Bi-objective Cost-sensitive Machine Learning: Predicting Stock Return Direction Using Option Prices*

Robert James† Jessica Wai Yin Leung‡ Artem Prokhorov§

August 2023

Abstract

This paper studies cost-sensitive loss functions for training machine learning models that predict the direction of future equity market index movement. In particular we design a bi-objective loss function that combines the log-loss with a second objective which asymmetrically penalizes individual false-positive and false-negative misclassification errors. We argue that put and call option prices are natural measures of the misclassification costs. The bi-objective optimization framework permits us to isolate the effect of cost-sensitivity and to study the information channels connecting the spot and options markets. Using a comprehensive suite of classification performance metrics we investigate how training an elastic-net logistic model and a non-linear gradient boosting model with cost-sensitive loss functions improves predictions of return direction. A long/short investment strategy that uses the predictions from our cost-sensitive models improves risk-adjusted investment performance and reduces downside risk.

JEL Codes: C53, C58
Keywords: Return Prediction, Machine Learning, Gradient Boosting, Bi-objective Optimization

*Helpful comments from the participants of the International Symposium on Forecasting 2023 are gratefully acknowledged. The use of the University of Sydney's high performance computing cluster, Artemis, is acknowledged. This research was supported in part by the Monash eResearch Centre and eSolutions-Research Support Services through the use of the MonARCH HPC Cluster.

†The University of Sydney Business School; email: r.james@sydney.edu.au

‡Department of Econometrics and Business Statistics, Monash University, Melbourne, VIC 3800; email: jessica.leung@monash.edu

§The University of Sydney Business School & CEBA & CIREQ; email: artem.prokhorov@sydney.edu.au

1 Introduction

Forecasting the direction of stock returns is an important elementary task in asset management. For example, predictions about the direction of future stock returns can inform short and medium-term asset allocation decisions or can be used as a direct input into signal-driven market timing strategies. Accurate forecasts allow investment managers to better allocate scarce resources and improve economic welfare. Recently, machine learning approaches to modeling the time series of stock return movements have become increasingly popular (see, e.g., Fischer and Krauss, 2018; Iworiso and Vrontos, 2020; Mascio et al., 2021). These machine learning models often outperform traditional econometric models or trend-following methods in terms of their predictive accuracy (Kelly et al., 2022). However, prior literature has predominantly applied off-the-shelf binary machine learning algorithms to the prediction task. These standard algorithms minimize objective functions that do not necessarily recognize the dynamic and asymmetric nature of financial markets. For example, traditional binary machine learning algorithms that minimize the log-loss objective function assume that the cost of making a false-positive or a false-negative classification error is constant and equal. In practice, this assumption is unlikely to hold, since time-varying market volatility and dynamic market regimes mean that investor preferences about prediction errors are likely to change over the business cycle.

In this paper, we propose to train binary machine learning models that predict the future return direction using cost-sensitive learning and bi-objective optimization. Specifically, we test two objective functions namely, the average expected cost function which originates from the field of cost-sensitive example-dependent learning, and a novel bi-objective loss function that augments conventional log-loss optimization with the average expected cost objective. The key feature of the average expected cost is that it asymmetrically penalizes individual false-positive and false-negative misclassification errors. Our bi-objective function reflects the idea that a practical financial forecasting model for the equity market return direction should target both classification accuracy and the dynamics of misclassification costs when making decisions. We expect that using an equally-weighted combination of the log-loss and average expected cost will produce models that generalize better to unseen data[1].

As measures of the misclassification costs we use at-the-money put and call option prices. These are natural measures of forward looking costs because they facilitate hedging and embed the market's expectations about future returns, investor preferences, and volatility. Moreover, informed trader's are attracted to option markets because of the implicit leverage and trading cost advantages that are embedded in these contracts (Kacperczyk and Pagnotta, 2019; An et al., 2014; Lin and Lu, 2015). For example, an investor who expects positive returns on an equity market index over the next month could buy a call option. In the event that the investor is wrong, that is the investor makes a false-positive prediction error, and the option expires out-of-the money the investor's misclassification costis the option price. By incorporating average expected options based miss-classification costs into the learning process we impose a degree of economic structure related to prevailing costs of a binary decision implied by the market.

We consider two types of machine learning models to predict the time series of stock return movements, namely the elastic-net regularized logistic regression and gradient boosting using the LightGBM framework (Ke et al., 2017). To prevent over-fitting noisy financial returns we use a robust cross-validation procedure that eliminates information leakage between the training and testing sets. Both our machine learning models include L^1 and L^2-regularization. Moreover, we implement stochastic gradient boosting and also use the DART framework of (Vinayak and Gilad-Bachrach, 2015) to reduce the likelihood that base learners added in the subsequent boosting iterations overfit.

Our decision to study logistic regression models is motivated by the observation that they remain a popular choice both within the industry and in academic literature (see, e.g., Mascio et al., 2021). Elastic-net logistic regressions are fast, interpretable, and often used as benchmarks. More recently, a growing

[1]Similar arguments have been put forward in the forecast combination literature (see, e.g., Timmermann, 2006; Wang et al., 2022)

literature demonstrates the utility of non-linear ensemble machine learning algorithms for solving financial machine learning tasks (Rasekhschaffe and Jones, 2019; Christensen et al., 2021; Krauss et al., 2017). Within this class of models, algorithms that use ensembles of decision trees, such as gradient boosting have emerged as popular choices, because these non-parametric models can naturally identify non-linear relationships and model complex higher order interactions among predictor variables. We are particularly interested in studying how the inclusion of average expected costs in a bi-objective optimization framework improves gradient boosting models. From a practical perspective, our approach is attractive because we only require at-the-money put and call option prices which can be easily obtained, in contrast to the entire spectrum of option prices across different strikes[2].

We evaluate the out-of-sample one-month ahead and 10-day ahead return direction forecasts for the S&P500, NASDAQ100 and Dow Jones Industrial Average stock market indices using a wide range of classification performance metrics and return statistics. To ensure the robustness of our results we also examine return direction forecasts for a sample of 24 individual U.S. equities. We find that our cost-sensitive models correctly predict a larger fraction of positive future returns,compared to using only the log-loss objective.

For the sample of individual equities we find that both precision and recall are higher for our bi-objective cost-sensitive models, demonstrating that they correctly predict more days where the future returns are positive with a higher accuracy as compared to the benchmark models. Incorporating cost-sensitive learning with option price based misclassification costs disproportionately improves predictions about positive future returns. Importantly, our results are robust to benchmark models that incorporate information about option prices and implied volatility information directly in the set of predictors, without cost sensitivity. At the same time, the bi-objective models tend to outperform their counterparts that use only average expected cost as the objective. Therefore, we find that designing multi-objective functions for financial machine learning models improves predictive performance.

To assess the economic value of our cost-sensitive machine learning models we backtest a long/short investment strategy that trades an equally-weighted portfolio of the S&P500, NASDAQ100 and Dow Jones Industrial Average indices based upon the return sign predictions. We find that our bi-objective models earn a higher Sharpe ratio than the benchmark models net of transaction costs. Hence, cost-sensitive learning from option prices is a novel way of generating superior risk adjusted returns. Moreover, strategies that use our cost-sensitive models have a lower downside risk. Specifically, relative to their respective benchmarks, strategies associated with our models tend to have a smaller maximum drawdown and a shorter maximum drawdown period, and higher Sortino ratios underpinned by higher annualized returns and lower downside deviations. These results remain robust to a long-only strategy and an alternative portfolio construction method where weights are based upon the predicted probabilities, rather than set equal to one another.

