A Development & Evaluation of a Machine Learning Based Value Investing Methodology

Jun Yi Derek He
Academy of Science and Technology
The Woodlands College Park High School
The Woodlands, TX
derekhe99@yahoo.com

Joseph Ewbank
Academy of Science and Technology
The Woodlands College Park High School
The Woodlands, TX
jewbank@conroeisd.net

Abstract – *Throughout most of the attempts to develop machine learning algorithms to accurately forecast stock market behavior, the majority of approaches are plagued with scalability, consistency, and profitability issues. The present work comprehensively developed and evaluated 4 machine learning algorithms to overcome these problems using fundamental analysis. Random Forest was found to obtain the highest return on investment, at a roughly 23.60% annual gross return over 12 test years. However, it suffers from a relatively high volatility, while a 3-model ensemble approach achieves slightly lower annual gross return but a lower volatility. An innovative process involving ranking and identifying the top 20 stocks to invest in based on the algorithms' confidence was found to enhance returns by approximately 30%. Metrics such as precision as well as combining feature importance ranking with feature distribution visualization suggested that the out-of-sample data may be reasonably used for forecasting. Decision tree visualization allows for potentially novel insights into how to perform fundamental analysis, benefitting the knowledge base as a whole. Finally, the returns of the algorithms were found to outperform most mutual and hedge funds in the real world. Overall, the machine learning algorithms have been suggested to outperform benchmarks and most human investors while being scalable for reliable future usage.*

1 Introduction

1.1 Investing Background

There are 4 main categories for investing analysis: technical, fundamental, top-down, and bottom-up (Chen, 2019). Actual implementation may vary, where investors or portfolio managers may choose from the following styles of investing: growth funds, value funds, income funds, tax-efficient funds, tax-exempt funds, market capitalization funds, and index funds (Kennon, 2020).

It has been suggested that value investing can be the most effective form of investing if the investor is competent and executes the strategy without emotional bias (Riley, 2010). Many skeptics of value investing argue that it underperforms against other strategies during many time periods. However value investing usually outperforms if given enough discipline, skill, and time (Riley, 2010).

The underlying principle behind value investing is to purchase stocks when the investor believes it is undervalued by the market, so that its price will eventually appreciate to its minimum intrinsic value over time, in which it can be sold for a profit (Riley, 2010). An investor uses the company's fundamental data found in their income statement, balance sheet, and cashflow statements to estimate its intrinsic value. Different value investors choose to analyze companies differently, because they make different assumptions, judgements, and projections based on the fundamental and business data.

Below is an example of a value investing situation, where an underpriced company in 2015 reaches its estimated intrinsic value 3 or more years later.

Figure 1: Value Investing Example [base image obtained from yahoo finance]

Given that value investing has historically been suggested to be highly lucrative and consistent, and that it requires specific financial knowledge and expertise that most individuals do not have, it would be useful to develop machine learning models that learn value investing so anyone can use the algorithm and benefit from the investing technique.

1.2 Relevant Work

There has been relatively little attention in academia to utilize computers and data science to understand and optimize value investing. Below is an overview of the research the author has surveyed.

One of the most comprehensive related works is by Rasekhschaffe & Jones in 2019. The researchers used multiple machine learning algorithms, ensembles, and feature generation to identify stocks that underperform and outperform the overall market. Some limitations of the research was that their data was imbalanced because they defined the prediction targets as either outperform or underperforming the overall market. Also the targets do not discriminate between large outperformance and small outperformance. The returns of the algorithms also were not compared to the overall market return, so we aren't given context of the algorithms' performance.

A similar work was conducted by Quah in 2008. The work analyzed soft-computing models on picking stocks from a pool of 30 U.S. stocks as defined by the Dow Jones Industrial Average from 1995 to 2016. They defined the prediction targets to stocks that have an 80% price appreciation within one year, which the researcher stated created an imbalanced dataset that they attempted to resolve using oversampling. The researchers identified optimal training parameters and algorithm evaluation techniques that achieved returns that were above the average return (32.63% compared to 22.99% and 14.93% compared to 11.22% on different test datasets) and an above guessing-level accuracy for stock picking (61.66% and 74.85% respectively). The limitations include the fact that DJIA has only 30 stocks at one time, meaning the data that was used for training could not be representative of the entire ideal sample space. Additionally, the DJIA tracks some of the largest 30 stocks in the U.S. economy, which are known to not have the highest returns due to their large market capitalization.

Another work was conducted by Sim, K., Gopalkrishnan, V., Phua, C., & Cong, G. in 2014. The researchers used 3D Subspace modeling with stock fundamental data to generate models that can identify profitable stock to purchase, based on the criteria of: average return, sortino ratio, and percentage of stocks bought. The results suggested that certain models could consistently outperform Graham's method of value investing. The forecast horizons were defined on an annual basis and thus relatively could confound value investing with short term market trends.

1.3 Contributions

1) Obtain high returns, low volatility, and algorithm accuracy
2) Utilize data from a large time frame and a large selection of stocks
3) Establish prediction targets that are balanced and discriminate large outperformance from small outperformance.
4) Build prediction targets that emphasize long term price change due to valuation correction as opposed to short term market trends
5) Gather evidence using various metrics to suggest the model can reliably predict the profitability of future stock samples
   a. Precision of In-Sample data and Out-Sample data comparison
   b. Data Distribution Analysis

2 Experimental Setup

2.1 Dataset Description & Preprocessing

The dataset was obtained from gurufocus.com and is comprised of 30 years of fundamental data from 1990 to 2019 on the domestic Consumer Discretionary industry according to the S&P and the Russell 2000, totaling approximately 200 companies and 4000 data entries. This industry was chosen for its large dataset size and because it is domestic.

The prediction targets for any given company at a certain year were either 0, 1, or 2. These were assigned by finding the return of a company three years after the present year in the data. The returns of all the companies in the dataset were separated into three percentiles, and 0 was assigned to the bottom 33 percentile, 1 was assigned to the middle 33 percentile, and 2 was assigned to the top 66 percentile. This allows the models to discriminate a large outperformance from a small outperformance (3rd percentile versus 2nd percentile).

Data whose change in price was greater than 200% were considered outliers and removed from the first two experiments. The third experiment allowed them to be used for training and testing. The dataset had missing values, so data affected by this was removed. The data was scaled proportionally to values between 0 and 1. A sample of the dataset can be found in the appendix.

