Application of Machine Learning on Fundamental Stock Price Analysis

Albina Cako, BSc *York University, Certificate in Machine Learning*
Colin Green, BSc *York University, Certificate in Machine Learning*
Lucy Zhang, BSc *York University, Certificate in Machine Learning*
Sean X. Zhang, MSc *York University, Certificate in Machine Learning*

Abstract:

Keywords: stock price, fundamental analysis, machine learning, R

Introduction

Background

The stock market is a marketplace where investors can purchase or sell shares of publicly traded companies. As of 2019, the amount of money invested in the global stock market has surpassed over $85 trillion. Since the inception of the stock market, investors have continuously sought to develop methods of improving their returns. Currently, there are two main schools of thought when it comes to stock market analysis: technical analysis and fundamental analysis.

Technical analysis looks at buying and selling trends of a particular stock. The core theory of technical analysis assumes that all information is already factored into the stock price. As such, technical analysis prioritizes identifying patterns or trends in time-series data to predict stock price at a particular time point.

Fundamental analysis attempts to measure the intrinsic value of a company by studying information from that company's balance sheet, such as revenue or debt. Fundamental analysis attempts to identify companies that appear to be 'undervalued' or 'overvalued' to inform buy or sell recommendations.

Previous machine learning models that simulated stock market returns have largely focused on using time series data to predict stock trends, which is more akin to technical analysis. However, such models have run into challenges such as overfitting or a lack of interpretability. One benefit of fundamental analysis is that it allows the investor to learn about which aspects of a company's financials will influence that company's stock price; it is more interpretable. As there are dozens to hundreds of variables on a company's balance sheet, the use of machine-learning approaches may augment fundamental analysis by pinpointing important markers of a company's financials and their relationship with the stock price.

Objective

In this project, we apply machine learning and data science techniques to predict the market capitalization, which is how much company is worth on the stock market. Stock price can then be calculated by dividing market capitalization by total number of stocks issued. We also create an application using R shiny to be used as a guide for investors. This application would be used individuals interested in checking their stock analyses with a machine learning prediction. The application could be used by financial analysts, portfolio managers, or non-professional investors with an interest in fundamental analysis.

Methodology

Data Preprocessing

The original dataset was obtained from Kaggle. Five datasets were combined together containing stock information for different years: 2014, 2015, 2016, 2017 and 2018, respectively. There were 225 columns in the original dataset. However, after curating the data, only 66 columns were chosen as fundamental columns and were included in the project.

Missingness

The dataset was assessed for missing values. Any columns that had more then 1/3 of the data as missing values were removed. For the rest of the columns, data imputation was performed using the MICE package in R. We used the CART method to impute the data. CART imputes values by using classification and regression trees. Four columns were left with missing values after imputation. Those columns were removed leaving a dataset with a total of 62 columns.

Feature Selection

It is important to note that this project contains both unsupervised and supervised learning. Decision Tree was used for feature selection. Decision tree is a classification algorithm used for classification problems, as well as detecting variable importance in a dataset. The top 10 important variables from the decision tree were selected as the features. They were used to run k-means unsupervised learning, which determined 4 clusters of data. Then, supervised learning dataset was selected as the top 10 variables selected from the decision tree plus the cluster # (as a categorical variable) and the Sector of the stock. Thus, the unsupervised learning data contained 10 features, while the supervised learning data contained 12.

Table 1: Common Predictor variables

Variable	Type
Consolidated.Income	numeric
Dividend.payments	numeric
Stock.based.compensation	numeric
Income.Tax.Expense	numeric
Retained.earnings deficit	numeric
Operating.Cash.Flow	numeric
Operating.Expenses	numeric
R.D.Expenses	numeric
Total.debt	numeric
Long.term.debt	numeric

Principle Component Analysis

We applied Principle Component Analysis (PCA) to our feature dataset for dimension reduction before doing unsupervised learning using the k-Means clustering algorithm. PCA creates orthogonal 'principle components' of the feature set, reducing multicollinearity within the data. Although the k-means algorithm is non-parametric, the reduction in multicollinearity by PCA could lead to greater discrimination between the observations.

Unsupervised Learning

The k-Means algorithm was performed in order to cluster the data before supervised learning. The number of clusters was evaluated by plotting the within-cluster sum of squares (WSS) against the number of clusters (k). The optimal number of clusters was chosen based on a combination of the 'elbow method' and domain knowledge.

Supervised Learning

Supervised learning was performed using four algorithms: XGBoost, Random Forest, Lasso Model and GBM Model. XGBoost is a very powerful algorithm which drives fast learning and offers efficient usage of storage. XGBoost uses ensemble learning, which is a systematic solution that combines the predictive power of multiple learners. It outputs a single model that gives the combined output from many models. This allows the opportunity to not rely on the results of a single machine learning model.In this particular model, the trees are built sequentially, such that the next tree focuses on reducing the errors of the previous tree. Random forest is another supervised learning model that uses "ensemble" method to fit many decision trees by using a subset of the rows and then taking the "mode" of the predicted class. GBM, which stands for Gradient Boosting Machine, is also a gradient boosting algorithm that works similar to XGBoost. However, XGBoost has more tuning parameters, thus both algorithms were chosen for comparison. All models were ran and they were evaluated using the k-fold cross validation method. Three accuracy metrics: Root Mean-Squared Error (RMSE), Pearson correlation (R^2), and Mean Average Error (MAE) were used to chose the final model.

Deployment

Results

Table 2: Data Dictionary

Variable	Type
X.1	Index of the records
X	Stock ticker symbol
Consolidated.Income	Describe all changes in equity except investments made by owners in a period of time
Dividend.payments	A dividend payment to shareholders
Stock.based.compensation	Describe the rewords to employees in lieu of cash made by stock or stock options
Income.Tax.Expense	Total amount of tax
Retained.earnings deficit	Represent the negative or debt banlance
Operating.Cash.Flow	Measuremnent of the amount of cash the company generated
Operating.Expenses	The amount of expense of a company
R.D.Expenses	Research and development of tax return

Variable	Type
Total.debt	Sum of long term debt and short term debt
Long.term.debt	Value of long term debt
Market.Cap	market capitalization for a company

Data Preparation

Missing Values

After we finished the first step of data cleaning, we want to do the data validation. For missing values, as the plot shown, a lot of observations make up the majority of the missing data and we decided to remove observations that have more than a third of the columns NA. After we removed those observations, we set the sector and year columns as a factor and saved the new data set into a new CSV files for further data exploration.