Our study is related to the literature that designs machine learning models for empirical asset pricing tasks, for example, to predict equity market returns (Basak et al., 2019; Fischer and Krauss, 2018; Iworiso and Vrontos, 2020; Mascio et al., 2021). In particular, Kelly et al. (2022) both theoretically and empirically demonstrate that machine learning models can outperform traditional approaches to predict equity market returns. We are interested in the subset of this literature that predicts return directions, which is a binary classification task (see, e.g., Fischer and Krauss, 2018; Iworiso and Vrontos, 2020; Mascio et al., 2021). Our contribution is to demonstrate that augmenting the loss function to better suit the dynamics of the actual forecasting task at hand improves classification performance. Consequently, our study is also related to a recent literature which advocates for augmenting off-the-shelf machine learning algorithms with information about economic structure and objectives (see, e.g. Chen et al., 2023; Brogaard and Zareei, 2022; Jensen et al., 2022).

Using a rolling window estimation scheme and a training dataset of approximately 6-years of daily data We find that gradient boosting does not significantly outperform elastic-net logistic regression in terms of

[2]There is no guarantee that out-of-the-money or in-the-money option prices are available at a daily frequency with an expiry matching a desired investment horizon.

performance metrics. However, in a long/short investment strategy that uses the return sign forecasts, gradient boosting models earn higher Sharpe ratios. Consequently, we also add to a growing literature studying the behaviour of non-linear and ensemble machine learning models for financial prediction tasks (see, e.g., Gu et al., 2020; Rasekhschaffe and Jones, 2019; Christensen et al., 2021).

There is a sizeable literature studying the informational interaction between stock and option markets (see, e.g., Pan and Poteshman, 2006; Johnson and So, 2012; An et al., 2014). This literature identifies several channels by which option markets can provide information about future stock returns. We contribute by designing a novel application of the information in option prices, namely by incorporating it within cost-sensitive bi-objective prediction problem. More recently, Cremers et al. (2019) find that several option price based measures of stock mispricing can identify undervalued stocks and these errors can be exploited via a long-only portfolio. Our result that the average expected cost and bi-objective models outperform the benchmarks for positive future returns appears to be broadly consistent with this work.

The remainder of this paper is structured as follows. Section 2 describes the building blocks of our bi-objective function and discusses the practicalities of using it to train logistic regression and gradient-boosting trees. Section 3 describes the data, predictors, and design of our forecasting experiments. We discuss the classification performance in Section 4 and results from a long-short investment strategy in Section 5. Section 6 concludes.

2 A Cost-sensitive Learning Framework for Stock Return Direction Forecasting

2.1 Cost Sensitivity and Options Prices

In a binary classification problem we have data $D = f(y_t; \mathbf{x}_t)g_{t=1}^{T}$ where $y_t \in f0; 1g$ is the ground truth binary outcome and $\mathbf{x}_t \in \mathbb{R}^p$ is a p-dimensional vector of predictor variables on day t. Without loss of generality, we assume that the predictor data has been standardized to zero mean and unit variance. In our application $y_t = \mathbb{I}(\sum_{i=1}^{h} r_{t+i} > 0)$, where $r_t = (\log(P_{t+1}) \quad \log(P_t))$, P_t is the daily price/level of the equity market index at time t, $\mathbb{I}(\ )$ is the indicator function and h is the forecast horizon. That is, we are interested in whether or not the cumulative log return[3] over the next h-days is positive. We study $h = 10$ and $h = 30$ days.

We consider logistic regression (LR) and gradient boosting machines (GBM), two supervised binary classification algorithms which estimate the probability that an observation belongs to the positive class. Both algorithms compute a real-valued pre-activation score, denoted by $f(\mathbf{x}_t; \ )$, that is converted into a probability using the sigmoid function

$$p(y = 1|\mathbf{x}_t; \ ) = \frac{1}{1 + e^{\ f(\mathbf{x}_t; \ )}}; \tag{1}$$

where contains the model parameters. The class label is then determined by comparing this probability to a fixed threshold[4] of 0.5.

We study improvements over the traditional binary classifiers that can be achieved by using objective functions from the field of example-dependent cost-sensitive machine learning. To understand why this ap-

[3] Augmented Dickey-Fuller tests reject the null hypothesis that this cumulative return series has a unit root at the 1% level of significance for the equity market indices that we study.

[4] Cost sensitivity can also be imposed by changing the threshold used to assign class labels so that it reflects the relative difference in costs. However, we consider that incorporating costs directly into the objective function of a machine learning model is a more flexible way of obtaining a cost sensitive classifier. Moreover, our method permits the design of bi-objective functions that can be used to train a machine learning model. It is not clear how such a bi-objective learning approach could be incorporated into the decision threshold.

proach could improve classification, note that in binary classification, the traditional log-loss (cross-entropy) measures the difference between the predicted probability and the true class label, that is

$$L_1(\mathbf{y}; \mathbf{x}; \ ) = \sum_{t=1}^{T} \ell_1(y_t; f(\mathbf{x}_t; \ )) = \sum_{t=1}^{T} [y_t \log(p(y_t = 1 | \mathbf{x}_t; \ )) + (1 \quad y_t) \log(1 \quad p(y_t = 1 | \mathbf{x}_t; \ ))]; \quad (2)$$

where $(\mathbf{y}; \mathbf{x})$ contains all $(y_t; \mathbf{x}_t)$. The log-loss objective function penalizes both false-positive and false-negative classification errors equally. However, in many real-world settings, these costs are asymmetric and time-varying. A classic financial example is credit card fraud or transfers fraud detection (Höppner et al., 2022; Hand et al., 2008).

With this idea in mind, we consider objective functions that are designed to penalize observations based on their observation-specific misclassification cost of each of the two types of prediction errors. This learning framework is example-dependent and cost-sensitive in the sense that the algorithm explicitly incorporates the asymmetry of the observation-specific costs. This strand of literature advocates for using the average expected cost (AEC) function, rather than expected loss, as the objective to be minimized when training a machine learning model (Elkan, 2001; Bahnsen et al., 2015; Höppner et al., 2022). The AEC function is defined as

$$L_2(\mathbf{y}; \mathbf{x}; \ ) = \frac{1}{T} \sum_{t=1}^{T} \ell_2(y_t; f(\mathbf{x}_t; \ )) = \frac{1}{T} \sum_{t=1}^{T} y_t(1 \quad p_t(y_t = 1 | \mathbf{x}_t; \ ))c_t^{(fn)} + (1 \quad y_t)p_t(y_t = 1 | \mathbf{x}_t; \ )c_t^{(fp)}; \quad (3)$$

where $c_t^{(fn)}$ is the cost of making a false-negative error and $c_t^{(fp)}$ is the cost of making a false-positive error. Note that we allow these costs to vary for each observation at time t.

Our application of an example-dependent cost-sensitive objective function to the return direction prediction task recognizes that the cost of misclassification in financial forecasting is likely to be dynamic and asymmetric. For example, during periods of heightened volatility, misclassification costs could be significantly higher because we are more likely to observe larger price movements in both the positive and negative directions. Moreover, asymmetries in the misclassification cost are likely to change during market cycles, such as periods of financial market distress or bullish market rallies.