2.2 Feature Selection, Feature Generation, and Supervised Learning

To prevent overfitting, only a relatively small amount of fundamental data could be used as features to train the model. Based on a combination of references, including Quah, 2018 and Buffet & Clark, 2011, eleven features were selected: Month End Stock Price, PB Ratio, PE Ratio, YoY Rev. per Sh. Growth, Gross Margin, Net Margin %, Retained Earnings, ROE %, Total Current Assets, Current Ratio, and Capital Expenditure.

The researcher needed to determine how many years in each row of data used to train an algorithm should be given as context before the algorithm makes a prediction of how the stock will perform three years later. This was done by adding the previous X-1 years to the subsequent row, where X is the number of years of context to be provided to an algorithm. For example, if 3 years of context is to be provided, then the data attributed to a company from 1990 and 1991 are added to the data at 1992.

Features were generated by describing the percent change of each corresponding feature from the first year in a row from the last year. This helps the algorithms to understand that those features are correlated with each other.

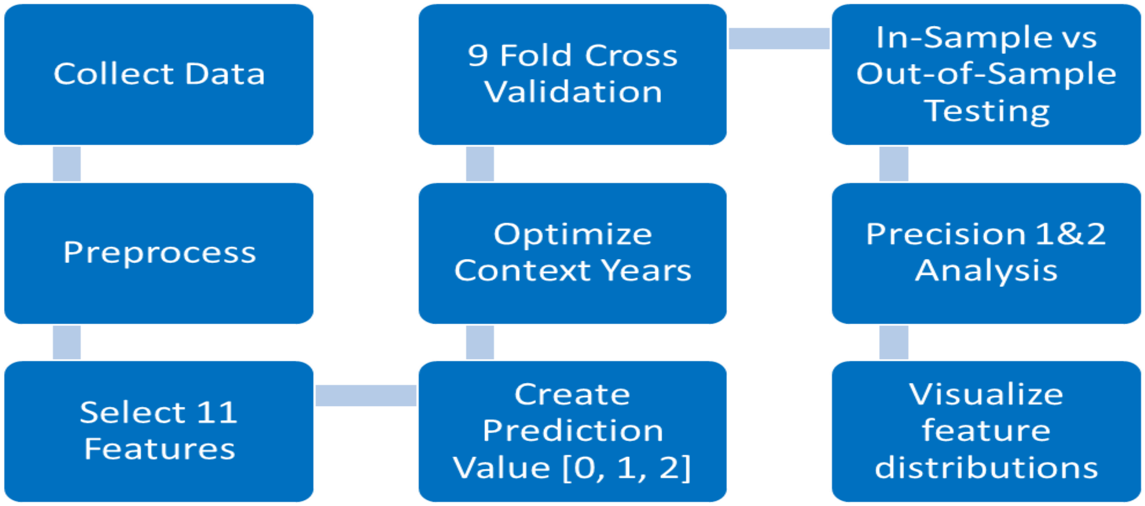

Figure 2: Methodology Overview

2.3 Data Structuring

There were multiple ways the dataset was divided and multiple metrics used to evaluate the models. Below are the four ways the dataset was divided.

2.3.1 Train Test Split

To create the training dataset, 89% of the data were randomly sampled using 42, 24, and 69 as the random states. The remaining data were used as test data. However, the training data which featured a company and year that overlapped with the test data (due to shifting the data to create context) were removed to avoid data leakage.

2.3.2 2013 Split & Train Test Split

All data before 2013 was treated as "in-sample" data, where the train test split method described in section 2.3.1 was used. All data after 2013 was treated as "out-of-sample" data and was used for further validation.

2.3.3 9 Fold Cross Validation & Train Test Split

9 Folds, each containing three years of data starting from 1990 to 2019 were created. 8 folds were used for training at any moment in time and the excluded fold was used for validation. The 8 folds were additionally treated with the train test split method described in 2.3.1. A potential problem with this methodology is that data that occurred after the excluded fold could be used for training, which is unrealistic. The benefit of this method is that years from prior to 2005 are being used to evaluate the model.

2.3.4 15 Year Train to 12 Year Test & Train Test Split

The 15 years between 1990 and 2004 were used for training the algorithms, and the remaining 12 years were used for testing the algorithms. The 15 years for training were also treated with train test split described in section 2.3.1. When comparing the performance on the test portions of the in-sample data to the performance on the train portions, the researcher can ensure that there is not significant overfitting on the in-sample data.

2.4 Testing Criteria

2.4.1 Return

The most important metric is the return of the portfolio on a certain set of data. This is calculated by subtracting the original price from the final price, divided by the original price.

$$price\ change = \frac{|final\ price_{t+3}\ - initial\ price_{t}|}{initial\ price} \quad (1)$$

The forecast horizon is 3 years, so the final price occurs 3 years after the original one. The current year is represented by the variable "t".

2.4.2 Volatility

Volatility was also measured by finding the interquartile range (IQR) of the companies' returns in a given set of data. Standard deviation was not used because it has been repeatedly suggested to not be applicable in the context of stocks, most notably due to its assumption that the return distribution must be normal, which is often times not the case (Adkins, 2020).

IQR however, is appropriate for measuring the variation from a skewed data distribution.

2.4.3 Precision 1 & Precision 2

Precision is a common metric in machine learning, defined as the number of true positives over all data points predicted to be a positive. In this problem, the formula is:

$$precision = \frac{number\ of\ stocks\ that\ are\ class\ X}{number\ of\ stocks\ predicted\ as\ class\ X} \quad (2)$$

Precision 1 is the number of stocks that are class 2 over the number of stocks predicted as class 2. Precision 2 however is the number of stocks that are class 1 or 2 over the number of stocks predicted as class 2. Precision 2 effectively calculates the percentage of predicted stocks that have at least average return, while Precision 1 calculates that for above average return.

2.4.4 Benchmark

A benchmark was also calculated for each year in the dataset to compare the algorithms with. The benchmark values were calculated by conducting the appropriate return, volatility, precision 1, and precision 2 calculations on the entire set of data in a certain year. This effectively represents a portfolio that does not have any rules when making an investment.

2.5 Algorithms

Different algorithms were used to learn from this data. Those were Support Vector Machines (SVM), Random Forests (RF), AdaBoost (Ada), an Artificial Neural Network (ANN), and an ensemble of the first three algorithms. The ANN was found to be unsuitable for this dataset, so the results are not reported.

