On the Directional Predictability of Equity Premium Using Machine Learning Techniques

Jonathan Iworiso Spyridon Vrontos [1]

Department of Mathematical Sciences, University of Essex, United Kingdom

Abstract

This paper applies a plethora of machine learning techniques to forecast the direction of the U.S. equity premium. Our techniques include benchmark binary probit models, classification and regression trees (CART), along with penalized binary probit models. Our empirical analysis reveals that the sophisticated machine learning techniques significantly outperformed the benchmark binary probit forecasting models, both statistically and economically. Overall, the discriminant analysis classifiers are ranked first among all the models tested. Specifically, the high dimensional discriminant analysis (HDDA) classifier ranks first in terms of statistical performance, while the quadratic discriminant analysis (QDA) classifier ranks first in economic performance. The penalized likelihood binary probit models (Least Absolute Shrinkage and Selection Operator, Ridge, Elastic Net) also outperformed the benchmark binary probit models, providing significant alternatives to portfolio managers.

Key words: Directional Predictability, Recursive Window, Forecasting, Binary Probit, CART, Penalized Binary Probit

[1] Corresponding author. Email: svrontos@essex.ac.uk

1 Introduction

Stock market participants aim at maximising returns on portfolio investments at minimal risk. Consequently, forecasting stock market returns has received considerable attention in recent years. The majority of papers have focused on the forecast accuracy of competing models and examined if there is evidence of predictability, which can lead to economic gains. However, devising successful trading strategies is contingent on the directional accuracy of the underlying models. The literature on directional predictability is sparse, and the empirical findings offer limited support. For example, the findings in (Chevapatrakul, 2013; Christoffersen and Diebold, 2006; Nyberg and Pönkä, 2016) provide weak evidence of directional stock market predictability. Although the predictive power of the models employed so far are shown to be weak in statistical terms, they seem to provide economic value. Thus, the emphatic challenge lies in the development of a suitable directional predictive model involving the relevant financial and economic variables.

The application of some benchmark econometric models used in previous findings are shown to be weak in terms of predictive performance. The introduction of out-of-sample estimation and forecasting techniques used by Nyberg (2011), Pönkä (2016) provide statistically significant evidence of the directional predictability of stock market returns, but the predictive power of the models are shown to be relatively weak, and hence, there is a need to introduce sophisticated machine learning techniques, as proposed in this paper, to improve the predictive task of the models.

This paper focuses on the application of sophisticated machine learning techniques on binary probit and classification models to forecast the direction of the U.S. excess stock market returns. The machine learning techniques employed include classification and regression trees (CART), such as Bagging, Boosting and Discriminant Analysis classifiers, Bayesian classifiers, Neural Networks and regularization techniques, such as Ridge, Least Absolute Shrinkage and Selection Operator (LASSO), and Elastic Net. To compare our findings with the previous literature, we also include four variants of the benchmark binary probit models, namely, the static, stepwise static, dynamic and stepwise dynamic models. The application of CART forecasting models aims to explore all covariates as ensembles to learn the data, train the classification model, recognize patterns, classify instances and to forecast future binary outcomes. With respect to penalised binary probit models, we should note that the presence of shrinkage penalty vector norms could result to a bias in coefficient estimates, reduction in the forecast errors and improvement in predictive performance via the so-called bias-variance trade off. Thus, the proposal of CART and penalized predictive models in this paper aims at yielding superior statistical predictive performance and economic significance compared to the benchmark econometric models typically employed in the literature to date.

The remaining structure of the paper is laid out as follows: Section 2 discusses the relevant literature; Section 3 describes the research methodology; Section 4 presents the data and the empirical findings; and Section 5 concludes the paper.

2 Literature Review

A notable quest in modern financial econometric literature is the application of suitable techniques to predict the sign of stock market returns. A review of relevant empirical literature has revealed that the use of econometric models for the directional predictability of excess stock returns are known to produce weak predictive power, poor statistical goodness of fit and low predictive accuracies, among others; see (Pesaran and Timmermann, 1995; Nyberg, 2011; Leung et al., 2000; Chevapatrakul, 2013; Leitch and Tanner, 1991; Pönkä, 2016), even though the empirical results seems to provide economic significance.

The previous findings on directional predictability by Anatolyev and Gospodinov (2010), and Hong and Chung (2003) have employed a logistic regression model to predict the sign of U.S. stock market returns using relevant financial variables as the key predictors, and their results provide evidence of predictability, but the overall predictive power is relatively weak as compared to a rule of thumb. In an attempt to determine market timing and asset allocation decisions between stocks and risk-free assets, some researchers considered the role of conditional mean and volatility while predicting the sign of asset returns. Christoffersen and Diebold (2006) have opined that the direction of asset returns is predictable, as volatility dependence produces sign dependence, so long as expected returns are nonzero. This notion seems to be true, as other existing papers have also provided significant statistical evidence of the sign predictability of the U.S. stock market returns and economic recession status by application of static, dynamic, autodynamic and error correction models, both in-sample and out-of-sample (Nyberg, 2011; Kauppi and Saikkonen, 2008; Nyberg and Pönkä, 2016; Nyberg, 2013).

The static and dynamic probit models proposed by Nyberg (2011) to predict the direction of monthly U.S. excess stock returns recursively appears to have outperformed the autoregressive moving average with exogenous inputs models (ARMAX), vector autoregressive-generalized autoregressive conditional heteroskedasticity models (VAR-GARCH), etc. used by previous researchers. The idea was based on the approach used by Kauppi and Saikkonen (2008), Estrella and Mishkin (1998) to obtain U.S. economic recession forecasts using variables such as the U.S. term spread and lagged stock returns, among others.

However, according to the Nyberg (2011) paper, the Estrella's statistical goodness of fit values in the various probit models are very low in all cases. The positive values of the Sharpe Ratios signified that investors are likely to have positive returns on portfolio investments. The percentage of correct matches as a statistical performance evaluation measure in the existing papers are relatively low, hence the need to employ more advanced sophisticated models that can yield a better degree of accuracy with the smallest prediction error.

The underlying challenges associated with the use of financial and economic variables to predict stock market returns has prompted researchers to introduce sophisticated statistical or machine learning algorithms to improve the predictive task and the overall performance evaluation of the resulting models under consideration. It is noticeable from the empirical literature that statistical learning techniques, which include Random Forest, Linear Discriminant Analysis (LDA), k-Nearest Neighbour, Tree-based Classification, Recursive Partitioning, Bagging and Boosting, Logistic Regression, Support Vector Machine (SVM), Ridge Regression, Least Absolute Shrinkage and Selection Operator (LASSO), Least Angle Regression and Elastic Nets, are useful for the analysis of financial econometric time series (Roy et al., 2015; Sermpinis et al., 2017; Li and Chen, 2014; Inoue and Kilian, 2008; Zhou et al., 2015; Hsu et al., 2008; Park and Sakaori, 2013; Chen, 2016; Stock and Watson, 2012; Lin and McClean, 2001; Kim and Swanson, 2014; Hajek et al., 2014; Shen et al., 2014; Pahwa et al., 2017; Swanson and White, 1997). Khaidem et al. (2016) used the Random Forest method to predict the direction of stock market prices. The algorithm appears to be robust in predicting the future direction of the stock market movement, thus minimizing the risk of investment in the stock market with good predictive accuracy.

The ridge regression introduced by Hoerl and Kennard (1970), and the least absolute shrinkage and selection operator (LASSO) introduced by Tibshirani (1996) are found to be useful statistical or machine learning techniques for econometric models. The ridge regression reduces multicollinearity and minimizes the model prediction error but does not perform feature selection; the LASSO shrinks the model coefficients towards zero and performs feature selection and model interpretability. The aim is to introduce bias in the model coefficient estimates and minimize the prediction error.

The empirical analysis in Inoue and Kilian (2008) revealed that bagging has large reductions in prediction mean square errors (PMSEs) in inflation forecasting. Kim and Swanson (2014) suggest that the model averaging does not dominate other well designed prediction model specification methods, and that the use of hybrid combination factor and shrinkage methods produced the best predictions as compared to principal components, bagging, boosting, least angle regression, among others. On the other hand, the empirical results from Zhou et al. (2015) showed no statistically significant difference between the best classification performance of the models with yearly feature selection guided by data mining techniques and the one involving domain knowledge; hence, their predictive accuracies seems to be the same.

The use of the LASSO linear regression model for stock market forecasting by Roy et al. (2015) using monthly data revealed that the LASSO method yield sparse solutions and performs extremely well when the number of features is less than the number of observations, and that the LASSO linear regression model outperforms the ridge linear regression model. Modelling the market implied ratings using LASSO variable selection techniques by Sermpinis et al. (2017) and forecasting macroeconomic time series using LASSO-based approaches and their forecast combinations with dynamic factor models by Li and Chen (2014) all reflect statistical evidence of the superior predictive power of LASSO.

The outperformance of the aforementioned statistical learning algorithmic techniques over the benchmark econometric and statistical modelling techniques has prompted modern researchers to proceed into a more advanced concept, i.e., the deep learning techniques based on artificial intelligence, which encompasses support vector machines (SVM) and neural networks (NN). However, the contrasting arguments of various scholars on the predictive performance by SVM and NN as compared to the previous literature has placed this notion pending for further statistical investigation. The application of artificial intelligence neural networks in forecasting financial markets and stock prices by Shahpazov et al. (2014) demonstrated the outperformance of the NN over previous techniques used in the existing literature. Again, the findings in de Oliveira et al. (2013) also revealed that the application of artificial neural networks yielded the minimum mean square prediction error (MSE) and excellent correct direction rates. Controversially, the analytical results by Moreno and Olmeda (2007) show that the ANN do not provide evidence of superior performance over the conventional linear models. The findings of Ding et al. (2013), applying the concept for daily data and market sentiment, show the outperformance of SVM over NN and logistic regression. The SVM seems to be the most accurate machine learning model for predicting stock market movement, but the statistical tests do not provide significant statistical evidence of better performance over NN and logistic regression. Patel et al. (2015) confirmed the outperformance of Random Forest over ANN, SVM and the genetic algorithm (GA) for input data with continuous values. Ballings et al. (2015) also presented random forest as the top machine learning algorithm over others and recommended the inclusion of ensembles in algorithmic sets when predicting the direction of stock market prices. The findings in Zheng (2006) demonstrated the superiority of boosting and bagging of NN over SVM and logistic regression when forecasting the daily directional movements of stocks.

It is obvious, based on the reviewed existing empirical literature, that machine learning techniques played an enormous role in financial econometric time series. Thus, the application of the proposed sophisticated machine learning recursive out-of-sample forecasting models for the directional predictability of the U.S. stock market returns in this paper aimed to yield significant results and outperform the benchmark econometric models and aimed to enrich the empirical literature for further relevant scholarly research work.

3 Research Methodology

3.1 Equity Premium Direction Modelling as a Binary Time Series

Let R_t be the monthly U.S. excess stock market return over the risk-free interest rate denoted by Rf_t, and let I_t^s denote the binary-valued dependent variable. The sign of the monthly equity premium is modelled as the return sign binary indicator, as follows:

$$I_t^s = \begin{cases} 1, \text{ if } R_t > 0 \text{ i.e., there is positive excess stock market return} \\ 0, \text{ if } R_t \leq 0 \text{ i.e., there is negative or zero excess stock market return.} \end{cases}$$

R_t is calculated as follows

$$R_t = \ln\Big(\frac{P_t}{P_{t-1}}\Big) - Rf_{t-1}$$

where P_t is the price of the stock index at period t and Rf_{t-1} is the risk-free interest rate at period $t-1$. The distribution of the return sign binary indicator I_t^s conditional on $\Re_{t-1}$ follows Bernoulli with probability p_t, as follows:

$$I_t^s|\Re_{t-1} \sim Bernoulli(p_t),$$

where $\Re_{t-1}$ is the information set of the covariates.