We consider two approaches to incorporate classification error costs into binary financial machine learning models. The first approach is to use the average expected cost function (L_2) directly as the objective to be minimized during the learning process. This is the standard approach within the cost-sensitive machine learning literature, although the application to stock return prediction tasks has not been explored yet. The second approach is to use a novel bi-objective function that is the equal weighted combination of the log-loss objective and the average expected cost objective. Our decision to combine these two objective functions is motivated by the idea that a practical return direction forecasting model should consider both predictive accuracy and the prevailing dynamics of costs derived from financial market estimates that are associated with the predicted binary decision. The decision to use the equal-weighted combination scheme is supported by the well-studied empirical results from the forecast combinations literature (see, e.g., Timmermann, 2006). Moreover, by using the equal-weighted combination we alleviate the need to further estimate optimal, and possibly time-varying, combination weights, reducing the likelihood that we over-fit noisy financial returns.

To combine the L_1 and L_2 objectives we use weighted-sum multi-objective optimization. This approach involves constructing a single objective function from the sum of the individual objective functions using user-defined weights u_1 and u_2. The weights express the relative importance of each objective as determined by the investor's preferences. Normalization of the individual objectives is important for ensuring consistency of the optimal solution in the weighted-sum method. For example, in our case the log-loss objective is a sum over T observations on the logarithmic scale while the AEC objective is the average over T observations. To ensure that the contribution of each individual objective to the weighted sum aligns with the user-defined

weights we introduce the normalization factors $_1$ and $_2$ so that the actual objective weights are computed as $w_1 = {}_1u_1$ and $w_2 = {}_2u_2$. While several normalization schemes exist, we follow the approach that uses the nadir and utopia points (see, e.g., Mausser, 2006).

Formally, the utopia points are the values obtained from minimizing each objective function individually,

$$s_i^U = \min L_i(\mathbf{y}; \mathbf{x}; \ ); \quad i \ 2 \ f1; 2g \tag{4}$$

The utopia points provide a lower bound on the values of the Pareto optimal set of solutions. The Nadir points are obtained as follows

$$s_i^N = \max_{j \ 2 \ f1;2g} L_i(\mathbf{y}; \mathbf{x}; \ _j) \tag{5}$$

where $L_i(\ ; \ ; \ _j)$ means that we evaluate the i^{th} objective function at the model parameters obtained by minimizing the j^{th} objective. The Nadir points represent an upper bound on the Pareto optimal set. Then, the normalization factors are given by

$$_i = \frac{1}{s_i^N \quad s_i^U} \tag{6}$$

Taken together, we propose to train bi-objective LR and GBM models using

$$L(\mathbf{y}; \mathbf{x}; \ ) = w_1 L_1(\mathbf{y}; \mathbf{x}; \ ) + w_2 L_2(\mathbf{y}; \mathbf{x}; \ ) \tag{7}$$

We would like to use measures of misclassification costs that are forward-looking and match the desired investment horizon. To do so, we use standardized at-the-money put and call option premiums (prices) as such costs. Since options are contingent claims on the underlying asset, they provide forward-looking information about the market's expectations of future price movements of the asset. Moreover, asymmetries in misclassification costs may reflect the market's expectations about risk and returns. To understand why option prices are a relevant measure of misclassification costs we consider a simple case, where an investor who thinks that the return on the S&P500 index will be positive over the next month buys a call option written on the index. If the investor is wrong then the option will finish out-of-the-money and the cost of this prediction error is the call option price. Likewise, if an investor thinks returns will be negative they can buy a put option, and the misclassification cost will be limited by the put option price.

Prior literature identifies several channels by which options markets may be informative about future price movements in the underlying asset. First and foremost, a wealth of literature documents that informed traders transact in options markets because of the implicit leverage embedded in the contracts (see, e.g. Roll et al., 2010; Lin and Lu, 2015; Li et al., 2017; Augustin and Subrahmanyam, 2020; An et al., 2014). This option-based leverage is often significantly larger than the leverage that could be gained from buying a stock on margin (Cremers et al., 2019). In particular, Kacperczyk and Pagnotta (2019) use a sample of illegal insider transactions with precise time and date information to show that informed traders make extensive use of use option markets to trade on their private information, on average accounting for 30% of daily trade volume. Shirvani et al. (2019) incorporate stock return predictability as an input into the Black-Scholes-Merton option pricing framework and find that an option trader's ability to predict stock returns affects option prices. Taken together these ideas suggest that changes in put and call option prices can lead changes in the underlying equity market index price, particularly when information asymmetry is high.

Goncalves-Pinto et al. (2020) find that option prices can act as a fundamental anchor during periods of transitory price pressure. Hence option prices can provide a clearer signal about future equity market prices during periods of elevated noise. Demand for downside protection by risk-averse investors and short selling constraints, such as stock lending fees, may also strengthen the information content of option prices with regard to future equity market movements. For example, when market makers hedge using options, they incorporate the expected short selling fee into the put option price. Since it is widely documented that short

sellers are informed, option prices may reflect this information via the size of the lending fee (Jones et al., 2018; Atmaz and Basak, 2019).

All else equal, put and call option prices will increase when implied volatility increases (Sundaram and Das, 2011). Consequently, our bi-objective function implicitly incorporates time-varying information about implied volatility into the machine learning process. This works via the costs $(c_t^{(fn)}; c_t^{(fp)})$ and scales the relative contribution of each observation to the loss. The resulting dynamic formulation is useful because spikes in implied volatility typically coincide with periods of elevated downside risk and financial distress.

2.2 Cost-Sensitive Logistic Regression Models

First we consider the LR model

$$p_t(y_t = 1 | \mathbf{x}_t; \quad) = \frac{1}{1 + e^{\quad \mathbf{x}_t^T}} \tag{8}$$

Such models have previously been applied to the equity market return prediction task (Ballings et al., 2015; Mascio et al., 2021). Traditionally, LR models are trained by maximizing the log-likelihood. This process is identical to minimizing the log-loss objective function over the training data. We use this traditional model as a benchmark in our empirical analysis.

Instead, we consider two penalized LR models that are trained by minimizing either the AEC function in (3) or the bi-objective function in (7). We can write these problems as follows

$$\text{AEC LR:} \quad \hat{\quad} = \arg\min \; L_2(\mathbf{y}; \mathbf{x}; \quad) + \quad [(1 \qquad) k \quad k_2 / 2 + \quad k \quad k_1]; \tag{9}$$

$$\text{bi-objective LR:} \quad \hat{\quad} = \arg\min \; w_1 L_1(\mathbf{y}; \mathbf{x}; \quad) + w_2 L_2(\mathbf{y}; \mathbf{x}; \quad) + \quad [(1 \qquad) k \quad k_2 / 2 + \quad k \quad k_1]; \tag{10}$$

where $jj \quad jj_1$ and $jj \quad jj_2$ denote the L^1- and L^2-norms of a vector.

We incorporate elastic net regularization ($[(1 \qquad) k \quad k_2 / 2 + \quad k \quad k_1]$) into the objective functions because financial markets are a low signal-to-noise environment with many features and with time-varying feature importance (see, e.g., James et al., 2023; Timmermann, 2018). This regularization involves two hyper-parameters, namely and . The former hyper-parameter controls the regularization strength while the latter hyper-parameter bridges the gap between the logistic lasso regression ($= 1$) and logistic ridge regression ($= 0$).