The grid search for the SVM used between 10 and 1000 for C, 0.01 and 10 for g. The RF had 100 trees that were a maximum of 2 inner nodes deep. The Ada had 15 weak classifiers. The ensemble was built as follows: Each algorithm assigns a predicted class to a test sample, and samples with a combined score of 5 or 6 were assigned a 2, either 3 and 4 were assigned a 1, and between 0 and 2 inclusive were assigned a 0. This reduced the number of 2's assigned, and increased the number of 0's assigned. Intuitively, requiring more algorithms to assign a company a 2 or a 1 before predicting it is class 2, the more likely the sample will have big returns.

2.6 Custom Prediction Methodology

An innovative method for an algorithm to make a prediction is to first assign a confidence value of each class to a certain stock. Next, rank the stocks from greatest to least in confidence for class 2. The top 20 ranked stocks are selected to invest in. This methodology was hypothesized to produce higher returns due to emphasizing investing in companies that the algorithm is more confident will have high returns.

2.7 Data Visualization

The dataset was visualized to understand the distribution of the features according to the 9 Fold Cross Validation technique, i.e. there were 9 different sets of graphs for each of the 9 folds, each with 12 histograms for each feature and prediction target. Different colors were used to represent the test fold and the training folds.

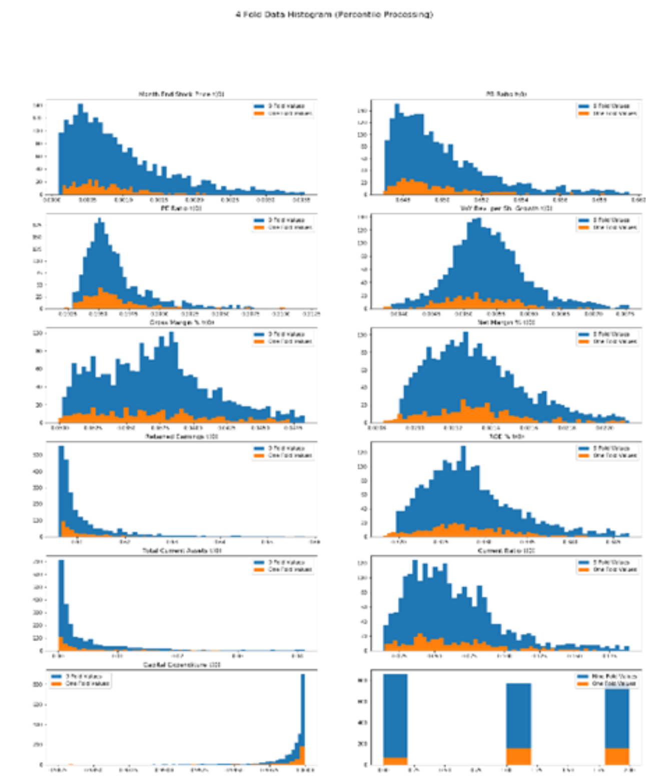

Figure 3: Data Visualization by Fold 4

3 Experimental Results

3.1 Experiment 1: Determining Optimal Context Years

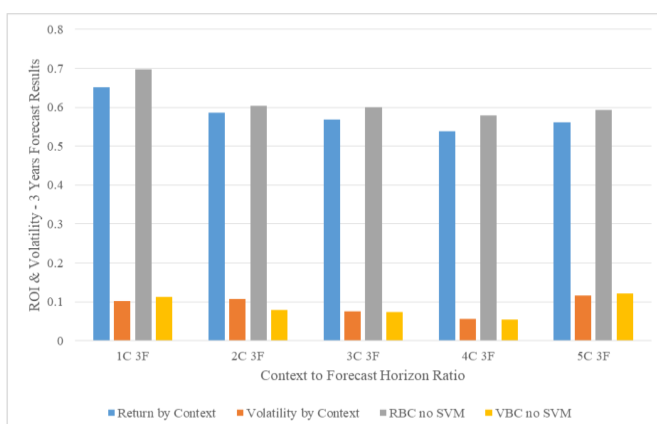

Figure 4: Average ROI & Volatility (Pre2013 and Post2013)

The averages were calculated by finding the mean of the returns of the four algorithms (blue bars) and three algorithms (gray bars) from the in-sample (before 2013) test data and the out-of-sample (post 2013) test data. In other words, sections 2.3.1 and 2.3.2 were combined together in experiment 1, and the average performance was recorded.

3.2 Experiment 2: 9 Fold Cross Validation

Fiures 5.1 to 5.4 illustrate the returns and volatility of the 9 Fold Cross Validation experiment. Each graph recorded the 4 algorithms under investigation, both their normal predictions of all stocks with class 2, and the confident predictions outlined in section 2.6. The means of three trials of the values were recorded, with error bars signaling the standard deviation of each value.

3.3 Experiment 3: 15 Years (1990-2005) Train - 11 Years (2005-2016) Test

The figures 6.1 to 6.3 showcase the returns and volatility obtained from the third experiment. A hypothetical trading situation using the predictions from the algorithms was graphed in 6.3.

Figures 7.1 to 7.4 are the precision 1 and precision 2 values obtained during experiment 3. Figure 7.1 compares the precision 1 of 4 algorithms. Each algorithm has 4 values tracked: the benchmark or average precision, the algorithm's precision on the training data, the algorithm's precision on the "in-sample" test data, and the algorithm's precision on the "in-sample" test data using confidence predictions outlined in section 2.6. The same values were tracked in Figure 7.2 but using precision 2. Figure 7.3 details the precision of the algorithm during the training data, the test data of the out-of-sample data, and the test data of the out-of-sample data using confidence predictions. Figure 7.4 tracked the same values, but using Precision 2.

