EC4308 GROUP PROJECT

IMPROVING BULL/BEAR MARKET PREDICTIONS WITH ML TOOLS

Chan Zhe Ying A0167629N
Daniel Ng Jun Rong A0183604H
Leong Kar Hoe, Edmund A0166810H
Wong Junying Jolene A0173783U
Daniel Adipranoto A0170547A
Hung Ding Liang A0169959Y

Table of Contents

1. Introduction

The stock market is a complicated network with many different elements that interact to affect prices, causing it to move in a rather unpredictable manner. A bull market refers to a market where investment prices are on an upward trend for a sustained period of time, and this is usually an indication of a thriving economy where conditions are favourable, accompanied by rising investor confidence. Bull markets are often further propelled by economic strength and low unemployment rates, resulting in a buyer's market. On the contrary, a bear market refers to one where an economy is on a decline and stock prices are decreasing in value. This occurs usually in a time of economic slowdown and higher unemployment, with low investor confidence, resulting in a seller's market. As a result, how the stock market performs has major implications on an investor's portfolio. It is important to be able to forecast bull/bear markets accurately as wrong predictions are extremely costly.

However, market prediction is inherently tricky – one can never be too sure about what happens in the future and how markets would respond. Moreover, while bear markets are usually referred to when stock prices drop 20% or more for a sustained period of time, there is no universal definition of a bull/bear market (Gonzalez, Powel, Shi and Wilson, 2005). Hence, using various machine learning (ML) techniques such as penalized logistic regression, bagging and boosted trees, we aim to analyze if these ML methods would be able to best predict stock market performance and generate the best return, as compared to our benchmark model of the dynamic probit autoregressive model.

For this paper, we seek to conduct classification predictive modelling to predict bull/bear markets for the US stock market based on the S&P 500 index. We examine dynamic macroeconomic and financial trends and variables to forecast the probability of a bull/bear trend. For our research, we follow the classification of bull/bear trends as identified by the R package '*bbdetection*'. Thereafter, we will compare our models across a set of evaluation metrics and choose the one that performs the best in the right directional accuracy.

Therefore, the objective of this paper is to generate improvements on bull/bear market predictions using machine learning methods. This is because a buying or selling decision in the stock market is a binary decision. We wanted to see if we would be able to generate the informative signals that would help generate excess returns in the stock market over a passive "buy and hold" strategy.

The rest of the paper would be structured as follows: we would introduce the dataset used for our predictions in section 2, followed by giving a brief overview of the models we intend to use in our methodology in section 3. Next, we would look at the evaluation metrics used in determining which model performs best in forecasting the bull/bear market performance in section 4 and move on to analyzing our results in section 5. We would then comment on the limitations and scope for further research, as well as give our concluding remarks in sections 6 and 7 respectively.

2. Data

The dataset used is obtained from FRED-MD and 4 variables, *Stock Variance*, *Book to Market ratio, Net Equity Expansion Ratio,* and *Risk Free Rate* from a study by Goyal and Welch (2007). The FRED-MD dataset consists of monthly data across 8 themed groups, of which interest rates and stock market data are considered for the purposes of our paper. The details of the dataset used are listed below:

Variable	Name	Description
1st Set of Independent Variables (Stock):		
uhatsp	Monthly stock returns	S and P common stock price index: Composite (Fred-MD)
uhatpe	Equity premium	Monthly stock returns - risk free rate
uhatdy	Dividend Yield aka Dividend price ratio	S and P composite common stock: dividend yield (Fred-MD)
uhatep	Earnings price ratio	S and P composite common stock: price-earnings ratio (Fred-MD)
uhatsvar	Stock variance	Sum of squared daily returns on the S&P 500 (Svar Goyal)
uhatbm	Book to market ratio	Ratio of book value to market value for the Dow Jones Industrial Average (b/m Goyal)
uhatntis	Net equity expansion ratio	Ratio of 12-month moving sums of net issues by NYSE listed stocks divided by the total end-of-year market capitalization of NYSE stocks (ntis Goyal)
2nd Set of Independent Variables (Interest Rate):		
uhatfed	Fed funds rate	Effective Federal Funds Rate (Fred-MD)
uhat3ms	3 month Treasury bill rates	TB3MS (Fred-MD)
uhatgs10	10 year treasury rate	GS10 (Fred-MD)
uhat3smffm	3 month treasury - fed fund rate	TB3SMFFM 3-Month Treasury C Minus FEDFUNDS (Fred-MD) Term spread between 3 months treasury and fed funds rate
uhat10yffm	10 year treasury - fed funds rate	T10YFFM 10-Year Treasury C Minus FEDFUNDS (Fred MD) Term spread between 10 year treasury and fed funds rate
uhatAAAffm	AAA - fed funds rate	Moody's Aaa Corporate Bond Minus FEDFUNDS
uhatBAAffm	BAA - fed funds rate	Moody's Baa Corporate Bond Minus FEDFUNDS

uhatts	10 year treasury - 3 month treasury (term spread)	Term spread between 3 month and 10 year treasury bill
uhatAAA	AAA	Moody's Seasoned Aaa **Corporate** Bond Yield 20 years and above(Fred-MD)
uhatBAA	BAA	Moody's Seasoned Baa **Corporate** Bond Yield 20 years and above (Fred-MD)
uhatjunkspr	Default yield spread (Junkspr) BAA - AAA	Difference between BAA and AAA-rated **corporate** bond yields (derive)
uhatdrs	Default return spread AAA - 10 year treasury	Difference between long-term **corporate** bond and long-term **government** bond returns (derive)
uhatinfl	Inflation	CPI: All items (CPIAUCSL Fred-MD)
uhatrf	Risk free rate	(Goyal)

Table 1: Description of dependent and independent variables in dataset

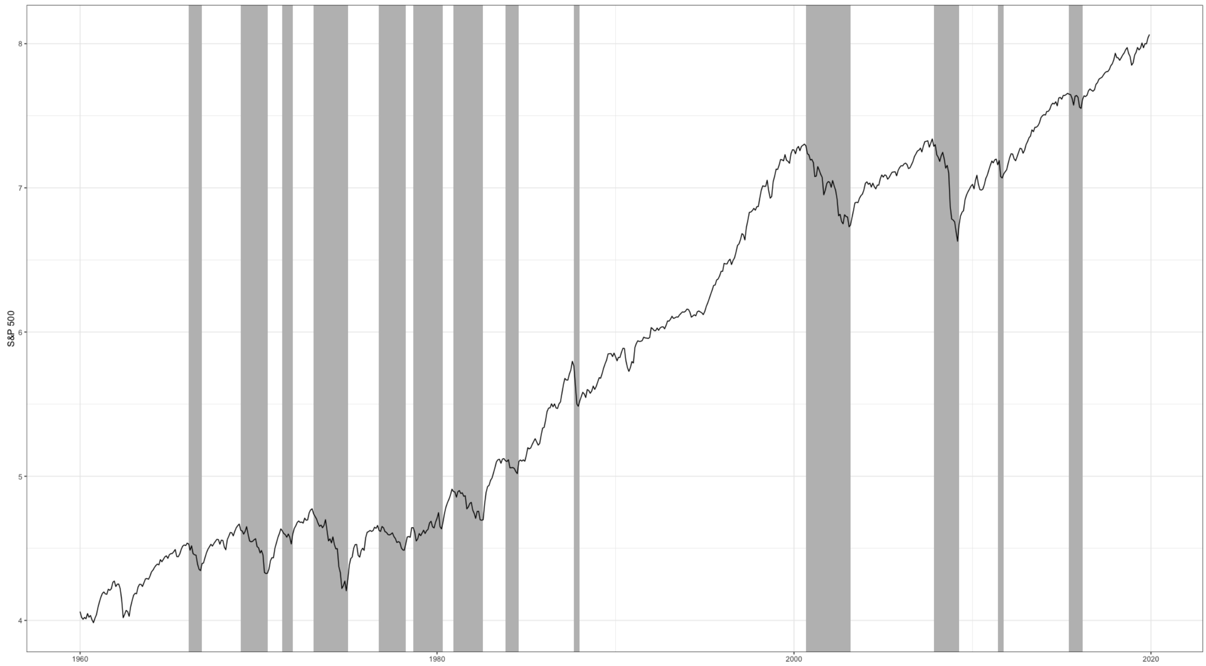

Figure 1: S&P 500 with Bull-Bear Markets (according to bbdetection package)

Figure 1 shows the S&P 500 monthly stock returns along with the identification of the bull and bear states of the market, where the shaded regions are the bear states. Using the *bbdetection* package to detect the bull and bear markets, we were able to classify each data point to a bull/bear market state.

