Group Project: Predicting bull and bear stock markets with traditional time series econometrics model and machine learning models

EC4308: Machine Learning and Economic Forecasting

12 Nov 2024

Choo Qi En Jonathan

Jessica Widyawati

Toh Kai Lin

Wang Tingyu Kelly

Yang Shu Ting

Table of Contents

1. Introduction

The primary aim of this research is to determine whether machine learning (ML) models can improve predictions of stock market states, such as bull and bear markets, compared to traditional econometric models. Our research question specifically evaluates whether modern ML models—LASSO logistic regression, random forests, bagging, gradient boosting, and XGBoost—offer predictive advantages over dynamic probit models, particularly the "error correction" probit3 model (Basak et al., 2019). Accurate stock market predictions are beneficial for investors and policymakers as they support effective asset allocation and risk management (Adebiyi, 2014).

Nyberg's foundational work (2010, 2013) introduced the probit3 model, a dynamic binary probit model with an "error correction" mechanism designed to forecast stock market direction by leveraging economic indicators like term spreads and dividend-price ratios. Nyberg found that the probit3 model was effective in predicting bull and bear market periods both in-sample and out-of-sample, highlighting the importance of strategic timing in asset allocation across different economic regimes. (Nyberg, 2013) However, the probit3 model's assumption of linearity may limit its ability to capture more complex, nonlinear interactions in the stock market.

Building on Nyberg's work, our study applies advanced ML classification models, including logistic regression as a baseline and LASSO logistic regression for feature selection, as well as tree-based models like random forests, bagging, gradient boosting, and XGBoost, which have shown strong predictive performance in past research (Basak et al., 2019). By using these models, we address two primary limitations in Nyberg's study: (1) the linear assumptions of the probit3 model, which may overlook complex patterns in market data, and (2) the lack of ML models, which could enhance predictive accuracy by capturing nonlinear variable interactions (Zhong & Enke, 2019).

To measure each model's real-world applicability, we also implement a profit-based portfolio strategy, which directly assesses financial impact, adding a practical dimension beyond standard accuracy metrics (Choudhry & Garg, 2008). Preliminary findings suggest that advanced ML models, especially at shorter forecast horizons, demonstrate superior performance, potentially offering enhanced returns through improved predictive accuracy.

2. Data

2.1 Data source

Following the footsteps of Nyberg, we first obtained the data from Robert Shiller's website (https://shillerdata.com/). To enrich our dataset, we obtained another dataset from Amit Goyal's website (https://sites.google.com/view/agoyal145). The datasets used in this analysis exhibit significant missing values due to several factors. Firstly, the two datasets have different start and end points, which leads to discrepancies in the availability of data across the two sets. Secondly, certain variables have missing values because of their specific definitions. One easy example would be Shiller's self-defined variable "Cyclically Adjusted Price-to-Earnings Ratio" (CAPE) and its variants. This variable uses the average real earnings for the last 10 years in the calculation, easily causing a loss of over 100 data points. Finally, some variables are just not regularly updated leading to more missing data. We also removed variables that are only available quarterly and yearly, since we are doing a monthly forecast.

While the predictors used by Amit Goyal are used to predict equity premiums, we felt that stock related predictors, and other macroeconomic related predictors will still be useful for predicting bull-bear market states. Hence, we also decided to supplement our data set with more macroeconomic predictors, such as unemployment rate, federal funds rate and industrial production growth rate, which are obtained from the FRED MD database (https://fred.stlouisfed.org/). According to Chen (2008), it has been found macroeconomic variables can be useful in predicting bear markets.

2.2 Data transformation

To ensure stationarity of the data, we conducted unit-root tests for all the variables. For price, dividend-price ratio, book market, CAPE, total return (TR) CAPE and excess CAPE yield, we took the natural logarithm of the variable before first differencing it. For the remaining variables which are mostly price-related, we took the first difference of the variable if it was not stationary and double-checked to ensure stationarity after the transformation.

2.3 Forecast horizon

Forecast horizons of 1, 3, 6 months were selected because stock investors typically are interested in future returns and whether the market will be bullish or bearish in the short to medium term. According to Nyberg, when the forecast horizon is long, the iterative forecasting approach becomes computationally expensive or unfeasible. The 12-month forecast horizon is not included because it will be difficult to produce out-of-sample forecasts with accuracy in the longer term.

2.4 Predictors selection

For the machine learning models, we chose to include up to 6 lags for each predictor to incorporate their potential delayed effect on the market state. Additional lags beyond that are likely to be less informative in predicting and make the models unnecessarily complex. For each forecast horizon, we kept the number of lags for each predictor at 6, which means that we would use lags 1 to 6 for h=1, lags 3 to 8 for h=3, and lags 6 to 11 for h=6 to prevent any data leakage.

Similarly, we kept the number of lags at 6 for the dependent variable to be included as predictors in our models. Since the Bry and Boschan Algorithm was used in Nyberg's paper to determine the bull or bear market state based on a two-sided moving average, it means that we would have to know the S&P500 return in the next 6 months as well as the last 6 months to locate peaks and troughs of the stock index. We used the bbdetection package in R which implements the algorithm and assigns market states following Nyberg's paper, where the minimum length of a bull or bear market must be at least 6 months, and the duration of a complete cycle is assumed to be at least 15 months. This means that the first 6 lags of market states will be unknown at the time of forecast and thus cannot be used as predictors. Only lag 7 and above can be added to our model. Therefore, we used lags 7 to 12 for h=1, lags 9-14 for h=3, and lags 12-17 for h=6. This makes up to 276 predictors being used for each forecast horizon.

2.5 Train-test split

After accounting for the missing data and aligning the start points of all the variables, we were left with 594 observations. As we used the Bry and Boschan algorithm to determine the state of the market, the last 6 values of the market state would be inaccurate due to the lack of data on the S&P 500 return for the next 6 months. Thus, we chose to remove the last 6 observations, leaving us with 588 observations. Finally, we had to remove the first 17 observations to account for the lags of the predictors that we were using, ending up with 571 observations in total.