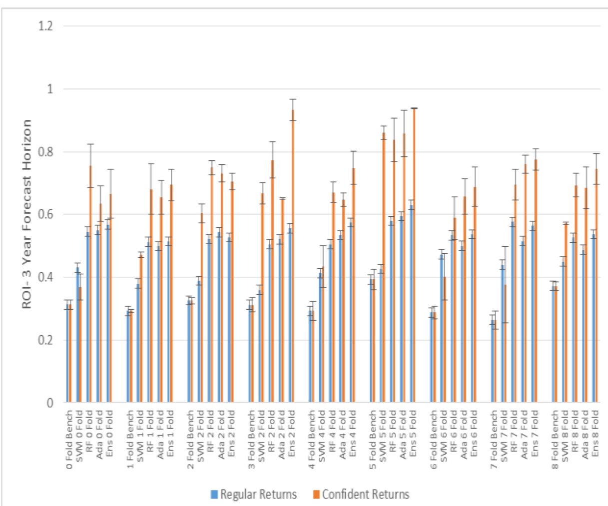

Figure 5.1) Returns of In-Sample Test Data based on Excluded Fold

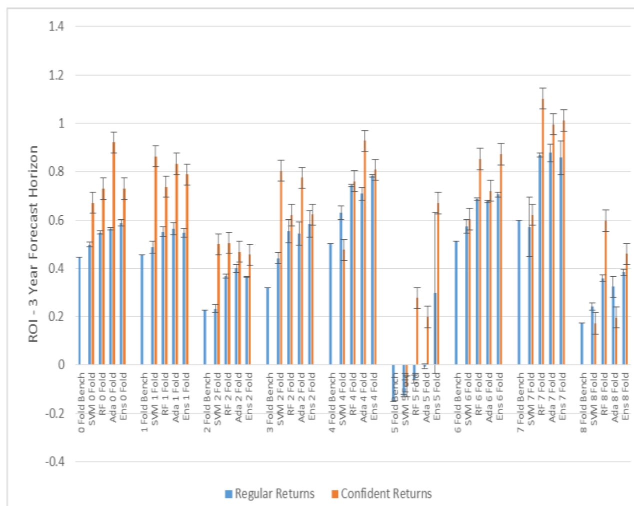

Figure 5.2) Returns of Out-of-Sample (Excluded Fold)

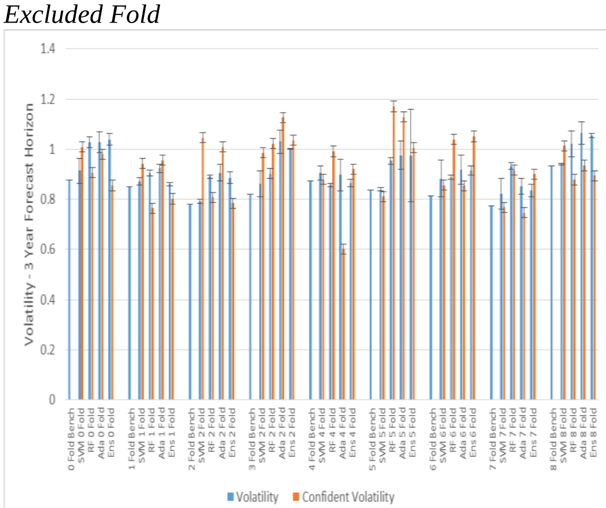

Figure 5.3) Volatility of Test Data based on Excluded Fold

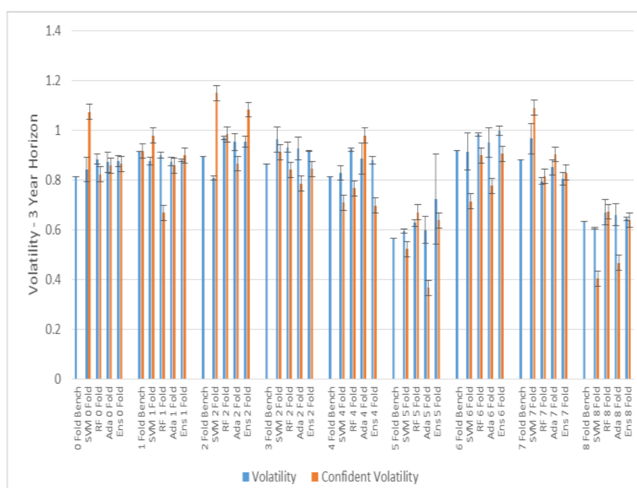

Figure 5.4) Volatility of Out-of-Sample (Excluded Fold)

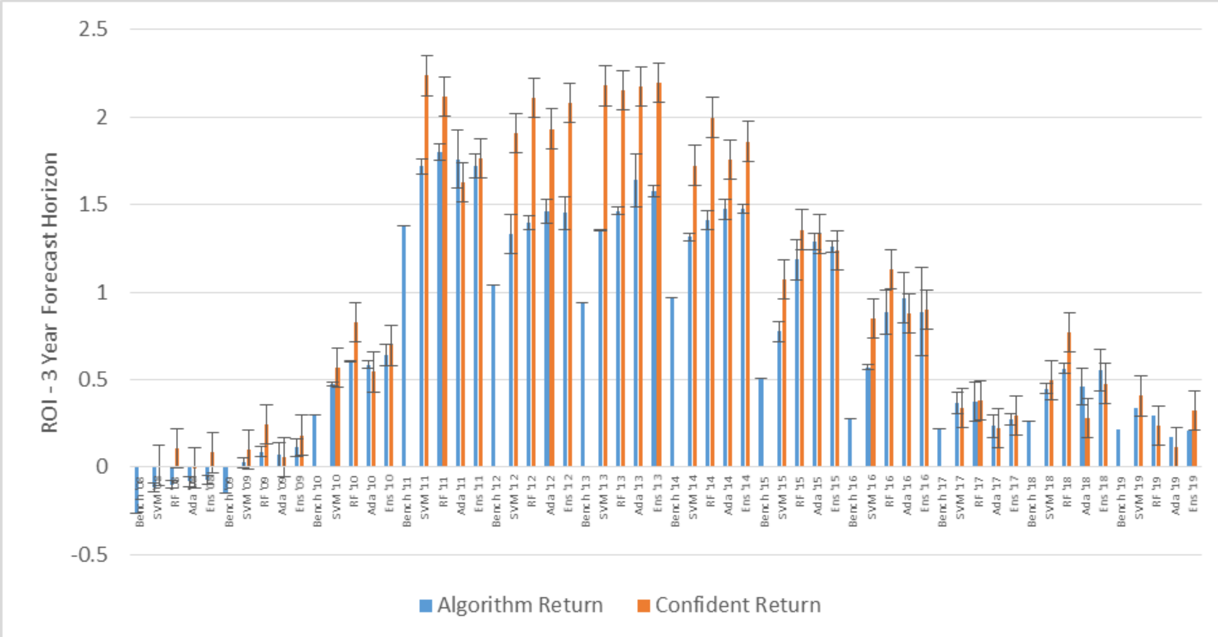

Figure 6.1) ROI of Year Forecasted (3 Year Forecast Horizon)