2.3 Cost-Sensitive Gradient Boosting Models

Recent literature that applies machine learning models in finance emphasizes the importance of non-linear relationships and interactions among predictors (Freyberger et al., 2020; Gu et al., 2020; Bianchi et al., 2021). As a consequence, the application of GBMs, a non-linear ensemble machine learning algorithm, to empirical asset pricing and financial forecasting is becoming increasingly popular (Krauss et al., 2017; Leippold et al., 2022). In particular, we are interested in how the two cost-sensitive objective functions can be used to improve standard GBM models.

Boosting is an ensemble learning concept whereby K weak base learners $h_k(\mathbf{x}_t; \quad_k); k = 1; \ldots; K$, are sequentially trained to minimize the aggregated prediction error of the ensemble at the current iteration. As base learners, we use binary decision trees. A binary decision tree divides the feature space into non-overlapping hyper-rectangles via a recursive sequence of binary splits. James et al. (2013) and Biau and Scornet (2016) provide a formal treatment of decision trees, and Varian (2014) provide a discussion of decision trees within the context of financial economics.

At each step in boosting, a new decision tree seeks to explain the residuals from the aggregated model at the previous iteration, thereby improving the prediction at the current iteration. The final boosted model is the sum of the sequential predictions from the individual base learners,

$$f(\mathbf{x}_t; \quad) = \sum_{k=1}^{K} h_k(\mathbf{x}_t; \quad_k). \tag{11}$$

In gradient boosting we treat the predictions $h_k(\mathbf{x}_t; \quad_k)$ as the ensemble model parameters (Friedman, 2001). Let $f^{(k)}(\mathbf{x}_t; \quad)$ denote the ensemble model at iteration k where $\quad$ contains all base learner parameters, and let the gradient of a loss function evaluated at the $(k \quad 1^{th})$ iteration be defined as

$$g_{tk} = \frac{@`(y_t; \quad)}{@} \Bigg|_{= f^{(k-1)}(\mathbf{x}_t; \quad)} \tag{12}$$

We seek to train a base learner in the direction that approximates the negative value of this gradient. The boosting updates can be written as follows

$$f^{(k)}(\mathbf{x}_t; \quad) := f^{(k \quad 1)}(\mathbf{x}_t; \quad) + \quad h_k(\mathbf{x}_t; \quad_k); \tag{13}$$

where $\quad 2\ [0; 1]$ is the learning rate hyper-parameter that shrinks the contribution of the k^{th} weak learner. The predictions from a ridge logistic regression model are used as the initial values in the first iteration of the boosting ensemble when we construct our gradient boosting models.

We use the LightGBM framework introduced by Ke et al. (2017). This framework is a highly efficient implementation of the general gradient boosting algorithm and also offers several techniques that can improve performance. In particular, LightGBM features histogram binning where the number of bins is a hyper-parameter which can be tuned to reduce over-fitting and speed up computations. Moreover, LightGBM grows the individual decision tree base learners leaf-wise, meaning that nodes of the binary decision tree are expanded in a best-first order and not the traditional so-called depth-wise ordering. This approach has been shown to achieve lower global values of the loss function when compared to the depth-wise approach (Shi, 2007).

We use the log-loss objective LightGBM model as a benchmark in our empirical analysis. For our AEC and bi-objective functions the expressions for the first and second-order gradients for observation t required by LightGBM are

$$\frac{@`_1(y_t; \quad)}{@} \Bigg|_{= f(\mathbf{x}_t; \quad)} = \frac{1}{1 + e^{\quad f(\mathbf{x}_t; \quad)}} \quad y_t \tag{14}$$

$$\frac{@`_2(y_t; \quad)}{@} \Bigg|_{= f(\mathbf{x}_t; \quad)} = \frac{c_t^{(fn)} y_t e^{\quad f(\mathbf{x}_t; \quad)} + c^{(fp)}(1 \quad y_t) e^{\quad f(\mathbf{x}_t; \quad)}}{(1 + e^{\quad f(\mathbf{x}_t; \quad)})^2} \tag{15}$$

$$\frac{@^2`_1(y_t; \quad)}{@^{\ 2}} \Bigg|_{= f(\mathbf{x}_t; \quad)} = \frac{1}{1 + e^{\quad f(\mathbf{x}_t; \quad)}} \quad 1 \quad \frac{1}{1 + e^{\quad f(\mathbf{x}_t; \quad)}} \tag{16}$$

$$\frac{@^2`_2(y_t; \quad)}{@^{\ 2}} \Bigg|_{= f(\mathbf{x}_t; \quad)} = \frac{c_t^{(fn)} y_t (e^{f(\mathbf{x}_t; \quad)}) e^{\quad 2f(\mathbf{x}_t; \quad)} \quad c_t^{(fp)} (1 \quad e^{f(\mathbf{x}_t; \quad)})(y_t \quad 1) e^{\quad 2f(\mathbf{x}_t; \quad)}}{(1 + e^{\quad f(\mathbf{x}_t; \quad)})^3} \tag{17}$$

To reduce the likelihood of over-fitting in the low signal-to-noise environment, we employ various algorithmic novelties. Specifically, we always sub-sample observations (a process known as stochastic gradient boosting) and only fit the base learner to a random sample of $\bar{p}$ features at each boosting iteration. We

also impose both L^1-norm and L^2-norm regularization explicitly within the boosting process. In classical gradient boosting, the base learners added in the latter iterations have a tendency to affect the predictions for a small number of observations (Vinayak and Gilad-Bachrach, 2015). In other words, these late-stage base learners over-fit. To address this issue we use the DART approach of Vinayak and Gilad-Bachrach (2015) which involves randomly dropping a fixed fraction, such as 10%, of the previously estimated base learners when computing the gradient in (12). This means that the predictions from the ensemble model at the i^{th} boosting iterations are computed without using this random fraction of base learners. The DART framework reduces the likelihood of over-fitting noisy financial data.

2.4 Hyper-parameter Optimization

Prior literature has typically used simple train/test splits when tuning model hyper-parameters (see, e.g., Gu et al., 2020; Chen et al., 2023; Bali et al., 2023; Goyenko and Zhang, 2020), while others have simply fixed model hyper-parameters at intuitively reasonable values or values used in prior literature (see, e.g., Krauss et al., 2017). Both of these methods may result in erroneous conclusions about model performance since hyper-parameters are critical to the machine learning process. Instead, we tune hyper-parameters using the K-fold purged and embargoed cross-validation method of De Prado (2018) and the Optuna framework of Akiba et al. (2019).

The K-fold purged and embargoed cross-validation method addresses issues associated with information leakage when traditional K-fold cross-validation is applied to financial forecasting tasks. Moreover, K-fold purged and embargoed cross-validation should be better at preventing over-fitting relative to the simple train/test split approach. This technique first splits the training data into K contiguous folds. Then, we remove (purge) observations in any of the training folds where there is known information overlap with observations in the test fold that were used to compute y_t, the forward-looking binary return direction variable, in the training fold. Additionally, we also remove a fixed number of observations from the beginning of the training fold that immediately follows the test fold to alleviate information leakage associated with serial correlation in predictor variables, a process referred to as embargoing by De Prado (2018). We set $K = 5$ and choose to embargo 2.5% of the training data.