Our data consists of 720 observations (monthly periods) split into 480 used for training our models, and 240 observations, reserved for a testing set to make predictions with our models. After preparing the training and test sets, we performed 1 month, 3 months, 6 months, and 12 months-ahead predictions for the implemented models. We used a 240-month rolling window to maximise the number of relevant data points available for use by our models.

3. Methodology

We examine a variety of machine learning methods in order to determine the best classification model for predicting bull/bear market. This section highlights the four different algorithms that we would be utilizing to forecast.

3.1 Dynamic Probit Model (Benchmark)

We referenced Nyberg's 2013 paper to reproduce the dynamic probit model as our benchmark model to determine the forecast performance for classifying bull/bear markets from monthly stock indices. This model is essentially a modified autoregressive model, with a full vector set of lagged predictors.

3.2 Penalised Logistic Regressions (*rlassologit*)

We utilise penalised logistics regressions to determine potential predictors for our prediction of a bull or bear market. Since we are trying to classify our predictions into bull or bear markets, we run the rlassologit command to obtain our predictors. The function estimates the coefficients of a logistic Lasso regression with data-driven penalty.

3.3 Bagging and Random Forests

We use classification tree models as well to further improve our predictions. We used hyperparameter tuning to find the optimal number of predictors to run the random forest on. Based on the tuneRF package, the suggested number of predictors was 22, which was all the available predictors. Thus, we ran bagging to select our best models.

3.4 Boosting

We implemented a popular ensemble learning method, boosting, which is used to fit small and low variance models. Although individual models do not provide a good prediction themselves, boosting helps improve performance by aggregating these models. The boosting variation that we use in this paper is Gradient Boosting Machines (GBM). GBM is a decision-tree based ensemble machine learning method that trains many models in a gradual, additive and sequential manner.

4. Evaluation Metrics

We examine a list of standard metrics for classification as shown below which we use to compare between the models we have implemented to determine which model best predicts bull/bear markets. This is important as investors would face substantial losses and lose confidence, should the predictions of the stock market be inaccurate. Hence, we seek to choose the model(s) that gives us the lowest false negative rates and false positive rates, which would serve as an indication of its accuracy in predicting stock market performance.

4.1 Sensitivity (True Positive Rate)

The true positive in this paper refers to correctly predicted bull markets. The sensitivity, or true positive rate, then provides the proportion of correct predictions amongst all positive instances. Thus, the higher the sensitivity, the better the model performance.

$$Sensitivity = TPR = \frac{True\ Positives}{False\ Negatives + True\ Positives}$$

4.2 Specificity (True Negative Rate)

The true negative in this paper refers to correctly predicted bear markets. The specificity, true negative rate, then provides the proportion of correct predictions amongst all negative instances. Thus, the higher specificity, the better the model performance.

$$Specificity = TNR = \frac{True\ Negatives}{False\ Positives + True\ Negatives}$$

4.3 False Alarm Rate (False Positive Rate)

The false positive rate provides the proportion of incorrect predictions amongst all negative instances. In our case, it refers to the rate at which a bull market was incorrectly predicted to be a bear market. Thus, the higher the false alarm rate, the worse the model performance.

$$False\ Alarm\ Rate = FPR = \frac{False\ Positives}{False\ Positives + True\ Negatives} = 1 - TNR$$

4.4 False Negative Rate

The false negative rate refers to the proportion of incorrect predictions amongst all positive instances. In other words, it refers to the number of times that a bear market was incorrectly predicted as a bull market. Similar to the FPR, the higher the false negative rate, the worse the model performance.

$$False\ Negative\ Rate = FNR = \frac{False\ Negatives}{False\ Negatives + True\ Positives} = 1 - TPR$$

4.5 Precision-Recall

Precision measures the proportion of true predicted bear markets amongst all predicted bear markets. Recall is equivalent to specificity, which also measures the true positive rate as explained above. The precision-recall curve measures the tradeoff between precision and recall. A high area under the curve represents both high recall and high precision, where high precision relates to a low false positive rate, and high recall relates to a low false negative rate.

$$Precision = \frac{True\ Positives}{False\ Positives + True\ Positives}; Recall = TPR = \frac{True\ Positives}{False\ Negatives + True\ Positives}$$

4.6 Lift

Lift is a measure of the effectiveness of a predictive model calculated as the ratio between the results obtained with and without the predictive model. It compares the proportion of true positives to the proportion of total positive predictions and a higher lift value implies that the model has performed better at distinguishing bull/bear markets compared to another model with low lift value.

$$Lift = \frac{True\ Positive\ Rate}{Rate\ of\ Positive\ Predictions}, \text{ where Rate of Positive Predictions} = \frac{True\ Positives + False\ Positives}{All\ Observations}$$

4.7 'Naive' Accuracy and Balanced Accuracy

'Naive' accuracy measures the percentage of correctly predicted outcomes. Balanced accuracy measures the average accuracy per class of predicted outcome, calculated by averaging the sum of true positive rate and true negative rate.

$$'Naive'\ Accuracy = \frac{True\ Positives + True\ Negatives}{All\ Predictions};\ Balanced\ Accuracy = \frac{Sensitivity + Specificity}{2}$$

4.8 Receiver Operating Characteristics (ROC) & Area Under Curve (AUC)

ROC summarizes the prediction performance of a model by comparing the proportion of true positives to the proportion of false positives. The ROC curve shows the performance of a classification model at all classification thresholds. This curve plots two parameters, True Positive Rate on the y-axis and False Positive Rate x-axis. Each point on the ROC curve represents a sensitivity/specificity pair which corresponds to a particular decision threshold. The ROC curve passing around the upper left corner implies 100% sensitivity and 100% specificity, meaning that the test has perfect discrimination. Thus, the closer the curve is to the upper left corner, the better the accuracy of the test. On the other hand, the AUC measures the entire two-dimensional area underneath the entire ROC curve. It provides an aggregate measure of performance across all possible classification thresholds. One way to interpret AUC is to treat it as the probability that the model ranks a random positive example more highly than a random negative example. Hence, the larger the AUC, the more accurate the model.

4.9 Diebold Mariano (DM) Test

The DM test is used to examine the accuracy of different model forecasts by testing the differences in predictive ability for the respective models. This is done by testing the loss differential between 2 different forecasts. As seen in the equation below, a more positive loss differential between forecasts 1 and 2 would mean that forecast 1 is more accurate. While a more negative loss differential would mean that forecast 2 is more accurate.

$$d_t = L(e^{(1)}_{t+h\,|\,t} - e^{(2)}_{t+h\,|\,t})$$

5. Results

5.1 ROC Curves

Figures 2 to 5 below show the comparisons between the models based on the ROC curves. In Figure 2, all the models perform relatively better than our baseline model (dynamic probit) for 1-step ahead forecast. However, in Figures 3 and 4, we can see that the LASSO Logistic model's ROC curves flatten and approach the 45 degree line and pass the line for Figure 5, indicating a poorer performance of the model compared to the rest as the forecast horizon increases.

On the other hand, the Bagging model's ROC curves remain above the other models' ROC curves for all forecast horizons, indicating its relatively better performance compared to the rest. Thus, by inspection, the Bagging model outperforms other models for the area under the ROC curve metric.

