Wake Forest University

Stock Growth Prediction and Alpha Factor Mining

Based on 10-k Filings Financial Dataset

Patrick Fan

STA 381: Applied Statistics Capstone

Dr. Nicole Dalzell

Dec 2, 2021

**Abstract**

In this research, our team tackles on the task of predicting stock price growth and alpha factor mine. While significant amount of work has been conducted in this field, we put our focus on the influence the 10-K filings financial indicators may have on stock price growth. Through a rational analysis on the explanatory variables and also fitting logistic regression and gradient boosting decision tree, we are able to give predictions with 60% of accuracy on unknown data and describe the significant variables at hand. Our work presents significance in that it relates 200+ financial indicators with the price of stock.

Table of Contents

[1. Introduction 4](#_Toc89437507)

[2. Data Description 4](#_Toc89437508)

[3. Data Cleaning / Processing 5](#_Toc89437509)

[4. Literature Review 7](#_Toc89437510)

[5. Explanatory Variable Analysis 7](#_Toc89437511)

[6. Method: 9](#_Toc89437512)

[6.1 Logistic Regression Model 9](#_Toc89437513)

[6.2 Gradient Boosting Decision Tree 10](#_Toc89437514)

[7. Result 11](#_Toc89437515)

[8. Conclusion / Future work 14](#_Toc89437516)

[9. Citation 15](#_Toc89437517)

1. Introduction

Statistics and machine learning models have entered the financial field for decades and have been used to acquired extraordinary amount of revenue by many hedge funds and asset management enterprises. Machine learning can be defined as “a subset of data science that uses statistical models to draw insights and make predictions” (Didur). Machine learning models are broadly used to predicting the stock price, given the corresponding data in history. In this research, my team and I endeavor to apply appropriate classification models to predict whether or not the stock price of a stock grows over the year based on the 200+ financial indicators that are defined in the 10-K fillings (Carbone). In addition, we would like to conduct statistical inference on our models to identify the effective financial indicators (alpha factors) in predicting the stock price.

2. Data Description

The data we use is the *200+ Financial Indicators of US stock (2014-2018)* dataset by Nicola Carbone (Carbone). This dataset by Carbone is dedicated to exploring how valuable a company is through providing the financial information of these companies. Our data have 5 datasets, each representing the stocks’ information of a specific year ranging from 2014 to 2018. Within each, we have information on a plethora of US stocks—on average 4000 companies—and their financial information. Besides the class variable which indicates whether or not the value of a stock increase over the year, we also have 223 descriptive features that are financial indicators released in the 10-k fillings. Several important descriptive features among them are: Free Cash Flow, Dividend Payment, Revenue Growth, etc.

In order to better understand our data, we applied the dataset *2014\_Financial\_Data.csv* to display some basic characteristics of our datasets. To start with, our raw data have 222 continuous features on various financial disciplines and only 1 categorical feature that indicates the sector this company belongs to. Beyond that, Figure 1 shows the class distribution of the entire dataset. There is a slight difference in the number of stocks that grew over the year compared to the number that dropped. Figure 2 shows the stock counts grouped by the sectors. It is shown that there are 11 sectors in total and the distribution is not entirely uniform.

Chart, bar chart

Description automatically generatedChart, bar chart

Description automatically generated

Figure 2. Sector distribution

Figure 1. Class distribution

The information above allows us to have some intuition about the type of data we are dealing with—datasets that can be slightly unbalanced. The next step is to preprocess our datasets in order to ensure we are working on reliable information.

3. Data Cleaning / Processing

Even though we have the financial data (Carbone) from more than 200 features, significant number of missing values and zero-values prevent us from conducting many of our intended research. Therefore, we started with processing the data entries with missing values or zero-values. Imputing missing values is an intuitive effort as no useful information is presented in the missing data entries. Zero-values, on the other hand, may present suspicious information since large proportion of zero-values in a column can itself be the subsequence of some imputation process. Figure 3 in below shows the distribution of missing values (top) and zero-values (bottom) for the 225 features (for all 5 datasets).

A picture containing text, measuring stick, different

Description automatically generated

Figure 3. Missing values / zero-values distribution

As shown in the above plots, our data have quite a few features with large proportion of missing values and zero-values. The columns “operatingCycle” and “cashConversionCycle” for instance almost have no useful information at all—close to 100% of missing values. On the other hand, variables such as “Preferred Dividends”, “Deposit Liabilities” have 80% plus zero-values, which are also suspicious. In total, there are about 17.24% of the data was either “NaN” or 0-valued. For the best quality of our analysis, we decide to trim these features with excessive amount of unrelated information. Consequently, appropriate thresholds on the proportion of missing values and zero-values need to be set so that we may conclude on whether the features present useful information or not. The option should allow the selected features to have large proportion of useful information so that the missing values may be imputed. Following this criterion, we decided to set the missing value threshold to be in 5% to 7% and 0-value threshold to be in 5% to 10%. In addition, we decided to remove the top and bottom 3% of values for all features as they represent the extreme and possibly outliers for each feature.