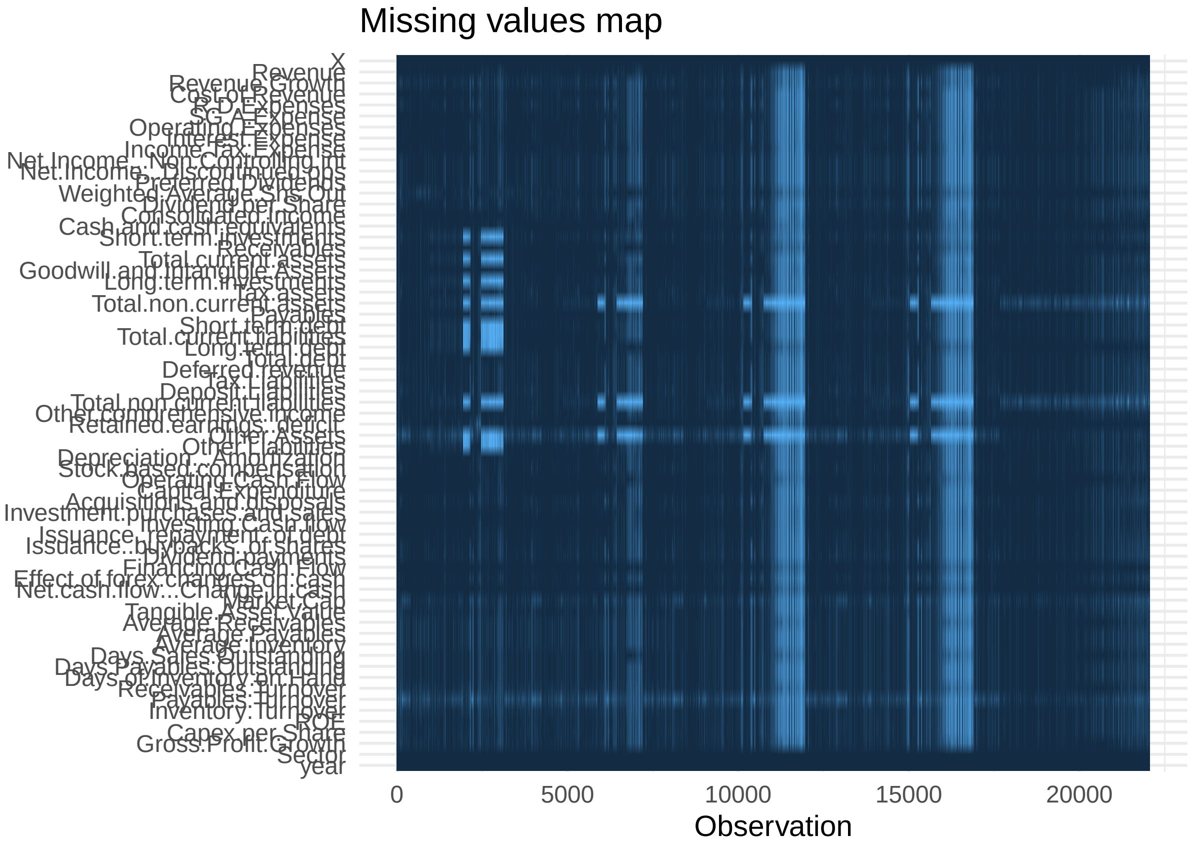


Figure 1: Missing values map before imputation

To account for missing values, we chose to use the CART (Classification and Regression Trees) method of imputation. After imputation, 4 columns still have missing values, which were then subsequently removed.

Correlation Plot

There were 62 columns after we finished data cleaning, and we want to select the important features to do modeling. We performed a correlation analysis based on Pearson's coefficient between each numeric predictor first. We considered a correlation > ∣0.8∣, with p < 0.05 as a significant correlation. Figure 3 demonstrates significant correlation between many of our predictor variables. However, due to the large amount of variables, the correlation plot was uninterpretable if we try to plot all the variables, therefore, we tried to filtered the correlation plot, keeping only variables that had a correlation with absolute value greater than 0.8.

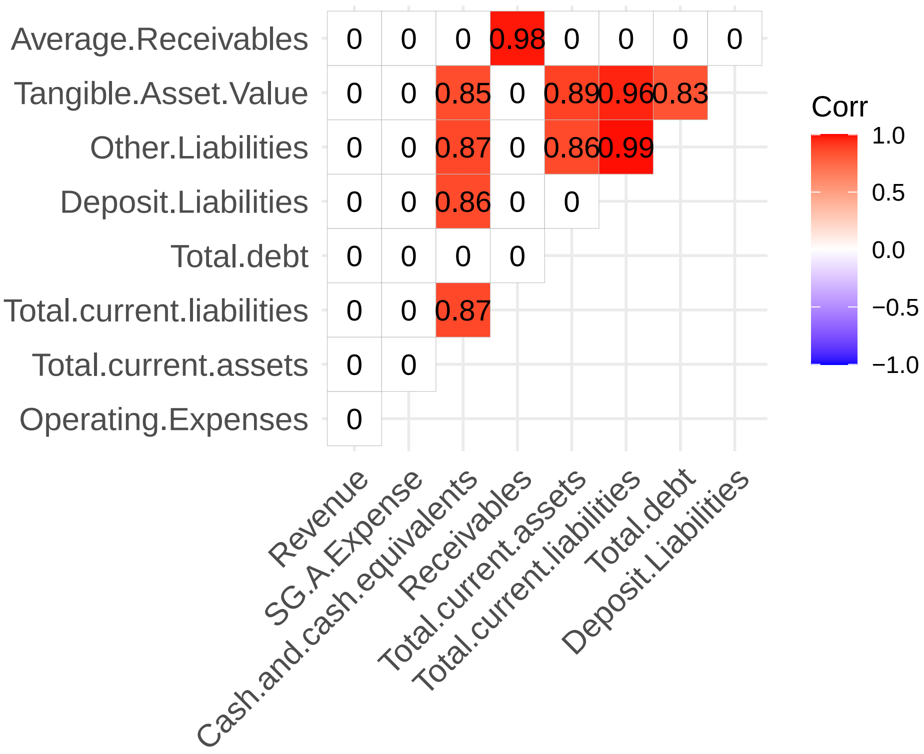


Figure 2: Correlogram of variables with R>∣0.8∣

Data Distribution

we then performed the distribution plot for all the columns to check the data distribution, to observe the means, and check for outliers. Most variables were not normally distributed and had a clear skew. In Figure (FILL_IN), we show a subset of the variable distributions.

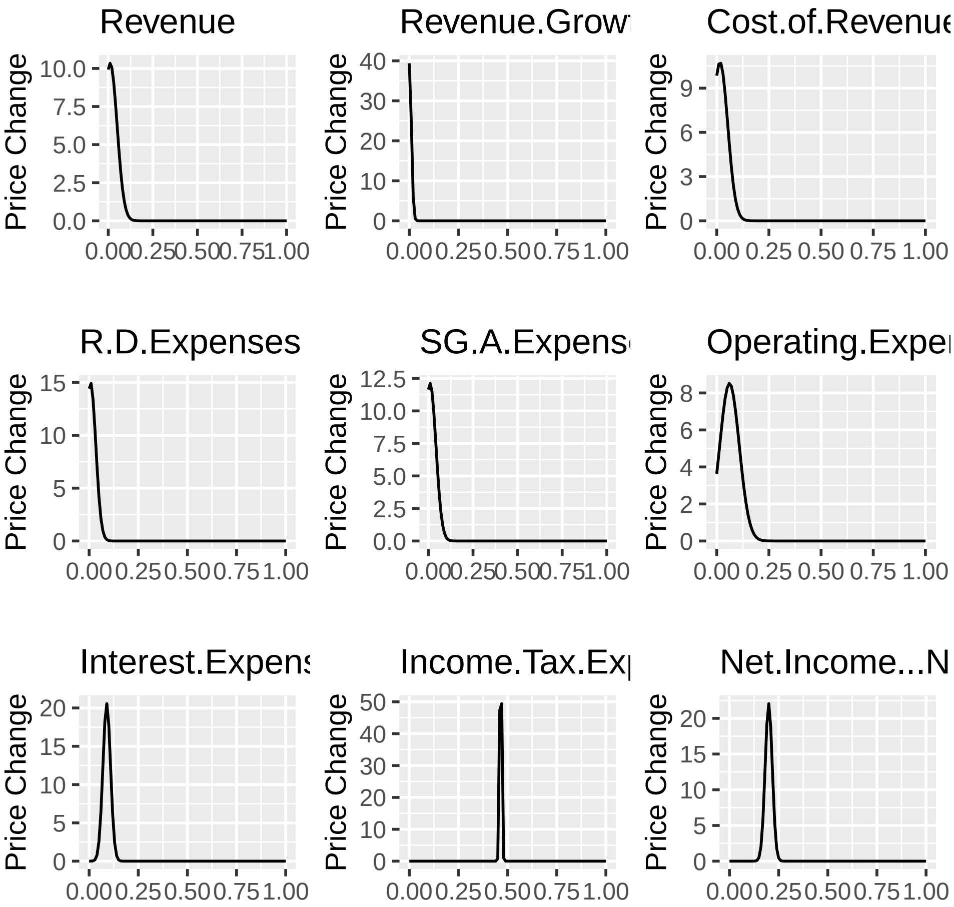


Figure 3: Column distributions

Feature Selection

In order to do our feature selection, we ran a decision tree model to determine important variables within our dataset. Below are plots showing the importance of each variable as well as their correlations with one another.

