https://doi.org/10.26219/heal.aueb.5657

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

MODELING AND FORECASTING THE S&P 500 'S BOOK VALUE GROWTH DIRECTION USING ECONOMETRIC AND MACHINE LEARNING MODELS

Written by Olympia Paschalidi

M.Sc. Thesis Submitted to the Department of Statistics of the Athens University of Economics and Business in partial fulfillment of the requirements for the degree of Master of Science in Statistics

Athens, Greece

August 2023

https://doi.org/10.26219/heal.aueb.5657

CONTENTS

https://doi.org/10.26219/heal.aueb.5657

CHAPTER 1

INTRODUCTION

1.1The aim of the analysis

This study focuses on the vital aspect of directional forecasting and predictability within financial markets, which holds immense importance for making well-informed investment decisions and effectively managing portfolios. Our research endeavors to forecast the direction of book value growth in the S&P 500 index and explore the predictive ability of multiple variables by employing classification models. The literature review highlights the importance of directional forecasting and the use of different econometric and statistical models for our aim. However, the existing literature also points out the limitations of traditional models in terms of predictive power and accuracy.

To overcome these limitations, we propose techniques that integrate machine learning algorithms such as Random Forest. By combining advanced techniques with traditional fundamental analysis, we aim to enhance the accuracy and effectiveness of directional forecasting in the stock market. This innovative approach holds the potential to provide more reliable and robust predictions for investors and analysts.

We explore the predictive ability of multiple variables using machine learning techniques and Logit/Probit models in our analysis. The primary objective is to identify the critical factors that contribute to the predictability of book value growth, offering valuable insights to investors and financial analysts. Our findings have practical implications in directional forecasting and underscore the effectiveness of diverse modeling approaches in this context.

1.2 An introduction to the S&P 500 index

The S&P 500 is a renowned stock market index that evaluates the performance of the 500 largest publicly traded companies in the United States. As a highly regarded and extensively monitored index, it serves as a crucial indicator of the overall status and progress of the U.S. equity markets. Investors and financial professionals often use it as a benchmark to gauge the health and performance of the stock market in the country. Book value growth refers to the increase in the book value of a company or an investment over a specific period of time. It is a financial metric used to assess the growth of a company's net worth or the value of its assets. The book value of a company is calculated by subtracting its total liabilities from its total assets. In our

https://doi.org/10.26219/heal.aueb.5657

case, we aim to predict the direction of book value growth of the S&P 500 which represents the aggrerate book value growth of the companies within the index. The sign of book value growth (+ or -) indicates the direction of the change in the company's net worth over the specified period. Positive book value growth signifies growth and value creation, while negative book value growth signifies a decline in value. Modeling and analyzing this topic can be important for several reasons in relation to the Market Performance, Economic Health, Investment Decisions and Risk Assessment.

https://doi.org/10.26219/heal.aueb.5657

CHAPTER 2

LITTERATURE REVIEW

Directional Forecasting Methods for Business Cycle Sign Prediction: Directional forecasting, a pivotal aspect of economic forecasting, is concerned with predicting the direction or trend of variables, especially in the context of business cycles. This review delves into the various methodologies employed to forecast the directional movement of economic indicators, with a specific focus on predicting business cycle turning points.

Historical Context and Foundational Approaches: The pursuit of early-warning indicators capable of identifying changes in the business cycle has a long-standing history. Pioneering the field, Burns and Mitchell (1946) sought to delineate the cyclical nature of time series data by identifying specific cycles and their start and end dates. Their groundwork paved the way for the U.S. Department of Commerce's widely used economic indicators for business cycle prediction.

Statistical Frameworks and Classification*:* Stock and Watson (1989, 1991, 1993) introduced a modeling framework centered around co-movements in economic activity components, offering an alternative index to traditional indicators. By inducing symmetric shocks to a stable time series system, this model generates economic cycles, capturing recessions and expansions.

Probit/Logit Models and Extensions: Early applications of Probit/Logit models examined financial crises and U.S. recessions. Estrella and Mishkin (1998) showcased the predictive power of financial variables, especially the yield curve slope, in forecasting recessions. Dynamic extensions incorporating lagged recession parameters and time variation in coefficients improved predictive performance (Dueker, 1997; Chauvet and Potter, 2005; Kauppi and Saikkonen, 2008).

Markov Switching Models for Regime Changes: Markov switching models have gained prominence for identifying and forecasting economic regime changes. Hamilton's (1989) Hidden Markov model enabled describing changes through latent Markov processes. Chauvet and Hamilton (2006) intertwined binary recession indicators with GDP rates, Nyberg (2018) introduced a regime-switching vector autoregressive model, and Tian and Shen (2019) extended Markov switching models in recession prediction.

https://doi.org/10.26219/heal.aueb.5657

Machine Learning Approaches: Recent advancements incorporate machine learning algorithms for business cycle forecasting. Gogas et al. (2015) used Support Vector Machines (SVM) to predict GDP growth based on the U.S. yield curve. Berge (2015) highlighted the superiority of boosting models and Bayesian Model Averaging (BMA). Dopke et al. (2017) demonstrated boosted regression trees' superiority in predicting German recessions.

News-Based Uncertainty Indices and Real-Time Forecasting: Pierdzioch and Gupta (2019) explored news-based uncertainty indices' effectiveness in predicting U.S. recessions using boosted regression trees. Davig and Hall (2019) showcased the superiority of Naive Bayes models over traditional models and survey-based forecasts in real-time recession forecasting.

In conclusion, directional forecasting for business cycle sign prediction has a rich history and has evolved with advancements in statistical methodologies and machine learning techniques. Probit/Logit models, Markov switching models, and machine learning algorithms are instrumental in identifying early-warning indicators for business cycle transitions. Recent developments, including news-based indices and real-time forecasting, provide new avenues for accurate and timely directional forecasts.

https://doi.org/10.26219/heal.aueb.5657

CHAPTER 3

ECONOMETRIC MODELS AND MACHINE LEARNING TECHNIQUES

At this stage, we will conduct a comprehensive analysis of the Econometric models and machine learning techniques utilized for modeling and prediction of the direction of book value growth (BVG).

3.1 Logit regression models

Let Yt is our dependent variable which takes values 0 if there is a decrease in the stock return or 1 otherwise in month t, and Xt is the corresponding set of predictors. Since Yt is a random variable which follows a Bernoulli distribution with probability function $p_t^{y_t}(1-p_t)^{1-y_t}$, where pt is the probability of a positive stock return, the aim of logistic regression is to model and predict the conditional probability of increase of a stock price $\Pr(Y_t = 1 \mid X_{t-h})$ given a set of covariates at lag h and a function pt. Logistic regression uses the linear form

$$\ln\left(\frac{p_t}{1-p_t}\right) = X'_{t-h}\,\mathrm{b} = b_0 + \sum_{i=1}^{n} b_i x_{i,\mathrm{t-h}}$$

where:

- $p_t = \Pr(Y_t = 1 \mid X_{t-h})$ is the conditional probability of increase of a stock price at time t given the covariates at time t-h
- $\ln\left(\frac{p_t}{1-p_t}\right)$ is the logit transformation
- X'_{t-h} is a vector which has elements the variables with values $x_{i,\mathrm{t-h}}$, i=1,…,N, t=1,…,T
- b is the parameter vector to be estimated b=(b0 ,b1 , b2, …, bN)

The above equation can be written otherwise as

$$p_t = \frac{\exp\left(b_0 + \sum_{i=1}^{n} b_i x_{i,t-\mathrm{h}}\right)}{1 + \exp\left(b_0 + \sum_{i=1}^{n} b_{ixi,t-\mathrm{h}}\right)}$$

The parameters are estimated using Maximum Likelihood (ML) methods. Let T represent the total sample size, and $(Y_t, X_{1,t}, \ldots, X_{N,t})$ denotes the corresponding observations taking place at time t, where t=1,…,T. Assuming that each Yt follows a

https://doi.org/10.26219/heal.aueb.5657

Bernoulli distribution characterized by the ascension probability p_t, the log-likelihood function is written as:

$$\begin{aligned}\ln f(Y \mid X, \beta) &= \sum_{t=1}^{T} [y_t \ln p_t + (1 - y_t) \ln(1 - p_t)] = \\ &= \sum_{t=1}^{T} \ln[1 - p_t] + \sum_{t=1}^{T} y_t \ln\left[\frac{p_t}{1 - p_t}\right] = \\ &= -\sum_{t=1}^{T} \ln\left[1 + \exp\left(\beta_0 + \sum_{i=1}^{N} \beta_i x_{i,t-h}\right)\right] + \sum_{t=1}^{T} y_t \left(\beta_0 + \sum_{i=1}^{N} \beta_i x_{i,t-h}\right)\end{aligned}$$