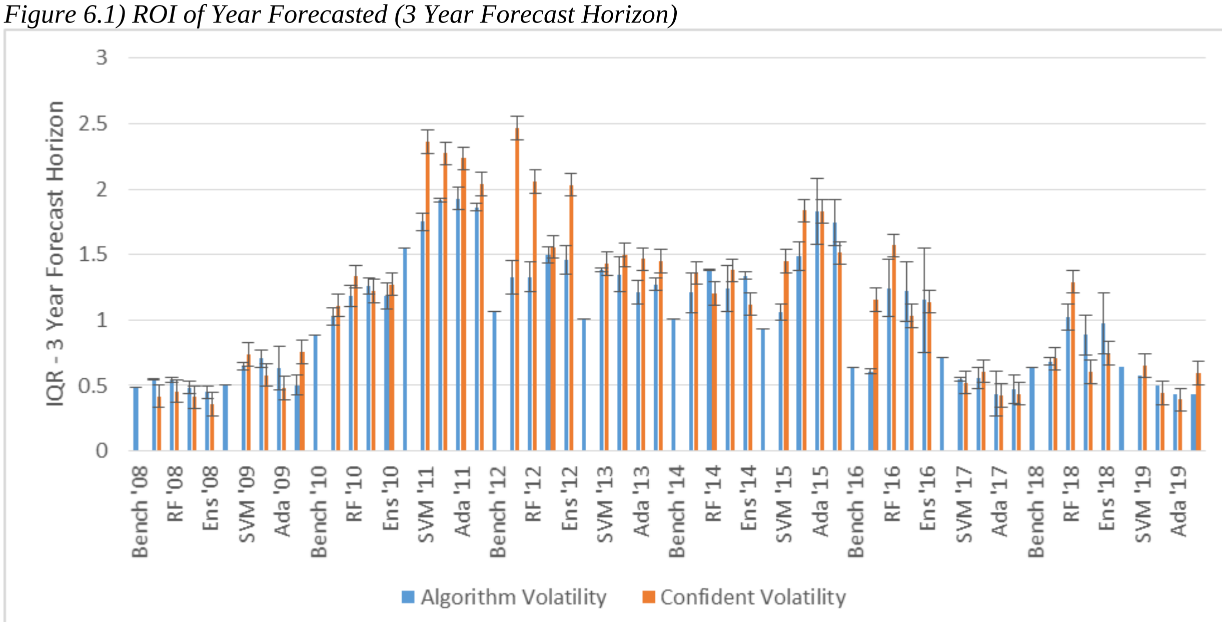

Figure 6.2) Volatility of Year Forecasted (3 Year Forecast Horizon)

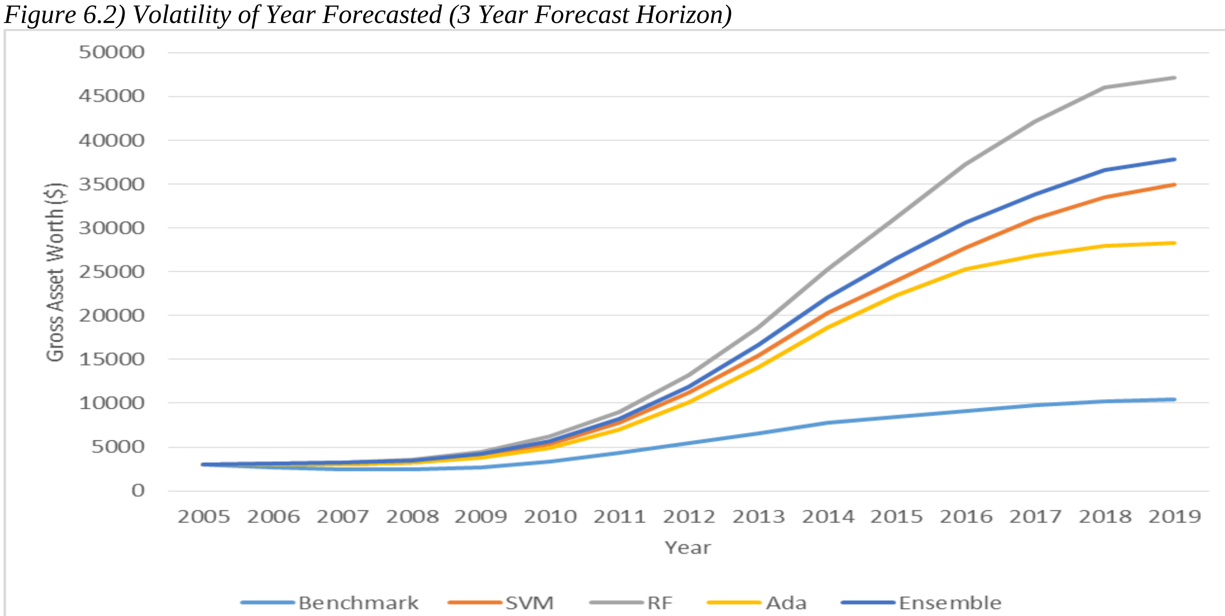

Figure 6.3) Gross Asset Worth of Confident Predictions from 2005-2019 (out-of-sample)