Optuna implements efficient sampling strategies that focus the hyper-parameter optimization search on regions that are most likely to achieve the smallest value of the loss function over the test set. Within Optuna we use the Tree-structured Parzen Estimator algorithm to search for the best set of hyper-parameters (Bergstra et al., 2011). By using this state-of-the-art hyper-parameter tuning framework we can be confident that any differences in model performance are likely attributable to our cost-sensitive objective function rather than sub-optimal hyper-parameters. We allow Optuna to use 200 iterations to find the best set of model hyper-parameters.

For the logistic regression model, we find the optimal value of within a log domain search space between 0.1 and 10, and the optimal value of between 0.1 and 1. For the gradient boosting model, we search for the number of leaf nodes between 6 and 254, the value of the learning rate between 0.01 and 0.25, the fraction of training data used to fit each base learner between 0.3 and 0.5, the number of bins used to construct the histogram based approximations of each feature between 20 and 256, and the number of boosting iterations between 50 and 200. The values of the L^1 and L^2 regularization penalties are chosen within the log domain search space between 0.01 and 1. For simplicity we use the log-loss objective to select model hyper-parameters for all models that we study. As such, any gains in performance observed for our cost-sensitive models likely represents a lower bound and could be further improved by tailoring the hyper-parameter tuning specifically to their individual AEC and bi-objective functions.

3 Data and Experimental Design

Our models are evaluated using the S&P500, NASDAQ100, and Dow Jones Industrial Average (DJIA) stock market indices. To ensure the robustness of our results we also evaluate the models using a sample of 24 individual U.S. equities that have remained constituents of the S&P500. The ticker symbols of these individual equities are; AAPL, XOM, MSFT, GE, JNJ, WMT, JPM, PG, PFE, CVX, WFC, BAC, IBM, C, KO, INTC, CSCO, VZ, ORCL, MRK, PEP, HD, DIS and UNH, and we estimate separate LR and GBM models for each asset. We make rolling h-day ahead return direction forecasts using a training sample of the last 1500 trading days, which is approximately six years of historical data. Our results are entirely robust to an expanding window forecasting scheme but to save space we do not report these results. We prefer the rolling window forecasting scheme because this limited memory estimator can provide a better local approximation when the target series is auto-correlated and/or the prediction model is potentially misspecified (see, e.g. Giacomini and White, 2006).[5]

Daily price data for each asset is obtained from Compustat and option prices are obtained from the OptionMetrics IvyDB database for the period 03/07/1996 to 31/12/2021. The length of available option price data differs for each asset that we consider. To make use of all available data, we evaluate out-of-sample return direction forecasts for the S&P500 from 02/08/2002, the NASDAQ100 from 11/05/2005, and the DJIA from 17/06/2008. Out-of-sample return direction forecasts for the sample of 24 individual U.S. equities are evaluated over the sample period beginning 26/10/2010, as these securities have shorter option price histories in the OptionMetrics IvyDB database.

Appendix A describes the predictor variables and their data source. Broadly speaking, our predictor variables describe the behavior of past returns, including the use of various technical indicators that measure trend and momentum of the market (see, e.g. Neely et al., 2014; Baetje and Menkhoff, 2016), financial market volatility, skewness, and kurtosis (Anatolyev and Gospodinov, 2010; Bollerslev et al., 2009), and economic and financial conditions (Aruoba et al., 2009; Kliesen et al., 2012; Long et al., 2022). We have purposefully kept the predictor set at a moderate size and used theoretically intuitive predictors since our contribution is to demonstrate the utility of a cost-sensitive learning framework, rather than to design the most comprehensive return direction forecasting model. Each variable is constructed to ensure that there is no look-ahead bias at the forecast time.

The hyper-parameters of all models are re-tuned at the beginning of every month. The performance of the cost-sensitive models is benchmarked against their traditional log-loss objective counterparts. The put-call spread (price differential) is included as a predictor in the benchmark models (LR (put-call spread) and GBM (put-call spread))[6]. We also include the CBOE VIX index as a predictor variable in all models (both the proposed cost-sensitive models and the benchmarks with the usual log-loss) when forecasting the S&P500 index and the 24 individual equities, and the CBOE VXN and VXD indices when forecasting the NASDAQ100 and DJIA respectively. This would be the simplest and most obvious way to incorporate option prices and implied volatility information into a predictive model.

4 Classi cation Performance

To evaluate the classification performance of the proposed prediction models we use precision, recall, specificity, negative predictive value (NPV), F1 scores for both classes (F1 (+1) and F1 (0)), and balanced accuracy. The precision score measures the proportion of h-day ahead returns predicted as positive that were positive. The recall score measures the proportion of all positive returns that were correctly predicted

[5]The codes for our models are available at ???

[6]We do not include the put and call option prices separately as predictor variables in the benchmark models because they are highly correlated.

as positive. Specificity and negative predictive value scores are akin to the precision and recall scores for the zero class, respectively. The F1 (+1) (F1 (0)) score is a harmonic mean of precision and recall (specificity and negative predictive value), and it measures the trade-off between precision and recall (specificity and negative predictive value). For example, a classifier that achieves high precision at the expense of recall will have a low F1 (+1) score. Balanced accuracy is an average of the true-positive and true-negative classification rates, essentially measuring classification accuracy for both classes.

The above measures depend on the threshold used for classification. As threshold-invariant measures, we report the area under the ROC curve (AUROC), area under the precision-recall curve (AUPRC), and Brier loss function. AUROC evaluates how well a classifier distinguishes between positive and negative returns and is a measure of the probability that the classifier will assign a higher score to a randomly chosen day where the future return direction is positive than a randomly chosen day where the future return is negative. The Brier loss is defined as the mean squared difference between the probability forecast and the realized binary outcome and summarizes the accuracy of the probability forecasts from each classification model.

We also report the mean, median, minimum, maximum, and standard deviation of 30-day ahead returns when $\hat{y}_t = 1$ and when $\hat{y}_t = 0$. Finally, we report the number of predictions for each class and % Gain/Annum, which is the annualized return earned when $\hat{y}_t = 1$ and $\hat{y}_t = 0$. These statistics provide an indication of the potential investment returns from each model; we provide a more thorough analysis of a long-short investment strategy in a later section.

4.1 Equity Index Classification Results

Table 1 reports the mean performance measures for S&P500, NASDAQ and DJIA. The average ROC and precision-recall curves are computed using threshold averaging for 100 unique values of the threshold between zero and one (see, e.g., Fawcett, 2006). The area under these average curves is computed using the trapezoidal rule.

We find that both our cost-sensitive LR and GBM models achieve a higher F1 score for the positive class. This is primarily underpinned by a higher recall score. However, both the AEC and the bi-objective logistic regression models tend to have slightly lower specificity, NPV and F1 (0) scores then benchmark models. Collectively, these results suggest that adjusting the loss function used to train an LR model to incorporate option-based costs disproportionately improves the prediction of positive future returns. The bi-objective LR model outperforms the AEC LR model in terms of F1 scores and balanced accuracy scores.

The AEC GBM model has a larger F1 (+1) score but lower F1(0) and balanced accuracy than the bi-objective GBM model. However, we note that balanced accuracy scores of our cost-sensitive models are marginally smaller than the benchmark model scores. The Brier loss demonstrates that the AEC and bi-objective LR models provide a more accurate probabilistic signal about the direction of future equity market returns compared to benchmark LR models. Both the AEC LR and bi-objective LR models achieve a higher AUROC and marginally lower AUPRC. This result is broadly consistent with the observation that the cost-sensitive models tend to perform better for predicting positive future returns (the majority class) than the benchmark models.