The maximum likelihood estimation (MLE) method seeks to find the values of parameter vector b that maximize the log-likelihood function. This is often done using numerical optimization techniques, such as gradient descent or specialized optimization algorithms, or Newton-Raphson type algorithms of the form:

$$\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_t - [\mathrm{H}\ell(\boldsymbol{\beta}_t)]^{-1} \nabla \ell(\boldsymbol{\beta}_t),\ t = 1,2, \ldots$$

3.2 Probit regression models

The second model used in empirical applications is the Probit regression model which is similar to Logit models, but it is based on probit link function. In Probit model, the cumulative standard normal distribution function Φ(·) is used to estimate the conditional probability of an increase of a stock price $\mathrm{Prob}\ (Y_t = 1 \mid \mathrm{X}_{t-h})$ as follows:

$$\mathrm{Prob}\ (Y_t = 1 \mid \mathrm{X}_{t-h}) = \Phi(\Psi_t)$$

where Φ(·) is the cumulative standard normal distribution function, and

$$\Psi_t(\beta) = X'_{t-h}\beta.$$

The parameter vector β =(β0, β1, …, βN) is estimated by maximizing the likelihood function as follows:

$$\hat{\beta}_{ML} = \ \mathrm{argmax}_{\beta} \left\{ \sum_{\{Y_t=1\}} \ln\ \Phi(\Psi_t(\beta)) + \sum_{\{Y_t=0\}} \ln\left(1 - \Phi(\Psi_t(\beta))\right) \right\}.$$

https://doi.org/10.26219/heal.aueb.5657

3.3 Linear Discriminant Analysis (LDA)

The main idea of discriminant analysis is to classify observations in to populations with known distributions by creating discriminant rules. LDA aims to find a linear combination of features that best discriminate between different classes. This linear combination is known as a discriminant function. Supposing that we knew fk(xt-h) = P(X = xt-h|Y = k) for all of our classes k, where fk is the density function for a particular x for a particular class we can use Bayes theorem as follows:

$$\left.\mathbb{P}(Yt = k \mid X = xt - h) = \frac{\mathbb{P}(Y = k) f_k(x_{t-h})}{\sum_{j=0}^{1} \mathbb{P}(Y = j) f_j(x_{t-h})}\right\} \text{ probability}$$

$\mathbb{P}(Y = k) := \pi_k$ prior probabilities k=0,1 the two classes

We need to estimate $\mathrm{P}(X = \mathrm{x_{t-h}} \mid Y = k)$ and prior probabilities π_k in order to use the generative form of the Bayes classifier. Linear discriminant analysis (LDA) does this by assuming that the data within each class are normally distributed:

$$f_j(x_{t-h}) = \mathrm{P}(X = x_{t-h} \mid Y = j) = N(\mu_j, \Sigma) \text{ density}$$

We allow each class to have its own mean $\mu_j \in \mathbb{R}^p$, but we assume a common covariance matrix $\Sigma \in \mathbb{R}^{p \times p}$. Hence

$$f_j(x_{t-h}) = \frac{1}{(2\pi)^{p/2} \det(\Sigma)^{1/2}} \exp\left\{-\frac{1}{2}(x - \mu_j)^T \Sigma^{-1}(x - \mu_j)\right\}$$

What does the decision rule look like? We want to find j so that $\mathrm{P}(Y = j \mid X = \mathrm{x_{t-h}}) \cdot \pi_j = f_j(\mathrm{x_{t-h}}) \cdot \pi_j$ is the largest. We can define the rule:

$$f^{\mathrm{LDA}}(x) = \underset{j=1,\dots K}{\operatorname{argmax}} \frac{1}{(2\pi)^{p/2} \det(\Sigma)^{1/2}} e^{-\frac{1}{2}(x_{t-h} - \mu_j)^T \Sigma^{-1}(x_{t-h} - \mu_j)} \cdot \pi_j$$

$$= \underset{j=1,\dots K}{\operatorname{argmax}} -\frac{1}{2}(x_{t-h} - \mu_j)^{\mathrm{T}} \Sigma^{-1}(x_{t-h} - \mu_j) + \ln \pi_j$$

$$= \underset{j=1,\dots K}{\operatorname{argmax}} \delta_j(x) = x_{t-h}{}^{\mathrm{T}} \Sigma^{-1} \mu_j - \frac{1}{2} \mu_j^{\mathrm{T}} \sum^{-1} \mu_j + \ln \pi_j$$

We call $\delta_j(x), j = 1, \dots K$, the discriminant functions.

3.4 Quadratic Discriminant Analysis(QDA)

Quadratic Discriminant Analysis (QDA) is a statistical classification technique that extends the principles of Linear Discriminant Analysis (LDA) to situations where the

https://doi.org/10.26219/heal.aueb.5657

assumption of equal covariance matrices among classes does not hold. In QDA, each class is assumed to have its own covariance matrix, allowing for more flexibility in capturing the underlying data distribution. All the same math goes through, giving Quadratic Discriminant Functions (and QDA)

$$\begin{aligned}\delta_k(x_{t-h}) &= -\frac{1}{2}(x_{t-h}-\mu_k)^T\Sigma^{-1}(x_{t-h}-\mu_k) - \frac{1}{2}\log|\Sigma_k| + \log\,\pi_k \\ &= -\frac{1}{2}x_{t-h}{}^T\Sigma_k^{-1}x_{t-h} + \mu_k\Sigma_k^{-1}x_{t-h} - \frac{1}{2}\mu_k^T\Sigma_k^{-1}\mu_k - \frac{1}{2}\log|\Sigma_k| + \log\,\pi_k\end{aligned}$$

The Σ matrices are now different in each function, and the boundaries $\delta_k(x_{t-h}) > \delta_{k'}$ are no longer linear. Instead, they are quadratic. This allows the boundaries to curve around groups to account for different patterns of spread.

3.5 Ridge Logit regression models

Ridge regression, also known as Tikhonov regularization or L2 regularization, is a linear regression technique used to solve multicollinearity and overfitting problems in predictive modeling. It extends ordinary least squares (OLS) regression by adding a penalty term to the regression coefficients that encourages them to remain small without setting them exactly to zero. This penalty term helps avoid large coefficients that can lead to overfitting and makes the model more stable. It is suitable for datasets with high dimension and many predictors. This method modifies the log likelihood function of the Logit regression model by adding a shrinkage penalty based on the Euclidean norm (L2 norm) of the coefficients β and a tuning parameter λ (lambda) that controls the strength of the penalty term. The estimation of the Ridge's coefficients is given by the form:

$$\hat{\beta}_R^{\text{RIDGE}} = \text{argmax}_\beta \left\{\ln\, f(Y \mid X,\beta) - \lambda\textstyle\sum_{j=1}^{N}\beta_j^2\right\}.$$

3.6 Lasso Logit regression models

When the set of predictors is large, we would like to make variable selection at the same time. The Lasso (Least Absolute Shrinkage and Selection Operator) is also a regularization technique which differs in how the penalty terms are added to the regression coefficients from Ridge. Specifically the shrinkage penalty is based on absolute values of the coefficients β so that some coefficients of the model to become exactly zero

$$\hat{\beta}_L^{\text{LASSO}} = \text{argmax}_\beta \left\{\ln f\,(Y \mid X,\beta\,) - \lambda\sum_{j=1}^{N}|\beta_j|\right\}$$

https://doi.org/10.26219/heal.aueb.5657

3.7 Elastic Net Approach

Elastic Net is a regularization technique used in linear regression and other machine learning models, combining the strengths of both Lasso (L1 regularization) and Ridge regression (L2 regularization). It aims to address the limitations of each method while providing a balance between feature selection and coefficient stability. Here is the form of Elastic Net:

$$\hat{\beta}^{ELASTICNET} = \text{argmax}_{\beta} \left\{ \ln f(Y \mid X, \beta) - \lambda \left((1-\alpha) \sum_{j=1}^{N} \frac{\beta_j^2}{2} + \alpha \sum_{j=1}^{N} |\beta_j| \right) \right\}$$

where $\alpha = 0.5$.

3.8 Decision Trees

It is a supervised machine learning algorithm that can be used for both regression (predicting continuous values) and classification (predicting categorical values) problems. They also serve as a basis for more advanced techniques such as bagging, boosting, and random forests.

A decision tree begins with a root node representing the entire population or sample, which is then divided into two or more uniform groups by a method called splitting. When the child nodes undergo further splitting, they are called decision nodes, while the nodes that do not split are called terminal nodes or leaves. A segment of a complete tree is called a branch.