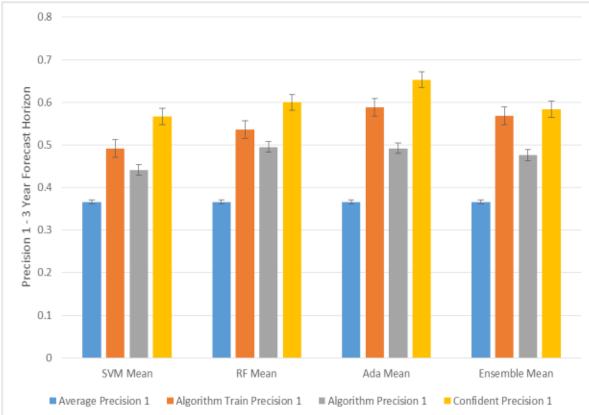

Figure 7.1: In-Sample Precision 1 Comparison

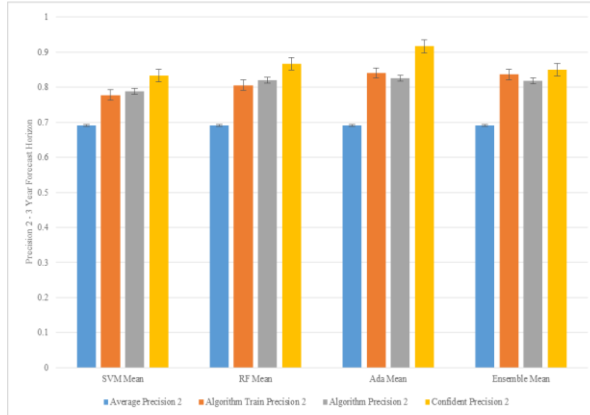

Figure 7.2: In-Sample Precision 2 Comparison

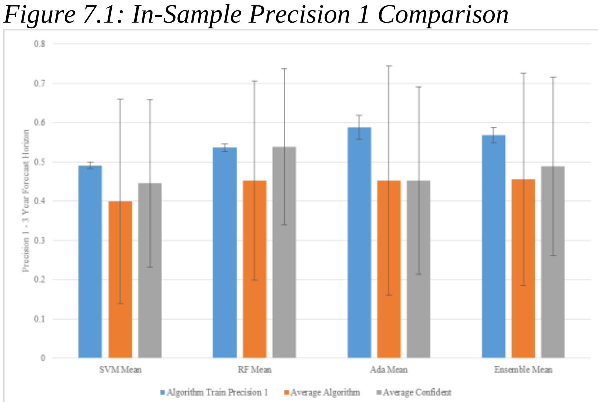

Figure 7.3: In-Sample vs Out-of-Sample Precision 1 Comparison

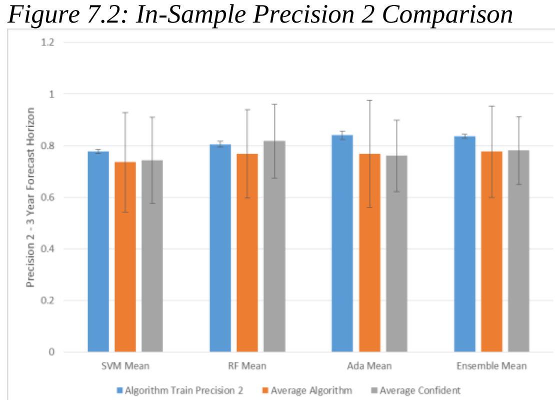

Figure 7.4: In-Sample vs Out-of-Sample Precision 2 Comparison

Below is table that summarizes the hypothetical returns based on figure 6.1 and used to generate figure 6.3.

Year	Benchmark	SVM	RF	Ada	Ensemble
2005	3000	3000	3000	3000	3000
2006	2711.182	3009.139	3101.816	2996.29	3079.913
2007	2489.846	3047.014	3249.594	3011.63	3193.252
2008	2440.495	3245.903	3607.474	3183.901	3486.15
2009	2726.113	3952.971	4497.737	3760.374	4230.448
2010	3351.084	5359.922	6188.841	4944.366	5684.514
2011	4316.025	7834.088	9049.571	7038.166	8194.717
2012	5420.461	11182.33	13167.75	10083.89	11865.62
2013	6615.738	15479.25	18644.55	14128.72	16634.67
2014	7751.905	20272.63	25337.33	18675.52	22035.36
2015	8480.619	24015.78	31121.41	22356.93	26469.13
2016	9131.158	27693.59	37317.95	25248.49	30567.5
2017	9786.677	31007.05	42114.59	26826.21	33879.82
2018	10198.66	33509.63	46032.37	27900.27	36562.24
2019	10399.02	34914.41	47144.75	28257.04	37849.34
Percent Change	246.6341	1063.814	1471.492	841.9014	1161.645
Times Change	2.466341	10.63814	14.71492	8.419014	11.61645
Annual Growth	**0.100344**	0.207789	**0.236016**	0.188293	0.215311

Table 1.1) Gross Asset Worth Confident Predictions from 2005-2019

Below is table that summarizes the hypothetical cumulative volatility based on figure 6.2.

Year	Benchmark	SVM	RF	Ada	Ensemble
2005	3000	3000	3000	3000	3000
2006	3422.267	3370.061	3400.285	3363.955	3320.104
2007	3908.884	3873.888	3889.276	3792.333	3785.3
2008	4587.501	4658.796	4696.9	4506.875	4578.231
2009	5726.135	6150.449	6211.532	5886.205	6027.677
2010	7405.895	8797.019	8805.042	8149.046	8447.02
2011	9662.416	12816.4	12604.25	11441.57	11953.56
2012	12229.32	17924.21	17191.91	15435.81	16258.91
2013	15374.44	24042.62	23300.15	21024.64	21650.79
2014	18917.38	31908.71	31712.95	28070.06	28444.3
2015	22795.15	40319.96	41860.62	35273.87	36024.01
2016	27023.98	48652.21	54109.1	41792.05	43589.7
2017	32042.46	57287.64	65616.43	47541.57	51128.89
2018	35436.86	64901.1	77060.42	52164.29	58401.56
2019	37441.1	68451.98	80271.36	54199.93	61767.82
Percent Change	1148.037	2181.733	2575.712	1706.664	1958.927
Times Change	11.48037	21.81733	25.75712	17.06664	19.58927
Annual Growth	**0.214298**	0.271985	**0.287665**	0.249346	0.261971

Table 1.2) Cumulative Volatility Confident Predictions from 2005-2019

Below is a table that compares the returns of the machine learning algorithms to mutual funds and hedge funds' returns in the past decade. The algorithms use gross returns, while the mutual funds use net, so these are slightly optimistic results.