As we wanted to have sufficient data to both train and test the model, we decided to have 150 observations as the test set, and the remaining 421 observations as the training set. For the methods that have parameters that need to be tuned using cross-validation (gradient boosting and xgboost), we further split the training set into 100 observations for validation and 321 observations for training the model. This split gives us a good balance between leaving enough data to train the models adequately, and also having enough data to test and evaluate the models well.

As our data is a time series data, the observations have dependence across time, so when validating and testing our models, we had to be sure to preserve the order of the observations to prevent any data leakage and to keep the forecast horizon the same. Thus, we chose to use recursive estimation with a rolling window. To do this, we fixed the number of observations to train the model to be 421 observations. Then, we predicted the market state h periods ahead, where h is the forecast horizon. After that, the estimation window was rolled forward by 1 observation to predict the market state of the next period, still h periods ahead. This process was repeated for all the observations in the test set.

3. Methodology

3.1 Models

Each model was selected based on the properties of our dataset, which includes financial time series data with both linear and non-linear relationships. For each method, we evaluated the alignment between the data characteristics and the model's assumptions and strengths.

3.1.1 Logistic regression

Logistic regression was selected as a benchmark model due to its ability to handle binary outcomes, making it suitable for predicting bull or bear market states. Given the stability of stock market states over extended periods, the logistic model is well-suited for our dataset's binary classification of market regimes. Logistic regression also allows for straightforward interpretability, offering insights into the influence of predictor variables on the probability of each market state. Our data includes variables that logistic regression can handle effectively, such as lagged returns and macroeconomic indicators, ensuring the model can exploit stable, linearly separable relationships for market predictions.

3.1.2 Penalized regression (LASSO logistic regression)

LASSO logistic regression is logistic regression with L1-norm regularization. In Zhang et al. (2019), it is found that LASSO can parsimoniously select a few predictors given a large set of predictors and the selected variables are ones with high predictive power (in the context of forecasting crude oil prices). Given our large set of predictors and the feature selection ability, we decided to focus on using LASSO. The model was ran using rlassologit using the hdm package and the optimal lambda given by the function was used to predict the probabilities. A rolling window forecast was used and for each shift in the data, the model was trained again to obtain the optimal lambda before making the prediction.

3.1.3 Random forest

Random forest is a machine learning model derived from bagging that can be used for both regression and classification problems. Bagging works by fitting many decision trees, then aggregating over all the trees to get a prediction. Each tree is built using a bootstrap sample drawn from the data and is grown big to ensure that the signal from the data will be captured. Aggregating over all the trees reduces overfitting and reduces the variance in the final forecast while keeping bias low. In our classification problem to predict whether the market is a bull or bear market, the majority vote is taken to be the forecast. For random forest, each decision is grown by using only m < P predictors, which causes the decision trees to be less correlated with each other. For our project, we used $m = \sqrt{P} = \sqrt{276} \approx 16$ predictors for each tree, which is the default choice for classification problems. To determine whether there are dominant predictors in the data, we tried both random forest and bagging models on the default settings (500 trees, minimum node size of 10) and compared their performances. In our case, random forest and bagging would be useful as our data contains many predictors, and we are unsure of the functional form of the data. Random forest and bagging also work well without any tuning, so we wanted to compare their performance to other models that require more tuning.

3.1.4 Boosting model

Boosting is an ensemble machine learning model that works well when there are many predictors. It combines hundreds of simple models and aggregates them together to produce a robust forecast. It can solve complex binary classification problems where the result is computed as the class with the majority vote. In our context of predicting the market state, boosting models can be powerful in capturing the non-linear relationships between variables. They are less prone to overfitting with proper tuning of the parameters and have the potential to perform well given how noisy the stock market data can be. By training models sequentially, boosting models learn from errors to reduce bias and enhance predictive ability. Both gradient boosting and XGBoost were implemented to predict bull or bear market states as it is possible for either model to outperform the other in this context.

Cross validation was used for tuning the tree size and finding the simplest model with the lowest misclassification rate. For gradient boosting, both default thresholds of 0.5 and the sample mean of market states in the training data were explored, as Nyberg’s paper pointed out that the sample mean method could be more effective, because the number of bull markets was greater than that of bear markets.

Rolling window was used for both cross validation and testing to construct forecasts. Across all forecast horizons, boosting with depth = 5 generally outperformed depth = 2, as the best model chosen by cross validation with depth=5 resulted in a smaller tree size with equal misclassification rates. Therefore depth = 5 is used for testing across all models for easier comparisons.

Summary table for the best tree sizes chosen by cross validation which were then run on the test set:

Model Type	Forecast Horizon	Best Tree Size	Misclassification
GBM	1	510	0.07
	3	496	
	6	488	
GBM (Sample Mean)	1	952	
	3	1058	
	6	1090	
XGB	1	174	
	3	179	
	6	145	

XGBoost is an optimized and highly efficient implementation of gradient boosting. It is faster and often results in better performance as the training can be parallelized across clusters with advanced regularization techniques (Subha, 2024). This is evident in the cross-validation result where XGBoost has a much smaller best tree size across forecast horizons while having equal misclassification rate.

3.2 Model performance metrics

3.2.1 F1 score

The F1 score was chosen to evaluate model performance, as it balances precision and recall. This metric is critical for stock market prediction, where false positives (incorrect bull predictions) and false negatives (missed bear predictions) each carry significant financial implications. A high F1 score thus indicates a model's reliability in predicting market direction, especially under volatile conditions (Zhong & Enke, 2019).

