many features are not linearly related to the annual mean of closing stock prices. In addition to that, based on the initial feature selections, we obtained a high-dimensional training set. Therefore, we chose the Random Forest model. Based on the resulting MSE, it demonstrated that this regression model performed best on our datasets compared to other models we had tried.

Although the resulting MSE was comparatively small, the error was still too large to be acceptable in a data-driven prediction. According to the above conclusion, we continued to use the Random Forest regression to do the model fitting on our datasets with some further transformations. We tried 3 different types of normalizations of data as stated above in **section 3.6**. Overall, the results from normalized data sets performed better model fitting compared to the original data set (**Figure 4.3.1)** using the test dataset. Eventually, the training set that was normalized by the $l_2$ normalization presented the highest accuracy in the Random Forest Regression shown in **Figure 4.3.2**. The MSE in the test set dropped to 10.02.

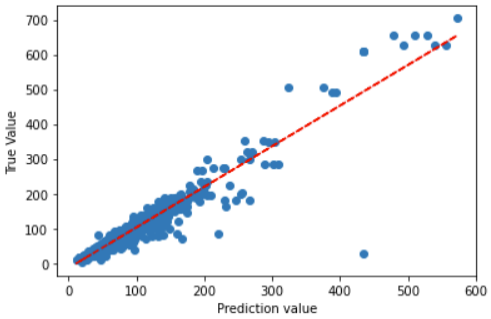


**Figure 4.3.1**, Before Normalization

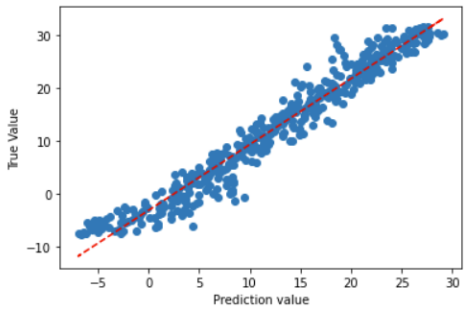


**Figure 4.3.2**, Normalized Data Result (l2)

## 5. Future Improvement

The datasets we used contain the data from 2014 to 2018, but the data used to train the model were selected only from 2014 to 2017, which was not sufficient in our situation as only the top 500 company stock prices were predicted. Based on this prediction goal, there are only 930 data points containing 5-year information of the top 500 companies in the training dataset. In the future, stock prices of more companies in the US stock market would provide a better data range to be considered and build a more accurate prediction model.

In our modeling part, since in the different sectors, such as financial services, healthcare, and technologies, it was discovered that they have comparatively different annual average close price distributions, it would be better to make the predictions and modeling on the basis of different sectors. We tried this concept in our modeling process by only focusing on the “Financial Services” sector which has the highest number of data points, and more solid distribution of the annual average close price compared to other sectors that have a similar number of points. The results of different prediction models were not desirable because of the limited number of data points compared to the number of features. To be more specific, there were 189 data points and more than 200

features to be selected. Models were easily overfitted and had huge MSE values. In the future, with more stock data of more companies in each sector, we believe that a sector-based analysis would be feasible, and prediction models regarding each sector would be more precise.

Besides, as we explored the data more, the indicators provided by the dataset were more company-related. More features about the US economic situation or the stock market features might be useful in future model prediction since these factors also impact a stock price's up and down from the financial aspect.

Although we applied the data transformation with normalization and the polynomial regression, there are some other possible ways to transform the data to better fit the relations between the dependent variables and the features. More techniques of the feature transformation could be tried with deeper research about the company financial indicators and the company stock prices.

## 6. Fairness Discussion

From our perspective, there is no variable that has the potential of discrimination or limitations in our final dataset . Our project goal is to predict the stock prices of firms in the US stock markets, which is not an aspect of people. Also, we didn't choose to evaluate the dependent variable based on any information about the biographies of investors. This can be clearly seen from the datasets selected. Last but not least, features are not nominal or ordinal variables and were collected regardless of grouping these companies into different categories based on any demographic information of each firm.

## 7. Conclusion

The next year's annual average close price prediction can be a valuable aspect for companies to make efficient business decisions, for example, investment in new areas. Besides, it can be helpful for traders while making stock trading decisions and strategies. Among all the linear regressions, we achieved the MSE of 167.10 which is considered to be good for the stock price prediction based on the limited observations but high dimensional data. From the model, we can get the conclusion that the average opening stock price in April and June positively contributes to the average annual close stock price performance and with significantly higher weight. Cash per share and Shareholder equity per share has a negative impact on the average annual close stock price. The average opening stock price in March and May has a more highly negative relation with the annual average close stock price. With these discoveries, companies and investors can form a blueprint of the next year's investment plans before June, this year. For our best prediction model, the Random Forest regressor, we achieved an MSE of 10.02 which is considered to be great for price prediction considering values in our datasets. As it can be seen from **Figure 4.3.2** in the Random Forest section, the true average close price aligns with the prediction of the annual average close price in the test set. Future improvement can be done as we mentioned above, with more data collections from US companies other than the S&P 500 companies. Besides, more valuable features can be added for a better prediction of the stock price

## References

1. Rodriguez-Nieto, J.A., Mollick, A.V. The US financial crisis, market volatility, credit risk and stock returns in the Americas. *Financ Mark Portf Manag* 35, 225–254 (2021). https://doi-org.proxy.library.cornell.edu/10.1007/s11408-020-00369-x