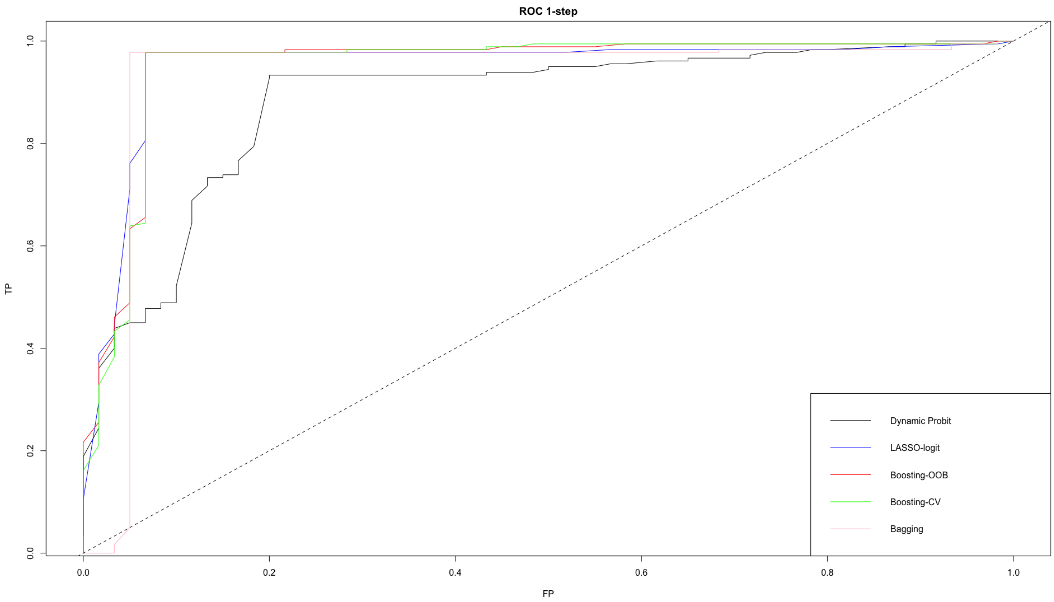

Figure 2. ROC Curve for 1-Step Models

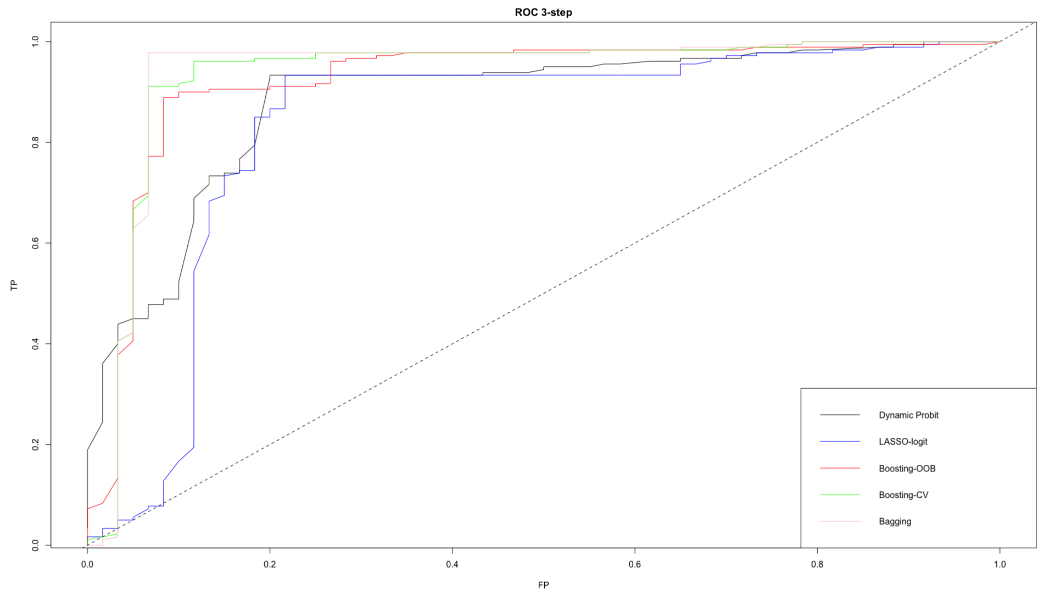

Figure 3. ROC Curve for 3-Step Models

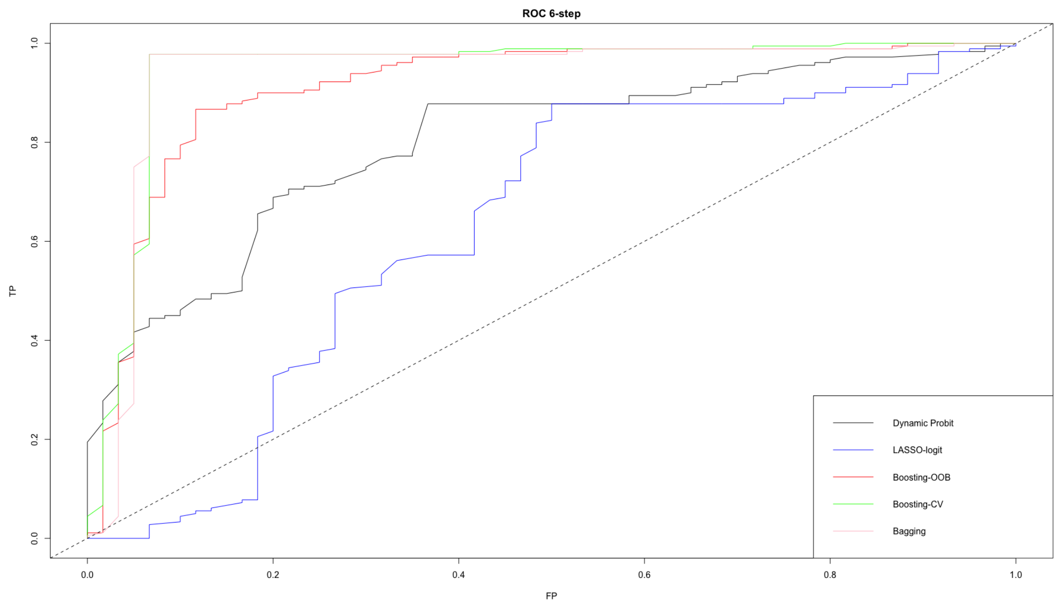

Figure 4. ROC Curve for 6-Step Models

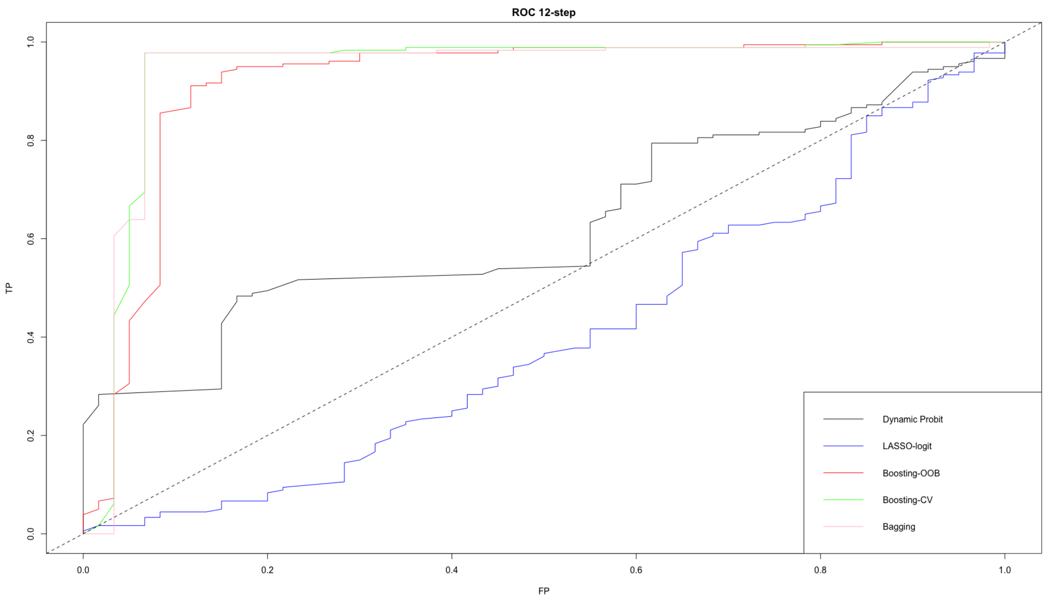

Figure 5. ROC Curves for 12-Step Model

5.2 Lift Curves

Moving on to the Lift curves as seen in Figures 6 to 9, the Bagging model's Lift curves for all forecast horizons remain above other models' Lift curves, indicating its better performance compared to other models. Furthermore, the LASSO Logistic model's Lift curves for 6-step and 12-step forecasts in Figures 8 and 9 flattens and approaches the 45 degree line, indicating that its performance decreases as the forecast horizon increases. Thus, we can see that the results from the Lift curves are in line with the results from the ROC curves, showing that the Bagging model outperforms other models and the LASSO Logistic model performs the worst.