3.2 The Static and Dynamic Binary Probit Models

Christoffersen and Diebold (2006) showed that if R_t is distributed as follows:

$$R_t|\Re_{t-1} \sim N(\mu, \sigma^2_{t|t-1})$$

and displays no conditional mean dependence and conditional variance dependence, then there exists a link between the volatility dynamics and the sign dynamics. The conditional probability of a positive excess stock market return $Prob_{t-1}(R_t > 0)$ is as follows:

$$Prob_{t-1}(R_t > 0) = 1 - \Gamma\Big(\frac{-\mu}{\sigma_{t|t-1}}\Big) = \Gamma\Big(\frac{\mu}{\sigma_{t|t-1}}\Big)$$

where $\Gamma(.)$ is the $N(0,1)$ cumulative distribution function, and the forecast horizon used is equal to 1. The conditional probability of a positive equity premium sign employing the binary indicator I_t^s is as follows:

$$E_{t-1}(I_t^s) = Prob_{t-1}(I_t^s = 1) = Prob_{t-1}(R_t > 0) = \Gamma(\Psi_t).$$

In the case of the static binary probit model, we have the following:

$$\Psi_{t+1}(\beta) = \alpha + Z_t'\beta \tag{1}$$

where α is the model intercept; Z_t is the vector of the predictors and β is the vector of the unknown coefficients (Nyberg, 2011; Nyberg and Pönkä, 2016). In the case of the dynamic binary probit model, we have the following:

$$\Psi_{t+1}(\beta) = \alpha + \sum_{i=1}^{k} \eta_i I^s_{t+1-i} + Z'_t\beta \tag{2}$$

where η denotes the unknown coefficients of the lagged equity premium sign indicator and k is the lag order of the equity premium sign indicator (Kauppi and Saikkonen, 2008).
The parameters of the binary probit models defined above are estimated by employing the maximum likelihood method, where the maximum likelihood estimator of β is as follows:

$$\hat{\beta}_{ML} = argmax_{\beta} \left\{ \sum_{(I^s_{t+1}=1)} \Gamma\Big(\Psi_{t+1}(\beta)\Big) + \sum_{(I^s_{t+1}=0)} \Big(1 - \Gamma\big(\Psi_{t+1}(\beta)\big)\Big) \right\}. \tag{3}$$

For more details, one can see Estrella and Mishkin (1998) and Pesaran (2015).

3.3 Penalized Likelihood Binary Probit Models

In this section, we will examine penalized likelihood binary probit models employing the relevant Ridge, LASSO and Elastic Net structures. The inclusion of a penalty vector norm in the log-likelihood function of the ordinary binary probit model results in the penalized binary probit model. It is worth noting that in the penalized likelihood binary probit models, the coefficients estimates are shrunk towards zero. The regularised coefficients have significantly reduced variances, resulting in smaller forecasting errors.

3.3.1 The Ridge Probit Model

The ridge probit model aims to reduce multicollinearity and minimize the prediction error of the model and is based on the ridge regression introduced by Hoerl and Kennard (1970). Given the log-likelihood function of the ordinary probit model (3), the ridge log-likelihood probit function introduces a shrinkage penalty employing the ℓ_2-norm of β, $\|\beta\|_2 = \sqrt{\sum_{j=1}^{p} \beta_j^2}$ and the ridge tuning parameter λ, $\lambda > 0$, which controls the amount of regularization. Thus, the maximum likelihood estimator of the ridge probit model is given by the following:

$$\hat{\beta}^{\lambda}_{RMLE} = argmax_{\beta} \left\{ \sum_{(I^s_{t+h}=1)} \Gamma\Big(\Psi_{t+h}(\beta)\Big) + \sum_{(I^s_{t+h}=0)} \Big(1 - \Gamma\big(\Psi_{t+h}(\beta)\big)\Big) - \lambda \sum_{j=1}^{p} \beta_j^2 \right\}.$$

3.3.2 The LASSO Probit Model

The Least Absolute Shrinkage and Selection Operator (LASSO) introduced by Tibshirani (1996) as a shrinkage and selection technique for linear regression models is extended to probit models. The proposed LASSO probit model aims to shrink the probit model coefficients toward zero, resulting in increased model interpretability and identification of the covariates most strongly associated with the equity premium direction. To obtain the LASSO probit coefficients $\hat{\beta}^{\lambda}_{LMLE}$, the maximization of the log-likelihood function of model (3) will include a shrinkage penalty on the ℓ_1-norm of β. The vector $\hat{\beta}^{\lambda}_{LMLE}$ is obtained by

$$\hat{\beta}^{\lambda}_{LMLE} = argmax_{\beta} \left\{ \sum_{(I^s_{t+h}=1)} \Gamma\Big(\Psi_{t+h}(\beta)\Big) + \sum_{(I^s_{t+h}=0)} \Big(1 - \Gamma(\Psi_{t+h}(\beta))\Big) - \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

where $\|\beta\|_1 = \sum_{j=1}^{p} |\beta_j|$ is the ℓ_1-vector norm of β and λ, $\lambda > 0$, is the LASSO tuning parameter, which controls the amount of shrinkage of β.

3.3.3 The Elastic Net Binary Probit Model

The elastic net (EN) is a regularized technique that linearly combines the ℓ_1 and ℓ_2 penalties of LASSO and Ridge. The elastic net probit coefficient estimates $\hat{\beta}^{\lambda}_{EMLE}$ are obtained by maximizing the log-likelihood function, which penalize the size of the model coefficients based on both the ℓ_1-vector norm and ℓ_2-vector norm of β. Thus, the parameter estimates of the elastic net probit model will be given by the following:

$$\hat{\beta}^{\lambda}_{EMLE} = argmax_{\beta} \left\{ \sum_{(I^s_{t+1}=1)} \Gamma\Big(\Psi_{t+h}(\beta)\Big) + \sum_{(I^s_{t+1}=0)} \Big(1 - \Gamma(\Psi_{t+h}(\beta))\Big) - \lambda\Big((1-\alpha) \sum_{j=1}^{p} \frac{\beta_j^2}{2} + \alpha \sum_{j=1}^{p} |\beta_j|\Big) \right\}$$

where λ and α are the EN tuning parameters (Zou and Hastie, 2005). We employ $\alpha = 0.5$.

To choose the tuning parameter λ in LASSO, Ridge and EN, we need a validation set in which the predictive value of the specific penalized binary probit model could be compared for various values of the tuning parameter, and the optimal tuning parameter should be chosen such that the error rate is minimal. We choose the best tuning parameter employing cross-validation.

3.4 Discriminant Analysis

In this section, we will discuss the discriminant analysis class of methods. This class includes linear, quadratic, regularised, heteroscedastic, sparse, and high-dimensional discriminant analysis methods. Discriminant analysis methods do not suffer from parameter instability as the binary models do when the classes are well separated.

3.4.1 Linear Discriminant Analysis

The discriminant function concept was first introduced by Fisher (1936). Linear discriminant analysis (LDA) uses Bayes' theorem to estimate output class probabilities given the input features, using the assumptions that the input data $Z = (Z_1, \cdots, Z_K)$ follow a multivariate Gaussian distribution with a class specific mean vector μ_c and a common covariance matrix $S_c = S$ for all c. If $f_c(z)$ is the class conditional density of the covariates Z, in class $Y = c$, i.e.,

$f_c(z) = Prob(Z = z|Y = c)$, and ψ_c is the prior probability of class c, then by Bayes' theorem, the class posterior probability is given by the following:

$$Prob(Y = c|Z = z) = \frac{f_c(z)\psi_c}{\sum_{c=1}^{C} f_c(z)\psi_c}, \quad for \quad c = 1,2,\cdots,C$$

and Z has a multivariate Gaussian density for each class given by the following:

$$f_c(z) = (2\pi)^{-\frac{p}{2}}|S_c|^{-\frac{1}{2}}exp\Big(-\frac{1}{2}(z-\mu_c)'S_c^{-1}(z-\mu_c)\Big).$$

The LDA classifier assigns an observation given by $Z = z$ to the class c given by the following:

$$\Psi_c^{LDA}(z) = argmax_c\left\{z'S^{-1}\mu_c - \frac{1}{2}\mu_c'S^{-1}\mu_c + log\psi_c\right\}. \tag{4}$$

For a proof of the above equation, see (James et al., 2013). The word linear in the LDA classifier stems from the fact that the discriminant function is a linear function of the input features Z.

3.4.2 Quadratic Discriminant Analysis

The quadratic discriminant analysis (QDA) classifier separates multi-class measurements by a quadratic surface. Unlike LDA, in the case of the QDA classifier, the input features in each class follow a multivariate Gaussian distribution with a class specific mean vector μ_c and a class specific covariance matrix S_c (James et al., 2013; Ou and Wang, 2009). The QDA classifier is given by the following:

$$\Psi_c^{QDA}(z) = argmax_c\Big(\Omega_c(z)\Big) = argmax_c\left\{-\frac{1}{2}log|S_c| - \frac{1}{2}(z-\mu_c)'S_c^{-1}(z-\mu_c) + log\psi_c\right\}. \tag{5}$$

The QDA classifier obtains its name from the fact that the QDA discriminant function is a quadratic function of the input features Z.

3.4.3 Regularized Discriminant Analysis

The regularized discriminant analysis (RDA) introduces regularization into the estimates of the covariance matrices and allows the shrinkage of the separate covariance matrices of QDA toward a common covariance, as in LDA. In this sense, RDA is a compromise between LDA and QDA. The regularized covariance matrices have the form

$$S_c(\lambda) = \lambda S_c + (1-\lambda)S$$

where S is the pooled covariance matrix used in LDA and S_c is the class specific covariance matrix of the input features used in QDA. Here, $\lambda \in [0,1]$ allows a continuum of models between LDA and QDA and needs to be specified. In practice, λ can be chosen employing cross-validation. Biasing the class covariance matrices toward commonality is not the only way to shrink them. An additional convex combination allows S_c itself to be shrunk toward a scaled identity matrix, using the shrinkage parameter γ as follows:

$$S_c(\lambda,\gamma) = (1-\gamma)S_c(\lambda) + \gamma\frac{1}{d}\,\mathrm{tr}[S_c(\lambda)]I$$

where $\frac{1}{d}\,\mathrm{tr}[S_c(\lambda)]$ is the mean of the diagonal elements of $S_c,(\lambda)$, so it is the mean variance of the class input features. The RDA classifier is given by the following:

$$\Psi_c^{RDA}(z) = \Big\{(z-\bar{z})' S_c^{-1}(\lambda,\gamma)(z-\bar{z}_k) + log|S_c(\lambda,\gamma)|\Big\} \tag{6}$$

where λ is the cross-validated parameter that controls the degree of shrinkage of the individual class covariance matrix estimates toward the pooled estimates and γ is an additional regularization parameter that controls shrinkage toward a multiple of the identity matrix for a given value of λ (Friedman, 1989).

3.4.4 Heteroscedastic Discriminant Analysis

The heteroscedastic discriminant analysis (HDA) is a generalized method of the LDA in that its feature space transformation does not require the imposition of equal within-class covariance assumptions as compared to the standard LDA. The HDA classifier is capable of handling different covariance structures of the class distributions (Kumar and Andreou, 1998).

Let $\Big\{z_i\Big\}_{i=1}^{N}$ denote a sequence of K-dimensional feature vectors, with each vector belonging to a single class $j \in \{1,...,C\}$, and let y denote a categorical response variable. If N_j, μ_j and Σ_j represent the sample count, mean and covariance, respectively, of the j^{th} class, then the between-class matrix M can be extracted in the following form:

$$M = \frac{1}{N}\sum_{j=1}^{C} N_j\mu_j\mu_j' - \mu\bar{\mu}'$$

where μ_j' is the transpose of μ_j of the j^{th} class; μ is a vector of overall means.