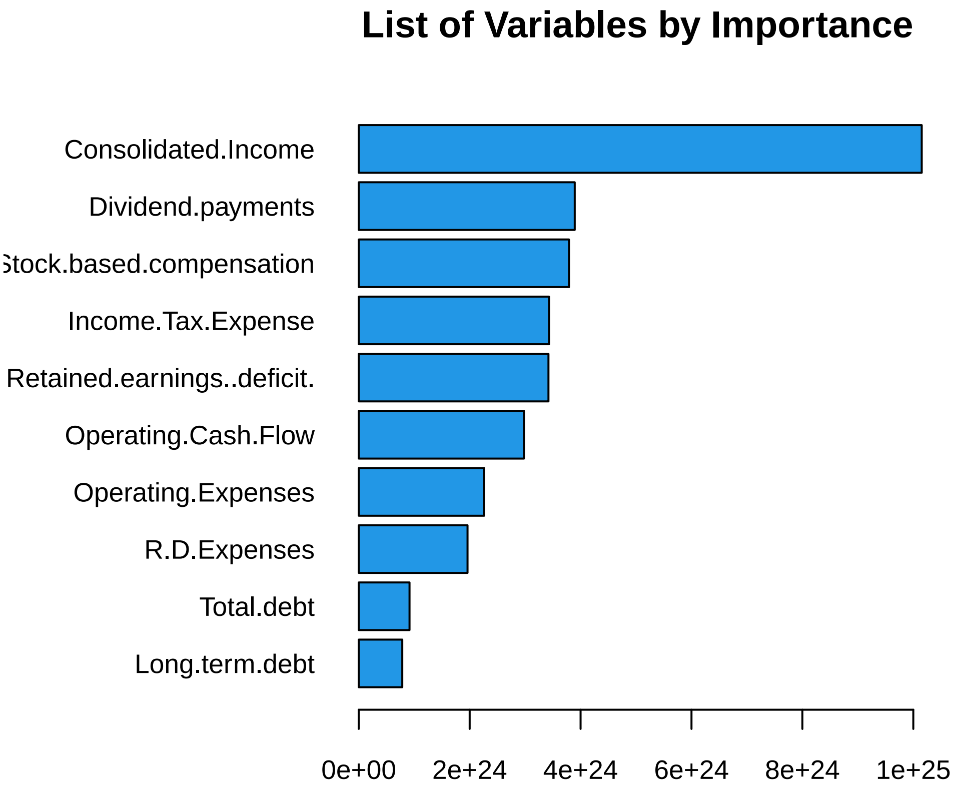


Figure 4: Top 10 variable importance, determined by Decision Tree

Principle Component Analysis

We performed PCA to reduce the dimensionality of our feature dataset. The Scree plot shows the overall variance explained by each principle component. The top 5 dimensions explained approximately 90% of the total variance within the data. Individual datapoints involving large technology companies (Google, Apple, Amazon) had high contributions to the overall variance. R&D Expenses and Stock-based compensation were two variables with high contribution to variance, while Income Tax Expense and Operating Cash Flow had more negligible contribution.

K Means Clustering

The ‘elbow method’ was first performed to determine an optimal number of k clusters. However, there was no significant drop in within-cluster sum of squares with k besides k=2. As two clus-