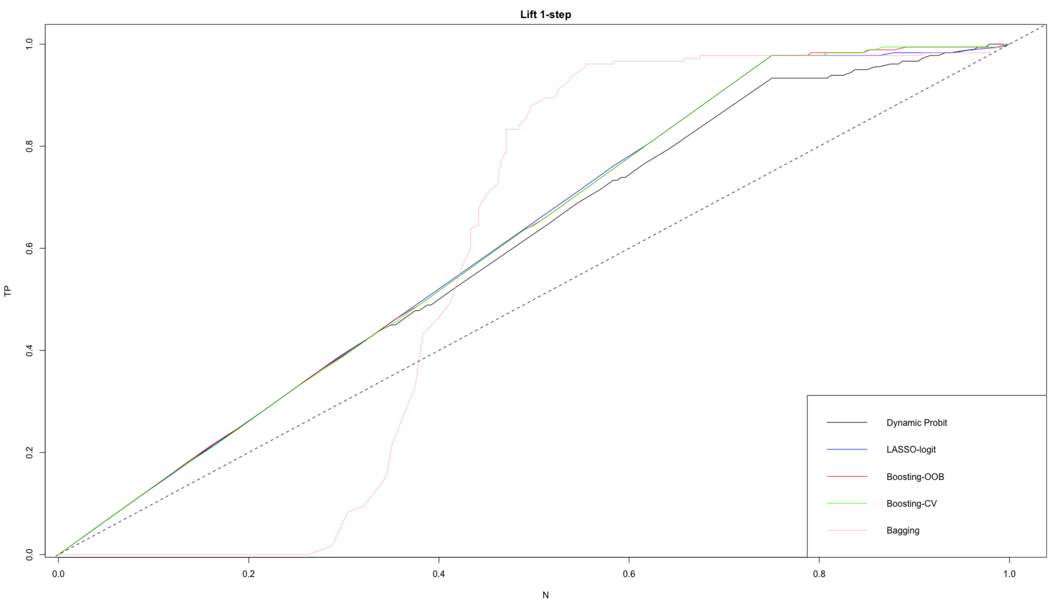

Figure 6. Lift Curves for 1-Step Models

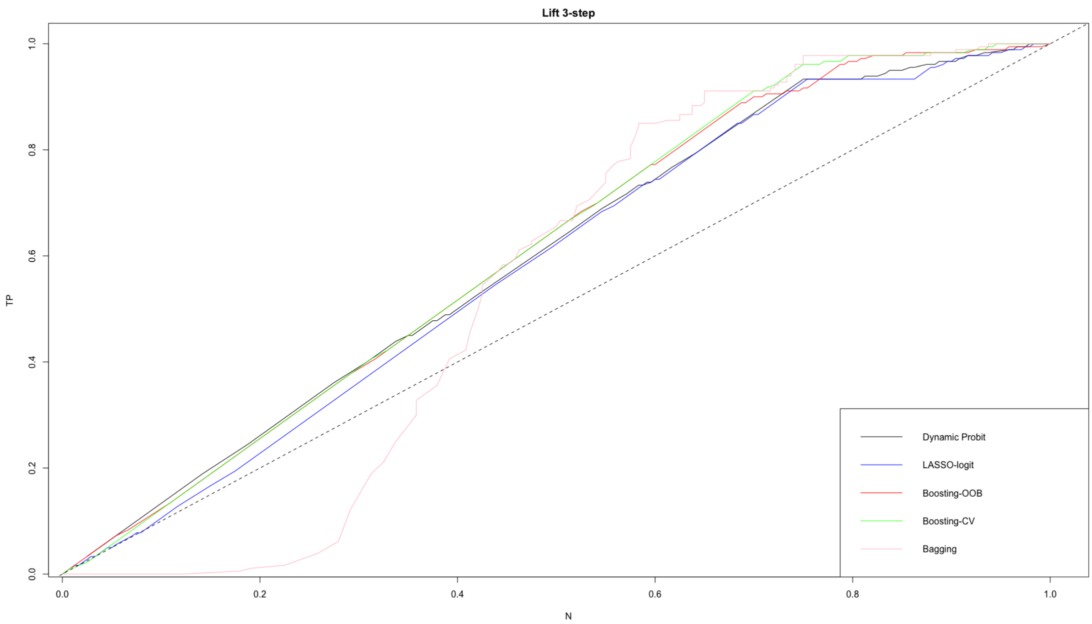

Figure 7. Lift Curves for 3-Step Models

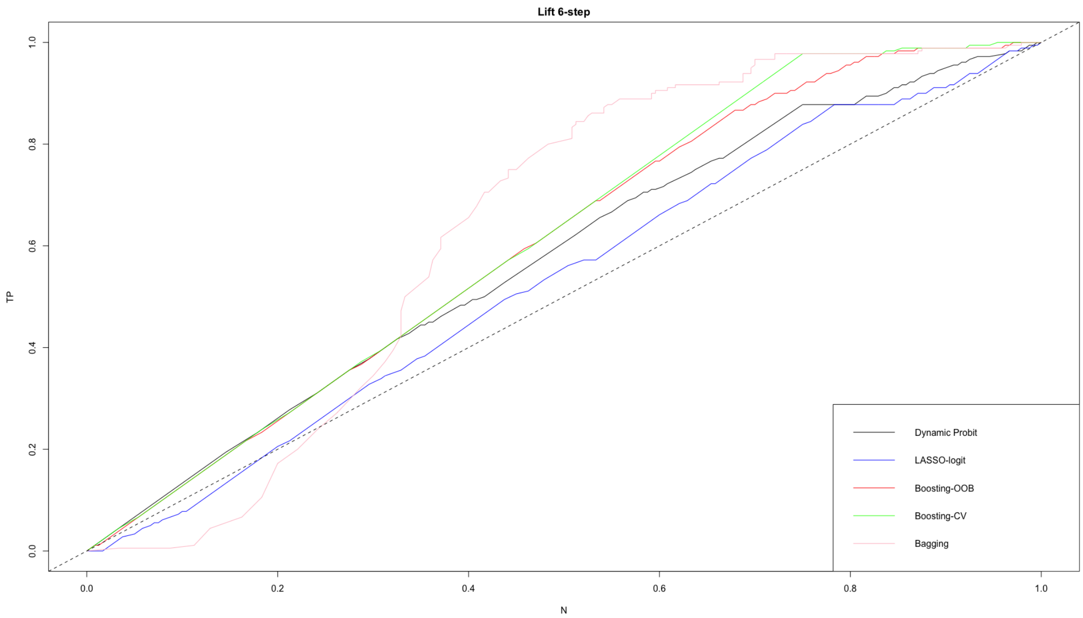

Figure 8. Lift Curves for 6-Step Models

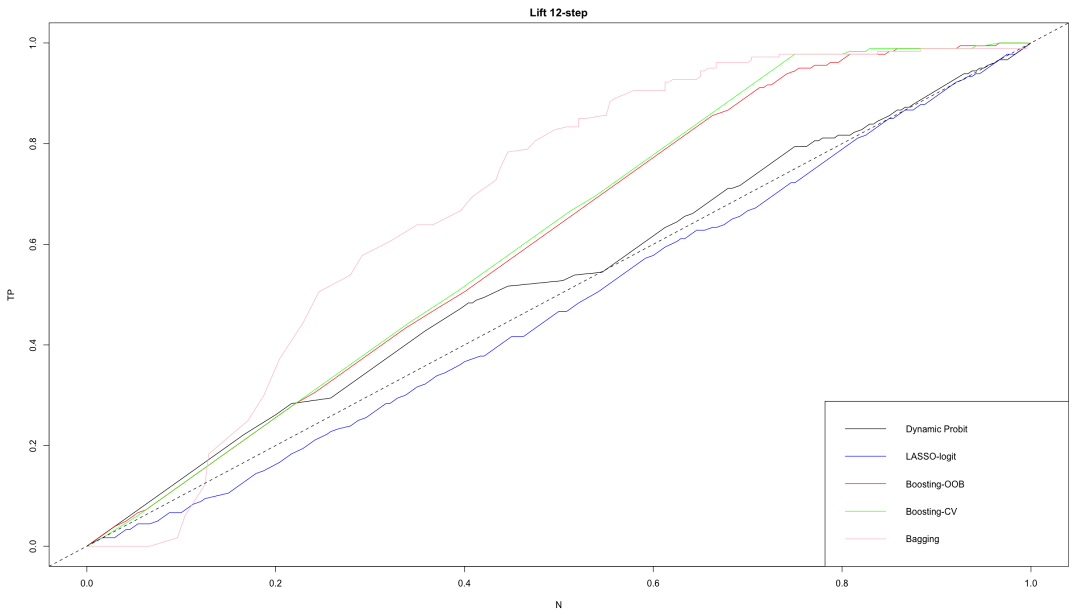

Figure 9. Lift Curves for 12-Step Models

5.3 Precision-Recall Curves

We can see from Figure 10 that the precision-recall curves for the 1-step ahead forecast are relatively close to the upper right corner which indicates that all models are generally predicting well. However, for 3-, 6-, 12-step ahead forecasts in Figures 11 to 13, we can see that the precision-recall curves are zig zag curves that frequently go up and down. Due to this occurrence, the precision-recall curves cross one another very frequently, making direct comparisons between curves challenging.