The HDA objective function seeks to find a projection matrix, denoted by β, that maximizes the likelihood in the Jacobian transformation space $y = \beta' z$ under the normality assumption, such that the ratio of the determinants

$$\Omega(\beta) = \frac{|\beta M \beta'|^N}{\prod_{j=1}^{C} |\beta\Sigma_j\beta'|^{N_j}} \tag{7}$$

is maximized, where β' is the transpose of β (Huang et al., 2000; Szepannek et al., 2009).

The HDA classifier is then given by the following:

$$\Psi^{HDA}(\beta) = \underset{\beta}{\mathrm{argmax}}\, log\Big\{\Omega(\beta)\Big\} = \underset{\beta}{\mathrm{argmax}}\Big\{\sum_{j=1}^{C} -N_j log|\beta\Sigma_j\beta'| + N log|\beta M\beta'|\Big\} \tag{8}$$

where M is the between-class matrix. See Kumar and Andreou (1998) for further details.

3.4.5 Sparse Discriminant Analysis

The sparse LDA introduces projection techniques that imposes zero entries in the feature matrix, aimed at reducing the dimensionality to produce a final parsimonious model. The sparse discriminant function involves the inclusion of an ℓ_1 penalty norm in the optimal scoring problem which results in the optimization problem, as follows:

$$max_{\beta_j}\beta_j' S\beta_j - \eta\,||\beta_j||_1 \quad \text{subject to} \quad \beta_j'(S_w+\Omega)\beta_j = 1, \quad \beta_j'(S_w+\Omega)\beta_m = 0 \quad \text{for all} \quad m < j \tag{9}$$

where β_j is the discriminant vector of class j, Ω is a positive definite matrix; S_w is the within class covariance matrix. The j^{th} sparse discriminant analysis solution pair (θ_j, β_j) is obtained by solving the problem, as follows:

$$min_{\beta_j,\theta_j}\left\{ \|y\theta_j - z\beta_j\|^2 + \eta\beta_j^{'}\Omega\beta_j + \lambda\,\|\beta_j\|_1 \right\} \tag{10}$$

$$\text{subject to} \quad \frac{1}{n}\theta_j^{'}y^{'}y\theta_j = 1 \quad \text{and} \quad \theta_j^{'}y^{'}y\theta_m = 0 \quad \text{for all} \quad m < j$$

and the sparse LDA is as follows:

$$\Psi^{SparseLDA}(\theta,\beta) = \underset{\beta_j,\theta_j}{\text{argmin}}\left\{ \frac{1}{n}\|y\theta_j - z\beta_j\|^2 + \eta\beta_j^{'}\Omega\beta_j + \lambda\|\beta_j\|_1 \right\}$$

where y is a matrix of dummy variables for the j^{th} classes; θ_j is a j-vector of scores; n is the sample size; η and λ are non-negative tuning parameters (Clemmensen et al., 2011). Thus the ℓ_1 penalty norm on β_j results in sparsity when the tuning parameter λ is large.

3.4.6 High Dimensional Discriminant Analysis

The high dimensional discriminant analysis (HDDA) is another important extension of the LDA most feasible for a dimensionality reduction model involving many features as compared to the sample size, and in which the LDA is weak in performance. Let Γ_i be an orthogonal matrix of eigenvectors of a covariance matrix S_i; let Φ_i be the basis from the eigenvectors of S_i, and assuming the class conditional densities follows Gaussian $\mathcal{N}(\mu_i, S_i)$ for all $i = 1, ..., C$. Then, the class conditional covariance matrix Ω_i, is defined by the following:

$$\Omega_i = \Gamma_i^{'}S_i\Gamma_i$$

where Ω_i is diagonal matrix with two distinct eigenvectors u_i and v_i, $u_i > v_i$.
If $\Pi_i(z) = \hat{\Gamma}_i\hat{\Gamma}_i^{'}(z-\mu_i)+\mu_i$ represents the projection of the input vector z on the affine space $\mho_i$, then the cost function will be as follows:

$$\theta_i(z) = \frac{\|\mu_i - \Pi_i(z)\|^2}{u_i} + \frac{\|z - \Pi_i(z)\|^2}{v_i} + d_i \ln u_i + (K - d_i)\ln v_i - 2\ln\pi_i \tag{11}$$

where $u_i = \dfrac{\sigma_i^2}{\alpha_i}$ and $v_i = \dfrac{\sigma_i^2}{1-\alpha_i}$ with $\alpha_i \in \{0,1\}$ and $\sigma_i > 0$ for all $i = 1, ..., C$; K is the K-dimensional input vector; d_i is the i^{th} diagonal of Γ_i (Bouveyron et al., 2007).
The posterior probability is defined as follows:

$$Prob(C_i|z) = \frac{e^{-\frac{1}{2}\theta_i(z)}}{\sum_{j=1}^{C} e^{-\frac{1}{2}\theta_j(z)}} \quad \text{for} \quad i \neq j \tag{12}$$

Thus, the maximum likelihood estimators of u_i and v_i are, respectively, as follows:

$$\hat{u}_i^{MLE} = \frac{1}{d_i}\sum_{j=1}^{d_i}\omega_{i,j} \quad \text{and} \quad \hat{v}_i^{MLE} = \frac{1}{K-d_i}\sum_{j=d_i+1}^{K}\omega_{i,j}$$

where $\omega_{i,1} \geq \omega_{i,2} \geq ... \geq \omega_{i,K}$ are the eigenvectors of $\hat{S}_i$.

Following this approach, the maximum likelihood estimators of α_i and σ_i^2 are

$$\hat{\alpha}_i^{MLE} = \frac{\hat{v}_i}{\hat{u}_i + \hat{v}_i} \quad \text{and} \quad (\hat{\sigma}_i^2)^{MLE} = \frac{\hat{u}_i\hat{v}_i}{\hat{u}_i + \hat{v}_i}$$

3.4.7 Distance Weighted Discrimination

The distance weighted discrimination (DWD) was introduced by Marron et al. (2007) to tackle high-dimensional datasets and to specifically improve the performance of support vector machines. It employs the concept of maximization, thereby maximizing the existing gap between an ordered pair of classes to make them more separable, introducing harmonic mean of the distances of all data vectors to the separating hyperplane (Huang et al., 2012). Given the training dataset $\{(y_i, z_i)\}_{i=1}^n$ with k-dimensional vector of covariates Z, Y the binary response variable $y \in \{-1, +1\}$, let $d_i = (z_i^{'}w + \theta)y_i + \alpha_i$ be the distance of the i^{th} data vector to the separating hyperplane. Then, the DWD is obtained by the following:

$$\underset{w,\theta,\alpha_i}{\operatorname{argmin}} \sum_{i=1}^{n} \left(\frac{1}{d_i} + C(\alpha_i)\right) \text{ subject to } d_i = (z_i^{'}w + \theta)y_i + \alpha_i;\ \ d_i, \alpha_i \geq 0;\ \forall\ ||w||^2 \leq 1 \tag{13}$$

where α_i is a positive slack variable included to boost the positivity of d_i; w is the weight vector (Qiao and Zhang, 2015). The slack variable serves as a correction measure, which corresponds to the amount of misclassification for the i^{th} vector. Thus, the DWD binary linear classification process employs gap minimization to improve the separability of the two classes and the minimization of the misclassification error.

3.5 Classification and Regression Trees

Classification and regression trees (CART) involve the use of decision tree learning procedures to build a model that can predict the value of a target variable based on several input variables, see Breiman et al. (1984). There are many classification algorithms, including decision trees, rule-based learners, support vector machines, neural networks and Bayesian networks. There are also ways of combining them into ensemble classifiers, such as bagging, boosting, and random forests. The consistent CART models in this study include the following: bagging, random forest, conditional inference tree, conditional inference forest, adaptive boosting, gradient boosting, generalized linear boosting, logitboost, recursive partitioning, k nearest neighbour, naive Bayes, learning vector quantization and neural networks.

3.5.1 Bagging

Bagging or bootstrap aggregating was introduced by Breiman (1996) to improve classification by combining classifications of randomly generated training datasets, to reduce the biases and variances in a tree-based analysis. Bagging implies fitting a model, including all potential points on the original training set. It appears to effectively remove the instability of a decision rule by averaging across resamples and to avoid overfitting (Zheng, 2006). Let $S = \{(z_1,y_1),\cdots,(z_T,y_T)\}$ denote the training sample, where T is the number of observations in the training sample, z_t is a vector of k covariates, and $y_t \in \{-1,1\}$ indicates a negative or positive return for each t. The classification into one of the two groups is defined as follows:

$$\hat{\Psi}(z) = sign\Big(\hat{\delta}(z_t) - \tau_B\Big), \hat{\Psi}(z) \in \{-1,1\}$$

where τ_B is the cut-off value; $\hat{\delta}(z_t)$ is the base classifier that learned the covariates in the training sample; $\hat{\delta}(z_t) > \tau_B$ implies a positive return classification, while $\hat{\delta}(z_t) < \tau_B$ implies a negative return classification (Lemmens and Croux, 2006). The decision tree classification score is given by the following:

$$\hat{\delta}(z) = 2\hat{\rho}(z) - 1,$$

where $\hat{\rho}(z)$ is the predicted probability of a positive return estimated by the tree. For each bootstrap sample S_b^*, a classifier can be estimated employing the score functions $\hat{\delta}_b(z)$ for $b = 1,2,...,B$. These functions are afterwards aggregated into a score, as follows:

$$\hat{\delta}_{bag}(z) = \frac{1}{B}\sum_{b=1}^{B}\hat{\delta}_b(z).$$

Thus, the final classification is obtained as follows:

$$\hat{\Psi}_{bag}(z) = sign\Big(\hat{\delta}_{bag}(z) - \tau_B\Big). \quad (14)$$

3.5.2 Random Forest

A random forest (RF) classifier, see Breiman (2001), is a specific type of bootstrap aggregating based on a random subset of the input features (Ballings et al., 2015; Kumar and Thenmozhi, 2006). A random forest classifier consists of an ensemble classification algorithm that involves the use of trees as base classifiers. It consists of a combination of classifiers in which each classifier contributes an individual vote for assigning the most frequent class to the input vector z, defined by the following:

$$\delta_{RF}^{\hat{B}} = majority\ vote\Big\{\hat{\delta}_b(z)\Big\}_{b=1}^{B} \quad (15)$$

where $\hat{\delta}_b(z)$ is the class prediction of the b^{th} random forest tree.

The Gini index, suggested by Breiman et al. (1984), is employed for selecting the best split at each node. For a given node τ with estimated class probabilities $Prob(j|\tau), j = 1,...,J$, the node impurity, $I(\tau)$, employing the Gini index is defined as follows:

$$I(\tau) = \sum_{j\neq i}^{J} Prob(j|\tau)Prob(i|\tau). \quad (16)$$

The Gini index is minimised when the node is pure (homogeneous) with respect to one of the classes.

3.5.3 Conditional Inference Tree

The conditional inference tree (CTree) enables the use of recursive partitioning and tree-structured models in a conditional inference framework. The use of the Gini index to determine the most favourable split induces a selection bias toward covariates with many possible splits and also cannot distinguish between a significant and an insignificant improvement in the information measure. Hothorn et al. (2006) proposed the conditional inference approach tree where the node split is selected based on how good the association is between the response and the covariates. The resulting nodes should provide a high association between the response and the covariates. The significance of the association is investigated by a χ^2 test and the covariate with highest association is selected for splitting. Moreover, multiple test procedures are applied to determine whether no significant association between any of the covariates and the response can be stated and the recursion needs to stop.