Type	13D/13G Date	13F/Portfolio Date	# stocks	10-Y Avrg Return %	Q/Q Turnover %	Value ($Mil)
RF	N/A	N/A	20	**~23.60%**	0%	N/A
Ensemble	N/A	N/A	20	**~21.53%**	0%	N/A
SVM	N/A	N/A	20	**~20.77%**	0%	N/A
Hedge Fund A	2/15/2017	9/30/2019	24	**20.10%**	42%	3,380
Ada	N/A	N/A	20	~18.83%	0%	N/A
Mutual Fund A	-	12/31/2019	29	16.10%	12%	3,339
Mutual Fund B	-	12/31/2019	31	14.80%	5%	2,623
Mutual Fund C	11/9/2019	9/30/2019	350	14.70%	4%	24,309
Mutual Fund D	1/10/2020	9/30/2019	144	14.60%	4%	7,506
Mutual Fund E	-	12/31/2019	39	14.50%	6%	5,982
Mutual Fund F	1/17/2019	9/30/2019	48	14.50%	6%	8,135
Mutual Fund G	-	9/30/2019	91	14.20%	7%	41,550
Mutual Fund P	5/8/2015	9/30/2019	895	11.60%	2%	11,998
Mutual Fund Q	11/17/2015	9/30/2019	35	11.50%	1%	459

Table 2: Comparison between proposed machine learning algorithms and mutual and hedge funds in the real world.

Figure 8.1 is visualization of each feature for 3 datasets. The blue represents the feature distributions of the in-sample training data. The orange represents those of the out-of-sample data. The green represents those of the out-of-sample data that any of the four algorithms predicted using the confidence ranking methodology.

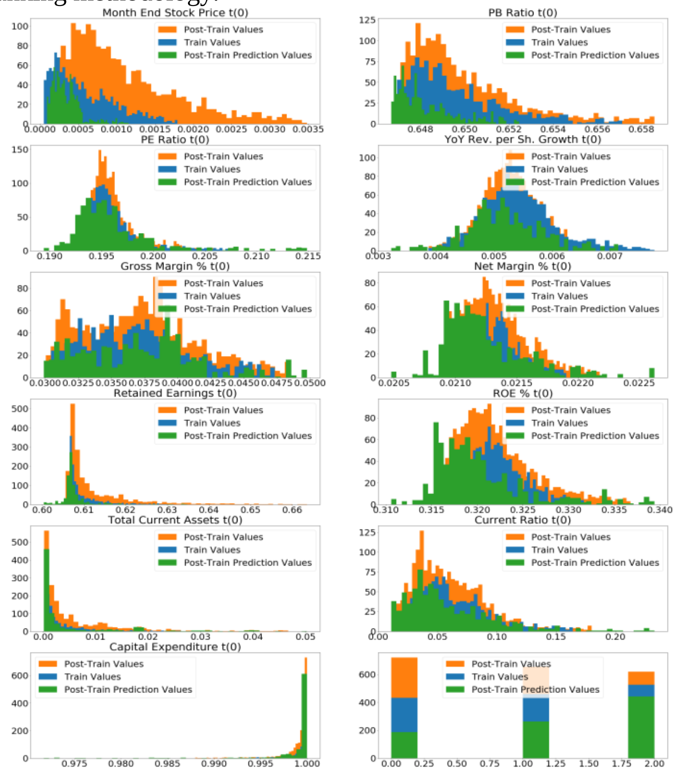

Figure 8.1) Data Visualization of Train vs Out-of-Sample Data vs Out-of-Sample Data Predicted using Confidence Methodology

Below are two graphs that indicate the importance two algorithms associate to the features used for training. These were gathered using the first trial, although the different trials had similar results, so they are excluded.

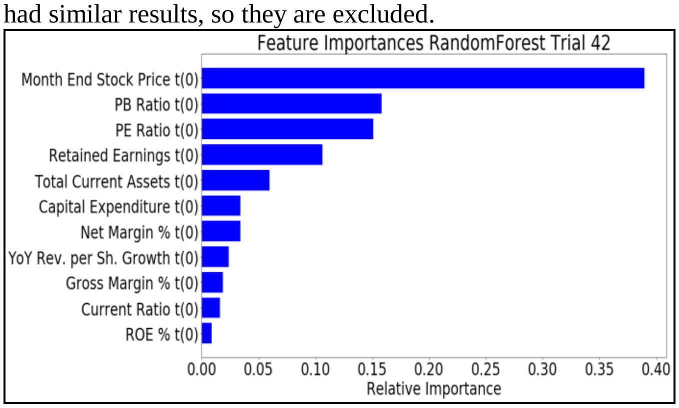

Figure 8.2) Feature Importance Determined by Random Forest

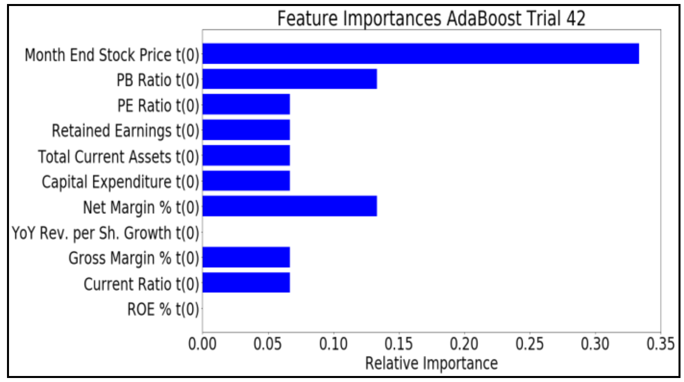

Figure 8.3) Feature Importance Determined by AdaBoost

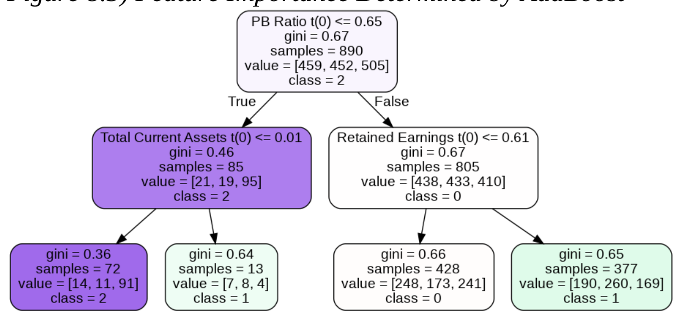

Figure 8.4) Sample Decision Tree from RF

4 Discussion

Experiment 1 indicated that one year of context was optimal for the algorithms to make predictions. Figure 3 shows that the average of the algorithms returns is the highest for one year of context and three years of forecast. This could be because having too many features while reducing the number of examples makes it difficult for algorithms to learn from the data. Additionally, the algorithms are not directly told some of the features are the same features but at different time stamps, which makes the learning process more difficult. This problem could potentially be solved using 3 dimensional data.