At this point, Unconditional Mean Imputation (UMI) technique is applied to impute for the other missing values. UMI refers to the data imputation technique of replacing each missing value with the mean of the observed data for the variables. This technique is achieved, in our case, by replacing the missing feature values of a specific stock with the mean of the feature values from the other stock of the same sector. This approach can be justified since stocks from the same industry are more likely to produce similar financial statements than stocks from different industries.

Eventually, our group believes that more analysis would be possible with five datasets combined compared to them being analyzed separately. Thus, we extracted the variables that are common in all five years of data and combined the stocks together with these variables. Through this step, we have a dataset with 18410 rows and 40 columns.

4. Literature Review

For this task, we will focus on classifying whether a stock grows or not over the year. Therefore, the model of interest to our problem should be some classification models. According to Garg, effective algorithms in the field of classification include logistic regression, random forest and many more (Garg). Among them, we decided to experiment using logistic regression and gradient boosting decision tree to classify our dataset.

5. Explanatory Variable Analysis

Since our dataset now have 30+ explanatory variables, we will only display the results of EDA that are significant. We will start with the only categorical variable “Sector”. Shown below in Figure 4 is the distribution of Sector variable against the class distribution. Simply examine the distribution allows us to hypothesize that stock from different Sector are likely to perform differently. This conforms to our intuition since different industries do perform differently across the year.

Chart, bar chart

Description automatically generated Chart, scatter chart

Description automatically generated

Figure 5. Net Income vs. Class

Figure 4. Sector Distribution

We conduct the rest of the EDA analysis based on the logistic regression. Detailed explanation of how logistic regression work can be found in part 6.1. For now, just the visual result suffices for providing us with some insights into the dataset. In Figure 5 above, we may see the distribution of the explanatory variable Net Income against the odds of the class variable. Positive linear relationship can be spotted, even though the trend is not deterministic.

Chart, scatter chart

Description automatically generated Chart, scatter chart

Description automatically generated

Figure 7. Free Cash Flow vs. Class

Figure 6. EPS vs. Class

Another logit plot (Figure 6) shows the relation between the variable Earning Per Share against the odds of class. Here we may observe a more apparent positive, convex relation between the two variables. Finally, we explore the relation between Free Cash Flow against the odds of class. The relation seems to be minor, but is to some extent concave, as shown in Figure 7.

6. Method:

Given the cleaned data, now we are facing a classification problem predicting the binary “Class” variable which indicate whether a stock grows or not over the year.

6.1 Logistic Regression Model

The embedded logic under the logistic regression model is deeply related to the concept of odds and odds ratios. For an event e with binary outcome, the odds of e is defined as the ratio of the probability that e happens divided by the chance that it does not happen, as shown in equation 1.

(1)

where:

~e = the event does not happen

The logistic regression model takes the natural logarithm of the odds as a regression function of the predictors (LaValley). This can be visualized by equation 2 below.

(2)

where:

xi = value of the ith explanatory variable

Y = binary response variable with possible values 0 and 1

**β**i = regression coefficient for the ith explanatory variable

n = number of explanatory variables in x

Another concept in logistic regression is the odds ratio. Assume the equation described by equation 2, then the odds ratio of xi, keeping all other variable values constant, is the amount of increase in odds of Y when xi increases by one unit.

A simple transformation of equation 2 gives equation 3, which is more intuitive according to our senses. This function then serves as the fundament of logistic regression.

(3)

6.2 Gradient Boosting Decision Tree

Gradient Boosting Decision Tree (GBDT), which is a subcategory of Gradient Boosting Machine (GBM) proposed by Friedman in 2001, is considered as one of the best machine learning classification models. GBDT can be described as the boosting type of ensemble technique that construct a set of base classifiers learned from the training data. And in this algorithm, the base classifiers we use are decision trees.

Boosting refers to an iterative procedure to adaptively change distribution of training data by focusing more on previously misclassified records. Initially, all training samples are assigned the same weight thus the same probability of being selected as the training samples for the next round. However, records that are misclassified will be given more weights (probability of being selected) in the next round. Consequently, the next classifier is more likely to predict this misclassified instance correctly.