As we are unable to note down the best performing model from the precision curves, we note that the LASSO Logistic model's precision-recall curves for figures 11 to 13 are consistently below the other models, this indicates the relatively poorer predictive performance compared to other models for 3-step and above forecasts. This supports the findings of the ROC and Lift curves in the previous subsections.

As precision-recall curves are often used to evaluate performances in the case of imbalance dataset, this is specifically relevant for our study as we note the presence of an imbalance in the observations between the two classes. Specifically, we have a total of 204 observations of bear markets and 516 observations of bull markets. Thus, it is also no surprise to us that the imbalance data set produces a much better ROC curve that curves closer to the upper left corner for all models in each forecast horizon but produces zigzagging precision-recall curves.

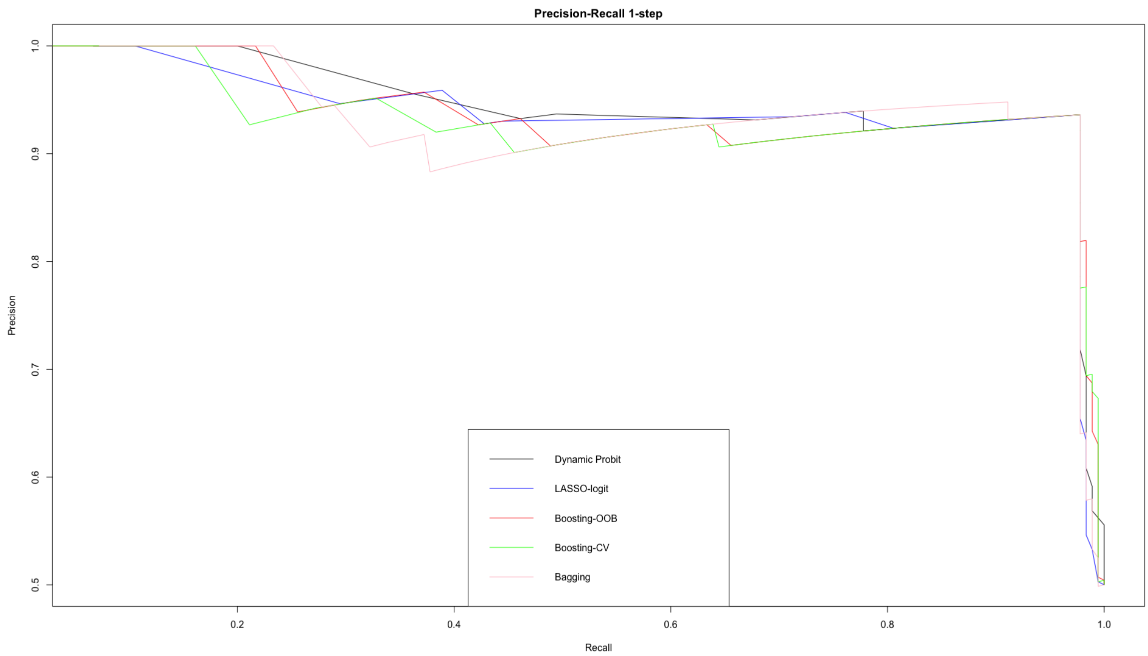

Figure 10. Precision-Recall Curve for 1-Step Models

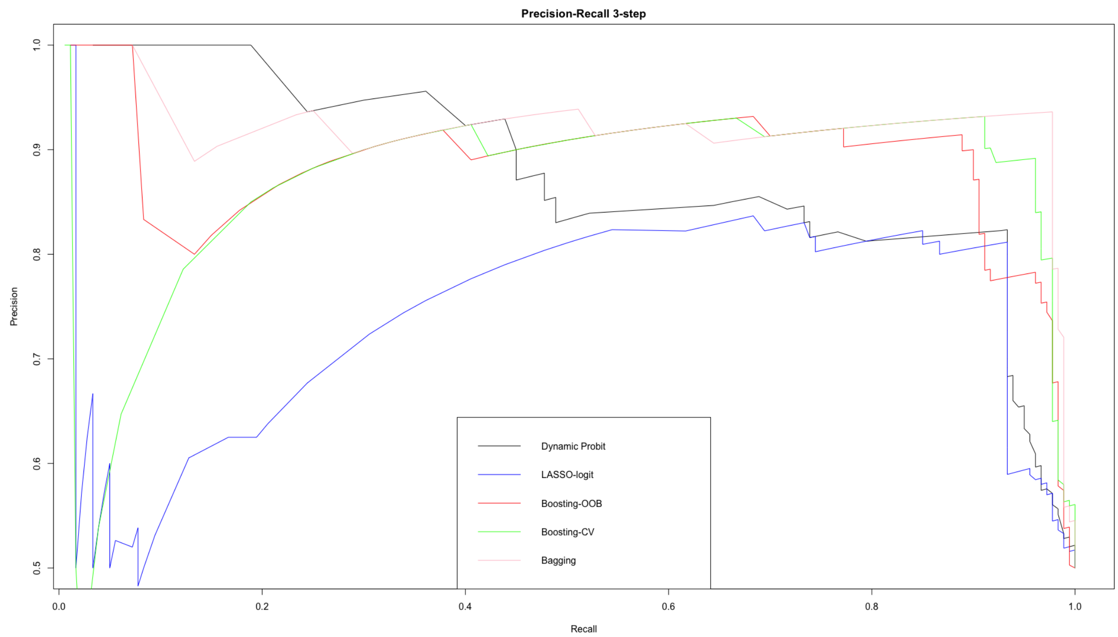

Figure 11. Precision-Recall Curve for 3-Step Models

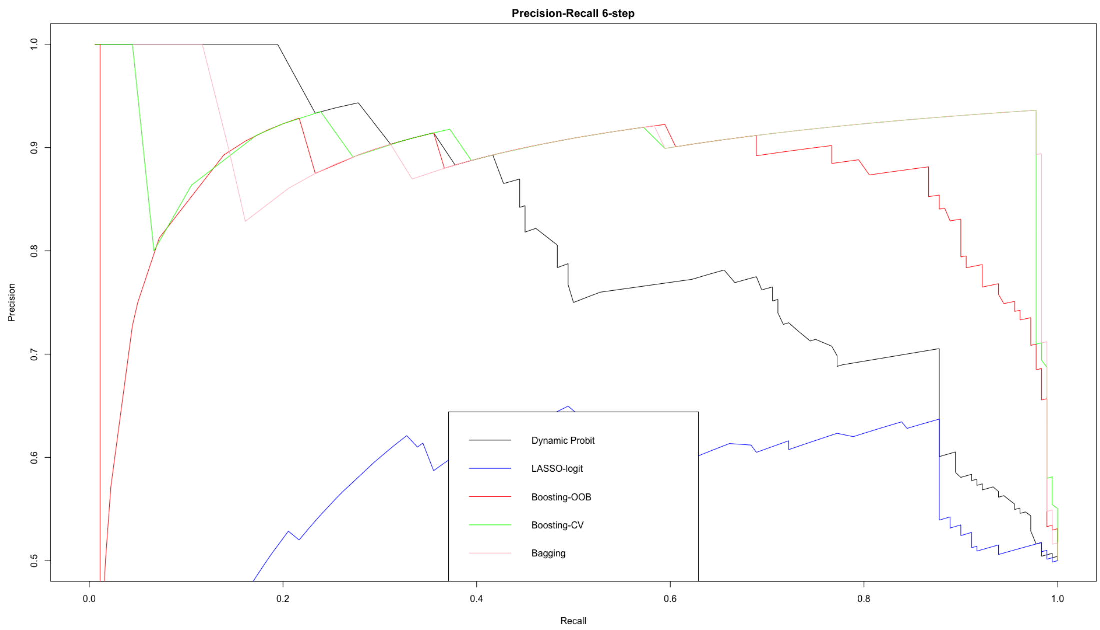

Figure 12. Precision-Recall Curve for 6-Step Models

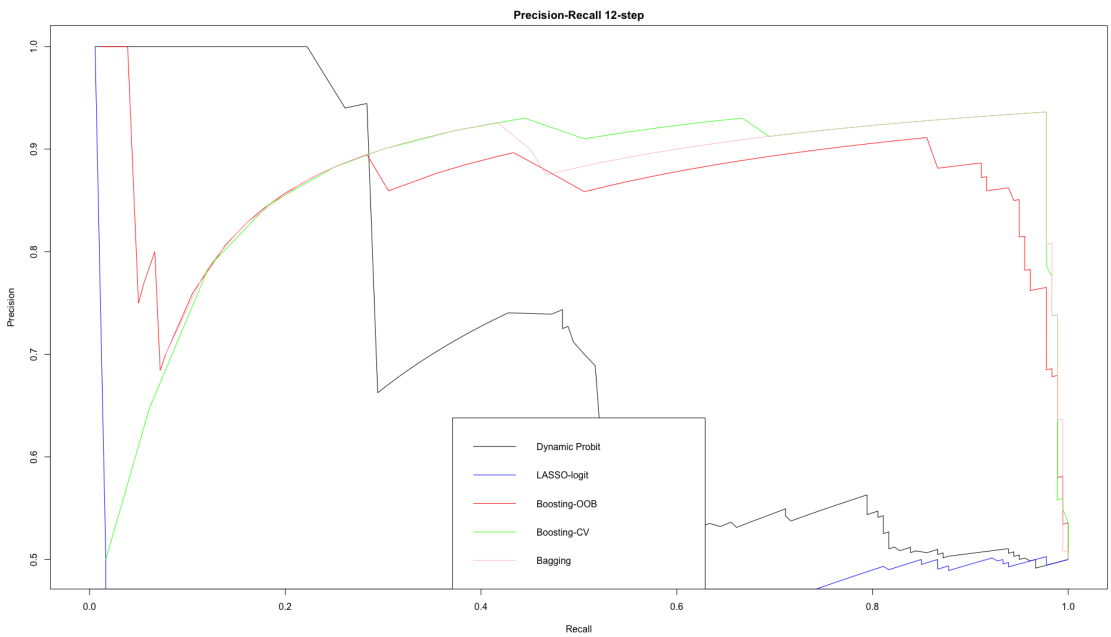

Figure 13. Precision-Recall Curve for 12-Step Models

5.4 Sensitivity, Specificity, Balanced and 'Naive' Accuracies

Finally, Table 2 below shows a summary of the other evaluation metrics used in our models. From the table, we can see that models using the 1-step ahead forecast perform quite similarly on sensitivity and specificity, implying that the models accurately predict bull/bear markets quite homogeneously. Models using the 1-step ahead forecast also provide better results than 3, 6, and 12-steps ahead forecasts overall. This is further supported by the balance accuracy and 'naive' accuracy tests, where their predictive power declines with longer lagged values. This is true for all the models except for Bagging, which managed to outperform the other models, even with longer lagged values.

Steps	Models	Sensitivity	Specificity	Balance Accuracy	Accuracy
1	Dynamic Probit	0.97778	0.93333	0.95556	0.96667
	rlassologit	0.97778	0.93333	0.95556	0.96667
	Bagging	0.97778	0.91667	0.94722	0.96250
	Boosting (OOB)	0.97778	0.93333	0.95556	0.96667
	Boosting (CV)	0.97778	0.93333	0.95556	0.96667
3	Dynamic Probit	0.93333	0.80000	0.86667	0.90000
	rlassologit	0.93333	0.60000	0.76667	0.85000
	Bagging	0.97778	0.93333	0.95556	0.96667
	Boosting (OOB)	0.93889	0.73333	0.83611	0.88750
	Boosting (CV)	0.96667	0.76667	0.86667	0.91667
6	Dynamic Probit	0.87778	0.63333	0.75556	0.81667
	rlassologit	0.89444	0.21667	0.55556	0.72500
	Bagging	0.97778	0.93333	0.95556	0.96667
	Boosting (OOB)	0.97778	0.56667	0.77222	0.87500
	Boosting (CV)	0.97778	0.83333	0.90556	0.94167
12	Dynamic Probit	0.96667	0.00000	0.48333	0.72500
	rlassologit	0.99444	0.00000	0.49722	0.74583
	Bagging	0.97778	0.91667	0.94722	0.96250
	Boosting (OOB)	0.98889	0.45000	0.71944	0.85417
	Boosting (CV)	0.97778	0.90000	0.93889	0.95833

Table 2: Evaluation metrics for All Models

5.5 Diebold Mariano (DM) Test Results

We ran a total of 10 DM tests across the 5 models that we generated. Comparing across models, we found that the best models for 1, 3, 6 and 12-steps ahead forecast are boosted trees (CV), bagging, and boosted trees (CV) respectively. Both these 2 models performed significantly better against most other models over most time frames. While the DM test shows that boosted trees (CV) performs significantly better than bagging for the 1 month horizon, bagging performs significantly better than boosted trees (CV) over the 6 month horizon.

6. Discussion

As seen in the previous section, all the ML models, with the exception of the LASSO logistic model, perform better than the Dynamic Probit benchmark model overall. Surprisingly, the LASSO logistic model performed the worst compared to all other models. Intuitively, this could be due to the fact that LASSO shrinks certain coefficients and some even to zero. Some of the coefficients that were shrunk to zero and dropped from the model might have become relevant again after a longer period of time. On the other hand, it seems that the Bagging model performs consistently better than all the other models, including the baseline model. Bagging is useful such that it aggregates over bootstrap replications to get rid of excess variance and gets rid of excess “signal”.