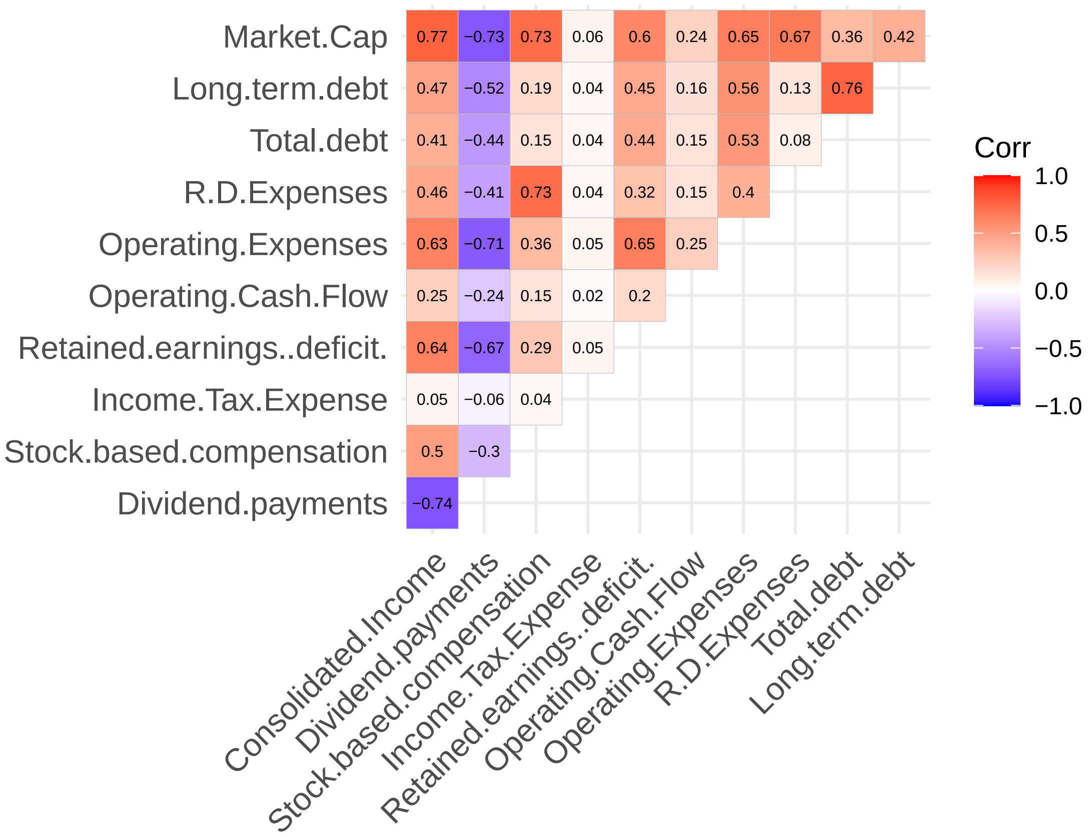


Figure 5: Correlogram of Top 10 Variables

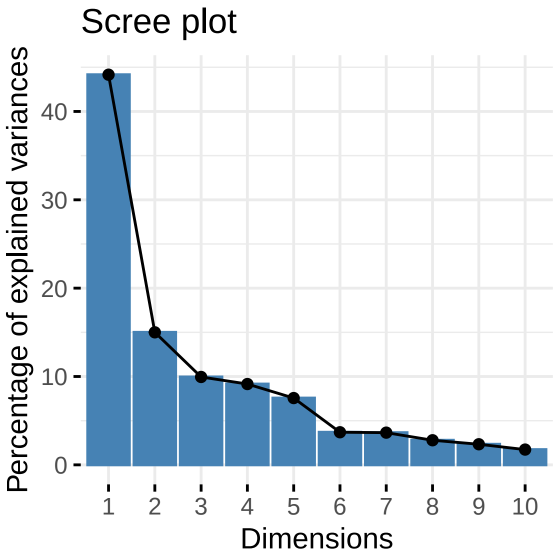


Figure 6: Scree plot

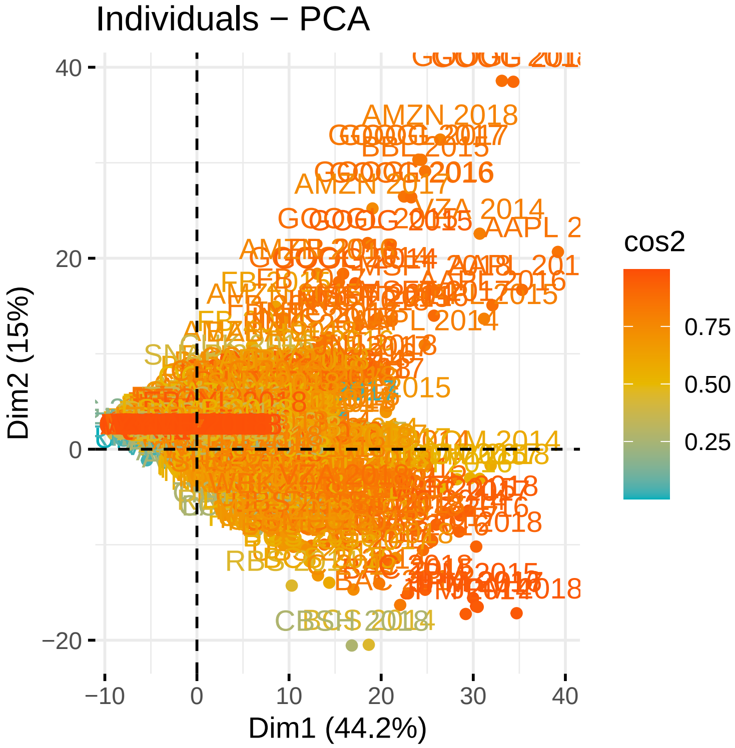


Figure 7: Effect of Individual points - PCA

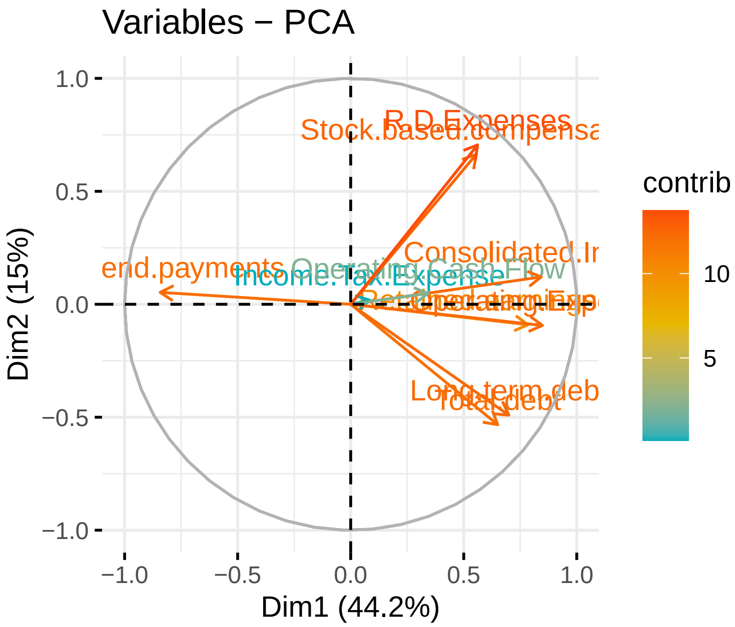


Figure 8: Effect of Variables - PCA

ters did not provide much discrimination for our observations, we instead used k=4 as the final number of clusters.

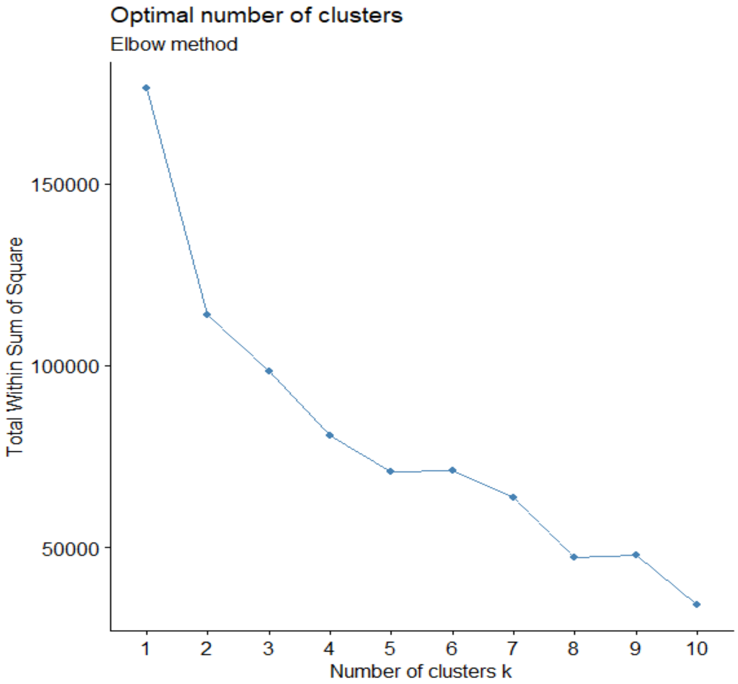


Figure 9: Elbow method

The following figure displays our datapoints in a 2-D space based on 4 clusters.