Gradient Boosting extends the concept of boosting by incorporating three essential elements: loss function, weak learner, and additive models. Firstly, a loss function shows the difference between the predictions given by a weak learner and the actual labels of the instances. The core property of the loss function is that it needs to be differentiable so that a weak learner can be fit based on the negative gradient value given by the previous weak learner. A common loss function used in the binary classification problem is the Binary Cross-Entropy Loss which is defined as in equation 4:

(4)

where:

= the actual label for the ith data object

= the predicted label for the ith data object

N = the number of instances in the dataset

Weak learners are individual classifiers each fitting a potentially different classification model. In the case of GBDT, the weak learner selected are simple decision tree, while hyperparameters can be set to prune the fitted decision trees. Lastly, the concept of additive model specifies how GBDT models perform gradient boosting. Initially, one decision tree is fitted for the dataset which will allow us to compute the predictions and then the cross-entropy loss. Subsequently, another decision tree is fitted to further minimize the loss. This can be achieved through setting the decision with an initial set of parameters and shifting the parameters to the correct direction (negative gradient of the loss) that will reduce the loss function. Therefore, each fitted decision tree will try its best to account for the instances that are not classified in the previous tree. Eventually, the output of all the weak learners are aggregated to produce predictions.

7. Result

In the literature review part, we decided to use classification models like logistic regression, gradient boosting decision tree to classify our dataset. We start with using the logistic regression model.

We started by fitting all explanatory variables (except the stock name and PriceVar which are irrelevant to our analysis) against the “Class” variable. This should provide us with some initial intuition about the effectiveness of each explanatory variable. The fitted model shows that most of the explanatory variables are insignificant (with a p-value greater than 0.05). The significant variables shown by the model are “Rev”, “GM”, “NCF”, “FCF”, “AT”, “GPG”, “OIG”, and “OCFG”. Beyond that, the sector indicator variables are also showing good p-values, leaving us to believe that the sector distribution can also be useful for explanation.

To validate this hypothesis, another logistic regression model is fit using the variables described above. The fitted model shows good significance for all variables. Still, to validate our decision of removing the insignificance variables, we need to use to likelihood ratio test, which compares the likelihood of the data under the full model against the reduced model. We choose ANOVA test to conduct this likelihood ratio test. Nevertheless, the ANOVA test shows very low p-value, thus we are highly confident in rejecting the null hypothesis that the two models show the same performance. This suggests that at least one of the explanatory variables removed is significant. Since the none of the explanatory variable alone shows high significance, this finding leads us to the hypothesis that some of the explanatory variables are highly correlated.

Reducing the explanatory variables as described above is inappropriate, but we also suffer from high dimensionality in the first logistic regression model we fit. With high dimensionality, logistic regression models tend to overfit the data, thus rendering high variance in the predictions in respect to the classification error rate. Still, we decide to test the performance of our first model (full model) on test dataset. Once again, the full model is trained on 70% of the total data while the remain 30% serves as the test data. Prediction on the test data using this fitted model gives an accuracy of 53%. Therefore, we conclude initially that the model is able provide some initial insights into our dataset yet is unable to give accurate predictions. We need another model that may work better with collinearity and high dimensionality.

Gradient Boosting Decision Tree can be a good substitute. One essential quality of GBDT is that it is more robust to collinearity than logistic regression model. In the case of several variables that are highly correlated with each other, each weak learner may just pick on the explanatory variable that shows the greatest contribution and ignores the rest. Also, with the aid of cross validation technique, we are also able to control the issue of overfitting quite handily.

In order to facilitate modelling the GBDT model, we decided to use the LGBMClassifier method in the lightgbm package of Python. The package was developed by Microsoft to provide a convenient and efficient GBDT implementation in Python. We also used the GridSearchCV facility provided by the sklearn.model\_selection package to tune the hyperparameters for the best performance of our model. Also, since lightgbm does not currently support categorical attribute as input, we drop the variable “Sector” for our model. The first step we took is to separate 70% of our dataset into the train set and rest being the test set. The GridSearchCV package allows us to choose several values for each hyperparameters and test the performance of each possible combination. For instance, the set of hyperparameters our group chose to test are [2, 3, 5, 10, 50] for the maximum depth (max\_depth) of individual decision tree and [2, 3, 5, 10, 50] for the number of estimators (n\_estimators) we want to fit. The GridSearchCV then searched for the set of hyperparameters that gives the best performance on the test dataset, which is when max\_depth = 50 and n\_estimators = 2. Using this set of hyperparameters, we fit the model which gives a prediction accuracy of approximate 60% on both the training data and the test data. This result demonstrates a good generation ability on this model. Nevertheless, simply ignoring the “Sector” variable means ignoring one significant feature. The next step we took is to account for the loss of information by ignoring the “Sector” variable.

We achieved this through transforming the “Sector” variable into a set of dummy variables where having a 1 on the dummy variable indicates that the sample belongs to this sector; and 0 otherwise. Even though introducing these dummy variables increases the dimensionality once again in our dataset, we still acquire a 0.5% to 1.5% increase in the accuracy of prediction if including “Sector”.