3.2.2 Profits-based portfolio strategy

In addition to conventional metrics, we implemented a profit-based portfolio strategy to evaluate real-world application. Unlike error-based metrics like Mean Squared Error (MSE), which assess prediction accuracy, this approach evaluates each model's practicality by simulating financial gains or losses based on the model's predictions (Choudhry & Garg, 2008). Our portfolio strategy classified each observation as either a bull or bear market using the Bry and Boschan dating rule. We then adjusted our investment allocations accordingly, moving into risk-free assets in bear markets and shifting back to stocks in bull markets. This method provides a clear measure of financial impact, reflecting the model's effectiveness in guiding investment decisions (Adebiyi, 2014). We selected this strategy as it is more relevant for financial applications, directly measuring profitability in a real-world scenario and providing insights beyond predictive accuracy. Here is a detailed breakdown of our portfolio strategy evaluation code, using prediction of bull market state using our logistic regression model for forecast horizon h = 1: The code begins by filtering the dataset to exclude any rows with missing values for predicted_prob_logit_h1, which represents the model's forecasted probability of a bear market. A threshold of 0.5 is applied to predicted_prob_logit_h1 to decide between investing in stocks or a risk-free asset (T-bill rate). If the probability of a bear market is above 0.5, the code assumes a bear market and invests in the risk-free asset (tbl); otherwise, it assumes a bull market and invests in stocks (ret). The cumulative return from this strategy is calculated with cumprod(1 + data$strategy_return) - 1, which compounds returns over time based on the daily or monthly strategy returns. The code also calculates an average bear market probability of the training set (sample_avg_threshold) and applies it as an alternative threshold. If the predicted probability exceeds this average, it invests in the risk-free asset; otherwise, it invests in stocks.

4. Analysis

4.1 Feature engineering

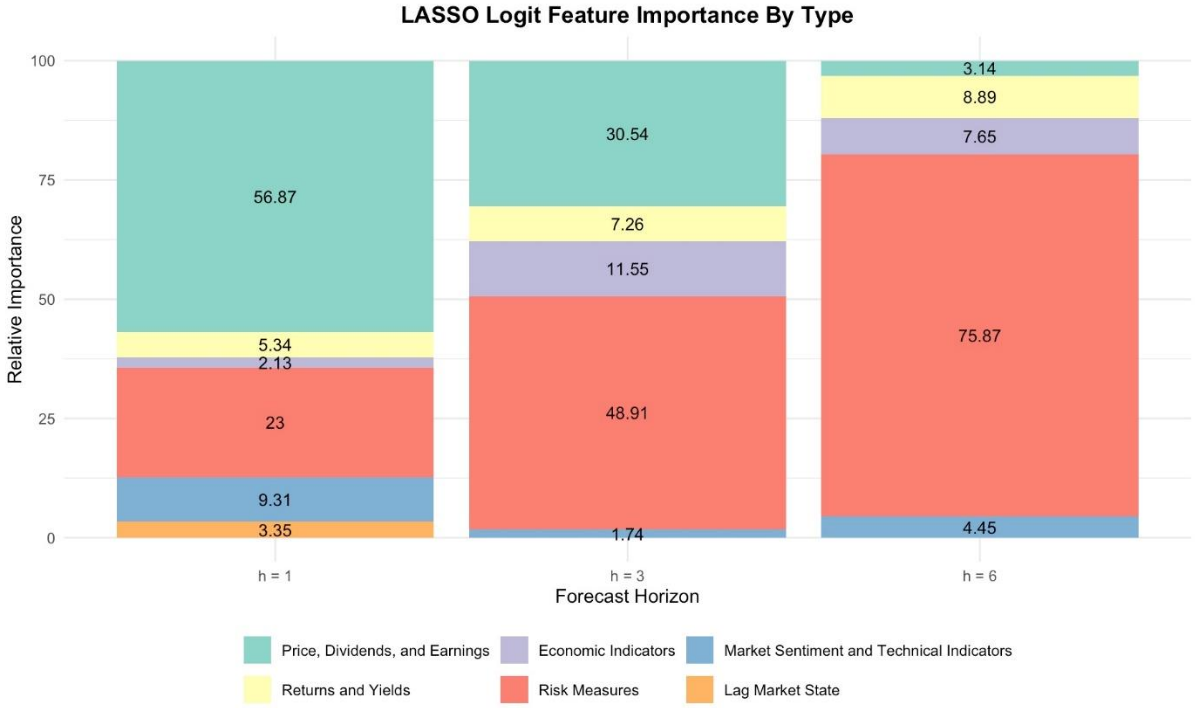
4.1.1 Logistic regression

Following Nyberg's 2013 paper, we constructed our benchmark logit model using lagged stock return, term spread and change in dividend-price ratio as predictors. We also included lags of long-term yield and inflation rate as possible predictors as they showed decent statistical significance in a study done by Goyal 2024. We used the lags of these variables corresponding to the forecast horizon (h). For 1-step ahead forecast, we used the 1st lag of all predictors. Similarly for h = 3 and h = 6, we use the 3rd and 6th lags of the predictors respectively.

For the machine learning models, the predictors were grouped into 6 groups to see which groups were the most important in forecasting the market state (refer to appendix for details). The 6 groups were:

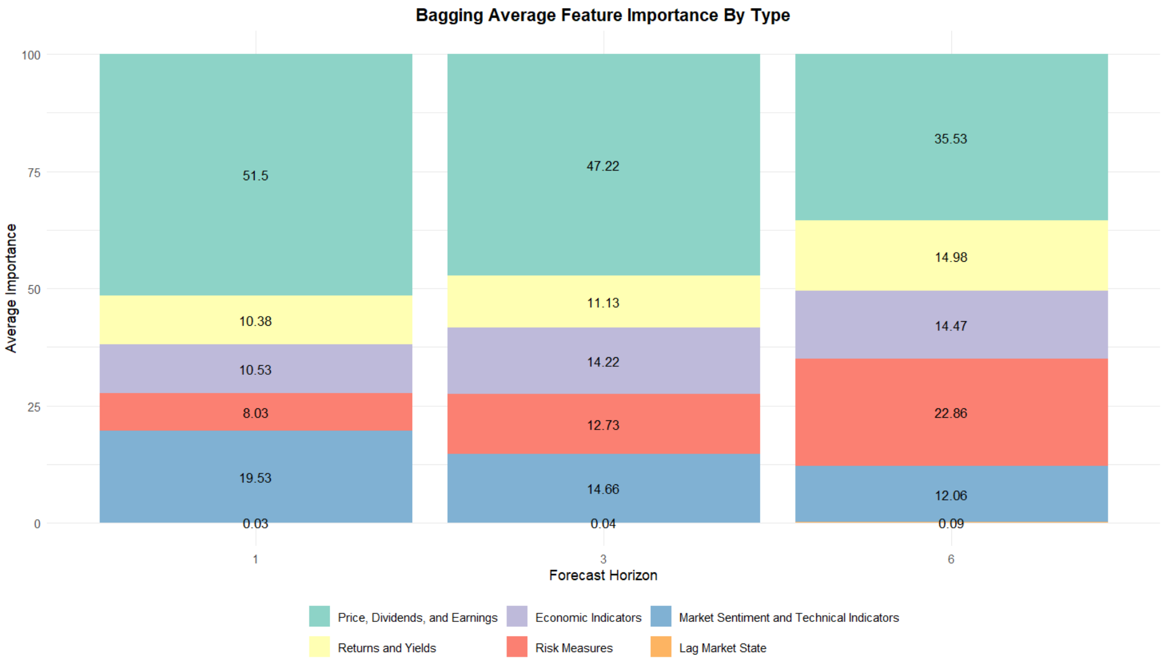
1) Price, Dividends and Earnings
2) Economic Indicators
3) Returns and Yields
4) Risk Measures
5) Market Sentiment and Technical Indicators
6) Lags of Market State (dependent variable)