In more detail, let $Z = (Z_1, \cdots, Z_k)$ be the k-dimensional vector of covariates and let Y be a categorical response variable. Z is taken from a sample space $\mathcal{Z} = \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_k$. We assume that the conditional distribution of Y given Z depends on the function f of Z as follows:

$$D(Y|Z) = D(Y|Z_1, \cdots, Z_k) = D(Y|f(Z_1, \cdots, Z_k)).$$

Thus, a generic algorithm for recursive binary partitioning for a given learning sample

$$\mathcal{L}_n = (Y_i, Z_{1i}, \cdots, Z_{ki}),\ i = 1, \cdots, n,$$

can be formulated using non-negative integer valued case weights $\omega = (\omega_1, \cdots, \omega_n)$.

Each node of the tree is represented by a vector of case weights having nonzero elements when the corresponding observations are elements of the node, and are zero otherwise. The following steps implement recursive binary partitioning:

1. Test the global null hypothesis of independence between any covariate Z and the categorical response variable Y for case weights ω. Stop if this hypothesis cannot be rejected. Otherwise, select the j-th covariate Z_j with the strongest association to Y.

2. Choose a set $A \subset \mathcal{Z}_j$ to split $\mathcal{Z}_j$ into two disjoint sets of A and A^c. The case weights ω_{left} and ω_{right} determine the two subgroups with $\omega_{left,i} = \omega_i I(Z_{j,i} \in A)$ and $\omega_{right,i} = \omega_i I(Z_{j,i} \notin A)$, for all $i = 1, 2, \cdots, m$, where $I(\cdot)$ is the indicator function.

3. Repeat steps 1 and 2 recursively with the different case weights ω_{left} and ω_{right}, respectively.

3.5.4 Conditional Inference Forest

Random forest has been criticised for the bias that results from favouring covariates with many split-points. The conditional inference forest (CForest) is known to correct this bias by separating the procedure for the best covariate to split on from that of the best split point search for the selected covariate. The conditional inference forest is an implementation of the random forest and bootstrap aggregating ensemble algorithms, utilising conditional inference trees as base learners.

To determine the variable importance in conditional inference forests, the vector of the predictor variables is randomly permuted and the initial association with the response variable is broken. When the permuted and the non-permuted variables are used to predict the response variable for the out of bag observations, the classification accuracy decreases substantially if the permuted variable is associated with the response. Hence, the variable importance is the difference in the prediction accuracy before and after permutation of the variable average over all trees (Strobl et al., 2008; Das et al., 2009).

3.5.5 Adaptive Boosting

Boosting is an ensemble technique aimed at increasing the strength of a weak learning classifier by improving its accuracy. The principle consists of sequentially applying the classifier to adaptively re-weighted versions of the initial dataset $S_b^*, b = 1, 2, \cdots, B$. In each step, the learning attention is focused on modified versions of the data, where the modifications give more weight, w_t, to misclassified points. Once the process has finished, the single classifiers obtained are combined into a final classifier by weighted majority vote. We employ the Adaptive Boosting (Adaboost) procedure proposed by (Freund and Schapire, 1996; Alfaro et al., 2013). The main steps of the Adaboost algorithm are as follows:

1. Initialize the observation weights $w_t = \dfrac{1}{T}$ for $t = 1, 2, \cdots, T$.
2. For $b = 1, 2, \cdots, B$:
   (a) Fit a classifier $c_b(z)$ to the training data using observation weights w_t.
   (b) Compute the weighted misclassification error for c_b:
   $err_b = \frac{\sum_{t=1}^{T} w_t I[y_t \neq c_b(z_t)]}{\sum_{t=1}^{T} w_t}$
   (c) Compute $\alpha_b = \frac{1}{2} ln[\frac{1-err_b}{err_b}]$
   (d) Update the weights $w_t \leftarrow w_t exp(\alpha_b I[y_t \neq c_b(z_t)])$, for $t = 1, 2, \cdots, T$ and normalize them.
3. Output the final classifier $\hat{\Psi}_{boost}(z) = sign\left[\sum_{b=1}^{B} \alpha_b c_b(z)\right], \hat{\Psi}_{boost}(z) \in \{-1, 1\}$.

3.5.6 Gradient Boosting

Friedman (2001, 2002) laid the groundwork for a new generation of boosting algorithms. Assume that we are interested in modelling $Pr(Y = 1|Z = z)$ for a Bernoulli response variable. The idea is to fit a model of the following form:

$$\lambda(z) = G_B(z) = \sum_{b=1}^{B} g_b(z;\gamma_b)$$

where

$$\lambda(z) = log\left(\frac{Pr(Y=1|Z=z)}{Pr(Y=0|Z=z)}\right)$$

and γ_b is parameter vector, which for the trees, captures the identity of the split variables, their split values and the constants in the terminal nodes. The main steps of the gradient boosting algorithm are as follows:

1. Start with $\hat{G}_0(z) = 0$, and set the shrinkage parameter $\varepsilon > 0$.
2. For $b = 1, 2, \cdots, B$:
   (a) Compute the pointwise negative gradient of the loss function at the current fit as follows:
   $r_t = -\frac{\partial L(y_t,\lambda_t)}{\partial \lambda_t}$
   (b) Approximate the negative gradient by a depth-d tree by solving the following:
   $minimise_\gamma \sum_{t=1}^{T}(r_t - g_b(z;\gamma_b))^2$.
   (c) Update $\hat{G}_b(z) = \hat{G}_{b-1}(z) + \hat{g}_b(z)$, with $\hat{g}_b(z) = \varepsilon g(z;\hat{\gamma}_b)$.
3. Return the sequence $\hat{G}_b(z)$, for $b = 1, 2, \cdots, B$.

3.5.7 Gradient Boosting With Component-Wise Linear Models

Gradient boosting with component-wise linear models (GLMBoost) employs component-wise (generalised) linear models as base-learners (Bühlmann and Yu, 2003; Bühlmann et al., 2006, 2007).

Let $z = (z_1, z_2, ..., z_K)^{'}$ be a set of K-dimensional covariates, from which the categorical binary response variable $y_i \in \{1, ..., C\}$ can be predicted. Then, a generalized linear model can be fitted in the following form:

$$\ell(\hat{\mu}) = \beta_0 + \beta_1 z_1 + ... + \beta_K z_K \tag{17}$$

where $\hat{\mu} = \mathbb{E}(y|z)$ is the conditional expectation of the binary response; ℓ is the link function; β is a vector of unknown parameters.

The boosted generalized linear model additionally performs variable selection and the effects are shrunken toward zero if early stopping is applied in the model (Hofner et al., 2014). The GLMBoost fits simple linear models separately for each column of the design matrix to the negative gradient vector, for each boosting iteration, using only the best fitting base-learner in the update step.

3.5.8 LogitBoost

The LogitBoost is an algorithm used to produce a logistic regression model at every node in the classification tree and each node is able to be split using a suitable splitting criterion (Friedman et al., 2000; Landwehr et al., 2005). It is designed to train the classification algorithm using stumps or one node decision trees as weak learners.

Let $\{(y_i, z_i)\}_{i=1}^{N}$ be the input dataset with N samples, $z_i \in Z$, $y_i \in Y \in -1, 1$. We use the transformation $y^* = \frac{1+y}{2}$ to represent the outcome with a $0/1$ response. We represent the probability of $y^* = 1$ with $p(z)$ where

$$p(z) = \frac{e^{F(z)}}{e^{F(z)} + e^{-F(z)}}.$$

The main steps of the LogitBoost algorithm are as follows:

1. Start with $w_t = 1/T, t = 1, \cdots, T$, $F(z) = 0$, and probability estimates $p(z_i) = \frac{1}{2}$.

2. For $b = 1, 2, \cdots, B$:

   (a) Compute the working response r_i and the weights as

   $$\begin{cases} w_i = p(z_i)(1 - p(z_i)) \\ r_i = \frac{y^* - p(z_i)}{w_i} \end{cases}$$

   (b) Fit the function $f_b(z)$ by a weighted least-squares regression of r_i to z_i using weights w_i.

   (c) Update $F(z) \leftarrow F(z) + \frac{1}{2} f_b(z)$, and $p(z) = \frac{e^{F(z)}}{e^{F(z)} + e^{-F(z)}}$.

3. Return the classifier $sign\left[F(z)\right] = sign\left[\sum_{b=1}^{B} f_b(z)\right]$, for $b = 1, 2, \cdots, B$.

3.5.9 Recursive Partitioning Algorithm

The recursive partitioning (RPart) algorithm builds a decision tree that attempt to correctly classify elements of the set by splitting it into subsets based on several features. The splitting process continues indefinitely, resulting in newer sub-samples and terminates after a specific stopping criterion is attained (Cook and Goldman, 1984).

Let y_t be a conditionally distributed dichotomous response variable given the k predictors, such that the k predictors are elements of a sample space $\Omega = \Omega_1 \times \Omega_2 \times .... \times \Omega_k$. Then, by tree-structured recursive partitioning, the conditional distribution of y_t given z_{t-1} depends on the function

$$\Psi(y_t|z_{t-1}) = \Psi(y_t|g(z_{(t-1)1}, z_{(t-1)2}, ..., z_{(t-1)k})) \tag{18}$$

from which the p disjoint cells $B_1, B_2, ..., B_p$ partitioning the predictor space

$$\Omega = B_1 \cup B_2 \cup ..... \cup B_p = \cup_{j=1}^{p} B_j$$

are obtained; where $g(.)$ is a function of the k predictors (Hothorn et al., 2006).
The fitted model is based on a learning sample defined by the following:

$$\ell_T = \{y_t; z_{(t-1)1}, z_{(t-1)2}, ..., z_{(t-1)k}; t = 1, 2, ..., T\} \tag{19}$$

The recursive algorithm proposed by Zeileis et al. (2008), Hothorn et al. (2006) is as follows:

1. Fit the model to all observations at once in the initial node and estimate the unknown parameters by minimizing the objective function;
2. Evaluate the stability or instability of the estimated parameters with respect to the ordering features;
3. Determine the splitting point that locally optimizes the objective function using a fixed or adaptive number of splits;
4. Split the node into sub-nodes and repeat the procedure recursively until no further splitting is feasible.

3.5.10 k Nearest Neighbour

The k nearest neighbour (kNN) is used for classifying objects based on the closest training instances in the feature space., Given the training data set $\{(z_1, y_1), (z_2, y_2), \cdots, (z_L, y_L)\}$, an object is to be classified based on a majority being assigned to the class most common to its corresponding k nearest neighbours. In more detail, given a positive integer k and a test observation z_{L+1}, the kNN classifier first identifies the k points in the training data set that are closest (using for example the Euclidean distance) to z_0, represented by N_0. It then estimates the conditional probability for class j as the fraction of points in N_0 whose response equals j:

$$Pr(Y = j|Z = z_0) = \frac{1}{k} \sum_{i \in N_0} I(y_i = j). \tag{20}$$

Finally, the Bayes rule is applied and the test observation is classified to the class with the largest probability. The k can be chosen by cross-validation, and the kNN model does not depend on the prior probabilities of the classes (James et al., 2013; Huang et al., 2008; Su, 2011).

3.5.11 Naive Bayes

The Naive Bayes classifier is a simple example of Bayes Networks. It combines the Bayes formula with a decision rule, and a common rule is to pick the most probable hypothesis, which is known as maximum posterior decision rule (Ripley, 1996; Ou and Wang, 2009).

Given the input of m features $z_1, \cdots, z_m$ and using the assumption that the features are independent given the class j, we have that Naive Bayes classifier is as follows:

$$\Psi^{NB}(z) = argmax_j \left\{ \psi_j \prod_{i=1}^{m} p(z_i|j) \right\} \tag{21}$$

where ψ_j is estimated from the sample proportion.