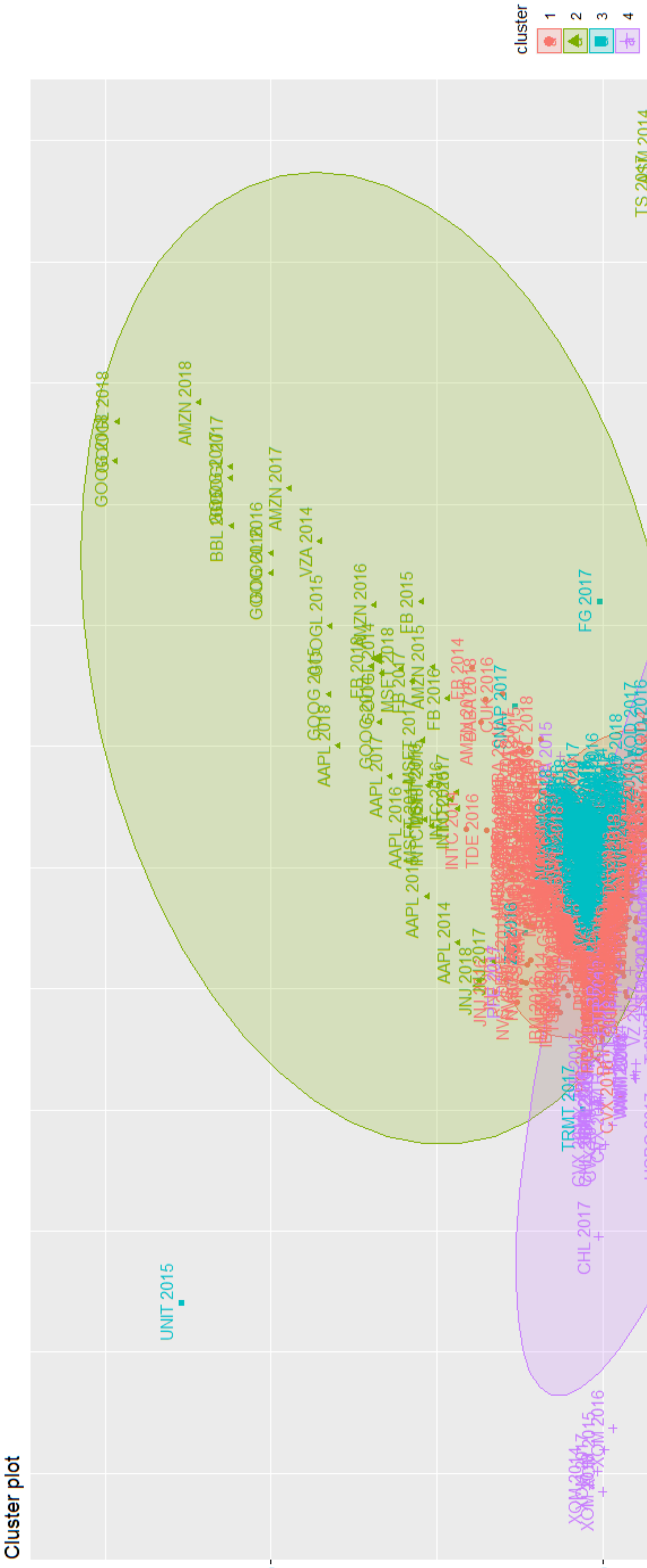


Figure 10: K means clustering, k = 4

Cluster Interpretation

We performed some exploratory visualizations to interpret how the data was clustered by k-means. Cluster 1 contained the majority of observations, with n=19759. Cluster 2 had 35 observations, cluster 3 had 126, and cluster 4 had 586. On average, cluster 1 contained more small- and medium-sized companies compared to other clusters, with 89% of observations falling under a market cap of $10 billion.

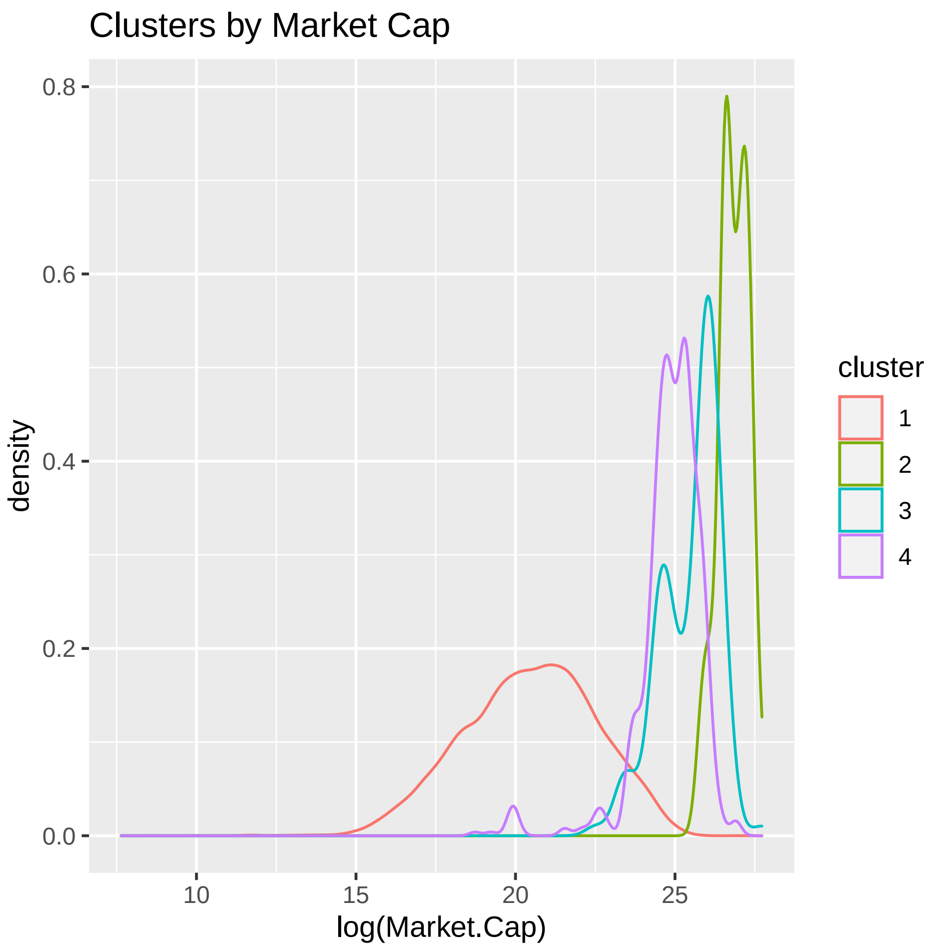


Figure 11: Clusters by Market Cap

K-means was able to segregate the large high-tech companies: Facebook, Apple, Amazon, Google, Intel, and Microsoft all into one cluster (Cluster 2). We noted that these companies trended towards large market capitalization, high stock compensation, and high R & D expenses. Cluster 3 contained a significant majority of big banks, such as JP Morgan, Wells Fargo, and Bank of America, as well as large energy corporations such as ExxonMobil and Chevron. Clusters 1 and 4 had similar sector distributions, although companies in cluster 4 were all large cap. Interesting to note that the top 20 big pharmaceutical and healthcare companies were mainly in cluster 4, such as Johnson & Johnson, Roche, and Abbvie.

Price-to-earnings ratio is a common method of determining how a company is valued by in-

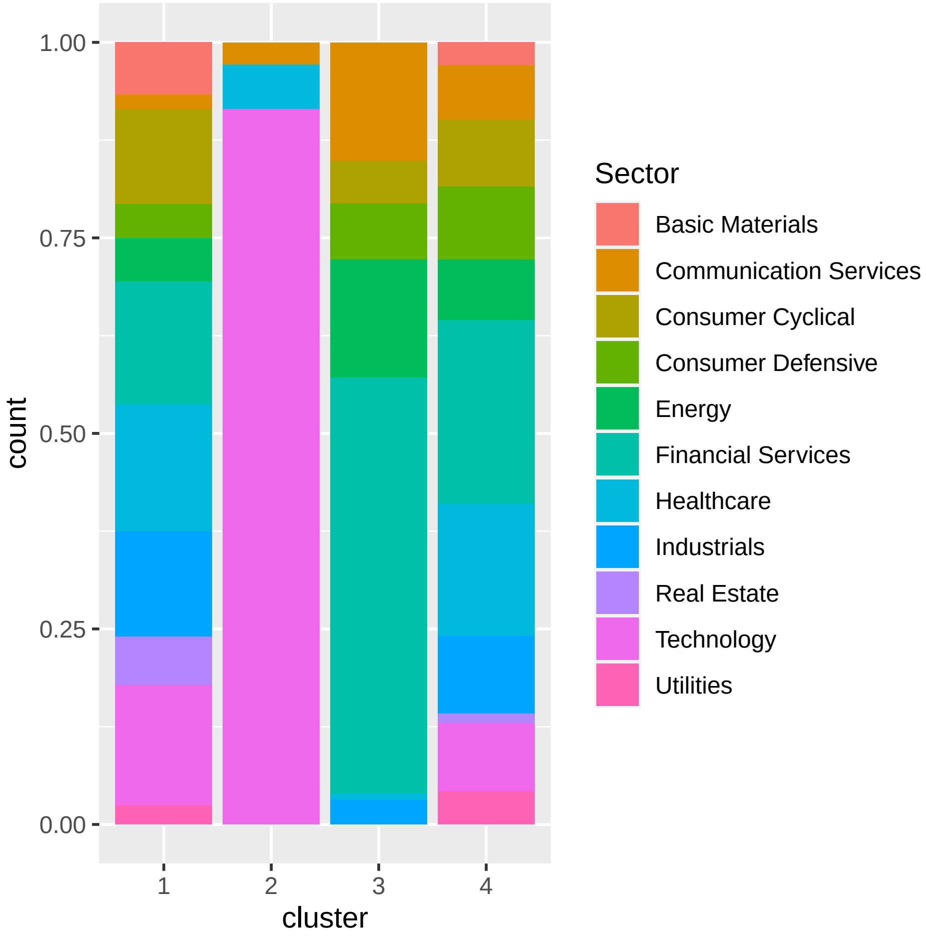


Figure 12: Clusters by Sector

vestors. A high P/E ratio may suggest that investors are willing to pay a higher price for that company's share price because of future growth expectations. Here, we see that cluster 2, composed largely of high-tech companies such as Google, had high P/E ratios - aligning with investor sentiments about the growth of the industry. Cluster 1 maintains an interesting bimodal distribution of positive and negative P/E ratios. Stocks with negative P/E ratio suggest that these companies are reporting a loss.