Implementing our best Bagging model, we are able to find out how much money an investor is able to make following the model, as compared to if he/she were to follow a passive "buy and hold" strategy or the perfect model where all bull and bear markets are predicted correctly with no lags. A naïve "buy and hold" strategy refers to when an investor uses all his capital at the beginning to buy into the S&P 500 and holds on to them for the entire period of the test time frame (i.e. January 2000 to December 2019) regardless of market fluctuations (Rosenberg, 2020). Based on our model, we came up with a simple trading strategy, investing half the capital at the beginning, then investing the other half when the market turns from bear to bull, and selling half when the market turns from bull to bear.

We start with \$100k in all scenarios, with the passive "buy and hold" strategy yielding \$222,837 at the end 2019, our best bagging model finishes with \$366,008, which gives us an alpha of 64.2% over the passive strategy. Lastly, if we were able to predict the bull and bear market with 100% accuracy and timing we would have done slightly better with a total portfolio of \$392,065. It seems that we were able to predict the bull and bear markets accurately with our Bagging model, however there are some lags in the execution of the trades which led to the loss of \$26057. While this is a favourable result that gives us further confidence that ML algorithms, and especially the Bagging model, is able to improve bull/bear market predictions*,* we must take into account that the forecast horizon of 10 years is relatively short and that there were not many fluctuations during this period of time.

We also considered the Granger Ramanathan (GR) forecast combinations to take advantage of the best parts of each of our 5 models to try and improve our forecast. Although GR forecast combination is only used for regression, we tried to implement it to see its performance on classification. However, upon experimenting, the combined model results were worse than individual models. This is likely due to the fact that our output was a binary 0 or 1, rather than a continuous regression model. One thing worth mentioning is that for our paper, the only ensemble method we utilized for this paper is boosted trees. As what we are analyzing is a classification problem, we are unable to make use of other forecast combination methods such as the GR combination. Such a combination introduces a regression method to combine forecasts.

Finally, there were also several limitations of our paper. We posit that predictions from models with longer lags might be less meaningful. The predictions of stock market performance further into the future is especially tricky as it would be more difficult to account for sudden changes in prices due to unexpected events, such as a pandemic or a supply shock. Based on these reasons, these models with longer lags might not be robust in accurately predicting bull/bear markets. Another limitation we faced is the lack of computational power to run more complex models. With higher computational power, our models can take in more data to provide us with higher predictive power for our analysis, thereby improving the accuracy of our forecast, and also reducing the time taken for each iteration to run.

7. Conclusion

To conclude, this paper looked at the prospects of improving predictions of bull/bear market predictions using various ML techniques. This problem is significant as wrong predictions are extremely costly and can have adverse impacts on an investor's portfolio. To analyse this classification problem, we compared the ML techniques used against a set of evaluation metrics, and we found that the Bagging model works best in terms of its accuracy of predicting the stock markets. Further comparing this with our Dynamic Probit benchmark model, we also see that the Bagging model generally performs better.

Although this paper does not provide insights as to how stock market performance would be like in the face of a large and sudden shock like that of COVID-19, and information about forecasts further into the future might be less meaningful, this paper is still useful in shedding light onto how ML methods are able to improve the predictions of bear/bull markets. Hence, our paper can be useful in informing investors on market performance to minimize their losses and maximize their gains.