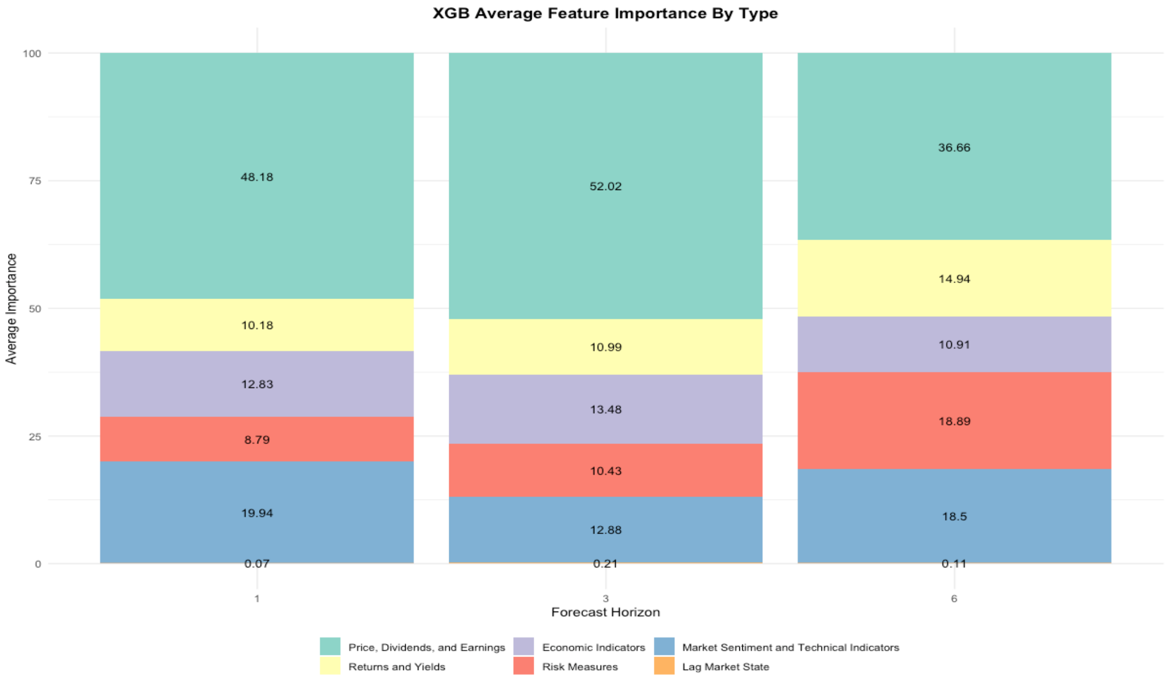
4.1.2 LASSO logit



With reference to the figure above, for LASSO logit, the Price, Dividends and Earnings group dominated the feature importance for 1-step ahead forecast. For the 3-step ahead and 6-step ahead forecast, the Risk Measures group dominated the importance. As the forecast horizon increases, the relative importance of the Price, Dividends and Earnings group drops and the relative importance of the Risk Measures group increases. This suggests that as the forecast horizon increases, the direct impact of short-term indicators on recent market conditions diminishes. Signals that provide a more macro view of the economy become more important, capturing long-term systemic risks and uncertainty in the economy, while signals that provide a more micro view of the economy become less important.

4.1.3 Tree-based models





For all tree-based models, the Price, Dividends and Earnings group had the highest importance across all 3 forecast horizons, but the importance generally decreased as forecast horizon increased. This group includes predictors such as lagged stock returns, lagged dividend yield and lagged dividend-price ratio, which likely give a lot of information about the state of the market. This ties in with the findings of important predictors from Nyberg's 2013 paper. Also, for all tree-based models, the Market Sentiment and Technical Indicators group, which includes predictors like distilled sentiment and the Dow Jones Industrial Average, had higher importance when h = 1 compared to other forecast horizons, and was the second most important group of predictors behind Price, Dividends and Earnings for h = 1. This may be because market sentiment reflects the general attitude of investors, which will be affected by what they think is about to happen in the market and affects how investors act in the short term. For longer forecast horizons of h = 3 and h = 6, while Market Sentiment and Technical Indicators remained important for boosting models, the importance of this group of predictors showed a consistent decrease as forecast horizon increased for the random forest and bagging models.

On the other hand, the Risk Measures group displayed a quite significant rise in importance as the forecast horizon increases for all tree-based models. This suggests that predictors in these groups, which includes lagged average stock skewness, lagged tail risk from cross-section, and lagged stock return dispersion, hold more predictive power for longer-term forecasts, which aligns with the result from lasso logit. The Returns and Yields group, consisting of predictors like lagged AAA and BAA bond yields, lagged long-term government yield and lagged term spread, also showed a smaller but consistent rise in importance with forecast horizon. This may be because these predictors have a more long-term or delayed effect on the market state. The feature importance of other tree-based models follows similar composition and pattern across forecast horizons, which can be found in the Appendix.