Tables 2 and 3 report summary statistics for realized returns categorized by the predicted class and model. The bi-objective LR model has the smallest mean and median realized returns when $\hat{y}_t = 0$. While the mean and median return for AEC and bi-objective GBM models are less favourable compared to the benchmark models, we note that our models outperform the benchmark GBM in terms of the % Gain/Annum. This may indicate an improved market timing ability of the AEC and bi-objective GBM models. Consistent with higher (lower) recall (specificity) from Table 1, the AEC and bi-objective models issue positive (negative) return5 forecasts more (less) often than the benchmark models.

Standardized 10-day ahead options are available for the the S&P500 index from 04/11/2005, and for the NASDAQ100 and DJIA indices from 20/08/2007. Therefore, in Tables 4 - 6 we report classification

Table 1: 30-day Ahead Equity Index Classification Performance Results
The table reports classification performance measures for 30-day ahead return direction forecasts for the S&P500, NASDAQ and DJIA equity market indices. The acronym LR stands for logistic regression. The acronym GBM stands for gradient boosting machine. The acronym AEC stands for Average Expected Cost.

	(i) Logistic Regression			
	LR	LR (put-call spread)	AEC LR	Bi-objective LR
Precision	0.7238	0.7238	0.7154	0.7178
Recall	0.6097	0.6100	0.6238	0.6790
F1 (+1)	0.6610	0.6611	0.6665	0.6976
NPV	0.3909	0.3917	0.3821	0.4005
Specificity	0.5168	0.5169	0.4829	0.4450
F1 (0)	0.4440	0.4443	0.4266	0.4212
Balanced Accuracy	0.5632	0.5634	0.5534	0.5620
Brier Loss	0.2929	0.2927	0.2635	0.2556
AUROC	0.5459	0.5471	0.5624	0.5604
AUPRC	0.7212	0.7208	0.6689	0.7161
	(ii) Gradient Boosting Machine			
	GBM	GBM (put-call spread)	AEC GBM	Bi-objective GBM
Precision	0.6991	0.6976	0.6863	0.6874
Recall	0.6844	0.6872	0.7939	0.7394
F1 (+1)	0.6916	0.6923	0.7361	0.7123
NPV	0.3712	0.3696	0.3645	0.3576
Specificity	0.3873	0.3806	0.2453	0.3011
F1 (0)	0.3790	0.3749	0.2931	0.3263
Balanced Accuracy	0.5358	0.5339	0.5196	0.5203
Brier Loss	0.2417	0.2395	0.2564	0.2485
AUROC	0.5528	0.5602	0.5252	0.5322
AUPRC	0.7090	0.7166	0.6946	0.6928

Table 2: 30-day Ahead LR Equity Index Return Statistics
The table reports summary statistics for the realized returns across the S&P500, NASDAQ and DJIA equity market indices. The acronym LR stands for logistic regression.

	(i) $\hat{y}_t = 1$			
	LR	LR (put-call spread)	AEC LR	Bi-objective LR
Mean	1.7715	1.7520	1.6938	1.6940
Median	2.4154	2.4261	2.3224	2.3672
Min	-35.4432	-35.4432	-35.4563	-35.2088
Max	23.1077	23.1077	18.3592	17.5322
Std	5.4234	5.4536	5.0103	5.0108
% of Time	56.8250	56.8385	58.8538	63.8627
% Gain/Annum	21.7630	21.0646	17.4068	17.5545
	(ii) $\hat{y}_t = 0$			
	LR	LR (put-call spread)	AEC LR	Bi-objective LR
Mean	0.4072	0.4239	0.4843	0.3314
Median	1.4038	1.3786	1.5123	1.3161
Min	-41.8663	-41.8663	-41.8663	-41.4685
Max	24.7162	24.7162	24.7162	24.7162
Std	6.6593	6.6383	7.1121	7.3916
% of Time	43.1750	43.1615	41.1462	36.1373
% Gain/Annum	-2.0422	-1.2836	2.3282	0.2848

Table 3: 30-day Ahead GBM Equity Index Return Statistics
The table reports summary statistics for the realized returns across the S&P500, NASDAQ and DJIA equity market indices. The acronym GBM stands for gradient boosting machine.

	(i) $\hat{y}_t = 1$			
	GBM	GBM (put-call spread)	AEC GBM	Bi-objective GBM
Mean	1.5039	1.4955	1.2965	1.3520
Median	2.1872	2.1841	2.0434	2.0958
Min	-37.3635	-37.1158	-41.8663	-36.1531
Max	24.7162	24.7162	23.5020	23.5020
Std	5.5229	5.4452	5.5913	5.4870
% of Time	66.0904	66.4965	78.1011	72.6319
% Gain/Annum	13.6431	12.0855	13.4910	15.0917
	(ii) $\hat{y}_t = 0$			
	GBM	GBM (put-call spread)	AEC GBM	Bi-objective GBM
Mean	0.5958	0.5981	0.8318	0.7767
Median	1.7126	1.6845	1.9688	1.9243
Min	-41.4685	-41.8663	-39.4850	-41.8663
Max	23.1077	23.1077	24.4726	24.4726
Std	6.8577	6.9807	7.3331	7.2499
% of Time	33.9096	33.5035	21.8989	27.3681
% Gain/Annum	5.6335	8.4510	2.5305	0.3173

performance metrics and return statistics for rolling 10-day ahead return direction forecasts using 10-day ahead put and call option prices. We find that the behavior of the AEC and bi-objective models at the 10-day ahead forecast horizon is consistent with the behavior at the 30-day horizon. That is, our cost-sensitive models tend to achieve higher F1(+1) scores, underpinned by a significantly higher recall scores and marginally higher precision scores relative to the benchmark models, and lower F1(0) scores primarily driven by a lower specificity score relative to the benchmark models.

Both the AEC and bi-objective LR models obtain a lower Brier loss static than benchmark models, consistent with the results for one-month ahead forecasts. Moreover, both the bi-objective LR and GBM models achieve higher AUROC and AUPRC than their respective benchmarks at the 10-day ahead forecasting horizon.

Table 4: 10-day Ahead Equity Index Classification Performance Results
The table reports classification performance measures for 10-day ahead return direction forecasts for the S&P500, NASDAQ and DJIA equity market indices. The acronym LR stands for logistic regression. The acronym GBM stands for gradient boosting machine. The acronym AEC stands for Average Expected Cost.

	(i) Logistic Regression			
	LR	LR (put-call spread)	AEC LR	Bi-objective LR
Precision	0.6764	0.6757	0.6677	0.6725
Recall	0.5937	0.5915	0.6354	0.7029
F1 (+1)	0.6322	0.6305	0.6494	0.6871
NPV	0.3865	0.3852	0.3874	0.4004
Specificity	0.4747	0.4749	0.4182	0.3659
F1 (0)	0.4257	0.4249	0.3995	0.3817
Balanced Accuracy	0.5342	0.5332	0.5268	0.5344
Brier Loss	0.2843	0.2847	0.2645	0.2625
AUROC	0.5415	0.5414	0.5270	0.5515
AUPRC	0.6745	0.6742	0.6432	0.6866
	(ii) Gradient Boosting Machine			
	GBM	GBM (put-call spread)	AEC GBM	Bi-objective GBM
Precision	0.6513	0.6399	0.6459	0.6491
Recall	0.5640	0.5752	0.8178	0.7808
F1 (+1)	0.6044	0.6055	0.7216	0.7087
NPV	0.3538	0.3405	0.3357	0.3517
Specificity	0.4416	0.4039	0.1702	0.2187
F1 (0)	0.3928	0.3692	0.2235	0.2684
Balanced Accuracy	0.5028	0.4896	0.4940	0.4998
Brier Loss	0.2485	0.2480	0.2539	0.2464
AUROC	0.5059	0.5009	0.5002	0.5038
AUPRC	0.6528	0.6461	0.6570	0.6489

The proposed bi-objective LR and GBM models earn a lower mean return and % Gain/Annum than benchmarks when predicting negative 10-day ahead returns, and a higher mean return than benchmarks when predicting positive 10-day ahead returns. Consistent with the one-month ahead forecasting results, we find that the AEC and bi-objective models issue fewer negative and more positive return forecasts relative to benchmark models.