We have established that decision trees can be used for both regression and classification tasks; therefore, let us now understand the algorithm behind the different types of decision trees.

To build a regression tree, first use recursive binary splitting to create a large tree on the training data that stops only when each terminal node has fewer than a minimum number of observations. Recursive binary splitting is a greedy top-down algorithm to minimize the residual sum of squares (RSS), an error measure also used in linear regression settings. The RSS is given by in the case of a partitioned feature space with M partitions:

$$\text{RSS} = \sum_{m=1}^{M} \sum_{i \in R_m} \left(y_i - \hat{y}_{R_m} \right)^2$$

Starting at the tree's root node, it's divided into two branches, thus forming two distinct regions. This initial division is replicated multiple times, and for each

https://doi.org/10.26219/heal.aueb.5657

replication, the features are chosen in a way that minimizes the residual sum of squares (RSS) for the current configuration.

Afterward, the approach incorporates cost complexity pruning, which involves refining the large tree into a sequence of optimal subtrees, taking into account a parameter known as α. This parameter introduces a balance between the tree's depth and its ability to accurately represent the training data. The purpose of this is to find the right compromise between tree complexity and its fit to the data.

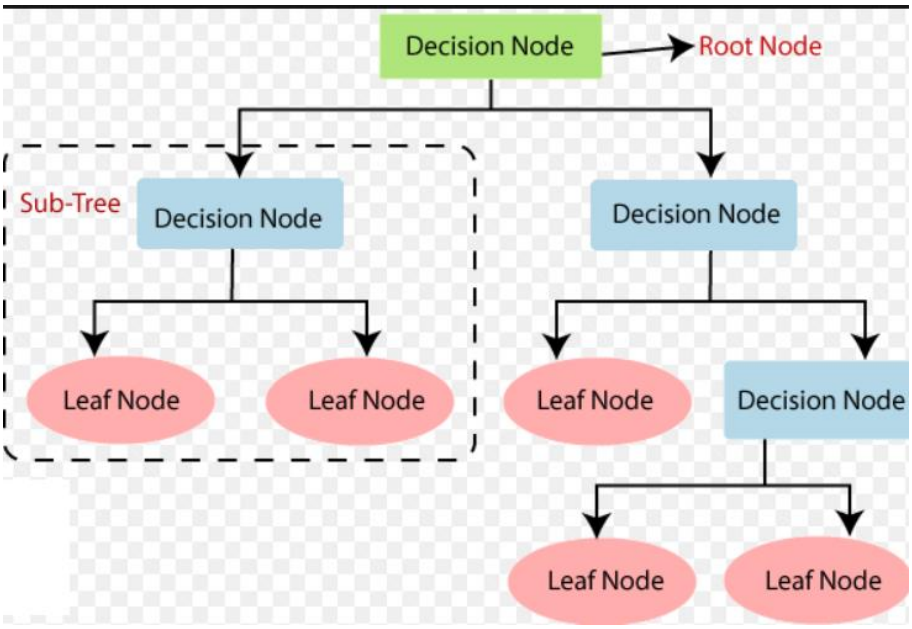


Figure 1: Decision tree structure

In a classification tree, the primary goal is to predict the class label of an observation, as opposed to predicting a numerical value in a regression tree. The process of constructing a classification tree shares similarities with building a regression tree, particularly the use of recursive binary splitting to segment the feature space. However, there are key differences in the way the splitting criteria are determined and the way predictions are made in a classification tree:

Prediction for a Terminal Node (Leaf): In a regression tree, the prediction for an observation in a terminal node is typically the mean of the response values of the training observations that belong to that node. In contrast, for a classification tree, the prediction is the class label that is most prevalent among the training observations within that terminal node. This means that the majority class in the terminal node becomes the predicted class for any observation that falls into that region.

https://doi.org/10.26219/heal.aueb.5657

Splitting Criterion: In regression trees, the split criterion is often based on minimizing the residual sum of squares. In classification trees, however, the Residual Sum of Squares is not applicable since we are dealing with categorical response variables. Instead, various methods can be used to determine the quality of a split. The most known criterions are:

- Classification Error Rate

The classification error rate is the proportion of training samples in a node that don't belong to the majority class. It represents the probability of misclassification if a sample is assigned the majority class of that node. Mathematically, it is calculated as:

$$\text{Classification Error Rate } (p) = 1 - \max(p_1, p_2, \dots, p_K)$$

- Gini Impurity (Gini Index)

The Gini impurity is a measure of the likelihood of misclassifying a randomly chosen element from a node if it were randomly labeled according to the distribution of classes in that node. For a node with K classes, let pi be the proportion of samples belonging to class i. The Gini impurity Gini(p) for that node is calculated as follows:

$$\text{Gini } (p) = 1 - \sum_{i=1}^{K} p_i^2$$

A Gini impurity of 0 indicates that the node is pure (contains only one class).

- Entropy

Entropy is a measure of the uncertainty or disorder in a node. For a node with K classes, let pi be the proportion of samples belonging to class i. The entropy Entropy(p) for that node is calculated using the formula:

$$\text{Entropy } (p) = -\sum_{i=1}^{K} p_i \log_2 (p_i)$$

3.9. Random Forests

Ensemble Methods offer a powerful approach to enhance the reliability and performance of decision-making through collaboration among multiple Decision Trees. Instead of depending solely on a single Decision Tree's judgments made at each split, ensemble Methods enable us to reduce overfitting and identify variables and splits that contribute in better predictions. Let's now look at tree-based ensemble methods that leverage the strengths of decision trees while balancing some of their limitations: Bagging and Random Forests.

https://doi.org/10.26219/heal.aueb.5657

Bagging (Bootstrap Aggregating) integrates two key techniques - Bootstrapping and Aggregation - into a single ensemble model. It operates by creating multiple bootstrapped subsamples from a given dataset. On each of these subsamples, a Decision Tree is constructed. After forming Decision Trees on each subsample, an aggregation algorithm is employed to combine the predictions generated by these individual Decision Trees, resulting in an optimized and powerful predictor. This process not only leverages the benefits of varied subsampling and Decision Tree construction but also ensures a cohesive and robust prediction outcome through intelligent aggregation of their results.

A random forest (RF) classifier stands as an ensemble classifier that generates multiple decision trees, employing a randomly chosen subset of training examples and variables. This classifier's popularity has surged among the remote sensing community due to its classification accuracy. The main aim of this study was to assess the application of the random forest classifier in remote sensing. This evaluation has uncovered that the random forest classifier adeptly manages high data dimensionality and multicollinearity. It exhibits both speed and resilience against overfitting. However, it does show sensitivity to the way samples are selected. The random forest classifier's measurement of variable importance (VI) has been extensively harnessed in various scenarios. For instance, it's been used to reduce the dimensionality of hyperspectral data, identify pertinent multisource remote sensing and geographic information, and determine the optimal season for classifying specific target categories. Further explorations are needed into less commonly explored uses of this classifier, such as employing it for analyzing sample proximity to detect and eliminate outliers within training samples.

The training procedure for random forests employs a fundamental method known as bootstrap aggregating, or bagging, in relation to the tree learning process. When presented with a training dataset $X = x_1, \dots, x_n$ alongside corresponding responses $Y = y_1, \dots, y_n$, the bagging process involves selecting a random subset from the training set with replacement, and then creating and fitting trees based on these subsets. This is done iteratively for a specified number of times, denoted as B.

1. Sample, with replacement, n training examples from X, Y; call these X_b, Y_b
2. Train a classification or regression tree f_b on X_b, Y_b. After training, predictions for unseen samples x' can be made by averaging the predictions from all the individual regression trees on x':

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x')$$

https://doi.org/10.26219/heal.aueb.5657

Alternatively, in the case of classification trees, the final decision is made by taking the majority vote among the predictions. This process of bootstrapping contributes to enhanced model performance as it reduces the model's variability without introducing additional bias. In practical terms, this implies that the predictions made by an individual tree can be greatly influenced by random fluctuations in its training data. However, when you consider the average prediction across numerous trees, this susceptibility to noise diminishes, granted that the trees are not closely related. If multiple trees were trained solely on a single training dataset, they would exhibit strong correlations (or potentially even be identical if the training process is deterministic). Bootstrapping counters this by presenting each tree with different training datasets, effectively reducing their interdependence.