It is noteworthy that even though there is a significant difference in the number of trees for bagging and XGB boosting (e.g. 500 and 174 for h = 1), and in algorithms where trees are built independently in bagging and sequentially in boosting, the average proportion of each group of variables picked up by the models remained relatively consistent across the board.

4.2 F1 score

Across all forecast horizons, the boosting GBM model using sample mean has the highest F1 score while the LASSO logit model had the lowest F1 score. However, the LASSO logit model still proves to be useful in predicting the movement of the stock market, as it has a relatively high F1 score of more than 0.90. It is worth noting that all the ML models as well as the benchmark logit model have similar F1 scores, of which all of them are above 0.90.

	Forecast Horizon		
Model	h=1	h=3	h=6
Logit	0.9547	0.9547	0.9547
Lasso Logit	0.9118	0.9170	0.9170
Bagging	0.9594	0.9520	0.9485
Random Forest	0.9562	0.9600	0.9565
Boosting GBM	0.9854	0.9854	0.9818
Boosting GBM (sample mean)	0.9854	0.9854	0.9854
Boosting XGB	0.9818	0.9854	0.9854

Nevertheless, high F1 scores might be misleading here due to the imbalance of market states in the test data. Since bull markets significantly outnumbered bear markets, a model can achieve high precision and recall by predicting mostly bull markets, making the F1 score for bull markets artificially high. However, this would come at a cost for identifying bear market which is the minority class. Therefore, a more comprehensive metric is needed to evaluate and compare models' performances.

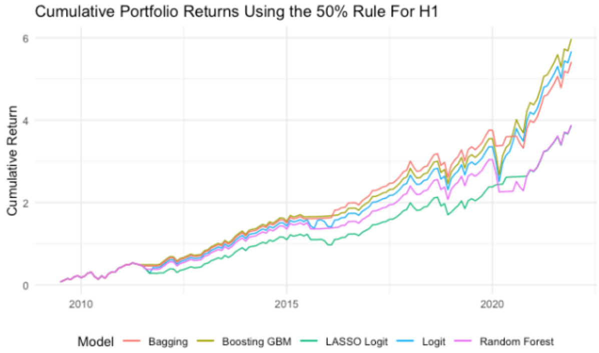
4.3 Market timing experiment

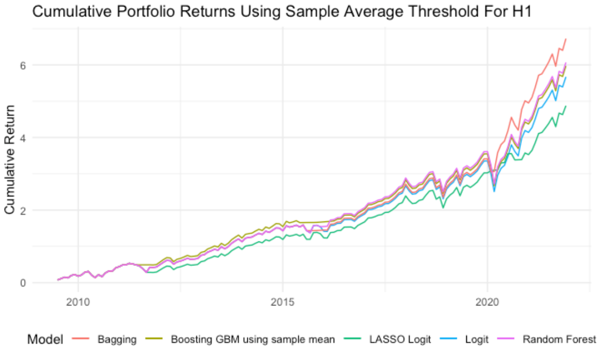
As mentioned in 3.2.2, we considered a small-scale market timing experiment between stocks and the risk-free interest rate to quantify the economic value of the out-of-sample forecasts. We consider 2 different portfolio weighting schemes to determine the asset allocation 1 period ahead, 3 periods ahead and 6 periods ahead. The benchmark model for all forecast horizons is the logistic regression model as we are interested in the performance of each ML model in comparison to the benchmark logit model.

4.3.1 Forecast horizon = 1

For 1-step ahead forecast, only the boosting models outperform the benchmark logit model when the threshold of 0.50 is employed. The boosting GBM and the boosting GBM using the sample mean had the best performance, as it yields cumulative returns of 597% when the threshold = 0.50. However, when the threshold employed is the sample average of bear market months, which is approximately 0.26, bagging had the highest cumulative returns of 672%. Amongst the boosting models, boosting GBM using sample mean had the best performance both when sample average threshold is used and when threshold = 0.50.

Model	Cumulative returns when threshold = 0.50	Cumulative returns when threshold = sample average
Logit	5.67031	5.67031
Lasso Logit	3.86681	4.87568
Bagging	5.41376	6.72304
Random Forest	3.88215	6.06708
Boosting GBM	5.97120	4.97808
Boosting GBM (sample mean)	5.97120	5.97120
Boosting XGB	5.83025	5.12975


Cumulative Portfolio Returns Using the 50% Rule For H1

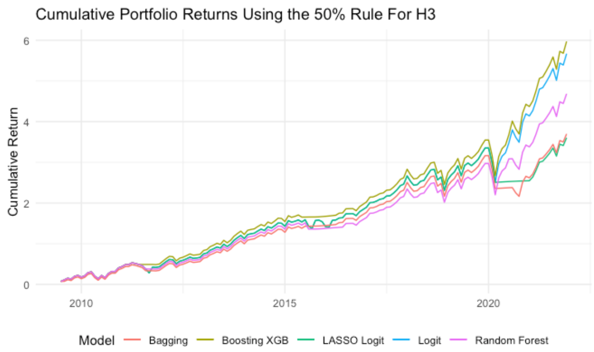

Cumulative Portfolio Returns Using Sample Average Threshold For H1

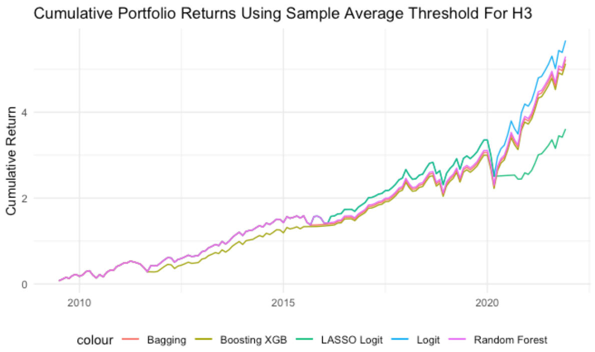
4.3.2 Forecast horizon = 3

For 3-steps ahead forecast, only the boosting models outperform the benchmark logit model for both thresholds. When the threshold = 0.50, all 3 boosting models had the highest cumulative returns of 597%. When the threshold employed is the sample average of bear market months, boosting GBM using the sample mean had the highest cumulative returns of 597%. It is worth noting that the bagging and random forest models had much higher cumulative returns when the threshold employed is the sample average of bear market months. Amongst the boosting models, boosting GBM using sample mean had the best performance both when the threshold = 0.50 and when sample average threshold is used.