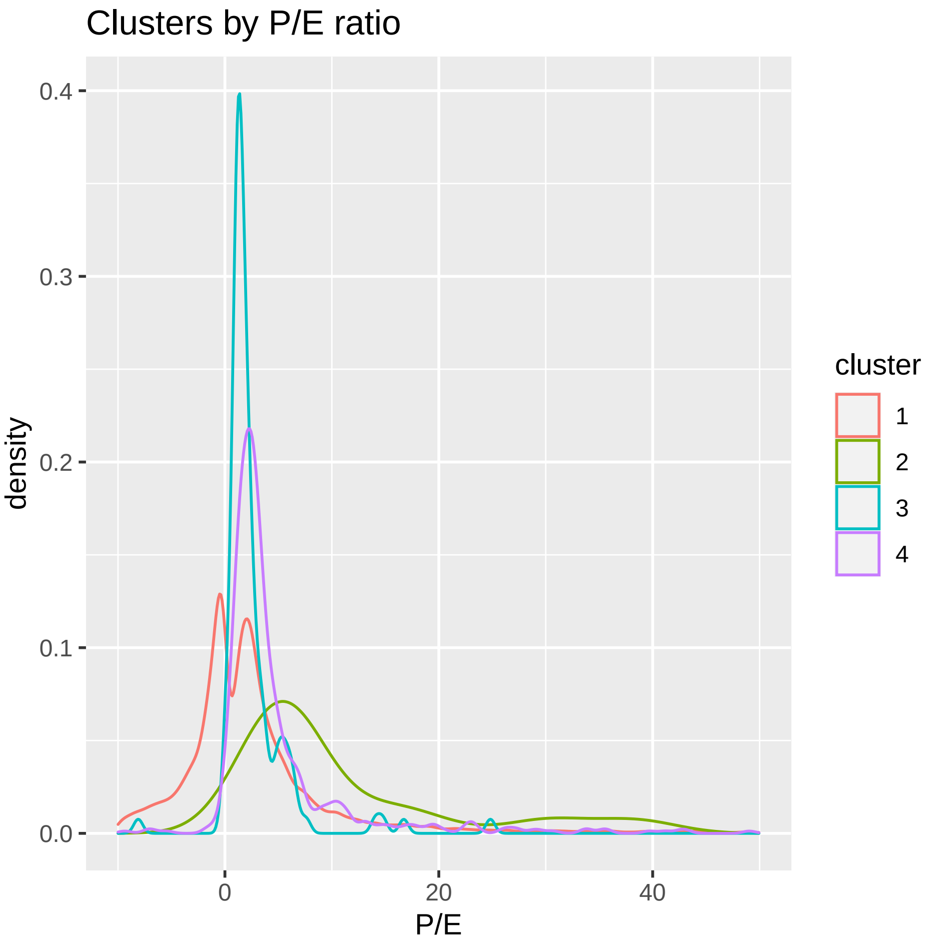


Figure 13: Clusters by Sector

Normalizing for the proportion of small-medium and large cap stocks in cluster 1, we noted that small to medium-sized companies were 2 to 3 times more likely to have a negative P/E ratio compared to large companies. The bimodal distribution in cluster 1 can therefore be partially explained by a split between the distribution of company size.

Modeling

The k-fold cross-validation method evaluates the model performance on different subsets of the training data calculates the average prediction error rate. We used k = 10 for our project,and this method was used instead of the simple train-test-split as it gives a more valid estimation of model effectiveness.

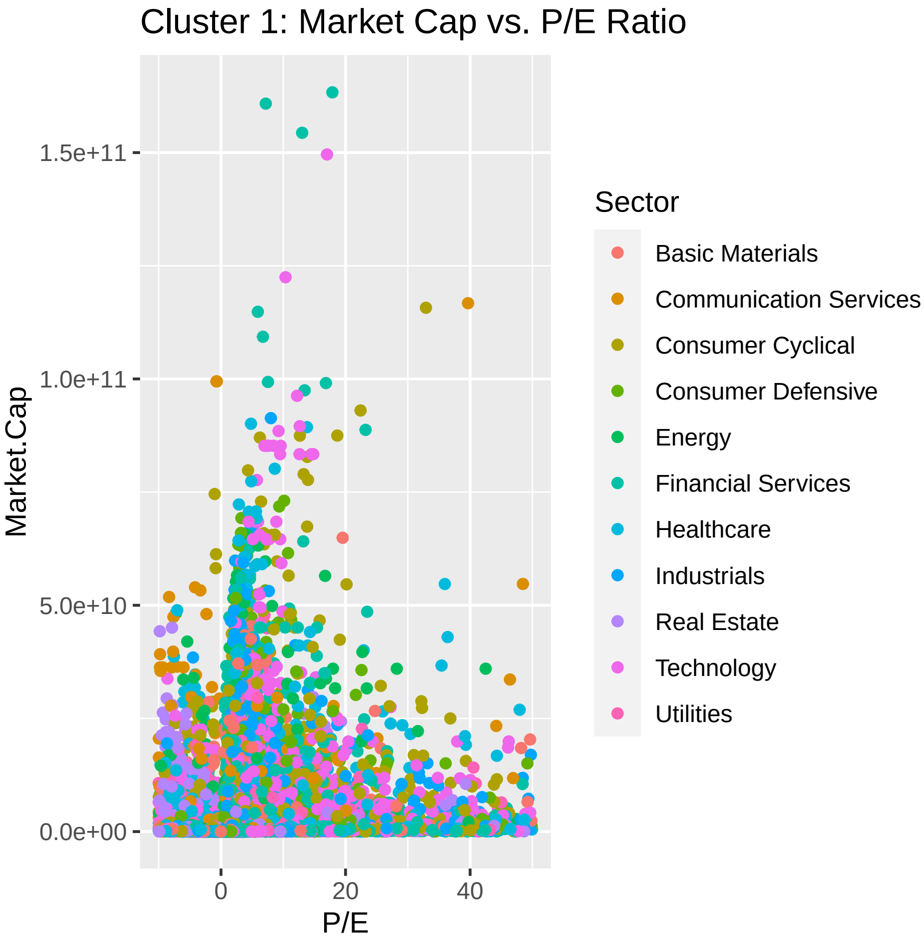


Figure 14: Cluster 1: Market Cap vs. P/E Ratio

Random Forest

XGBoost

For the XGBoost model, we used nrounds, eta, max_depth as tuning parameters to increase the accuarcy of the final model and reduce errors. The model used tuning parameter 'subsample' as a constant at a value of 0.8, and RMSE was used to select the optimal model using the smallest value. The final values used for the model were nrounds = 200, max_depth = 6, eta = 0.1, gamma = 0, colsample_bytree = 0.5, min_child_weight = 1 and subsample = 0.8.