Experiment 2 suggested that the ensemble could be the optimal portfolio manager. It had the highest number of years of which it had the highest returns, followed by RF, Ada, and SVM respectively. Ada, however, had the most years of the lowest volatility, followed by SVM, while RF and ensemble are tied for last.

Experiment 3 indicated that RF was the optimal portfolio manager. It achieved the highest return using confidence predictions, with a 23.60% gross annual return. This would turn a $3000 initial investment into $47,144.02, while the ensemble would only return $37,849.33. The ensemble had the second best return overall, and it had the best return using only the algorithm (without confident predictions). The benchmark has the lowest cumulative volatility, with an average of 21.42% annual volatility. The RF had the highest cumulative volatility with an average of 28.77%. The ensemble had the second lowest volatility, with a 26.19% annual volatility. However, due to the outlier nature of the RF and ensemble, this may not be a bad result. To illustrate, a 75^{th} percentile being at a 200% price change and a 25^{th} percentile being at a 50% change may not be a bad portfolio result, while being represented as volatile.

Figures 7.1 to 7.4 compare various precision measurements on different datasets. Foremost, from figure 7.1, it can be seen that RF shows the least deviation from the training precision and the in-sample testing (not using confidence predictions). This implies it is the most able to generalize to unseen data within the training sample. From figure 7.2, a similar conclusion can be drawn, but using precision 2. Figure 7.3 indicates that the RF has the least deviation from the training precision compared to the out-of-sample test set, also implying the algorithm is able to scale to unseen data. The high error bars are from the inherent volatile nature of the test years, including the bear market in 2008 and the bull market proceeding that.

Table 2 compares the machine learning algorithms to funds currently in operation, although they almost exclusively serve accredited and wealthy investors. The gross returns of the RF and ensemble outperform the net of the hedge fund A. It is difficult to estimate the net return of the algorithm considering the intricacies of tax rules, so these results are only approximations. However, it can be concluded that the algorithms overall perform in the top percentile of most mutual and hedge funds.

Figure 8.2 shows the relative feature importance the RF used to make its predictions in one trial. The top 4 factors were Month End Stock Price, PB Ratio, PE Ratio, and Retained Earnings across all trials. Figure 8.3 shows the relative feature importance Ada used to make its predictions in one trial. Ada preferred Month End Stock Price, PB Ratio, Net Margin %, and other factors were equally weighted.

This information is useful for understanding figure 7.1. The distribution of the most important features from the companies that were predicted to be most profitable in the out-of-sample data tend to fall within the distribution of the training data. This implies that the algorithms are confident that companies in the future will be class 2 if the features are also near the data they were trained on. Thus, out-of-sample data perhaps can be reliably predicted to be profitable if the algorithm is confident the company is class 2 and the features fall within the distribution of the original training data. Conversely, if features are out of the training distribution (as seen by the Post Train Values in orange being outside the scope of Train Values in blue), then they may be harder to predict.

Figure 8.4 contains one sample out of 100 decision trees in an RF. This allows the investor to understand the thinking process of the RF before it performs bootstrap aggregation. Eventually, the 100 decision trees can be studied to gather new insights for investing.

Overall, the researcher concluded that the random forest and ensemble are strong candidates for individuals who would like to utilize machine learning to help aid in identifying stocks that will likely appreciate in price according to the time-tested value investing principle. Near state-of-the-art returns were obtained, and evidence was gathered that the results are scalable to future years.

5 References

[1] Adkins, Troy. “Calculating Volatility: A Simplified Approach.” Investopedia, Investopedia, 25 Jan. 2020, www.investopedia.com/articles/basics/09/simplified-measuring-interpreting-volatility.asp.

[2] Buffett, M., & Clark, D. (2011). Warren Buffett and the interpretation of financial statements the search for the company with a durable competitive advantage. London: Simon & Schuster.

[3] Chen, J. (2019, April 23). Investment Analysis: The Key to Sound Portfolio Management Strategy. Retrieved from https://www.investopedia.com/terms/i/investment-analysis.asp

[4] Kennon, J. (2020, January 13). What Is an Equity Fund? Retrieved from https://www.thebalance.com/what-is-an-equity-fund-357477

[5] Keywan Christian Rasekhschaffe & Robert C. Jones (2019) Machine Learning for Stock Selection, Financial Analysts Journal, 75:3, 70-88, DOI: 10.1080/0015198X.2019.1596678

[6] Quah, T.-S. (2008). DJIA stock selection assisted by neural network. Expert Systems with Applications, 35(1-2), 50–58. doi: 10.1016/j.eswa.2007.06.039

[7] Sim, K., Gopalkrishnan, V., Phua, C., & Cong, G. (2014). 3D Subspace Clustering for Value Investing. IEEE Intelligent Systems, 29(2), 52–59. doi: 10.1109/mis.2012.24

[8] Riley, E. (2010, June). Advisor Today. Advisor Today.

6 Appendix

Date t(0)	Company Name t(0)	Month End Stock Price t(0)	PB Ratio	PE Ratio	YoY Rev. per Sh. Growth	Gross Margin	Net Margin %	Retained Earnings	ROE %	Total Current Assets	Current Ratio	Capital Expenditure	Month End Stock Price (t+3)	Percent Change	Price Category
15-Dec	NWSA	11.35	0.570495	37.83333	-2.71	42.98	2.16	150	1.5	3889	1.59	-256	13.49	0.188546	1
15-Dec	NWS	11.67	0.58658	38.9	-2.71	42.98	2.16	150	1.5	3889	1.59	-256	13.96	0.19623	1
15-Dec	CNTY	7.78	1.56	15.88	11.33	49.76	8.61	57.171	9.58	33.881	1.46	-18.875	7.39	-0.05013	0
15-Dec	CONN	12.32	0.84	14.67	12.72	48.35	1.91	452.766	5.18	1163	7.96	-63.405	22.23	0.804383	2
15-Dec	CVGI	2.76	1.26	11.5	-2.66	13.43	0.86	-122.431	11.32	308.277	2.68	-14.685	6.92	1.507246	2
15-Dec	ERI	11	1.9	4.55	26.53	37.93	15.86	99.758	54.09	116.179	1.1	-36.762	36.21	2.291818	2

Table 3: Sample Data from 2015