3.5.12 Learning Vector Quantization

The learning vector quantization (LVQ) algorithm (Kohonen, 1995; Ripley, 1996), is an artificial neural network designed to enable one to construct a modified training set iteratively. The modified training sets are called codebooks. We will describe the LVQ1 process based on Kohonen (1995). Assume that a number of codebooks m_i are placed into the input space to approximate various domains of the input vector z by their quantized values. Usually several codebook vectors are assigned to each class of z values, and z is then decided to belong to the same class to which the nearest m_i belongs. Let $c = argmin(||z - m_i||)$, define the nearest m_i to z, denoted by m_c.

Values for the m_i that approximately minimize the misclassification errors in the above nearest-neighbor classification can be found as asymptotic values in the following learning process. Let $z(t)$ be a sample of input and let the $m_i(t)$ represent sequences of the m_i in the discrete-time domain. The basic LVQ1 process is defined by:

$$m_c(t+1) = m_c(t) + \alpha(t)[z(t) - m_c(t)]$$

if z and m_c belong to the same class,

$$m_c(t+1) = m_c(t)\alpha(t)[z(t) - m_c(t)]$$

if z and m_c belong to different classes, and

$$m_i(t+1) = m_i(t)$$

for i not in c. Here $0 < \alpha(t) < 1$, and $\alpha(t)$ may be constant or decrease monotonically with time.

3.5.13 Neural Network

The neural network (NNET) is a system made up of a number of simple highly interconnected processing elements, which process information by their dynamic state response to external inputs. The NNET consists of layers made up of interconnected nodes that contain the activation function (Ripley, 1996; Hastie, 2005; Caudill, 1989). The NNET layers are as follows:

$$InputLayer \longmapsto Hidden \quad Layer \longmapsto Output$$

Given an input vector of covariates z, and a categorical output y, neural network can be modelled in the following form:

$$x_j = \Gamma(\theta_{0,j} + \theta_j^{'} z) \quad for \quad j = 1,2,...,m$$

$$\hat{y}_k = \Psi_k(\beta_{0,k} + \beta_k^{'} x) \quad for \quad k = 1,2,...,q$$

where $\Gamma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid activation function, used to introduce a nonlinearity at the hidden layer. The parameters $\theta_{j,l}$ and $\beta_{k,j}$ are known as the weights and define linear combinations of the input vector z and hidden unit output x. The intercepts $\theta_{0,j}$ and $\beta_{0,k}$ are known as biases. The function Ψ_k permits a final transformation of the output and a typical choice for binary classification is the inverse logit function.

3.6 Statistical and Economic Performance Evaluation

3.6.1 Confusion Matrix Metrics

The confusion matrix consists of true positives (TP), false positives (FP), false negatives (FN) and true negatives (TN), presented in Table 1. In this paper, we use the following metrics to evaluate the accuracy and correctness of the classification models:

	Positive (Predicted)	Negative (Predicted)
Positive (Actual)	TP	FN
Negative (Actual)	FP	TN

Table (1) The Confusion Matrix

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

The accuracy is also known as the correct prediction ratio (CPR).

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

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

$$Specificity = \frac{TN}{TN + FN}.$$

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

Kappa Statistic: The kappa statistic, denoted by κ, is computed as follows:

$$\kappa = \frac{p_0 - p_e}{1 - p_e} = 1 - \frac{1 - p_0}{1 - p_e}$$

where p_0 is the relative observed agreement among the raters; p_e is the hypothetical probability of chance agreement, which can be obtained from the following:

$$p_e = \frac{1}{N^2}\sum_m n_{m_1} n_{m_2}$$

for categories m with N items and n_{m_i} is the number of times rater i predicted category m.

McNemar's Test: The McNemar's test, as introduced by McNemar (1947), is used in this paper to investigate the marginal homogeneity between the row and column marginal frequencies in the 2×2 confusion matrix. The null hypothesis of marginal homogeneity (that is the two outcomes are marginally equiprobable) against the alternative hypothesis that they differ in probabilities is defined as follows:
$H_0 : Prob(FN) = Prob(FP)$
$H_A : Prob(FN) \neq Prob(FP)$
The $McNemar's$ test statistic is defined as follows:

$$\chi^2 = \frac{(FN - FP)^2}{FN + FP} \sim \chi_1^2(\alpha).$$

Thus, the McNemar's test statistic is asymptotically chi-square distributed with 1 degree of freedom at the α% significance level.

3.6.2 The Pesaran-Timmermann Directional Predictability Test

This test was first proposed by Pesaran and Timmermann (1992) and was employed also by Granger and Pesaran (2000) for evaluating directional forecasting or predictability performance and market timing. The null hypothesis H_0, which is "No statistically significant directional predictability" against the alternative hypothesis H_A, which is "There is statistically significant directional predictability" can be tested based on the Pesaran-Timmermann test statistic, as follows:

$$PT = \frac{\sqrt{T}KS}{\left(\frac{\bar{\tau}_I(1-\bar{\tau}_I)}{\bar{I}(1-\bar{I})}\right)^{0.5}} \overset{asymptotically}{\sim} N(0,1)$$

where $KS = TR - FR$ is the Hanssen-Kuiper skill score; $TR = \frac{\hat{I}^{uu}}{\hat{I}^{uu} + \hat{I}^{du}}$ is the true or hit rate; $FR = \frac{\hat{I}^{ud}}{\hat{I}^{ud} + \hat{I}^{dd}}$ is the false rate; T is the sample period in months;
and the forecasts' classifications are again obtained from the 2 x 2 confusion matrix showing:

$$\hat{I}^{uu} = \sum_{t=1}^{T} I(\hat{I}_t = 1, I_t = 1);$$

$$\hat{I}^{ud} = \sum_{t=1}^{T} I(\hat{I}_t = 1, I_t = 0);$$

$$\hat{I}^{du} = \sum_{t=1}^{T} I(\hat{I}_t = 0, I_t = 1);$$
$$\hat{I}^{dd} = \sum_{t=1}^{T} I(\hat{I}_t = 0, I_t = 0);$$

where u is an upward signal ($I_t = 1$) and d is a downward signal ($I_t = 0$); $I(.)$ is the indicator function taking the values 0 and 1; $\bar{I}$ is the sample mean of the sign indicator values I_t computed in the $T - month$ sample period; $\hat{I}_t$ is the predicted excess stock return sign indicator; I_t is the actual excess stock return sign indicator; $\bar{\tau}_I = \bar{I}TR + (1 - \bar{I})FR$ (Nyberg, 2011; Granger and Pesaran, 2000; Bergmeir et al., 2014).

Thus, the PT test statistic as stated above has the asymptotic standard normal distribution under the null hypothesis H_0 of no directional predictability.

3.6.3 Economic Performance Evaluation

Evaluating the economic performance of a forecasting model is of great importance to a profit oriented portfolio investor. The models proposed in this paper seem to provide useful evidence of economic significance.

We consider the following trading strategy: Let $Prob_t(R_{t+1} > 0)$ be the estimated probability of a positive excess stock return for the period $t + 1$. Then the trading strategy or decision rule can be expressed as follows:

If $Prob_t(R_{t+1} > 0) > 0.5$, then purchase the stock index.

Else if $Prob_t(R_{t+1} > 0) \leq 0.5$, then purchase the treasury bill.

The performance of the constructed portfolios is evaluated over the out-of-sample period (1991 to 2016: T=312 months) using a plethora of performance measures. First, we consider the realized returns of the constructed portfolios. Let $r_{p,t+1}$ be the realized return of the portfolio at time $t + 1$. We calculate the average return (AR) within the out-of-sample period, the cumulative return at the end of the period, and the volatility of the portfolio. We also compare the return per unit of risk by using the Sharpe Ratio.

Sharpe Ratio

We use the Sharpe Ratio (SR), which standardizes the realized returns with the risk of the portfolio and is calculated through the following equation:

$$SR_p = \frac{E(r_p) - E(R_f)}{\sqrt{Var(r_p)}},$$

where r_p is the average realized return of the portfolio over the out-of-sample period; R_f is the risk-free interest rate and $Var(r_p)$ is the variance of the portfolio over the out-of-sample period. Portfolios with high Sharpe ratios are most preferable to portfolios with low Sharpe ratios, owing to the fact that the higher the Sharpe ratio, the higher the return and the lower the volatility.

Maximum Drawdown

We also calculate the maximum drawdown (MaxDD). MaxDD determines the maximum sustained percentage decline (peak to trough), which has occurred in the portfolio within the period studied. MaxDD up to time T is the maximum of the drawdown over the history of the specific variable under consideration, and it is computed as follows:

$$MaxDD_p = \max_{T_0 \leq t \leq T-1} [\max_{T_0 \leq j \leq T-1} (PV_j) - PV_t],$$

where PV denotes the portfolio value; T_0, T denote the beginning and end of the evaluation period, respectively.

Omega Ratio

The Omega ratio, as a risk-return performance measure of a portfolio investment introduced by Keating and Shadwick (2002), gives the probability weighted ratio of gains versus losses for a stipulated threshold return target. We first define the n-th lower partial moment (LPM_n) of the portfolio return and the kappa function K_n, and used the concept to compute Omega, Sortino and the Upside Potential respectively, see (Harlow and Rao, 1989; Sortino and Van Der Meer, 1991; Sortino and Price, 1994) for detail studies. The n-th lower partial moment of the portfolio return is defined as follows:

$$LPM_n(r_b) = E[((r_b - r_p)_+)^n]$$

where r_b is the benchmark return.

The Kappa function $K_n(r_b)$ is defined as follows:

$$K_n(r_b) = \frac{E(r_p) - r_b}{\sqrt[n]{LPM_n(r_b)}} \quad \text{for} \quad n = 1, 2, ...$$

Thus, the Omega ratio is computed from the following formula:

$$Omega(r_b) = K_1(r_b) + 1.$$

Sortino Ratio

Unlike the Sharpe ratio, which penalizes both upside and downside volatility equally, the Sortino ratio penalizes only the returns that fall below a user specified target. The Sortino ratio measures the risk adjusted return of a portfolio. It can be computed from the following formula:

$$S(r_b) = K_2(r_b).$$

Like the Sharpe ratio, the higher the Sortino ratio, the better the risk adjusted performance and vice versa.

Upside Potential

Upside Potential is a measure of the return of an investment relative to the minimal acceptable return. The upside potential is calculated as follows:

$$UP(r_b) = \frac{E[(r_p - r_b)^+]}{\sqrt{LPM_2(r_b)}}$$

The economic importance of the upside cannot be overemphasized. It not only indicates an investor′s potential gain in value but also judges the success of a portfolio manager′s performance comparative to a benchmark.

Additionally, we investigate the tail-risk of the different proposed portfolios. A CVaR of λ% at the 100(1-α)% confidence level means that the average portfolio loss measured over 100α% of worst cases is equal to λ% of the wealth managed by the investor. To compute VaR and CVaR, we use the empirical distribution of the portfolio realized returns. VaR and CVaR are calculated at the 95% confidence levels.

In this paper, we employ the U.S. 3-month interest rate for the risk free rate R_f and for the benchmark rate of return (r_b) necessary for the calculation of *Omega*, *MaxDD* and *S*.

4 Data Analysis and Discussion

4.1 Sources of Data and Variables

The data used in this paper are obtained from Amit Goyal's webpage[2], covering monthly observations ranging from January 1960 to December 2016. These variables, presented in Table 2, have been used in the existing literature quite extensively for predictability of the equity premium, see (Rapach et al., 2010; Nyberg, 2011; Meligkotsidou et al., 2014, 2019) among others. The total number of observations is $T = 684$. An out-of-sample period of $T_2 = 312$ monthly observations ranging from January 1991 to December 2016 has been employed for the evaluation of the forecasting performance. The forecast horizon denoted by h is one month ahead for each of the forecasting models.