7.1 Possible Extensions

Perhaps an area of our paper that could be further improved is to include other possible ML methods such as Artificial Neural Networks (ANNs). Furthermore, daily or weekly data could have been used as well instead of monthly data. This would enable us to get more data points to allow for more accurate predictions. However, this would mean an added layer of complexity to our models and more computational power might be required to run such a model. Hence, further extensions of our paper could include more ML methods that make use of more frequent data points to improve the predictions of the bull/bear market even further.

We could have also attempted to forecast a longer time horizon with breaks in the data such as during the global financial crisis of 2007-2008 or the COVID-19 pandemic, to see how our model would perform in more robust environments. Perhaps our results would not have been so favourable considering that the test data time frame happens to be pretty tranquil. An alternative to our report would be predicting a bull or bear market in a binary output manner would be to attempt to predict stock returns which would have been a continuous variable output.

We can also further tune how we adjust how much of our portfolio we buy and sell each time we get a signal. Currently, we are using a very naive strategy where there is only 1 signal each time. Every time there is a bull market we buy in with $50,000 and every bear market we sell $50,000 worth of our portfolio. However, if we were to make our buying and selling signals more sophisticated, our portfolio could be further optimized. For example, if we predict a bull market for this month, we can buy in with \$10,000 this month, and if the bull market continues for the subsequent 2 months we will continue to buy in with \$10,000 each time. However, when we predict a bear market, we should sell \$30,000 immediately. This is just an example of how we can tune our parameters to maximise our returns. The rationale behind this is because an asymmetric loss function is more appropriate than an absolute loss function when it comes to stock returns. If we are unable to predict a bear market when a bear market happens and exit as soon as possible, the loss would be much more severe than in any other scenario.

8. References

Gonzalez, L., Powell, J. G., Shi, J., Wilson, A. (2005). Two centuries of bull and bear cycles. *International Review of Economics & Finance*, *14(4),* 469-486. Retrieved from: https://doi.org/10.1016/j.iref.2004.02.003

Nyberg, H. (2013). Predicting bear and bull stock markets with dynamic binary time series models. *Journal of Banking and Finance, 37(9),* 3351-3363. Retrieved from: https://www.sciencedirect.com/science/article/abs/pii/S0378426613002264

Rosenberg, E. (2020, July 29). *What is buy-and-hold investing? A long-term investment strategy*. Business Insider. Retrieved from: https://www.businessinsider.com/personal-finance/what-is-buy-and-hold-investing-strategy

Welch, I., Goyal, A. (2008). A Comprehensive Look at The Empirical Performance of Equity Premium Prediction. *Review of Financial Studies, 21(4),* 1455-1508. Retrieved from: https://www.researchgate.net/publication/5217192_A_Comprehensive_Look_at_The_Empirical_Performance_of_Equity_Premium_Prediction

9. Appendix

9.1 Confusion Matrix Results for All Models

Confusion Matrix (Dynamic Probit)				
h-step forecast	**True Positive**	**True Negative**	**False Positive**	**False Negative**
1-step	176	56	4	4
3-step	168	48	12	12
6-step	158	38	22	22
12-step	174	0	6	60

Table 3: Confusion Matrix for Dynamic Probit Baseline Model

Confusion Matrix (rlassologit)				
h-step forecast	**True Positive**	**True Negative**	**False Positive**	**False Negative**
1-step	176	56	4	4
3-step	168	36	12	24
6-step	161	13	19	47
12-step	179	0	1	60

Table 4: Confusion Matrix for LASSO Logistic Model

Confusion Matrix (Bagging)				
h-step forecast	**True Positive**	**True Negative**	**False Positive**	**False Negative**
1-step	176	55	4	5
3-step	176	56	4	4
6-step	176	56	4	4
12-step	176	55	4	5

Table 5:.Confusion Matrix for Bagging Model

Confusion Matrix (Boosting CV)				
h-step forecast	**True Positive**	**True Negative**	**False Positive**	**False Negative**
1-step	176	56	4	4
3-step	174	46	6	14
6-step	176	50	4	10
12-step	176	54	4	6

Table 6:Confusion Matrix for Boosting CV Model

Confusion Matrix (Boosting OOB)				
h-step forecast	**True Positive**	**True Negative**	**False Positive**	**False Negative**
1-step	176	56	4	4
3-step	169	44	11	16
6-step	176	34	4	26
12-step	176	27	2	33

Table 7:Confusion Matrix for Boosting OOB Model

9.2 Best forecast performance based on DM tests

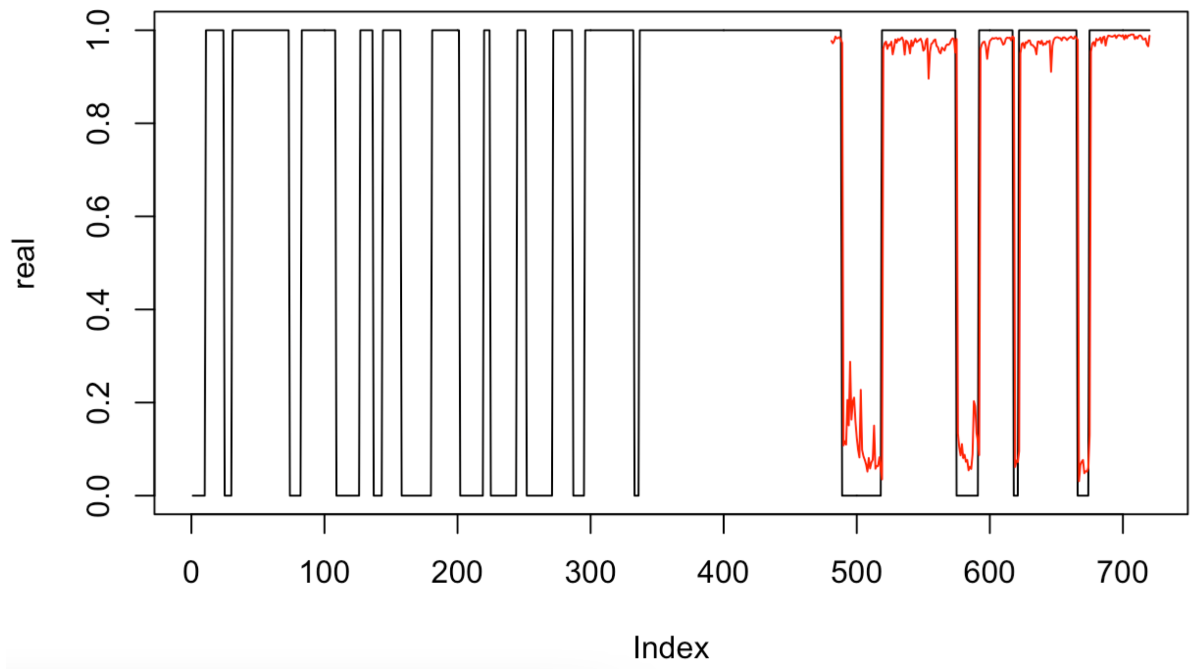

Figure 14: 1-step Forecast for Boosted Trees (CV) (DM best model)

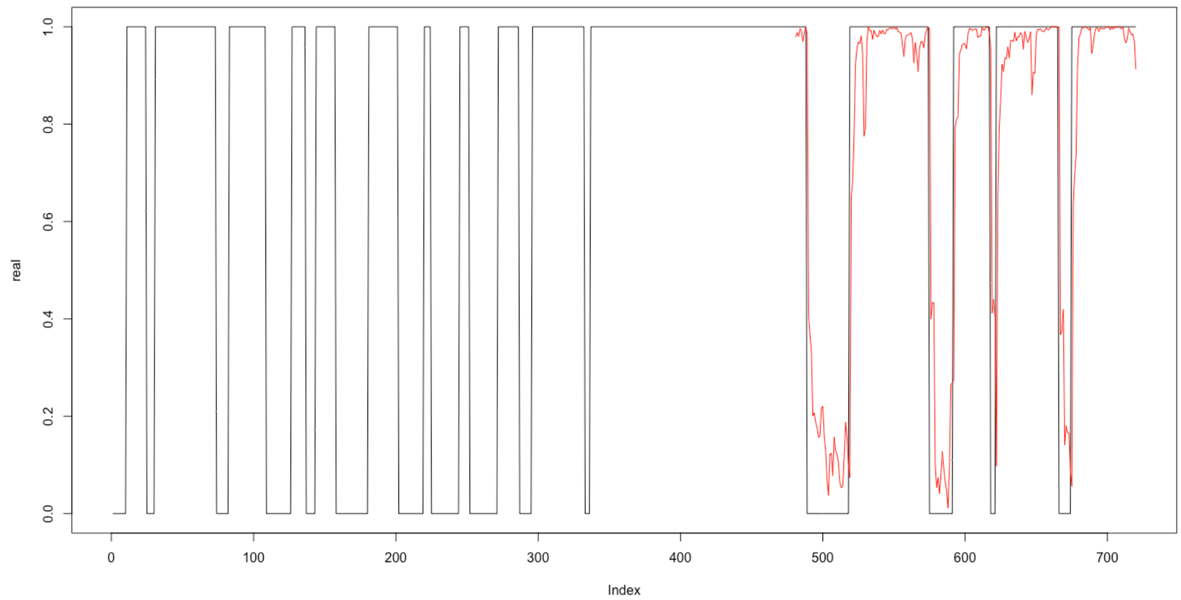

Figure 15: 3-step Forecast for Bagging Model (DM best model)