8. Conclusion / Future work

To conclude, we first noticed that the GBDT can not only generate more accurate prediction result but also give predictions on unknown data with less variance. Thus, for to our first research goal—predicting stock price growth over the year—we believe that the GBDT model generated can be most appropriate. Still, logistic regression has the advantage of interpretability. The significant variables we are able to extract from this dataset, such as “Rev”, “GM”, and “Sector”, are all extracted using the logistic regression model we have.

Overall, our study demonstrates some good understanding on the dataset provided and how it may be used to generate prediction result in real world problems. For the next step of our study, we decide to focus on incorporating the time series nature of stock market data to our study. In our study, we mix data ranging from all five years in order to extract information out of it. But in real world, we can only predict stock growth based on historic data. Therefore, we would like to explore more about predicting future growth using historic data.

9. Citation

Didur, Konstantin. “Machine Learning in Finance: Why, What & How.” *Medium*, 11 July 2018, <https://towardsdatascience.com/machine-learning-in-finance-why-what-how-d524a2357b56>.

Carbone, Nicolas. *200+ Financial Indicators of US Stocks (2014-2018)*. https://kaggle.com/cnic92/200-financial-indicators-of-us-stocks-20142018. Accessed 3 Dec. 2021.

Garg, Rohit. “7 Types of Classification Algorithms.” *Analytics India Magazine*, 7 Oct. 2021, analyticsindiamag.com/7-types-classification-algorithms.

LaValley, Michael. *Logistic Regression | Circulation*. https://www.ahajournals.org/doi/full/10.1161/CIRCULATIONAHA.106.682658. Accessed 3 Dec. 2021.

Friedman, J. (2001). *Greedy boosting approximation: a gradient boosting machine*. *Ann. Stat*. 29, 1189–1232. doi: 10.1214/aos/1013203451

Firke, Sam. *Janitor: Simple Tools for Examining and Cleaning Dirty Data*. 2021, https://CRAN.R-project.org/package=janitor.

Hester, Jim, and Hadley Wickham. *Readr: Read Rectangular Text Data*. 2020, https://CRAN.R-project.org/package=readr.

Bache, Stefan, and Hadley Wickham. *Magrittr: A Forward-Pipe Operator for R*. 2020, <https://CRAN.R-project.org/package=magrittr>.

Templ, Matthias, and Alexander Kowarik. *Imputation with the {R} Package {VIM}*. 2016, pp. 1--16, <https://doi.org/10.18637/jss.v074.i07>.

Hadley, Romain, Lione, and Kirill Müller. dplyr: *A Grammar of Data Manipulation*. 2021, <https://CRAN.R-project.org/package=dplyr>

Philipp, Thomas, and Jan Schulz. *repr: Serializable Representations*. 2021, <https://CRAN.R-project.org/package=repr>

Wickham, Hadley. “Ggplot2: Elegant Graphics for Data Analysis.” *Springer-Verlag New York*, <https://ggplot2.tidyverse.org>.

Terry and Beth Atkinson. rpart: Recursive Partitioning and Regression Trees. 2019, <https://CRAN.R-project.org/package=rpart>

Williams, Graham J. Data Mining with {Rattle} and {R}: The art of excavating data for knowledge discovery. Springer. 2011, http://www.amazon.com/gp/product/1441998896/ref=as\_li\_qf\_sp\_asin\_tl?ie=UTF8&tag=togaware-20&linkCode=as2&camp=217145&creative=399373&creativeASIN=1441998896

Wiener, Matthew, and Andy Liaw. “Classification and Regression by RandomForest.” *Springer-Verlag New York*, vol. 2, 2022, pp. 18–22.

Milborrow, Stephen. rpart.plot: Plot 'rpart' Models: An Enhanced Version of 'plot.rpart'. 2020, <https://CRAN.R-project.org/package=rpart.plot>

Kuhn, Max. caret: Classification and Regression Training. 2021, <https://CRAN.R-project.org/package=caret>

Silge, Julia, and Chow, Fanny, Et al. rsample: General Resampling Infrastructure. 2021, <https://CRAN.R-project.org/package=rsample>

Chen, Tianqi, and He, Tong, Et al. xgboost: Extreme Gradient Boosting. 2021, <https://CRAN.R-project.org/package=xgboost>

Wickham, Hadley, and Averick, Mara, Et al. Welcome to the {tidyverse}. vol. 4, 2019, pp. 1686. doi: 10.21105/joss.01686

McKinney, W., Et al. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference.* Vol. 445, pp. 51–56. *2010*

Pedregosa, F. **and** Varoquaux, G., Et al. Scikit-learn: Machine Learning **in** {P}ython. Vol. 12, pp. 2528-2830. 2011

*Welcome to LightGBM’s Documentation! — LightGBM 3.3.1.99 Documentation*. https://lightgbm.readthedocs.io/en/latest/. Accessed 3 Dec. 2021.