Table 5: 10-day Ahead LR Equity Index Return Statistics
The table reports summary statistics for the realized returns across the S&P500, NASDAQ and DJIA equity market indices. The acronym LR stands for logistic regression.

	(i) $\hat{y}_t = 1$			
	LR	LR (put-call spread)	AEC LR	Bi-objective LR
Mean	0.6829	0.6700	0.6142	0.7021
Median	1.0018	0.9973	0.9496	0.9740
Min	-23.9025	-23.9025	-24.2093	-26.4219
Max	10.9312	10.9312	11.0349	10.8908
Std	3.0889	3.1025	3.1721	2.9812
% of Time	57.0273	56.8756	61.7110	67.8716
% Gain/Annum	18.7978	18.6489	15.6448	21.0826
	(ii) $\hat{y}_t = 0$			
	LR	LR (put-call spread)	AEC LR	Bi-objective LR
Mean	0.4231	0.4432	0.4444	0.2768
Median	0.6612	0.6558	0.6245	0.5433
Min	-26.5290	-26.5290	-17.5890	-25.7820
Max	17.6675	17.6675	17.6675	17.6675
Std	3.2850	3.2654	3.2700	3.5570
% of Time	42.9727	43.1244	38.2890	32.1284
% Gain/Annum	11.8681	12.2042	15.3202	4.7751

Table 6: 10-day Ahead GBM Equity Index Return Statistics
The table reports summary statistics for the realized returns across the S&P500, NASDAQ and DJIA equity market indices. The acronym GBM stands for gradient boosting machine.

	(i) $\hat{y}_t = 1$			
	GBM	GBM (put-call spread)	AEC GBM	Bi-objective GBM
Mean	0.5156	0.5350	0.5225	0.5664
Median	0.8925	0.8381	0.8185	0.8731
Min	-26.7287	-26.9894	-26.9894	-26.7287
Max	13.1898	12.3756	13.5116	13.6623
Std	3.4561	3.3971	3.2311	3.2369
% of Time	56.2240	58.3113	82.2129	78.1096
% Gain/Annum	9.9665	13.8580	14.3707	15.7557
	(ii) $\hat{y}_t = 0$			
	GBM	GBM (put-call spread)	AEC GBM	Bi-objective GBM
Mean	0.6327	0.6041	0.7690	0.5795
Median	0.8039	0.8443	0.9734	0.7539
Min	-21.1336	-19.5025	-10.0618	-20.3866
Max	17.6675	17.3624	17.1950	17.1950
Std	2.8013	2.8646	2.9850	3.0090
% of Time	43.7760	41.6887	17.7871	21.8904
% Gain/Annum	23.6621	17.9208	20.6350	15.7779

Table 7 reports averages of the performance metrics for the 30-day ahead forecasts of the 24 individual U.S. equities.[7] Consistent with the results obtained for the U.S. equity market indices, we find that the AEC and bi-objective models achieve a higher F1 (+1) score and a lower F1 (0) score relative to their benchmark models. The larger F1 (+1) score is underpinned by both higher precision and recall.

The bi-objective models continue to outperform the AEC models in terms of balanced accuracy. However, now our bi-objective models achieve the highest balanced accuracy score, albeit by only a small margin. The AUROC and AUPRC scores for our bi-objective LR and GBM models are higher than for all other models that we consider, indicating that these models outperform the benchmarks regardless of the specific probability threshold used to assign binary class labels.

Table 7: Individual Equity Classification Performance Results
The table reports the average classification performance results for 30-day ahead return direction forecasts for the sample of 24 U.S. equities. The acronym LR stands for logistic regression. The acronym GBM stands for gradient boosting machine. The acronym AEC stands for Average Expected Cost.

	(i) Logistic Regression			
	LR	LR (put-call spread)	AEC LR	Bi-objective LR
Precision	0.5984	0.5981	0.6002	0.6019
Recall	0.5043	0.5037	0.5587	0.5630
F1 (+1)	0.5435	0.5430	0.5754	0.5788
NPV	0.4144	0.4138	0.4183	0.4222
Specificity	0.5086	0.5081	0.4585	0.4610
F1 (0)	0.4528	0.4523	0.4330	0.4370
Balanced Accuracy	0.5065	0.5059	0.5086	0.5120
Brier Loss	0.3406	0.3412	0.2598	0.2802
AUROC	0.4872	0.4865	0.5052	0.5104
AUPRC	0.5833	0.5825	0.5764	0.5938
	(ii) Gradient Boosting Machine			
	GBM	GBM (put-call spread)	AEC GBM	Bi-objective GBM
Precision	0.5837	0.5789	0.5894	0.5982
Recall	0.3694	0.3720	0.4951	0.4703
F1 (+1)	0.4429	0.4434	0.5338	0.5228
NPV	0.4041	0.4029	0.4068	0.4125
Specificity	0.6199	0.6137	0.5018	0.5408
F1 (0)	0.4869	0.4838	0.4458	0.4657
Balanced Accuracy	0.4946	0.4929	0.4984	0.5056
Brier Loss	0.2556	0.2561	0.2773	0.2699
AUROC	0.5013	0.5003	0.4892	0.5028
AUPRC	0.5754	0.5713	0.5755	0.5868

Tables 8 and 9 report the summary statistics for realized returns across the 24 individual U.S. equities, by model prediction. The benchmark LR models tend to outperform both the AEC and bi-objective logistic regression models in terms of realized returns for both class labels. However, our AEC and bi-objective GBM models outperform their respective benchmark model mean and median returns. Moreover, the AEC

[7] We did not consider 10-day ahead forecasts as standardized 10-day option data is only available from 2011, leaving an unreasonably short out-of-sample testing period available for analysis.

and bi-objective GBM models earn a higher (lower) % Gain/Annum when $\hat{y} = 1$ ($\hat{y} = 0$). Consistent with the results for the equity market index backtest, we find that the AEC and bi-objective models tend to issue more (less) positive (negative) class predictions than benchmark models.

Table 8: LR Individual Equity Return Statistics
The table reports averages of summary statistics for the realized returns across 24 individual U.S equities. The acronym LR stands for logistic regression.