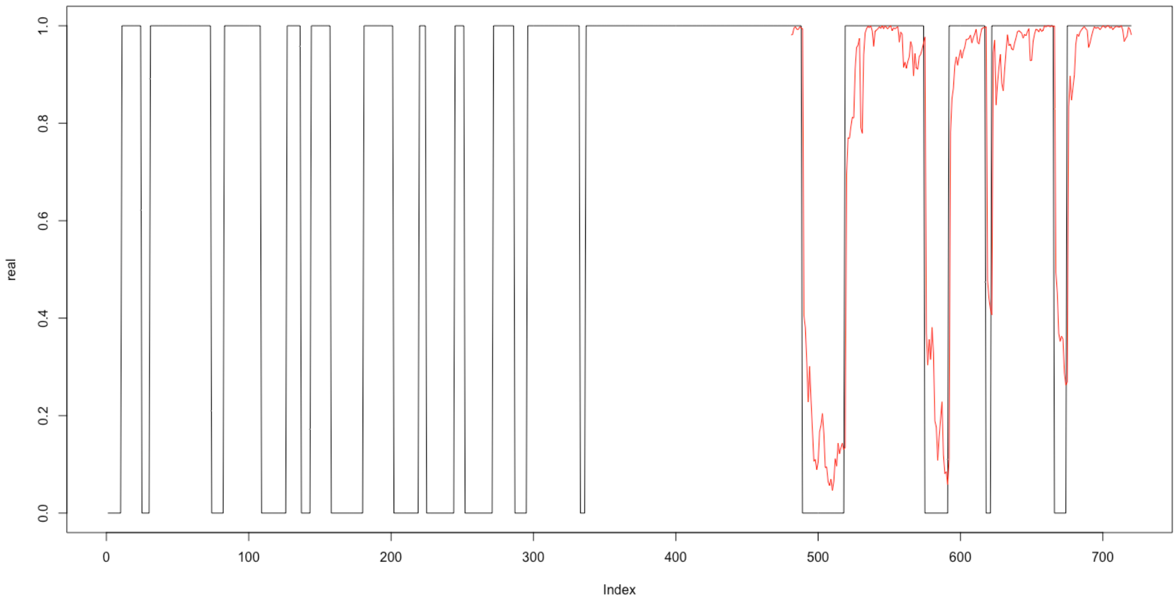

Figure 16: 6-step Forecast for Bagging Model (DM best model)

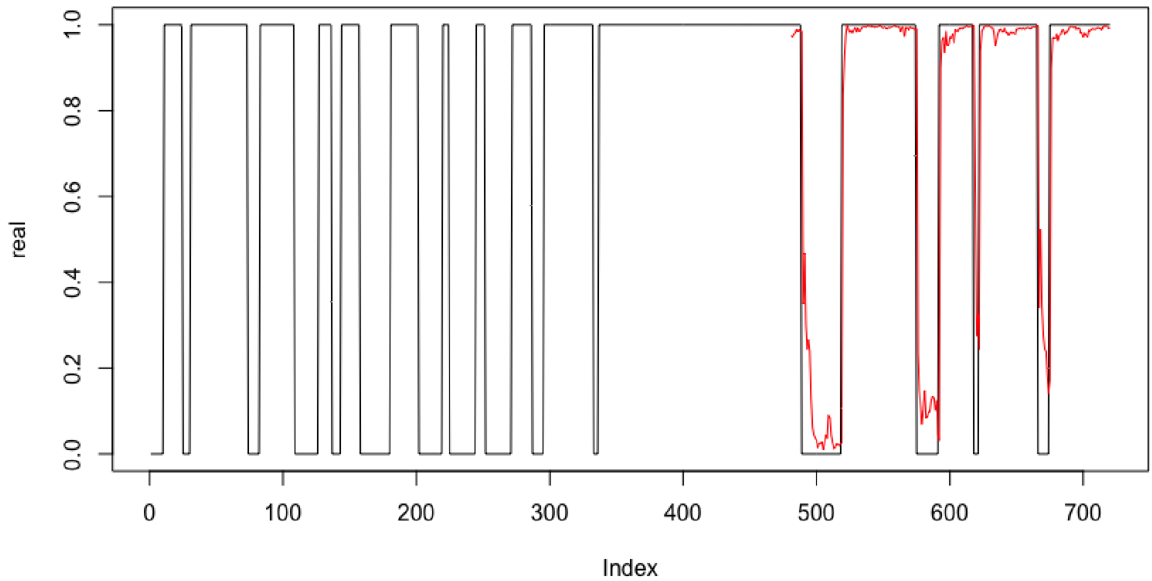

Figure 17: 12-step Forecast for Boosted trees (DM best model)

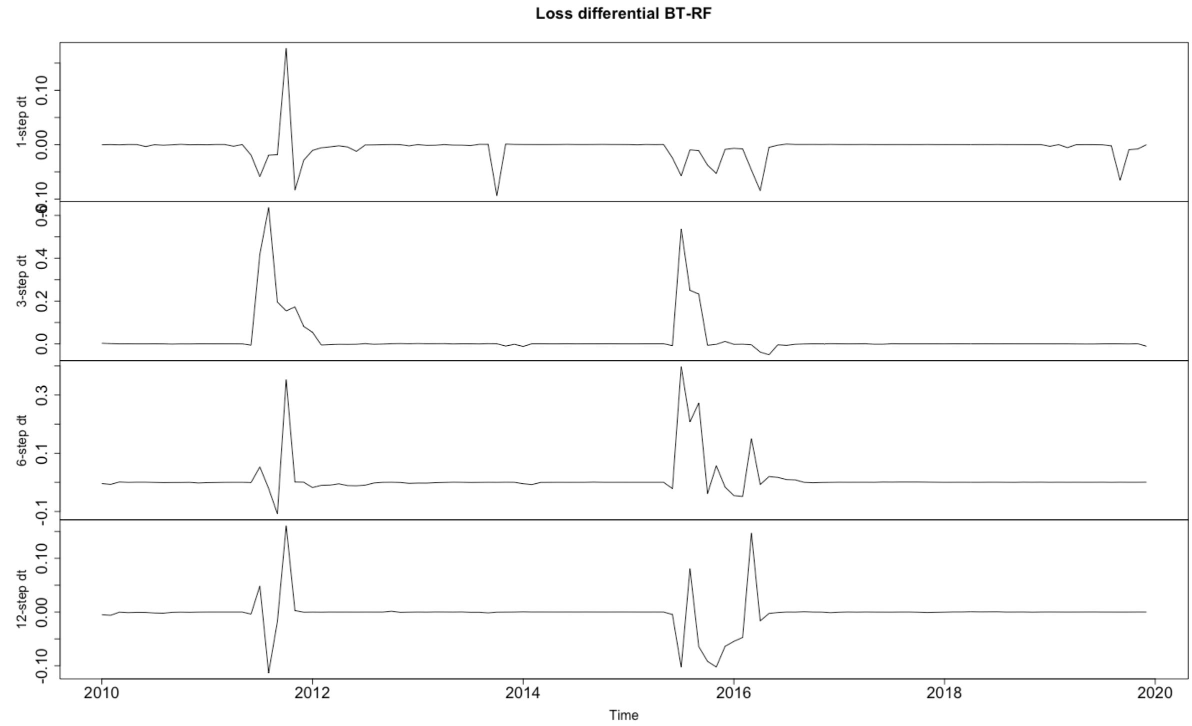

Figure 18: Loss differential between Boosted Trees and Bagging (2 best models)