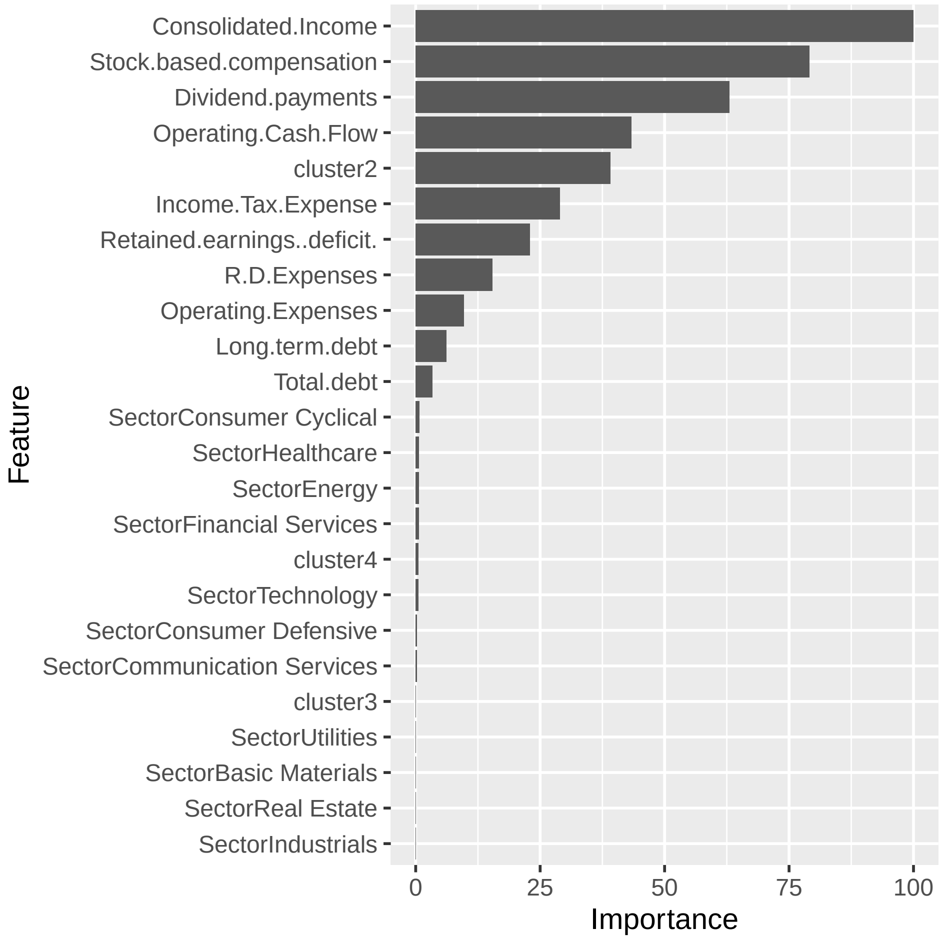


Figure 15: XGBoost Tuning Parameters

Lasso Regression

For the lasso regression model, RMSE was used to select the optimal model using the smallest value. The final value used for the model was fraction = 0.9.