Model	Cumulative returns when threshold = 0.50	Cumulative returns when threshold = sample average
Logit	5.67031	5.67031
Lasso Logit	3.60500	3.61037

Bagging	3.69823	5.22590
Random Forest	4.68467	5.29800
Boosting GBM	5.97120	5.69690
Boosting GBM (sample mean)	5.97120	5.97120
Boosting XGB	5.97120	5.12975

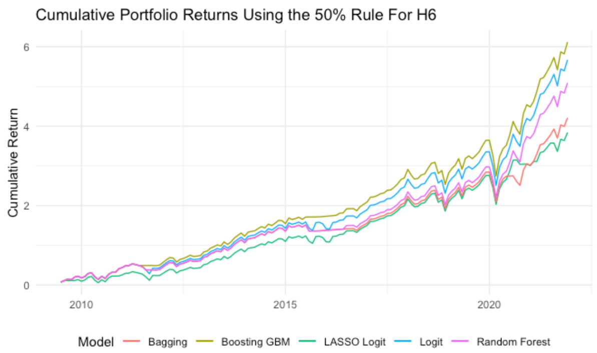


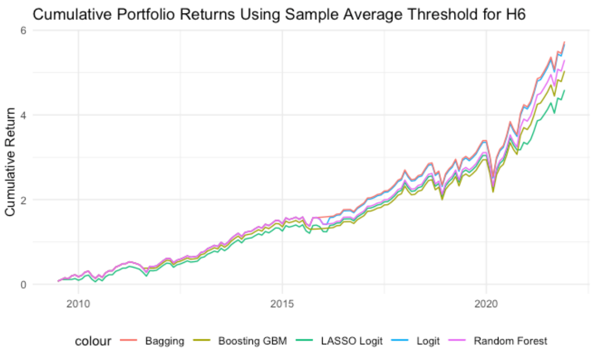


4.3.3 Forecast horizon = 6

For 6-steps ahead forecast, again only the boosting models outperform the benchmark logit model for both thresholds. The boosting GBM model performed the best, with cumulative returns of 612% when the threshold employed is 0.50. When the threshold employed is the sample average of bear market months, the boosting GBM using the sample mean had the highest cumulative returns of 597%, which is closely followed by the bagging model with 573% and the benchmark logit model with 567%. Amongst the boosting models, boosting GBM had the best performance when the threshold = 0.50 whereas boosting GBM using the sample mean had the best performance when sample average threshold is used.

Model	Cumulative returns when threshold = 0.50	Cumulative returns when threshold = sample average
Logit	5.67031	5.67031
Lasso Logit	3.83951	4.59110
Bagging	4.21127	5.73455
Random Forest	5.08841	5.29800
Boosting GBM	6.11753	5.03982
Boosting GBM (sample mean)	5.97120	5.97120
Boosting XGB	5.97120	4.75748





5. Conclusion

This study set out to address the question: *Can machine learning models improve predictions of stock market states (bull or bear) compared to traditional econometric models?* Our aim was to determine which model offers the best predictive performance across various forecast horizons and evaluate the practical financial impact of these predictions.

A critical aspect of our evaluation was determining the most effective threshold for model performance. Between the standard threshold of 0.50 and the sample average threshold, we argue that the sample average threshold provides a more accurate basis for predicting market states, as it adjusts for the inherent asymmetry in bull and bear markets. This threshold more effectively balances model sensitivity to both market states, reducing potential bias in cumulative return assessments.

Based on both F1 scores and cumulative returns using the sample average threshold, gradient boosting machines (GBM), particularly those employing the sample mean threshold, consistently demonstrated good performance across all three forecast horizons (h = 1, 3, and 6). For short-term (h = 1) predictions, the GBM model achieved the highest F1 score, capturing intricate market patterns that logistic regression and other traditional models may overlook. However, in terms of cumulative returns, the GBM model performed worse than the bagging and random forest models when using the sample mean threshold.

At medium-term horizons (h = 3 and h = 6), the GBM model continued to excel when threshold = 0.50 was used, and the performance of XGBoost follows closely behind. However, XGBoost and GBM consistently showed a fall in cumulative returns when the sample mean threshold was used instead, perhaps because too many months were wrongly classified to be bull months with the lower threshold. The discrepancy in cumulative returns from the 2 thresholds might also have resulted from the two models being cross validated with threshold = 0.50 but then are tested with a threshold derived from sample mean. However, it is noteworthy that XGBoost models are much more interpretable than GBM models due to smaller tree size, while having comparable performance. On the other hand, the bagging and random forest models showed a marked improvement when the threshold was changed from 0.50

to the sample average, indicating that these models might be more sensitive to the asymmetric proportion of bear and bull months.

While the logistic benchmark model performed very stably across all forecast horizons, generating the same cumulative returns of 567%, this was due to the model classifying all the months in the test set as a bull market for both thresholds. While this led to relatively good performance for the test set we used, this might not extend to datasets with many observations that are bear markets.

The GBM model's robustness across all forecast horizons and metrics suggests that its ability to capture complex, nonlinear relationships, alongside effective parameter tuning through boosting, makes it the most consistently reliable model for market state prediction. In contrast, the logistic regression benchmark, while simpler, lacks the adaptability to model nonlinearities inherent in market data, particularly at longer forecast horizons.

In terms of this paper's contributions, our study extends Nyberg's (2013) findings by demonstrating the predictive advantage of ML models in capturing non-linear market dynamics, which are not addressed by the probit3 model's linear structure. Our findings value-add to Nyberg's work by showing that ML models, especially tree-based ensemble methods, can yield higher accuracy and financial returns in a real-world trading strategy. However, limitations include potential overfitting in more complex models and reliance on historical data instead of real-time vintages, which may not fully adapt to real-time model performance.