Additionally, an estimate of the uncertainty of the prediction can be made as the standard deviation of the predictions from all the individual regression trees on x':

$$\sigma = \sqrt{\frac{\sum_{b=1}^{B} \left(f_b(x') - \hat{f}\right)^2}{B-1}}$$

The quantity of samples or trees, denoted as B, represents a flexible parameter. Generally, the selection of a few hundred to several thousand trees is common, with the actual number being contingent upon the characteristics and extent of the training dataset. Determining the ideal count of trees, B, can be achieved through techniques like cross-validation or by assessing the out-of-bag error. The out-of-bag error involves calculating the average prediction error for each training sample x_{i}, exclusively employing the trees that did not include x_{i} within their bootstrap sampling process. As the number of trees increases, both the training and test errors tend to stabilize and show diminishing returns.

From bagging to random forests

The previously outlined process outlines the original bagging technique for trees. Random forests, however, incorporate an additional form of bagging strategy: they employ an altered tree learning method that, during the learning phase's candidate split selection, picks a random subset of the available features. This approach is sometimes referred to as "feature bagging." This approach is motivated by the interrelatedness of the trees within a standard bootstrap sample. When a few features are notably strong predictors for the response variable (target output), these features tend to be chosen in a significant number of the B trees, leading to correlated outcomes among the trees.

For classification tasks involving p features, roughly $\sqrt{p}$ (rounded down to the nearest whole number) features are typically utilized in each split. In the case of

https://doi.org/10.26219/heal.aueb.5657

regression tasks, the developers suggest using $p/3$ (rounded down) features, while maintaining a minimum node size of 5 as the default. In real-world scenarios, determining the optimal values for these parameters should be done on a case-by-case basis for each problem.

3.10 Evaluation of directional forecast models

In this section we present a comprehensive evaluation of the directional forecasting models employed in this study. When evaluating classification models, several statistical metrics are commonly used to assess their performance. These metrics provide insights into how well the model is performing in terms of correctly classifying instances into their respective classes. In order to compute these metrics we should be construct a Confusion Matrix. It is presented as a table in which the predicted class is compared with the actual class as follows:

Confusion Matrix

		Predicted Value	
		Positive	**Negative**
Real Value	**Positive**	TP	FP
	Negative	FN	TN

where :

True Positive (TP): you predicted positive, the real value was positive

True Negative (TN): you predicted negative, the real value was negative

False Positive (FP): you predicted positive, the real value was negative

False Negative (FN): you predicted negative, the real value was positive

After constructing the confusion matrix, then the key statistical evaluation metrics for classification can be easily computed. We present several classification metrics:

1. Accuracy

Accuracy is defined as the proportion of correctly classified observations over the whole set of observations.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

2. Precision

https://doi.org/10.26219/heal.aueb.5657

Precision measures the proportion of correctly predicted positive samples among all predicted positive samples.

$$\text{Precision} = \frac{TP}{TP+FP}$$

3. Recall (Sensitivity, True Positive Rate)

The recall is defined as the number of true positives divided by the sum of true positives and false negatives. It expresses the ability to find all relevant instances in a dataset. Recall measures how good your model is at correctly predicting positive cases. It's the proportion of actual positive cases which were correctly identified

$$\text{Sensitivity} = \frac{TP}{TP + FN}$$

4. Specificity

Specificity is similar to sensitivity, only the focus is on the negative class. It is the proportion of true negative cases which were correctly identified as such. The equation for specificity is:

$$\text{Specificity} = \frac{TN}{TN + FP}$$

5. Mean classification error (MCE)

MCE calculates the average error rate across all classes in a dataset. It is given by:

$$\text{MCE}=1\text{-accuracy}$$

6. Balanced accuracy

The balanced accuracy takes the average of the sensitivity (true positive rate) and specificity (true negative rate) to provide a balanced assessment of the model's performance across different classes.

$$\text{BalancedAccuracy} = \frac{\text{Sensitivity} + \text{Specificity}}{2}$$

7. F1-Score

The F1-score is the harmonic mean of precision and recall. It balances both precision and recall and is especially useful when classes are imbalanced.

$$F_1 \quad \text{Score} = \frac{2}{\frac{1}{\text{Recall}}+\frac{1}{\text{Precision}}} = \frac{2TP}{2TP+FP+FN}.$$

8. The Area Under the Curve (AUC)

https://doi.org/10.26219/heal.aueb.5657

The Area Under the Curve (AUC) measures the model's ability to discriminate between positive and negative instances across various classification thresholds. The AUC is plotted on a ROC (Receiver Operating Characteristic) curve, where the x-axis represents the False Positive Rate (FPR), and the y-axis represents the True Positive Rate (TPR), also known as sensitivity.

- An AUC of 0.5 indicates that the model's performance is no better than random chance.
- An AUC greater than 0.5 indicates that the model's performance is better than random chance.
- An AUC of 1 signifies perfect classification, where the model's predictions are always correct.

https://doi.org/10.26219/heal.aueb.5657

CHAPTER 4

EMPIRICAL APPLICATION

4.1 About the data

This section discusses how real data are treated in this study for modeling based on the methods discussed, the evaluation of each model, and examines whether there are differences between Logit/Probit model with machine learning techniques let alone the existence of predictive ability of some variables.

First of all, lets look at the data description. As it mentioned our variable-outcome is the direction Book Vallue Growth of S&P 500 which takes values 0 (decrease or negative return of the index or 1 otherwise, and our predictors variables are 45.

https://doi.org/10.26219/heal.aueb.5657

Table 1

Variable No.	Variable Name	Description
X1	10_Year_Treasury_Yield	The yield on the 10-year U.S.
X2	V3_Month_Treasury_Yield	he yield on the 3-month U.S. Treasury bill,
X3	Term_spread	The difference between the yields on long-term and short-term government bonds.
X4	TED_spread	The difference between the 3-month Treasury bill yield and the 3-month LIBOR (London Interbank Offered Rate) yield.
X5	Credit_spread	The difference in yields between bonds with higher credit risk (like corporate bonds) and lower risk (like government bonds)
X6	Gold_price	The price of gold per unit. Gold is often considered a safe-haven asset and can be used as an indicator of market sentiment.
X7	ECU1	Not specified in the provided context. It could refer to a specific economic indicator or currency.
X8	MACVOL	Not specified in the provided context. It might refer to a measure of market volatility.
X9	VXO_Average	The average value of the VXO (Volatility Index - also known as the "fear index") over a certain period. It measures market expectations of near-term volatility.
X10	Industrial_production_yoy	The year-over-year percentage change in industrial production, indicating the health of the manufacturing sector.
X11	ISM_Manufacturing_Index_New_Orders_D	The Institute for Supply Management's Manufacturing Index, specifically focusing on new orders. It reflects the manufacturing sector's business activity and health.
X12	Chicago_Fed_National_Financial_Conditions_Index	An index that summarizes financial conditions in the U.S., indicating how easy or tight credit is in the economy.
X13	Chicago_Fed_National_Financial_Conditions_Leverage_SubIndex	A subindex of the Chicago Fed National Financial Conditions Index, focusing on leverage-related conditions.
X14	Chicago_Fed_National_Activity_Index	An index that combines multiple economic indicators to gauge overall economic activity and inflationary pressure.
X15	Core_PPI_Inflation_yoy	The year-over-year change in the Producer Price Index (PPI) for core goods and services, excluding food and energy. It measures inflation at the wholesale level.
X16	USD_	U.S. dollar performance
X17	S&P_500_ret	S&P_500 return

https://doi.org/10.26219/heal.aueb.5657

X18	S&P_500_3M	S&P_500 3 months return
X19	S&P_500_6M	S&P_500 6 months return
X20	S&P_500_12M	S&P_500 12 months return
X21	S&P_500_12_1M	S&P_500 12 months total return
X22	CONSSENT_D	Consumer sentiment index data.
X23	Real_Money_Supply_M2	The money supply measure M2 adjusted for inflation, providing insight into the growth of money in the economy.
X24	S&P 500 DY	Dividend yield of the S&P 500 index.
X25	S&P 500 PE	Price-to-earnings ratio of the S&P 500 index.
X26	S&P 500 EBYD	Earnings-based yield of the S&P 500 index.
X27	SMB	SMB measures the historical excess return of small-cap stocks over large-cap stocks
X28	HML	HML captures the historical excess return of value stocks over growth stocks
X29	RMW	RMW captures the historical excess return of stocks of companies with strong profitability over those with weak profitability
X30	CMA	CMA measures the historical excess return of stocks of companies with conservative investment policies over those with aggressive policies
X31	MOM	MOM captures the historical excess return of stocks that have shown recent strong performance over those that have shown recent weak performance
X32	STREV	STREV is the opposite of momentum. It captures the historical excess return of stocks that have recently performed poorly but have the potential to bounce back in the short term.
X33	RVAR	RVAR measures the historical excess return of stocks with low residual volatility (less sensitive to market movements) over those with high residual volatility
X34	BAB	BAB captures the historical excess return of low-beta stocks (stocks with lower market sensitivity) over high-beta stocks.
X35	HMLD	HMLD measures the historical excess return of stocks of companies that pay low dividends over those that pay high dividends.
X36	QMJ	HMLD measures the historical excess return of stocks of companies that pay low dividends over those that pay high dividends.
X37	TSMOM	TSMOM captures the historical excess return of assets that have shown strong