In the out-of-sample method, the parameters of the forecasting models are estimated recursively using an expanding window of observations, in which the fitted models are estimated using data from the start date of the dataset to the present time and obtain a one month-period-ahead forecast. The procedure is repeated iteratively until the end of the forecast sample period is attained. In the CART techniques, we train each classification model, pre-process the training dataset in a closed centre and scale form, tune the parameter(s) of each model by cross-validation and resampling, determining the variable importance before making the out-of-sample forecasts. The resampling approach seeks to determine the values of each of the model parameters (if any) and uses the best tuning parameter(s) based on fitted in-sample

[2] www.hec.unil.ch/agoyal/

Table (2) The Financial Variables used for the Study

Indicator	**Time Series Variable**
Equity Premium	$EquityPrem$
Default Return Spread	DFR
Excess Stock Return	ESR
Short Term Interest Rate	$\Delta ShortR$
Long Term Yield	$\Delta LongR$
Term Spread	$TermSpr$
Inflation	$\Delta Infl$
Return Spread	$ReturnSpr$
Yield Spread	$YieldSpr$
Book to Market Value	BMV
Net Equity Expansion	NEE
Dividend Price Ratio	DPR
Earning Price Ratio	EPR
Stock Variance	$SVar$

accurate measures to produce the out-of-sample forecasts. In each model, the best tuning parameter(s) were used to run the out of sample forecasts recursively, and their respective performance evaluation measures were obtained.

4.2 Statistical Performance Evaluation Results

The statistical performance evaluation results for the proposed techniques in this paper, presented in Table 3, are shown to be promising, owing to the empirical evidence of useful predictability. The out-of-sample positive class return forecasts are depicted in Figure 1. In the benchmark binary probit models, the predictive accuracy of the static binary probit model involving all covariates appeared to be very low with insignificant evidence of PT directional predictability, and the kappa statistic is extremely poor, indicating a poor inter-rater agreement between the actual and predicted values. Whereas the application of stepwise variable selection by the Akaike information criterion (AIC) on the static model seeks to improve the predictive accuracy, it does not provide statistically significant evidence of directional predictability and the kappa statistic is still low. The dynamic binary probit, which includes the lagged excess stock return indicator together with the other predictor variables, produced a slightly better predictive accuracy as compared to the static probit. The "$***$", "$**$" and "$*$" signified 0.1%, 1% and 5% significance, respectively.

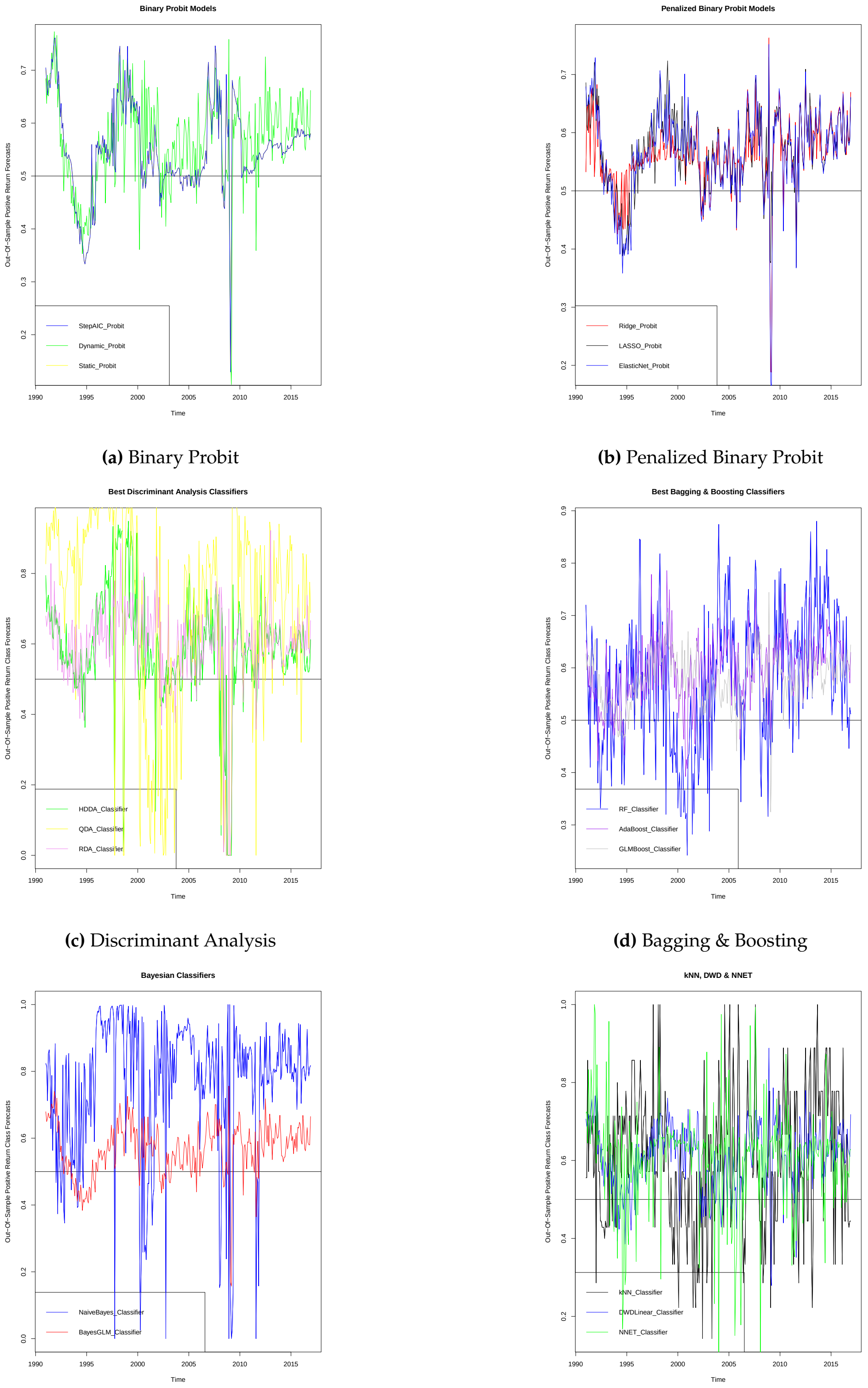
Again, the application of stepwise variable selection by AIC on the dynamic probit results in a slight increase in the predictive accuracy and the result equals the result of the stepwise

Table (3) Statistical Performance Evaluation Results

Model	CPR/Accuracy	MCE	PT	Precision	Specificity	Sensitivity	Kappa	McNemar	F_1 *Score*
Panel A: Binary Probit Models									
Static BP	0.561	0.439	-0.066	0.794	0.391	0.605	-0.003	0.000	0.686
Static BP StepAIC	0.596	0.404	1.533	0.820	0.477	0.628	0.079	0.000	0.711
Dynamic BP	0.580	0.420	0.794	0.810	0.452	0.617	0.014	0.000	0.700
Dynamic BP StepAIC	0.596	0.404	1.533	0.820	0.477	0.628	0.079	0.000	0.711
Panel B: Penalized Binary Probit Models									
Ridge BP	0.651	0.349	3.939***	0.947	0.706	0.644	0.163	0.000	0.767
LASSO BP	0.619	0.381	2.215*	0.894	0.545	0.631	0.101	0.000	0.740
Elastic Net BP	0.628	0.372	2.744**	0.894	0.574	0.638	0.127	0.000	0.744
Panel C: Bagging and Boosting Models									
Bagging	0.606	0.394	2.626**	0.735	0.500	0.656	0.147	0.047	0.693
RPart	0.612	0.388	1.674*	0.621	0.138	0.921	0.068	0.000	0.742
RF	0.654	0.346	4.339***	0.681	0.577	0.804	**0.240**	0.001	0.738
CTree	0.609	0.391	0.970	0.984	0.571	0.610	0.020	0.000	0.753
CForest	0.612	0.388	1.465	0.610	0.024	0.995	0.023	0.000	0.756
AdaBoost	0.644	0.356	3.668***	0.963	0.731	0.636	0.136	0.000	0.766
LogitBoost	0.583	0.417	1.294	0.627	0.293	0.773	0.070	0.000	0.692
GBM	0.615	0.385	2.001*	0.899	0.537	0.627	0.089	0.000	0.739
GLMBoost	0.625	0.375	2.408**	0.952	0.625	0.625	0.086	0.000	0.755
Panel D: Nearest Neighbour Model									
kNN	0.628	0.372	3.219***	0.799	0.542	0.659	0.175	0.000	0.722
Panel E: Neural Networks Model									
NNET	0.593	0.407	0.555	0.619	0.195	0.852	0.052	0.000	0.717
LVQ	0.606	0.394	2.626**	0.624	0.187	0.878	0.073	0.000	0.730
Panel F: Bayesian Models									
Bayes GLM	0.603	0.397	1.739*	0.836	0.492	0.629	0.088	0.000	0.718
Naive Bayes	0.660	0.340	4.402***	0.937	0.707	0.653	0.195	0.000	0.770
Panel G: Discriminant Analysis Models									
LDA	0.558	0.442	-0.241	0.794	0.381	0.602	-0.012	0.000	0.685
Sparse LDA	0.603	0.397	1.533	0.862	0.490	0.625	0.073	0.000	0.724
Step LDA	0.603	0.397	0.663	0.952	0.471	0.610	0.021	0.000	0.744
HDA	0.580	0.420	0.407	0.847	0.420	0.611	0.019	0.000	0.710
HDDA	**0.670**	**0.330**	5.244***	0.894	0.667	0.671	**0.241**	0.000	0.766
QDA	0.654	0.346	4.131***	0.857	0.609	0.667	**0.215**	0.000	0.750
Step QDA	0.603	0.397	0.663	0.952	0.471	0.610	0.021	0.000	0.744
RDA	0.636	0.365	3.015**	0.926	0.622	0.636	0.029	0.000	0.759
DWD Linear	0.609	0.391	1.581	0.905	0.514	0.622	0.067	0.000	0.737

static binary probit. The analysis of the static and dynamic binary probit models revealed that a parsimonious approach is preferable to incorporating many predictors in the models. The replication of the static and dynamic binary probit models used in the previous findings, as shown in the existing literature, such as in Nyberg (2011), had confirmed the feasibility of these models for excess stock return directional predictability. Interestingly, the empirical analysis of the static and dynamic binary probit models in this paper produced predictive accuracy (CPRs) equivalent to the CPRs of these models demonstrated by Nyberg (2008), Nyberg (2011) and investigate other important statistical performance measures, such as the kappa statistic, which determines inter-rater agreement between the actual results and the forecasts, and the McNemar's test for the detection of marginal homogeneity or equiprobability.

Turning to penalized binary probit models, the inclusion of penalty vector norm(s) in the ordinary binary probit models revealed a good improvement in predictive performance of the models. Specifically, the ridge, LASSO and elastic net provide higher predictive accuracy, which outperformed the benchmark binary probit models, with Ridge being statistically significant at 0.1%, EN at 1% and LASSO at 5%, with better inter-rater agreement between the



(a) Binary Probit

(b) Penalized Binary Probit

(c) Discriminant Analysis

(d) Bagging & Boosting

(e) Bayesian

(f) kNN, DWD & NNET

Figure (1) Graphical Representation of the Out-of-Sample Positive Class Return Forecasts actual results and the forecasts, as judged by the kappa statistic, and McNemar's p_{value} evidence of marginal heterogeneity. The penalized probit models also produced better precision, specificity, sensitivity and F_1 scores compared to the ordinary probit models. The ridge produced a better predictive accuracy and other statistical performance evaluation measures than the LASSO and elastic net, outperforming both the LASSO and the elastic net in this direction. Overall, the presence of the ℓ_1 and ℓ_2 penalty vector norms in the binary probit models appeared to improve the predictive task and the overall performance of the resulting models.