For future research, exploring additional market states (e.g., leading indicators especially for recession) and refining model interpretability would enhance the practical applicability of ML models in stock market predictions. Expanding analysis across various economic conditions could further validate these findings and optimize model selection for robust, actionable financial insights.

6. References

Ariyo, A. A., Adewumi, A. O., & Ayo, C. K. (2014). Stock Price Prediction Using the ARIMA Model. *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*. https://doi.org/10.1109/uksim.2014.67

Basak, S., Pavlova, A., & Shapiro, A. (2006). Optimal Asset Allocation and Risk Shifting in Money Management. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.879294

Chen, S.-S. (2009). Predicting the bear stock market: Macroeconomic variables as leading indicators. *Journal of Banking & Finance*, *33*(2), 211–223. https://doi.org/10.1016/j.jbankfin.2008.07.013

Choudhry, R., & Garg, K. (2017). *A Hybrid Machine Learning System for Stock Market Forecasting*. World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering. https://www.semanticscholar.org/paper/A-Hybrid-Machine-Learning-System-for-Stock-Market-Choudhry-Garg/94c33ce14bb84baa2dd49ad9ed49cbf680a15bfe

Krauss, C., Do, X., Anh, & Huck, N. (n.d.). *A Service of zbw Leibniz-Informationszentrum Wirtschaft Leibniz Information Centre for Economics*. https://www.econstor.eu/bitstream/10419/130166/1/856307327.pdf

Nyberg, H. (2013). Predicting bear and bull stock markets with dynamic binary time series models. *Journal of Banking & Finance*, *37*(9), 3351–3363. https://doi.org/10.1016/j.jbankfin.2013.05.008

Subha. (2024, March 29). *Boosting — Adaboost, Gradient Boost and XGBoost*. Medium; Medium. https://medium.com/@pingsubhak/boosting-adaboost-gradient-boost-and-xgboost-bdda87eed44e

Zhang, Y., Ma, F., & Wang, Y. (2019). Forecasting crude oil prices with a large set of predictors: Can LASSO select powerful predictors? *Journal of Empirical Finance*, *54*, 97–117. https://doi.org/10.1016/j.jempfin.2019.08.007

Zhong, X., & Enke, D. (2019). Predicting the daily return direction of the stock market using hybrid machine learning algorithms. *Financial Innovation*, *5*(1). https://doi.org/10.1186/s40854-019-0138-0

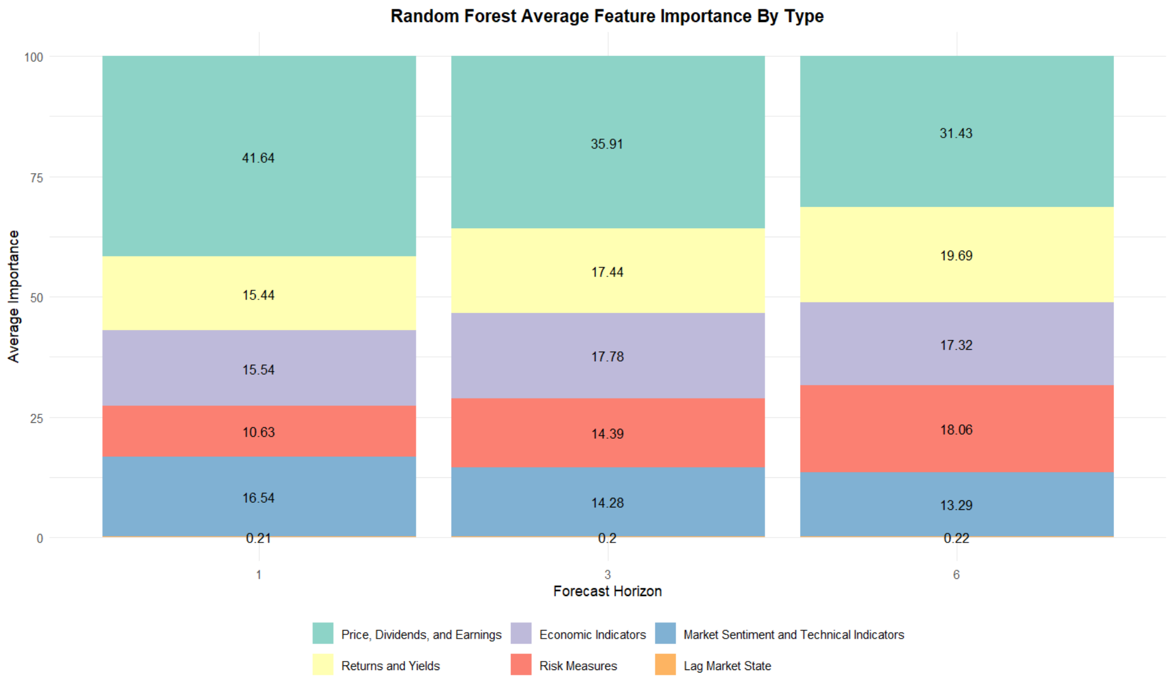
7. Appendix

Grouping of predictors:

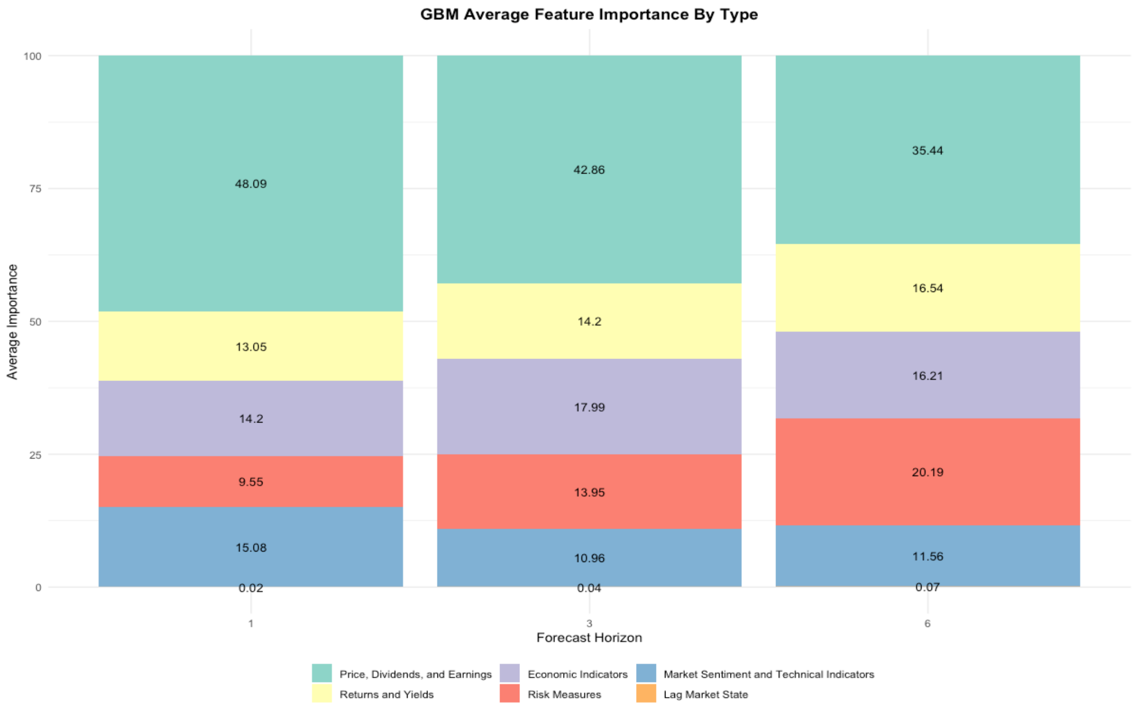
Price, Dividends and Earnings	Return with dividends, return without dividends, dividend price ratio, dividend yield ratio, earnings price ratio, dividend payout, book-to-market ratio, book-to-market cross section factor, price, dividend, earnings, total return CAPE, long interest rate GS10
Economic Indicators	Inflation, net equity issuance, production-output gap, oil price changes, CPI, unemployment rate,
Returns and Yields	AAA bond yield, BAA bond yield, long government yield, long government return, corporate bond return, risk free return, term spread, default yield spread, default return spread, monthly total bond returns
Risk Measures	Average stock skewness, tail risk from cross-section, 9 illiquidity measures, stock return dispersion
Market Sentiment and Technical Indicators	Distilled sentiment, 14 technical indicators, Dow 52-week high, Dow all-time high, short stock interest, average correlation of daily stock returns, stock-bond yield gap
Lags of Market State	

Feature Importance Plot

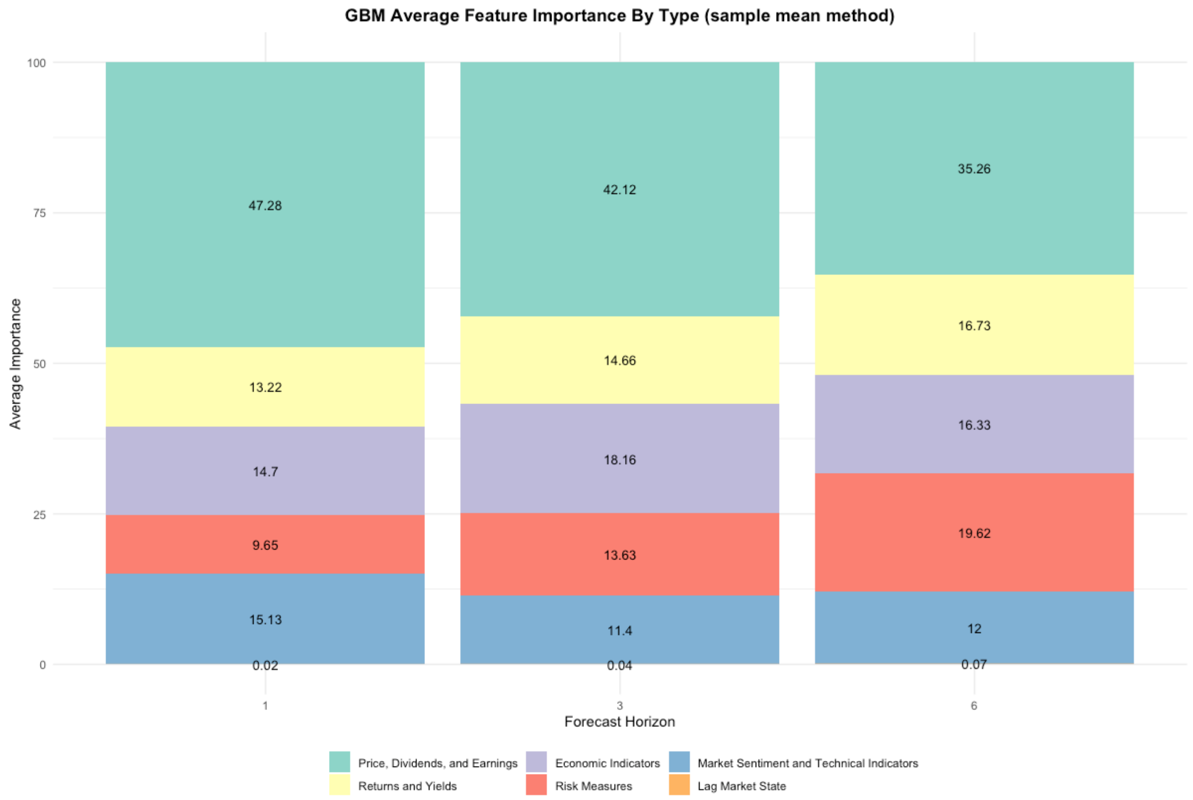
Random Forest



GBM



GBM using sample mean method


GBM Average Feature Importance By Type (sample mean method)
Average Importance
Forecast Horizon
Price, Dividends, and Earnings
Returns and Yields
Economic Indicators
Risk Measures
Market Sentiment and Technical Indicators
Lag Market State