https://doi.org/10.26219/heal.aueb.5657

X38	MOMCM	MOMCM captures the historical excess return of stocks that have experienced a recent momentum crash, i.e., a sudden drop in their positive momentum.
X39	MOMEQ	MOMEQ captures the historical excess return of stocks based on their equal-weighted momentum, which means that all stocks with momentum are given equal importance.
X40	MOMFI	MOMFI captures the historical excess return of stocks based on their momentum, weighted by some fundamental factor (e.g., revenue, earnings, etc.).
X41	MOMFX	MOMFX captures the historical excess return of stocks based on their momentum, where "excess return" refers to the return beyond a risk-free rate.
X42	S&P 500 EYDVG	S&P 500 earnings yield
X43	S&P 500 DYDVG	S&P 500 dividend yield
X44	S&P 500 FPE	S&P 500 Forward estimate P/E ratio
X45	Kansas City Financial Stress Index	An index measuring stress in the financial markets and economy, often used as an indicator of systemic risk.

The table reports all the predictors / independent variables that are used in all models.

The dataset consists of 293 monthly observations covering the period from July 1995 to November 2019. The number of months with a negative sign of BVG of S&P 500 is 156, while the number of months with a positive sign is 137.

4.2 Applying the methods in data.

Logit/Probit Models

As it concerned to classical econometric Logit/Probit models the dataset is divided into two samples the train set and test set. The train set is used to estimate the coefficients

https://doi.org/10.26219/heal.aueb.5657

of the models and examine which predictors are most statistically significant, and the test set for their predictive evaluation.

Since the dataset is a time series analysis, the order of the variables can play an important role in understanding the relationships between them and making accurate predictions. The general guideline is to arrange the variables in chronological order, starting with the earliest period and progressing to the most recent. For this reason, the train set and test set must include consecutive observations to preserve the time dependence of the observations. Thus, in this study, the train set includes observations from July 1995 to November 2014 (233 months), and the test set includes observations from December 2014 to November 2019 (60 observations).

Also, the data frame used for the analysis contains the variables in such a way that the independent variables Xt are mapped to Yt+1, and therefore the problem is to find out which variables have predictive ability, i.e. to investigate whether the explanatory or predictor variables at time t can predict the dependent directional variable Y at time t+1 (upward or downward move). After the models were exercised with the training data, the models were evaluated with the test data.

Penalized models: Lasso, Ridge, Elastic Net Logit models

As for the penalized models included in the machine learning techniques, the issue is the choice of their hyperparameter values. For this purpose, the dataset is divided into three subsamples: the training group, the validation group and the test group. The training dataset covers the period between July 1995 and November 2009 (173 observations), the validation dataset covers the period between December 2009 and November 2014 (60 observations), and the test dataset covers observations from December 2014 to November 2019 (60 months). For a valid assessment and accurate evaluation across all models, it is crucial that the same test set is consistently used. This ensures that the comparability and reliability of results are maintained throughout the analysis. The subsequent algorithm outlines the procedure for identifying the optimal hyperparameters for penalized models.

Hyperparameter tuning with penalized models for binary classification - without cross-validation

Initialize an array to store evaluation metrics for each candidate regularization parameter.

In our study, we use the following values of λ: $\lambda \in (10^{-5}, 10^{-1}$, with step : $10^{-4})$

For each candidate regularization hyperparameter λ, the following steps are adopted:
1. Fit a Lasso/Ridge/Elastic Net regression model on the training data using a given λ.

https://doi.org/10.26219/heal.aueb.5657

2. Predict probabilities on the validation data using the trained model.

3. Convert probabilities to binary predictions using a threshold (e.g. the value 0.5).

4. Calculate evaluation metrics such as the misclassification error.

Select Optimal Parameter: Identify the λ value that corresponds to the lowest misclassification error as the optimal hyperparameter.

This algorithm outlines the process of hyperparameter tuning using Lasso/Ridge/Elastic Net regression models for binary classification. It iterates through different candidate regularization parameters, trains models, and evaluates their performance on validation data. The best regularization parameter is chosen based on the misclassification error. This approach helps in selecting the model that provides the best trade-off between bias and variance, leading to improved generalization on unseen data.

After identifying the best value of lambda λ, we train a final penalized model using the optimal lambda in the train dataset, so the estimation of the coefficients be done. After that we evaluate the performance of the final model using appropriate metrics (accuracy).

Classification Trees

As for the penalized models, classification has many parameters that must be chosen, for this reason the dataset is divided into 3 subsamples in the same aspect as in penalized models. The algorithm used is presented below:

Define Hyperparameter Values:

These include the complexity parameter (CP), the minimum number of samples required to split an internal node (MinSplit), the minimum number of samples required for a terminal node (MinBucket), and the maximum depth of the tree (MaxDepth). In our study, we investigate the following possible values for these hyperparameters:

- $CP = \{0.01, 0.02, 0.03, \dots, 0.49, 0.50\}$
- MinSplit $= \{2, 5, 10\}$
- MinBucket $= \{1, 5, 10\}$
- MaxDepth $= \{2, 5, 10, 20, 30\}$

https://doi.org/10.26219/heal.aueb.5657

Hyperparameter Tuning and Model Training:

Iterate through the defined hyperparameter combinations, considering the above values of CP, MinSplit, MinBucket, and MaxDepth. For each combination:

1.Train a decision tree model on the training data using the current hyperparameters.

2.Evaluate whether the root node has been split into multiple nodes.

3.Model Validation and Evaluation: If the root node has been split, proceed with the following steps:

a. Make predictions on a validation set using the trained decision tree model.
b. Calculate the accuracy of the model's predictions by comparing them to the true labels in the validation set.
c. Compare and store best hyperparameters: Compare the accuracy achieved with the current hyperparameters to the best accuracy achieved so far. If the current accuracy is higher, update the variables storing the best hyperparameters (CP, MinSplit, MinBucket, MaxDepth) to the current hyperparameter values.
d. After evaluating all hyperparameter combinations, display the best hyperparameters that resulted in the highest accuracy. Also, present the corresponding accuracy achieved by the model with the best hyperparameters.

Random Forest

Random forest has two hyperparameters that should to be tuned. The steps executed in this study are as follows:

a. Define a range of hyperparameter values for ntree (number of trees) and mtry (number of features to consider at each split).

In our study we adopt the following hyperparameter values:

- ntree_values $= \{50, 100, 150, 300\}$
- mtry_values $= \{2, 3, 4, 5, \dots, 45\}$

b. For each ntree value and for each mtry value in the defined range:

- Train a Random Forest model on the training dataset with the current ntree and mtry values.
- Make predictions on the validation set using the trained model.
- Calculate the accuracy by comparing the predicted labels with the actual validation labels.

https://doi.org/10.26219/heal.aueb.5657

- Train a final Random Forest model on the training dataset using the best hyperparameters values for ntree and mtry.

4.3 Model comparison and evaluation of each model

Findings

➢ Logistic regression:

Logistic regression yielded an accuracy of 53.33% with a sensitivity of 56.25% and a specificity of 52.27%. Despite the moderate accuracy and sensitivity, the precision was remarkably low at 30%. The balanced accuracy value of 54.26% indicates that the model is able to maintain some balance between class predictions. The model achieved an AUC value of 63.67%, suggesting some degree of discriminatory power.

➢ Probit Regression:

The Probit regression model had an accuracy of 55%, accompanied by a sensitivity of 58.82% and a specificity of 53.49%. Notably, the precision was relatively higher at 33.3%, representing a more balanced ratio between true and false positives from than Logit. The balanced accuracy of 56.16% supports the potential of the system for balanced classification. The AUC of 64.44% underscores its ability to discriminate between classes.

➢ Logit Lasso:

The Logit Lasso model achieved 55% accuracy and had a sensitivity of 57.89% and a specificity of 53.66%. Its precision of 36.67% reflects an improvement in controlling false positives compared with the previous models. The balanced accuracy of 55.77% is consistent with its ability to maintain a balanced predictive distribution. The AUC value of 59.44% indicates satisfactory discrimination ability.