The models employed for forecasting the direction of the U.S. stock market in this paper demonstrate both the feasibility of the models and significant evidence of outperformance over the benchmark probit models. With the exception of LogitBoost, neural networks, LDA and HDA, all other classifiers in this paper are shown to have outperformed the benchmark binary probit models by statistical performance evaluation measures. In more detail, seven of the proposed methods outperform the benchmark probit models at 0.1%, five methods at 1% and four methods at 5%. It is noticeable that the introduction of the stepwise variable selection concept in the LDA model improved the predictive task and the resulting statistical performance of the LDA model, whereas the introduction of the stepwise concept in the QDA model worsened the predictive task and overall performance of the QDA model. The empirical analysis in this paper confirmed the superior outperformance of random forest (RF) over other forest based classification models in financial analysis, as shown in Ballings et al. (2015). Bagging and boosting, as demonstrated by Zheng (2006) in other aspects of stock market analysis, also appeared to have outperformed the neural networks in this paper. Unlike the benchmark binary probit models, the three sophisticated machine learning classification models, i.e., random forest, HDDA and QDA, provide fair inter-rater agreement between the actual results and the forecasts, as shown by their respective kappa statistic. The HDDA appeared to produce the best out of sample statistical performance evaluation results, followed by naive Bayes, and the QDA, with significant evidence of outperformance. Overall, the HDDA is the best model for predicting the direction of the U.S stock market in terms of statistical measures of predictability.

4.3 The Economic Performance Evaluation Results

As in the statistical case, the economic performance evaluation results, presented in Table 4, also revealed that the dynamic binary probit model produced better cumulative returns, Sharpe ratio (SR), MaxDD, Omega, Sortino Ratio and Upside Potential than the static binary probit, and the stepwise variable selection cases by AIC, in each case, appeared to yield even better economic evaluation results than the ordinary case. The penalized binary probit models (ridge, LASSO and elastic net) produced better cumulative returns, SR, MaxDD, Omega, Sortino and Upside than the ordinary binary probit models and, hence, demonstrate economically significant evidence of outperformance over the benchmark binary probit. For example ridge has SR equal to 0.642 while the static probit has SR equal to 0.271. Interestingly, all the penalized binary probit models outperformed the benchmark static and dynamic binary probit models in this paper. Again, the ridge outperformed the LASSO and elastic net in terms of the economic significance measure and seems to provide better economic information on future investment outcomes to a stock market investor than the LASSO and elastic net. All the CART models that are shown to be promising in terms of statistical predictability in this paper are also shown to be promising in terms of economic significance to portfolio investors. The effectiveness of Bagging (Bootstrap Aggregating), Boosting, Trees, Forests, Naive Bayes Discriminant Analysis models and other ensembles that were demonstrated to be useful in other concepts of financial analysis are also shown to be useful in forecasting the direction of the U.S. excess stock market returns and providing portfolio investors with better economic significance about the future outcome of investments in the stock market. The Random Forest method produced the highest SR (0.643) among the bagging and boosting models and by far greater than the static probit model (0.271). It is worth noting that a best performing model in terms of the statistical measure may not necessarily reflect the best performance in the economic significance measure.

Contrary to the statistical performance analysis, the HDDA does not correspondingly provide the best economic performance result; instead the QDA produced the highest cumulative return, SR, Omega, Sortino and Upside with a corresponding least MaxDD. QDA gives SR equal to 1.077 (almost four times the SR of static probit), while HDDA produces SR equal to 0.831. Although the HDDA also demonstrates good evidence of economic significance and appeared to have outperformed the other models in terms of some useful economic performance evaluation measures, another suitable benchmark comparative measure of economic significance on portfolio investment by investors is to compare the expected return on portfolio investment produced by the model with a buy and hold trading strategy of the SP500 index. In this case, we see that the simple probit models do not outperformed the buy and hold strategy. However, the penalized probit models and the prominent CART models (for example, HDDA, QDA, RDA and Naive Bayes) outperformed the buy and hold strategy, providing higher risk-adjusted returns.

Interestingly, the prominent CART models used in this paper have economically outperformed the benchmark binary probit models and the buy and hold trading strategy with a significant margin. Overall, the QDA appeared to be the best economically significant model for forecasting the direction of the U.S. stock market out of sample.

Table (4) Economic Performance Evaluation Results

Model	CumRet	ER	SD	SR	VaR0.05	CVaR0.05	MaxDD	Omega	Sortino	Upside Potential
Buy & Hold Strategy	1.914	0.074	0.144	0.326	-0.230	-0.342	34.655	1.274	0.130	0.594
Panel A: Binary Probit Models										
Static BP	1.576	0.061	0.125	0.271	-0.215	-0.308	38.973	1.256	0.107	0.525
Static BP StepAIC	2.145	0.083	0.130	0.430	-0.215	-0.308	20.784	1.436	0.177	0.582
Dynamic BP	2.012	0.078	0.126	0.428	-0.215	-0.308	26.342	1.374	0.163	0.574
Dynamic BP StepAIC	2.145	0.083	0.130	0.430	-0.215	-0.308	20.784	1.436	0.177	0.582
Panel B: Penalized Binary Probit Models										
Ridge BP	2.796	0.108	0.126	0.642	-0.181	-0.284	15.556	1.669	0.274	0.683
LASSO BP	2.459	0.095	0.130	0.524	-0.204	-0.301	17.362	1.526	0.219	0.630
Elastic Net BP	2.533	0.097	0.127	0.559	-0.189	-0.289	16.755	1.574	0.236	0.647
Panel C: Bagging and Boosting Models										
Bagging	2.137	0.082	0.109	0.511	-0.189	-0.253	13.475	1.563	0.223	0.618
RPart	1.655	0.064	0.140	0.265	-0.229	-0.338	38.932	1.231	0.105	0.559
RF	2.605	0.100	0.114	0.643	-0.168	-0.266	9.256	1.737	0.285	0.672
CTree	1.986	0.076	0.143	0.348	-0.229	-0.341	33.527	1.305	0.139	0.594
CForest	1.973	0.076	0.144	0.342	-0.230	-0.342	33.720	1.296	0.136	0.598
AdaBoost	2.383	0.092	0.137	0.475	-0.219	-0.319	21.728	1.450	0.195	0.628
LogitBoost	1.762	0.068	0.124	0.332	-0.201	-0.296	32.488	1.318	0.138	0.570
GBM	2.112	0.081	0.131	0.418	-0.218	-0.295	27.367	1.392	0.175	0.620
GLMBoost	2.161	0.083	0.138	0.409	-0.224	-0.326	27.669	1.376	0.166	0.607
Panel D: Nearest Neighbour Model										
kNN	2.266	0.087	0.124	0.487	-0.179	-0.298	17.423	1.519	0.203	0.594
Panel E: Neural Networks Models										
NNET	2.004	0.077	0.127	0.396	-0.218	-0.290	27.504	1.367	0.165	0.612
LVQ	1.861	0.072	0.143	0.314	-0.230	-0.342	34.198	1.270	0.124	0.585
Panel F: Bayesian Models										
Bayes GLM	2.230	0.086	0.124	0.477	-0.189	-0.289	21.103	1.487	0.198	0.603
Naive Bayes	2.807	0.108	0.125	0.653	-0.191	-0.278	15.844	1.673	0.281	0.699
Panel G: Discriminant Analysis Models										
LDA	1.481	0.057	0.126	0.240	-0.218	-0.311	38.973	1.223	0.094	0.515
Sparse LDA	1.972	0.076	0.130	0.380	-0.219	-0.309	27.066	1.361	0.152	0.575
Step LDA	2.007	0.077	0.142	0.356	-0.229	-0.338	33.202	1.315	0.143	0.595
HDA	1.927	0.074	0.127	0.375	-0.201	-0.308	25.157	1.360	0.151	0.569
HDDA	3.090	0.119	0.111	0.831	-0.162	-0.225	16.430	1.935	0.396	0.820
QDA	**3.517**	0.135	0.101	**1.077**	-0.125	-0.168	**6.072**	**2.378**	**0.598**	**1.032**
Step QDA	1.863	0.072	0.141	0.319	-0.229	-0.338	34.156	1.281	0.127	0.579
RDA	2.499	0.096	0.124	0.561	-0.204	-0.277	31.126	1.552	0.240	0.676
DWD Linear	1.921	0.074	0.133	0.355	-0.223	-0.322	37.314	1.331	0.141	0.569

5 Conclusion

The analysis of the benchmark binary probit models in this paper corroborates the empirical findings in previous studies, especially in Nyberg (2008), Nyberg (2011). In this paper, additional statistical and economic performance evaluation measures were introduced to investigate the long-run usefulness of these models in the financial stock market.

The empirical analysis in this paper revealed that the proposed sophisticated machine learning techniques outperformed the benchmark binary probit models both statistically and economically. In terms of the statistical predictive accuracy, the best penalized binary probit model outperformed the best binary probit model by 5.5% and the best CART model outperformed the best binary probit model by 7.4%.

In terms of statistical performance evaluation measures, the HDDA appeared to be the best model for forecasting the direction of the U.S stock market in this paper, owing to its highest predictive accuracy with minimum misclassification error (MCE) and other resulting statistical measures. Adding to the previous analysis in the existing financial and econometric literature, the Kappa statistic was used in this paper to investigate the inter-rater agreement between the actual values and forecasts produced by the various models. The Kappa statistic revealed that there is no inter-rater agreement between the actual values and the forecasts obtained by the static and the dynamic binary probit models. Interestingly, the RF, QDA and HDDA proposed in this paper provide evidence of fair inter-rater agreement between the actual values and the forecasts produced by the models. However, the QDA appeared to be the best model in terms of the measures of economic significance in this paper. The QDA seems to provide more economic value to guarantee the success of a portfolio manager in the stock market than the other models used in this paper.

Overall, the HDDA is the best model for forecasting the direction of the U.S stock market out of sample in terms of statistical predictability measures, while the QDA is the best economically significant model for a portfolio investor whose utmost goal is to minimise risk and maximize profit, based on the empirical analytical findings in this paper.

References

Alfaro, E., Gamez, M., Garcia, N., et al. (2013). Adabag: An R package for classification with boosting and bagging. *Journal of Statistical Software*, 54(2):1–35.

Anatolyev, S. and Gospodinov, N. (2010). Modeling financial return dynamics via decomposition. *Journal of Business & Economic Statistics*, 28(2):232–245.

Ballings, M., Van den Poel, D., Hespeels, N., and Gryp, R. (2015). Evaluating multiple classifiers for stock price direction prediction. *Expert Systems with Applications*, 42(20):7046–7056.

Bergmeir, C., Costantini, M., and Benítez, J. M. (2014). On the usefulness of cross-validation for directional forecast evaluation. *Computational Statistics & Data Analysis*, 76:132–143.

Bouveyron, C., Girard, S., and Schmid, C. (2007). High-dimensional discriminant analysis. *Communications in StatisticsTheory and Methods*, 36(14):2607–2623.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

Breiman, L. (2001). Random forests. *Machine Learning*, 45:5–32.

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). *Classification and Regression Trees*. CRC press.

Bühlmann, P. et al. (2006). Boosting for high-dimensional linear models. *The Annals of Statistics*, 34(2):559–583.

Bühlmann, P., Hothorn, T., et al. (2007). Boosting algorithms: Regularization, prediction and model fitting. *Statistical Science*, 22(4):477–505.

Bühlmann, P. and Yu, B. (2003). Boosting with the l 2 loss: regression and classification. *Journal of the American Statistical Association*, 98(462):324–339.

Caudill, M. (1989). Neural networks primer, part viii. *AI Expert*, 4(8):61–67.

Chen, J. (2016). The Chen-Tindall system and the LASSO operator: improving automatic model performance. *Federal Reserve Bank of Dallas, Occasional Paper*, 16:01.