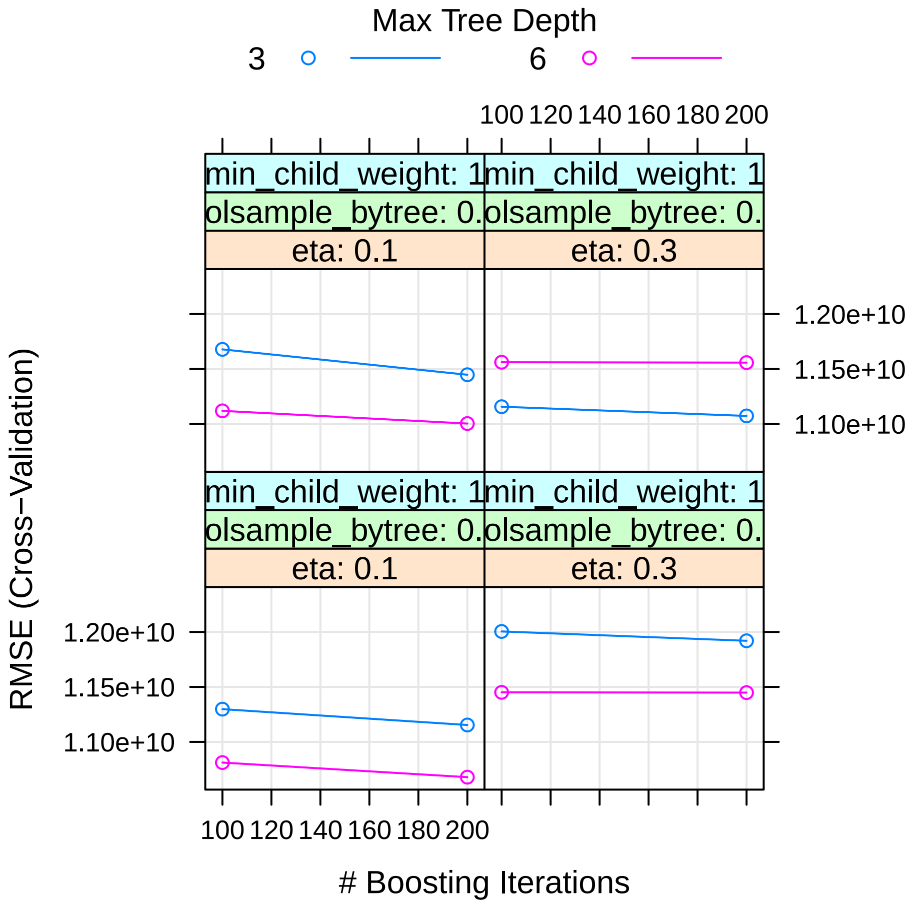


Figure 16: XGBoost Tuning Parameters

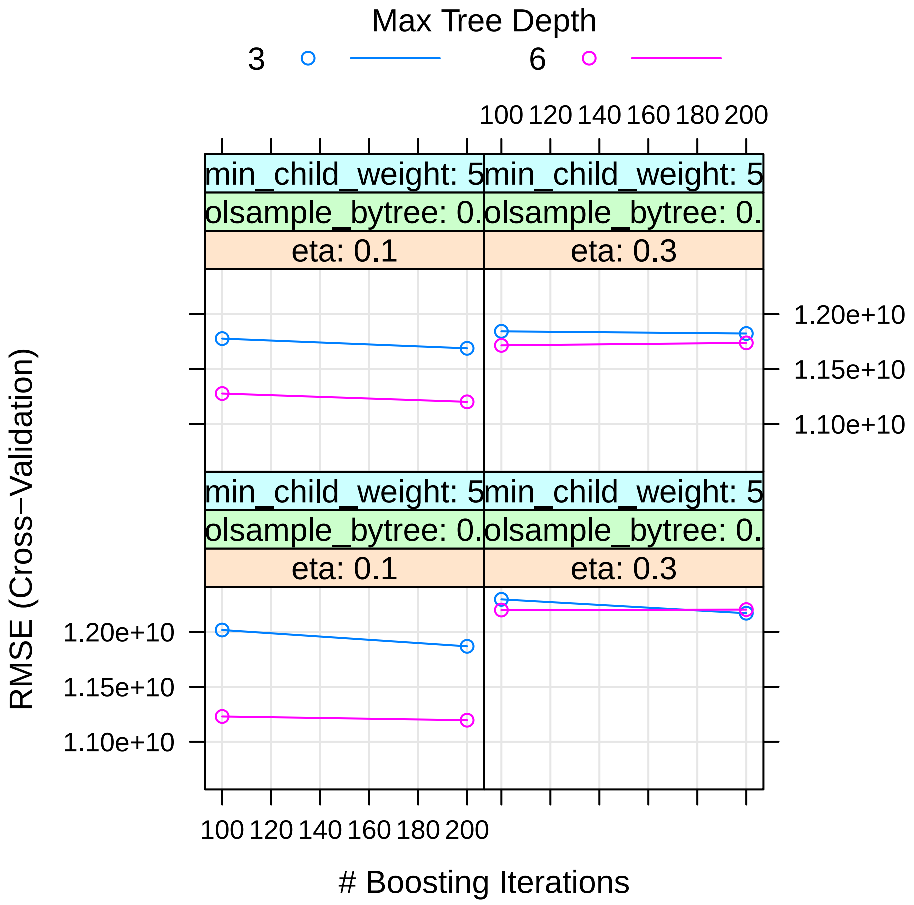
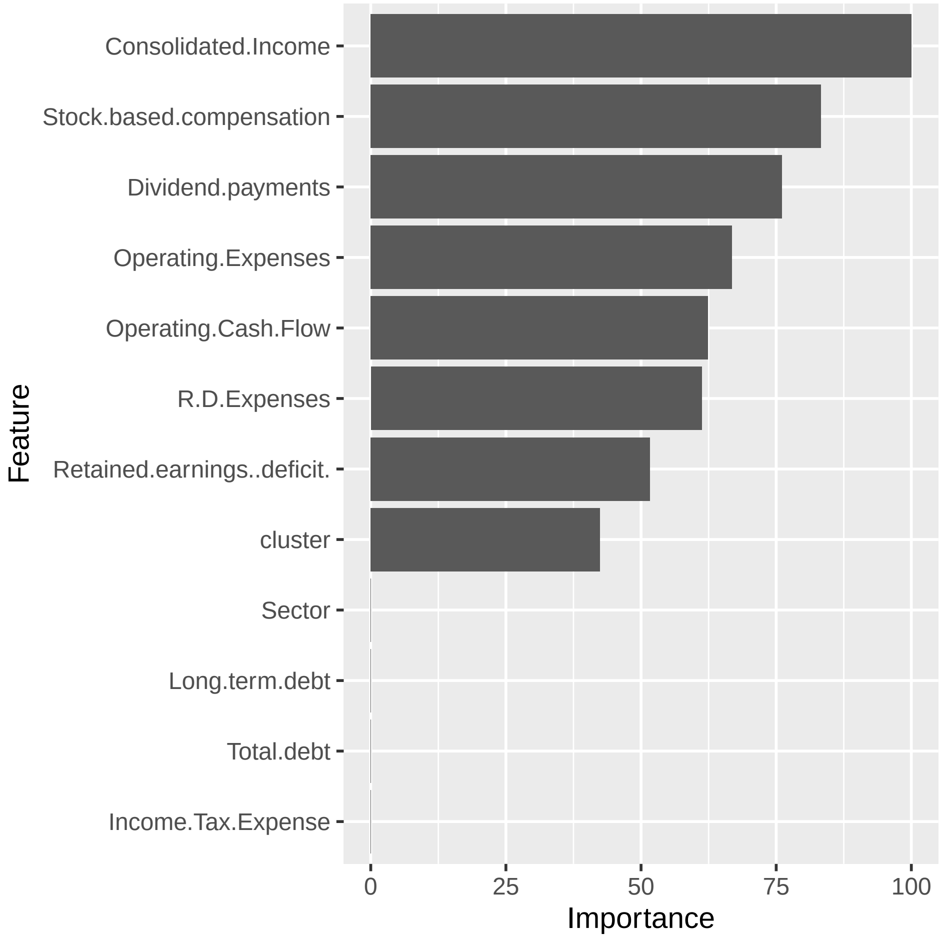
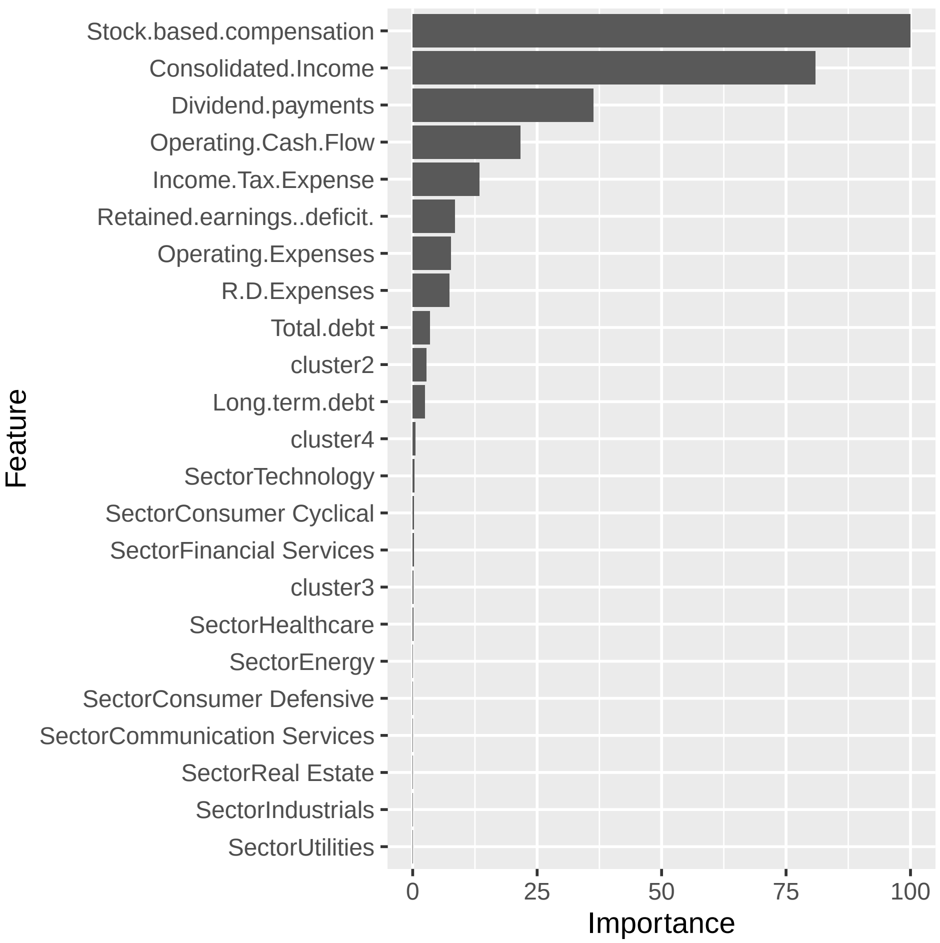


Figure 17: XGBoost Tuning Parameters



Gradient Boosting

The gradient boosting model was tuned by several different parameters. The final values used for the model were n.trees = 600, interaction.depth = 9, shrinkage = 0.1 and n.minobsinnode = 20



Model Selection

All models found *nmnmb* and *hghh* to be important predictors of Market.Cap. Mean Absolute Error (MAE) tells the average error of the variable we want to predict. Root Mean-Squared Error (RMSE) is similar with MAE but it is more useful when we are interested in fewer larger errors over many small errors. Overall, we prioritize model stability and thus prioritized RMSE over MAE. R^2 computes how much better the regression fits the data than the mean line, which gives an overall score.For predicting market cap, we desired a model with the lowest RMSE and MAE to keep the high accuracy of prediction. The XGBoost model had the highest R^2 as well as the lowest RMSE and MAE, thus, it was chosen for deployment.

Table 3: Model Accuracy

model	RMSE	R2	MAE
random_forest	2.75e+05	0.81	1.36e+05
extreme_gradient_boosting	1.08e+10	0.90	2.70e+09
Lasso_Regression	1.45e+10	0.82	3.75e+09
gradient_boosting	1.18e+10	0.88	2.92e+09

Discussion