➢ Logit Ridge:

Logit Ridge showed an accuracy of 51.67%, sensitivity of 52%, specificity of 51.43%, and precision of 43.33%. While its accuracy and sensitivity are comparable to the other models, its precision is relatively higher. The balanced accuracy of 51.72% and AUC of 49.33% suggest performance that may not be significantly above chance.

https://doi.org/10.26219/heal.aueb.5657

- Elastic Net:

Elastic Net showed an accuracy of 51.67%, sensitivity of 51.72%, specificity of 51.61%, and precision of 50%. The balanced accuracy of 51.67% and AUC of 56.56% indicate a balanced predictive capacity with a remarkable AUC value indicating a good ability to discriminate between classes.

- Decision tree:

The decision tree model achieved an accuracy of 61.67% as well as a sensitivity of 65.22% and a specificity of 59.46%. However, the precision of 50% indicates that there is room for improvement in reducing false positives. The balanced precision of 62.34% and AUC of 55% highlight the robustness of the model in capturing both true positives and true negatives.

- Random Forest:

Random Forest showed an accuracy of 61.67% and a remarkable sensitivity of 73.33%, although its specificity was slightly lower at 57.78%. However, the precision of the model was relatively low at 36.67%, suggesting the risk of false positives. The balanced accuracy value of 65.56% suggests a promising ability for balanced class prediction. The AUC value of 55% underscores the discriminatory ability of the model.

- Linear Discriminant Analysis (LDA):

LDA showed an accuracy of 53.3%, sensitivity of 57.14%, and specificity of 52.17%. Precision was significantly lower at 26.67%, indicating a higher rate of false positives. The balanced precision and kappa of 54.65% and 46.22%, respectively, indicate an overall performance that may not exceed chance.

- Quadratic Discriminant Analysis (QDA):

QDA yielded an accuracy of 53.3%, a sensitivity of 55%, and a specificity of 52.5%. Its precision of 36.67% improved compared to LDA, but its balanced accuracy and kappa of 53.75% and 46.22% remained the same, indicating alignment with the performance of LDA.

Decision trees and random forest models have best sensitivity and the best accuracy from all the other methods.

https://doi.org/10.26219/heal.aueb.5657

Identifying positive instances is of paramount importance and minimizing false negatives is a priority, then the Random Forest model stands out with the highest sensitivity at 73.33%. This means Random Forest is better at correctly identifying actual positive cases.

Balanced Sensitivity and Specificity: For a balance between correctly identifying both positive and negative instances, the Decision Tree model offers good sensitivity (65.22%) and specificity (59.46%), resulting in a balanced accuracy of 62.34%.

Discriminatory Ability: If the ability to discriminate between positive and negative instances is a significant concern, the Probit Regression model has an AUC of 64.44%, which is among the higher values.

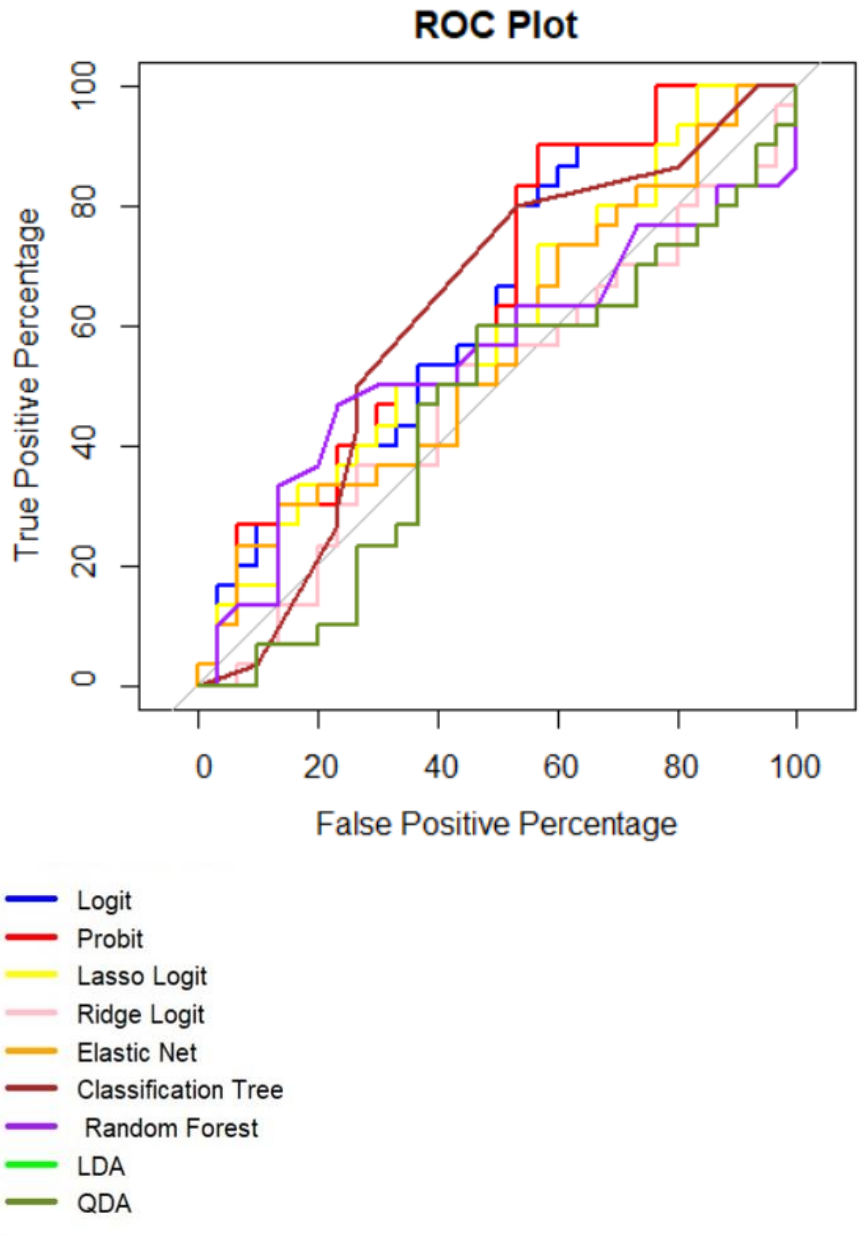
Table 2

PERFORMANCE EVALUATION MEASURES								
Model / Metric	**Accuracy**	**MCE**	**Sensitivity**	**Specificity**	**Precision**	**Kappa**	**Balanced Accuracy**	**AUC**
Logit	0.5333	0.4667	0.5625	0.5227	0.3000	0.3435	0.5426	63.67%
Probit	0.5500	0.4500	0.5882	0.5349	0.3330	0.2629	0.5616	64.44%
Logit Lasso	0.5500	0.4500	0.5789	0.5366	0.3667	0.3455	0.5577	59.44%
Logit Ridge	0.5167	0.4833	0.5200	0.5143	0.4333	0.1998	0.5172	49.33%
Elastic Net	0.5167	0.4833	0.5172	0.5161	0.5000	0.2479	0.5167	56.56%
Decision Tree	0.6167	0.3833	0.6522	0.5946	0.5000	0.3867	0.6234	62.06%
Random Forest	0.6167	0.3830	0.7333	0.5778	0.3667	0.1875	0.6556	55.00%
LDA	0.5330	0.4667	0.5714	0.5217	0.2667	0.2500	0.5465	46.22%
QDA	0.5330	0.4667	0.5500	0.5250	0.3667	0.2638	0.5375	46.22%

The table represents the performance evaluation of the 1 month ahead of forecasts done by Logit/Probit models and some machine learning techniques for the out of sample period December 2014 to November 2019.

https://doi.org/10.26219/heal.aueb.5657

Figure 2. Receiver Operating Characteristic (ROC) curves for the out-of-sample period December 31, 2014 to November 30, 2019 for all models.



https://doi.org/10.26219/heal.aueb.5657

4.4 Variable Importance and Variable Selection

In the aftermath of model estimation, it is essential to identify which variables have predictive ability and can lead to improved accuracy and reliability of forecasts, thus aiding in better decision-making, to simpler and more interpretable models, reducing overfitting and enhancing model generalization and to a deeper understanding of the relationships within the data and the factors affecting the outcome.

In both Logit and Probit models, the p-value associated with each predictor variable is found so that to investigate the statistical significance of that variable's contribution to the model. Also Feature Selection Techniques can be used to identify the most relevant predictors. Common methods include Forward Selection, Backward Elimination and Stepwise Selection.