Chevapatrakul, T. (2013). Return sign forecasts based on conditional risk: Evidence from the UK stock market index. *Journal of Banking & Finance*, 37(7):2342–2353.

Christoffersen, P. F. and Diebold, F. X. (2006). Financial asset returns, direction-of-change forecasting, and volatility dynamics. *Management Science*, 52(8):1273–1287.

Clemmensen, L., Hastie, T., Witten, D., and Ersbøll, B. (2011). Sparse discriminant analysis. *Technometrics*, 53(4):406–413.

Cook, E. F. and Goldman, L. (1984). Empiric comparison of multivariate analytic techniques: advantages and disadvantages of recursive partitioning analysis. *Journal of Chronic Diseases*, 37(9):721–731.

Das, A., Abdel-Aty, M., and Pande, A. (2009). Using conditional inference forests to identify the factors affecting crash severity on arterial corridors. *Journal of Safety Research*, 40(4):317–327.

de Oliveira, F. A., Nobre, C. N., and Zárate, L. E. (2013). Applying artificial neural networks to prediction of stock price and improvement of the directional prediction index–case study of PETR4, Petrobras, Brazil. *Expert Systems with Applications*, 40(18):7596–7606.

Ding, T., Fang, V., and Zuo, D. (2013). Stock market prediction based on time series data and market sentiment. *Working Paper*.

Estrella, A. and Mishkin, F. S. (1998). Predicting US recessions: Financial variables as leading indicators. *Review of Economics and Statistics*, 80(1):45–61.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188.

Freund, Y. and Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *In: Thirteenth International Conference on ML*. Citeseer.

Friedman, J., Hastie, T., Tibshirani, R., et al. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The Annals of Statistics*, 28(2):337–407.

Friedman, J. H. (1989). Regularized discriminant analysis. *Journal of the American Statistical Association*, 84(405):165–175.

Goyal, A. (2018). A comprehensive look at the empirical performance of equity premium prediction: Updated dataset. ”http://www.hec.unil.ch/agoyal/”.

Goyal, A. and Welch, I. (2008). A comprehensive look at the empirical performance of equity premium prediction. *Review of Financial Studies*, 21:1455–1508.

Granger, C. W. and Pesaran, M. H. (2000). Economic and statistical measures of forecast accuracy. *Journal of Forecasting*, 19(7):537–560.

Hajek, P., Olej, V., and Myskova, R. (2014). Forecasting corporate financial performance using sentiment in annual reports for stakeholders decision-making. *Technological and Economic Development of Economy*, 20(4):721–738.

Harlow, W. V. and Rao, R. K. (1989). Asset pricing in a generalized mean-lower partial moment framework: Theory and evidence. *Journal of Financial and Quantitative Analysis*, 24(3):285–311.

Hastie, T. (2005). *Neural Network, Encyclopedia of Biostatistics*. Wiley.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Applications to nonorthogonal problems. *Technometrics*, 12(1):69–82.

Hofner, B., Mayr, A., Robinzonov, N., and Schmid, M. (2014). Model-based boosting in R: A hands-on tutorial using the R package mboost. *Computational Statistics*, 29(1-2):3–35.

Hong, Y. and Chung, J. (2003). Are the directions of stock price changes predictable? Statistical theory and evidence. *Working Paper, Cornell University*.

Hothorn, T., Hornik, K., and Zeileis, A. (2006). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, 15(3):651–674.

Hsu, N.-J., Hung, H.-L., and Chang, Y.-M. (2008). Subset selection for vector autoregressive processes using LASSO. *Computational Statistics & Data Analysis*, 52(7):3645–3657.

Huang, C.-J., Yang, D.-X., and Chuang, Y.-T. (2008). Application of wrapper approach and composite classifier to the stock trend prediction. *Expert Systems with Applications*, 34(4):2870–2878.

Huang, H., Lu, X., Liu, Y., Haaland, P., and Marron, J. (2012). R/DWD: distance-weighted discrimination for classification, visualization and batch adjustment. *Bioinformatics*, 28(8):1182–1183.

Huang, J., Kingsbury, B., Mangu, L., Padmanabhan, M., Saon, G., and Zweig, G. (2000). Performance improvement in voicemail transcription. In *Proceedings of DARPA Speech Transcription Workshop*. Citeseer.

Inoue, A. and Kilian, L. (2008). How useful is bagging in forecasting economic time series? A case study of US consumer price inflation. *Journal of the American Statistical Association*, 103(482):511–522.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning*, volume 6. Springer.

Kauppi, H. and Saikkonen, P. (2008). Predicting US recessions with dynamic binary response models. *The Review of Economics and Statistics*, 90(4):777–791.

Keating, C. and Shadwick, W. F. (2002). A universal performance measure. *Journal of performance measurement*, 6(3):59–84.

Khaidem, L., Saha, S., and Dey, S. R. (2016). Predicting the direction of stock market prices using random forest. *Working Paper*.

Kim, H. H. and Swanson, N. R. (2014). Forecasting financial and macroeconomic variables using data reduction methods: New empirical evidence. *Journal of Econometrics*, 178:352–367.

Kohonen, T. (1995). Learning vector quantization. In *Self-Organizing Maps*, pages 175–189. Springer.

Kumar, M. and Thenmozhi, M. (2006). Forecasting stock index movement: A comparison of support vector machines and random forest.

Kumar, N. and Andreou, A. G. (1998). Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition. *Speech* C*ommunication*, 26(4):283–297.

Landwehr, N., Hall, M., and Frank, E. (2005). Logistic model trees. *Machine Learning*, 59(1-2):161–205.

Leitch, G. and Tanner, J. E. (1991). Economic forecast evaluation: Profits versus the conventional error measures. *The American Economic Review*, pages 580–590.

Lemmens, A. and Croux, C. (2006). Bagging and boosting classification trees to predict churn. *Journal of Marketing Research*, 43(2):276–286.

Leung, M. T., Daouk, H., and Chen, A.-S. (2000). Forecasting stock indices: A comparison of classification and level estimation models. *International Journal of Forecasting*, 16(2):173–190.

Li, J. and Chen, W. (2014). Forecasting macroeconomic time series: LASSO-based approaches and their forecast combinations with dynamic factor models. *International Journal of Forecasting*, 30(4):996–1015.

Lin, F. Y. and McClean, S. (2001). A data mining approach to the prediction of corporate failure. *Knowledge-Based Systems*, 14(3):189–195.

Marron, J. S., Todd, M. J., and Ahn, J. (2007). Distance-weighted discrimination. *Journal of the American Statistical Association*, 102(480):1267–1271.

McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Meligkotsidou, L., Panopoulou, E., Vrontos, I. D., and Vrontos, S. D. (2014). A quantile regression approach to equity premium prediction. *Journal of Forecasting*, 33(7):558–576.

Meligkotsidou, L., Panopoulou, E., Vrontos, I. D., and Vrontos, S. D. (2019). Out-of-sample equity premium prediction: A complete subset quantile regression approach. *The European Journal of Finance*, pages 1–26.

Moreno, D. and Olmeda, I. (2007). Is the predictability of emerging and developed stock markets really exploitable? *European Journal of Operational Research*, 182(1):436–454.

Nyberg, H. (2008). Forecasting the direction of the US stock market with dynamic binary probit models. *Discussion Paper*, (27).

Nyberg, H. (2011). Forecasting the direction of the US stock market with dynamic binary probit models. *International Journal of Forecasting*, 27(2):561–578.

Nyberg, H. (2013). Predicting bear and bull stock markets with dynamic binary time series models. *Journal of Banking & Finance*, 37(9):3351–3363.

Nyberg, H. and Pönkä, H. (2016). International sign predictability of stock returns: The role of the United States. *Economic Modelling*, 58:323–338.

Ou, P. and Wang, H. (2009). Prediction of stock market index movement by ten data mining techniques. *Modern Applied Science*, 3(12):28.

Pahwa, N. S., Khalfay, N., Soni, V., and Vora, D. (2017). Stock prediction using machine learning: A Review Paper. *International Journal of Computer Applications*, 163(5).

Park, H. and Sakaori, F. (2013). Lag weighted LASSO for time series model. *Computational Statistics*, 28(2):493–504.

Patel, J., Shah, S., Thakkar, P., and Kotecha, K. (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications*, 42(1):259–268.

Pesaran, M. H. (2015). *Time series and panel data econometrics*. Oxford University Press.

Pesaran, M. H. and Timmermann, A. (1992). A simple nonparametric test of predictive performance. *Journal of Business & Economic Statistics*, 10(4):461–465.

Pesaran, M. H. and Timmermann, A. (1995). Predictability of stock returns: Robustness and economic significance. *The Journal of Finance*, 50(4):1201–1228.

Pönkä, H. (2016). Real oil prices and the international sign predictability of stock returns. *Finance Research Letters*, 17:79–87.

Qiao, X. and Zhang, L. (2015). Distance-weighted support vector machine. *Stat*, 1050:8.

Rapach, D. E., Strauss, J. K., and Zhou, G. (2010). Out-of-sample equity premium prediction: Combination forecasts and links to the real economy. *The Review of Financial Studies*, 23(2):821–862.

Ripley, B. D. (1996). *Pattern recognition and neural networks*. Cambridge university press.

Roy, S. S., Mittal, D., Basu, A., and Abraham, A. (2015). Stock market forecasting using LASSO linear regression model. In *Afro-European Conference for Industrial Advancement*, pages 371–381. Springer.

Sermpinis, G., Tsoukas, S., and Zhang, P. (2017). Modelling market implied ratings using LASSO variable selection techniques.

Shahpazov, V., Doukovska, L., and Karastoyanov, D. (2014). Artificial intelligence neural networks applications in forecasting financial markets and stock prices. In *Proc. of the International Symposium on Business Modeling and Software Design–BMSD*, volume 14, pages 24–26.

Shen, W., Wang, J., and Ma, S. (2014). Doubly regularized portfolio with risk minimization. In *AAAI*, pages 1286–1292.

Sortino, F. A. and Price, L. N. (1994). Performance measurement in a downside risk framework. *The Journal of Investing*, 3(3):59–64.

Sortino, F. A. and Van Der Meer, R. (1991). Downside risk. *Journal of Portfolio Management*, 17(4):27–31.

Stock, J. H. and Watson, M. W. (2012). Generalized shrinkage methods for forecasting using many predictors. *Journal of Business & Economic Statistics*, 30(4):481–493.

Strobl, C., Boulesteix, A.-L., Kneib, T., Augustin, T., and Zeileis, A. (2008). Conditional variable importance for random forests. *BMC Bioinformatics*, 9(1):307.

Su, M.-Y. (2011). Using clustering to improve the kNN-based classifiers for online anomaly network traffic identification. *Journal of Network and Computer Applications*, 34(2):722–730.

Swanson, N. R. and White, H. (1997). A model selection approach to real-time macroeconomic forecasting using linear models and artificial neural networks. *The Review of Economics and Statistics*, 79(4):540–550.

Szepannek, G., Harczos, T., Klefenz, F., and Weihs, C. (2009). Extending features for automatic speech recognition by means of auditory modelling. In *2009 17th European Signal Processing Conference*, pages 1235–1239. IEEE.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288.

Zeileis, A., Hothorn, T., and Hornik, K. (2008). Model-based recursive partitioning. *Journal of Computational and Graphical Statistics*, 17(2):492–514.

Zheng, Z. (2006). Boosting and bagging of neural networks with applications to financial time series. Technical report, Working paper, Department of Statistics, University of Chicago, Tech. Rep.

Zhou, L., Lu, D., and Fujita, H. (2015). The performance of corporate financial distress prediction models with features selection guided by domain knowledge and data mining approaches. *Knowledge-Based Systems*, 85:52–61.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B*, 67(2):301–320.