In penalized techniques like Lasso that penalizes the magnitude of coefficients, encouraging some coefficients to be exactly zero, effectively selects a subset of predictors.

In a classification tree, the most important variables are those that lead to the best separation of the target classes. The tree algorithm assesses the impact of each variable on classifying observations correctly by measuring the decrease in impurity (e.g., Gini impurity or cross-entropy) achieved when a variable is used to split the data.

Random Forest's variable selection process occurs naturally as part of the ensemble learning process. It assesses the importance of features by evaluating their impact on impurity reduction or prediction accuracy across multiple decision trees. This approach helps identify and emphasize the most relevant features for making accurate predictions while reducing the risk of overfitting.

https://doi.org/10.26219/heal.aueb.5657

Table 2

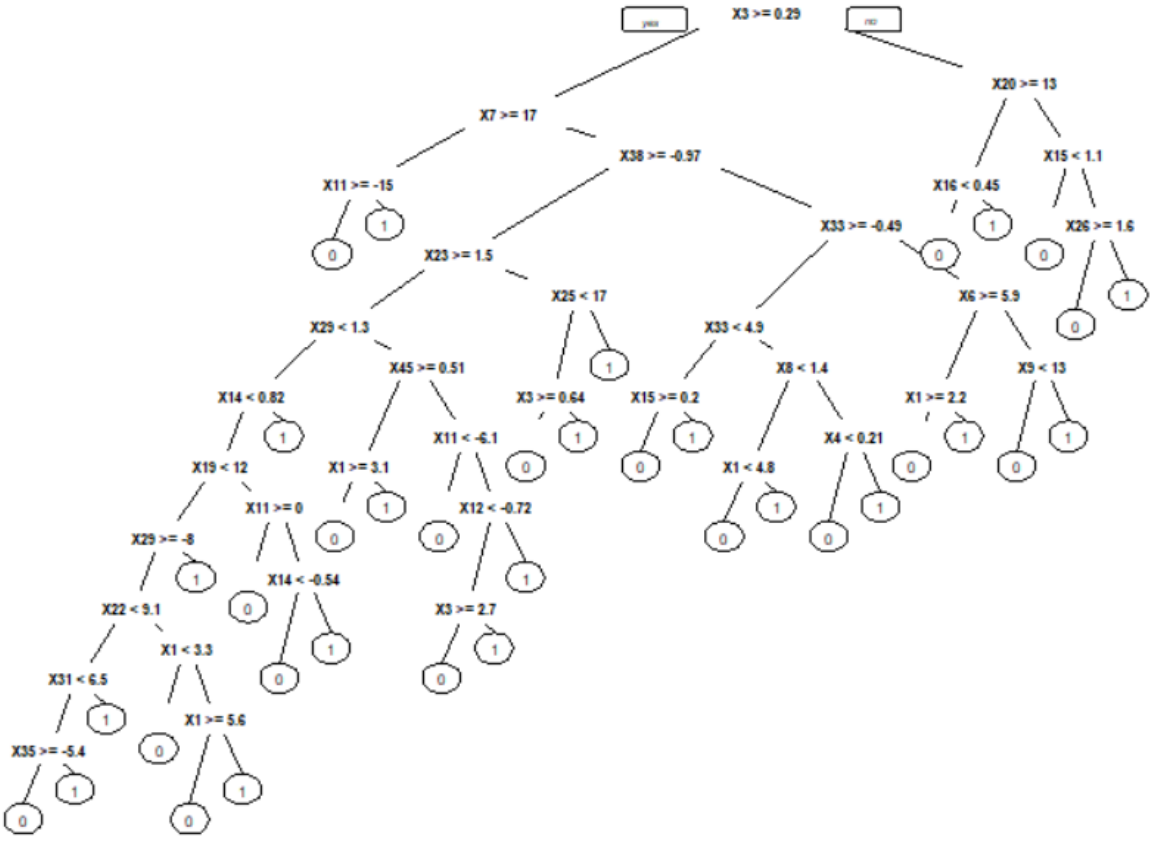
THE VARIABLES WITH PREDICTIVE ABILITY

Logit/Probit models	Penalized Lasso Logit Model
X10(Industrial_production_yoy) X14(Chicago_Fed_National_Activity_Index) X22(ISM_Manufacturing_Index_New_Orders_D X23 (Real_Money_Supply_M2) X24(S&P 500 DY) X25(S&P 500 PE) X26(S&P 500 EBYD) X34(BAB) X43(S&P 500 DYDVG)	X24: S&P 500 DY X23: Real Money Supply M2 X15: Core PPI Inflation yoy X17: S&P 500 ret X26: S&P 500 EBYD X14: Chicago Fed National Activity Index X12: Chicago Fed National Financial Conditions Index X5: Credit spread X43: S&P 500 DYDVG X10: Industrial production yoy
Decision Tree	Random Forest
X11-SM_Manufacturing_Index_New_Orders_D X14 - Chicago_Fed_National_Activity_Index X15 - Core_PPI_Inflation_yoy X16 - USD_ X19 - S&P_500_6M X20 - S&P_500_12M X23 - Real_Money_Supply_M2 X25 - S&P 500 PE X29 - RMW X3 - Term_spread X33 - RVAR X38 - MOMCM X45 - Kansas City Financial Stress Index X7 - ECU1 X8 - MACVOL	X3 - Term_Spread X17 - S&P_500_Ret X7 - ECU1 X28 - HML X23 - Real_Money_Supply_M2 X6 - Gold_Price X38 - MOMCM X4 - TED_Spread X29 - RMW X37 - TSMOM X2 - V3_Month_Treasury_Yield X26 - S&P 500 EBYD X33 - RVAR X32 - STREV X24 - S&P 500 DY X22 - CONSSENT_D

The table reports the most relevant predictive variables obtained by Logit/Probit model, Lasso Logit model, Decision Tree and Random Forest.

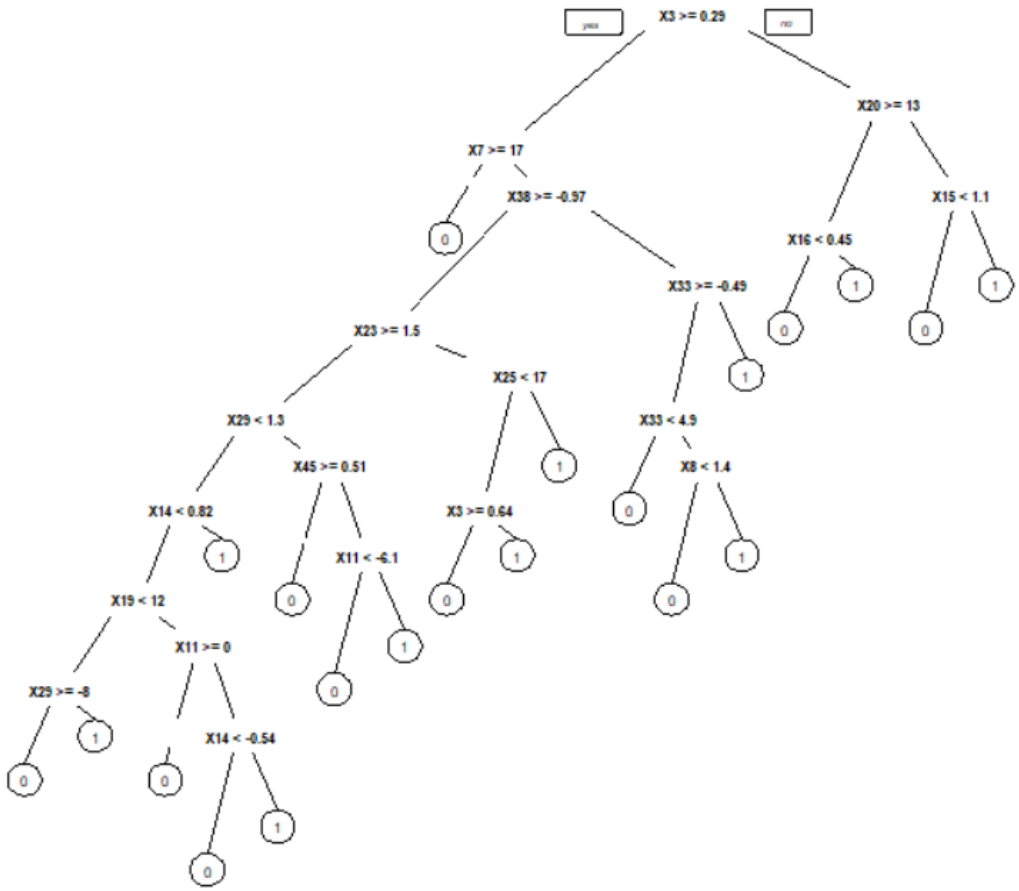
https://doi.org/10.26219/heal.aueb.5657

Figure 3. Decision tree structure



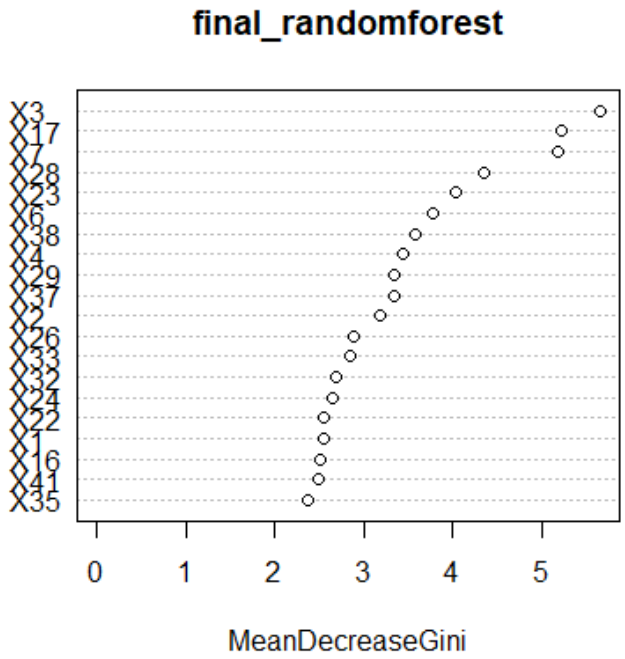
https://doi.org/10.26219/heal.aueb.5657

Figure 4.The Decision tree with optimal hyperparameters



https://doi.org/10.26219/heal.aueb.5657

Figure 4. Top 20 selected variables in the Random Forest with highest importance in descending order by a mean decrease in Gini. The top variables contribute more to the model than the bottom ones and also have high predictive power in classifying recession or ascension



Discussion about the findings of the selected variables

Among the plethora of variables examined, several common factors have consistently emerged as crucial contributors to the predictability of book value growth across various models. These variables exhibit robust predictive power and offer key insights into the intricate dynamics of the financial landscape. Some of the noteworthy common important variables include:

- S&P 500 Dividend Yield (X24): The dividend yield of the S&P 500 index is a fundamental indicator that reflects the income generated by owning the stocks within the index. A higher dividend yield can indicate a potential undervaluation of stocks or a shift in market sentiment.
- Real Money Supply M2 (X23): This variable measures the amount of money circulating in the economy, capturing changes in the money supply that can influence economic activity and asset valuation.

https://doi.org/10.26219/heal.aueb.5657

- Chicago Fed National Activity Index (X14): This index provides a comprehensive gauge of overall economic activity, reflecting trends in production, consumption, employment, and other economic indicators.
- S&P 500 Price-to-Earnings Ratio (X25) and Earnings Yield (X26): These metrics offer insights into the valuation of the index constituents. A higher P/E ratio might suggest overvaluation, while a higher earnings yield could point to undervaluation.
- ISM Manufacturing Index New Orders (X22): This index tracks new orders in the manufacturing sector, indicating shifts in business activity and future production levels.
- Industrial Production Year-on-Year (X10): Reflecting changes in industrial output, this variable can indicate economic expansion or contraction.
- Term Spread (X3): The difference between short-term and long-term interest rates can provide insights into market expectations for economic growth and inflation.
- USD Exchange Rate (X16): Fluctuations in the value of the US dollar can impact international trade and financial markets, influencing asset prices.
- Core Producer Price Inflation (X15): This indicator measures changes in the prices of goods and services, offering insights into inflationary pressures within the economy.
- Economic Uncertainty (ECU1 and X7): Indicators capturing economic uncertainty can reflect shifts in investor sentiment and risk perceptions.

These common important variables are representative of a diverse set of economic, financial, and market-specific factors that collectively contribute to the directional predictability of book value growth. By focusing on these variables, researchers and practitioners can gain deeper insights into the intricate relationships shaping the behavior of stock market indices.

https://doi.org/10.26219/heal.aueb.5657

CHAPTER 5

CONCLUSION

In this study, an analysis was conducted to explore the predictability of book value growth direction in the context of the S&P 500 index. The prediction was made one month ahead using logit/probit models and some of statistical and machine learning techniques. The features include economic, financial and market variables. The modeling and evaluation of the models are constructed using in sample, validation and out of sample sets.

The main findings can be summarized as follows based on their accuracy. Firstly, Random Forest and Decision Tree have higher Book Value Growth direction prediction accuracy than the other models, reaching the same accuracy 61.67%. The subsequent models are the probit and lasso logit model with accuracy 55%, while the models with the least accuracy, at 51.67%, were Ridge logit and Elastic Net. Secondly, the Random Forest model demonstrated the highest sensitivity at 73.73%, implying a strong ability to predict positive book value growth. In contrast, the Decision Tree model displayed the greatest specificity, indicating its proficiency in predicting negative book value growth. Overall, all employed models yielded accuracies surpassing the 50% mark. This implies that the directional forecasts are not random, showcasing the potential for refinement in capturing Book Value Growth movements.

Regarding the question which variables have the best predictive ability, Real Money Supply M2 and Chicago Fed National Activity Index are present in logit/probit models, penalized models, decision tree and random forest. Movements in the CFNAI can indicate whether the economy is expanding, contracting, or remaining stable. Including this variable in models allows the model to capture the impact of various economic activities on the target variable, contributing to predictive accuracy. Real money supply M2 is considered important in economic analysis and modeling because changes in the money supply can have a significant impact on economic activity, interest rates, and inflation. When the money supply increases, it can stimulate economic growth, but it might also lead to higher inflation. On the other hand, a decrease in the money supply can have the opposite effects.

https://doi.org/10.26219/heal.aueb.5657

BIBLIOGRAPHY

Juha Junttila, Juho Pesonen, Juhani Raatikainen. 2018. "Commodity market based hedging against stock market risk in times of financial crisis: The case of crude oil and gold." Journal of International Financial Markets, Institutions and Money, 2018, vol. 56, issue C, 255-280

Tobias J. Moskowitz, Yao Hua Ooi, Lasse Heje Pedersen. 2012 "Time series momentum." Journal of Financial Economics,Volume 104, Issue 2, Pages 228-250

Syed Abul Basher, Perry Sadorsky. 2022. "Forecasting Bitcoin price direction with random forests: How important are interest rates, inflation, and market volatility?" In "Machine Learning with Applications." Publisher: Elsevier.

Christoph Bergmeir, Rob J. Hyndman, Bonsoo Koo. 2018. "A note on the validity of cross-validation for evaluating autoregressive time series prediction." Computational Statistics & Data Analysis , Volume 120 , issue C, pages 70–83

Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. "An Introduction to Statistical Learning: with Applications in R." Springer Texts in Statistics

Henri Nyberg.2011. "Forecasting the direction of the US stock market with dynamic binary probit models." International Journal of Forecasting ,Volume 27, Issue 2, , Pages 561-578

John B. Guerard Jr., Harry Markowitz, GanLin Xua.2015. "Earnings forecasting in a global stock selection model and efficient portfolio construction and management." International Journal of Forecasting, Volume 31, Issue 2, Pages 550-560

J.C.K. Ash, D.J. Smyth, S.M. Heravi.1998. "Are OECD forecasts rational and useful?: a directional analysis." International Journal of Forecasting ,Volume 14, Issue 3, Pages 381-391

Oliver Blaskowitza, Helmut Herwartzb.2011. "On economic evaluation of directional forecasts." International Journal of Forecasting ,Volume 27, Issue 4, Pages 1058-1065.

https://doi.org/10.26219/heal.aueb.5657

Spyridon D. Vrontos, John Galakis, Ioannis D. Vrontos. (2021). "Modeling and predicting U.S. recessions using machine learning techniques." International Journal of Forecasting ,Volume 37, Issue 2, Pages 647-671

Loh, W.-Y. and Zhou, P. (2021), Variable importance scores. Journal of Data Science, vol. 19, 4, 569-592

M Umer Ghani, M Awais, Muhammad Muzammul.2019. "Stock Market Prediction Using Machine Learning (ML) Algorithms."

Shihao Gu, Bryan Kelly, Dacheng Xiu . 2020. "Empirical Asset Pricing via Machine Learning."

Pushpendu Ghosh, Ariel Neufeld, Jajati Keshari Saho. 2021. "Forecasting directional movements of stock prices for intraday trading using LSTM and random forests", Finance Research Letters