	(i) $\hat{y}_t = 1$			
	LR	LR (put-call spread)	AEC LR	Bi-objective LR
Mean	1.3288	1.3070	1.2696	1.2601
Median	1.6377	1.6283	1.6716	1.6487
Min	-37.5102	-37.4633	-40.2212	-37.3782
Max	28.1745	28.1745	29.0293	28.2126
Std	7.4353	7.4564	7.4323	7.3897
% of Time	49.8846	49.8698	55.0293	55.2779
% Gain/Annum	15.3441	14.9070	11.9485	12.8817
	(ii) $\hat{y}_t = 0$			
	LR	LR (put-call spread)	AEC LR	Bi-objective LR
Mean	0.9406	0.9649	0.9766	0.9502
Median	1.4698	1.4896	1.3967	1.4022
Min	-39.8185	-39.5483	-40.9332	-40.2716
Max	26.3318	26.0917	27.0770	27.8379
Std	7.7253	7.7055	7.8658	7.9618
% of Time	50.1154	50.1302	44.9707	44.7221
% Gain/Annum	6.2884	6.7067	10.7654	8.0063

Broadly speaking, the bi-objective models outperform the AEC models for both the equity market index and single stock 30-day ahead return direction prediction. This demonstrates the utility of combining a traditional machine learning objective function with a second objective that incorporates the dynamics of financial markets. This finding is also related to the forecast combinations literature, where it is often found that the equal-weighted combination of forecasts is a competitive benchmark that outperforms individual forecasts in noisy economic and financial environments (Timmermann, 2006). Including the put-call spread as an additional predictor variable only yields a very minor improvement in classification performance for backtests using both the equity market index and individual equities.

Some prior literature has found that non-linear and ensemble machine learning models outperform linear regression models for stock return forecasting (see, e.g., Gu et al., 2020; Bali et al., 2023; Leippold et al., 2022). However, we find no definitive evidence that gradient boosting comprehensively outperforms elastic-net logistic regression for 30-day-ahead stock return direction forecasting. Instead, our result is broadly consistent with that of Iworiso and Vrontos (2020) who show that elastic-net probit models outperform gradient boosting for one-month ahead forecasts of the U.S equity risk premium direction. To reconcile our results with some prior literature, we first note that Gu et al. (2020); Bali et al. (2023) and Leippold et al. (2022) studied point forecasts of returns, not return direction forecasts. Moreover, Leippold et al. (2022) find that the performance of GBM models is largely driven by small-cap stocks, whereas we study

Table 9: GBM Individual Equity Return Statistics
The table reports averages of summary statistics for the realized returns across 24 U.S individual equities. The acronym GBM stands for gradient boosting machine.

	(i) $\hat{y}_t = 1$			
	GBM	GBM (put-call spread)	AEC GBM	Bi-objective GBM
Mean	1.0860	1.0363	1.1341	1.2787
Median	1.3935	1.3223	1.4859	1.6197
Min	-38.7440	-36.8273	-39.1770	-37.4575
Max	27.6160	26.7197	28.4929	28.0217
Std	7.2950	7.3189	7.2815	7.1375
% of Time	37.4231	37.8344	49.6375	46.5584
% Gain/Annum	10.3855	8.5743	12.1812	13.6139
	(ii) $\hat{y}_t = 0$			
	GBM	GBM (put-call spread)	AEC GBM	Bi-objective GBM
Mean	1.1737	1.1781	1.1523	1.0713
Median	1.6718	1.6849	1.6077	1.5168
Min	-42.0164	-41.6720	-40.3937	-40.7594
Max	27.9253	27.9695	27.3085	27.9797
Std	7.7851	7.7927	7.8840	7.9091
% of Time	62.5769	62.1656	50.3625	53.4416
% Gain/Annum	9.4721	11.0972	8.7930	8.5937

equity market indices and large-cap stocks.[8] However, when our AEC and bi-objective models do outperform the benchmark models, the gains are larger for the GBM models. That is, the non-linear ensemble GBM algorithm tends to benefit more from cost-sensitive learning, relative to the traditional log-loss objective GBM model when there are gains to be extracted.

5 Economic Signi cance

Ultimately, an effective stock return direction prediction model is judged by its ability to generate superior returns net of transaction costs in a relevant investment strategy. Even small differences in classification performance between models have the potential to generate sizeable differences in long-run returns when the forecasts are used to make trading decisions. We backtest two long-short market timing strategies that trade an equal-weighted portfolio of the three equity market indices based upon the one-month and 10-day ahead return direction predictions. That is, we long the index when the model predicts the class label +1 (positive), and short the index when the model predicts the class label 0 (negative). We use the equal weight portfolio because it is a simple and robust benchmark portfolio construction method that outperforms more complex portfolio construction methods out-of-sample (see, e.g., DeMiguel et al., 2009; Swade et al., 2023).[9]

Each long/short strategy trades the equal-weighted combination of the State Street SPDR, State Street DIA, and the Invesco QQQ exchange-traded funds. These are the tradeable equity index instruments that track the S&P500, DJIA, and NASDAQ100 indices respectively. Importantly these tradeable instruments incorporate the real-world expense ratios that an investor would incur. All strategy performance statistics are computed net of transaction costs which are assumed to be 0.3% of the value traded (Petraki, 2020). We also include a cost equal to 0.1% of the prevailing mid price to account for slippage at the time of the transaction.[10]

To mitigate path dependence, that is the dependence of the strategy returns on a given starting date, we average results over all possible starting days between 18/06/2008 and 18/06/2009 for the strategy involving one-month ahead predictions, and between 20/08/2013 and 20/08/2014 for the 10-day ahead predictions. The exception is the maximum drawdown statistic, where we take the maximum instead of the average. Both backtests end on 31/12/2021. The portfolio re-balancing period is set to match the forecast horizon. Predictions generated by each model use information up to and including time t to forecast the direction of return at time $t+h$. Therefore, it is unrealistic to trade on any forecast at time t. We lag each forecast so that at time $t+1$, we trade on the forecast produced at time t. This ensures that our strategies do not have look-ahead bias.

Tables 10 and 11 present measures of investment performance for the one-month and 10-day ahead predictions respectively. We report three ratios that measure returns per unit of risk. The Sharpe ratio is the excess return divided by the standard deviation.[11] The Sortino ratio is the excess return divided by the standard deviation of only negative returns (downside deviation). The MAR ratio is the strategy's compound annual growth rate divided by the strategy's maximum drawdown. Additionally, we report the beta with respect to the equal-weighted market buy & hold portfolio.

[8] For the largest 70% of stocks by market-cap and the largest 70% average market capitalization per shareholder Leippold et al. (2022) find that elastic-net linear regressions exhibit a higher out-of-sample R^2 than GBM models.

[9] Our results are robust to using a weighting scheme based upon the confidence in the predicted probabilities, that is upon how far away from 0.5 the predicted probabilities are for each equity market index. In this weighting scheme more confident predictions receive larger weights.

[10] We checked sensitivity of our results to using the so-called square-root market impact model for transaction costs, in which the costs are computed as $C = 0.35\sigma(Q/V)^{0:4}$, where Q is the order size, V is the average daily traded volume, σ is daily volatility, and 0.35 and 0.40 are default parameters of the model. Our results are fully robust to this alternate transaction costs model.

[11] We always define excess returns as the strategy return above the return on cash.

Table 10: 30-day Ahead Long/Short Investment Strategy Backtest Results
The table reports investment performance statistics for the long/short investment strategy backtest that uses 30-day ahead predictions from the logistic regression and gradient boosting models. The acronym LR stands for logistic regression. The acronym GBM stands for gradient boosting machine.

	(i) Logisitic Regression			
	LR	LR (put call spread)	AEC LR	Bi-objective LR
Annualised Return (%)	6.9455	6.8990	7.0799	8.9388
Annualised Std	14.8760	14.8667	14.0180	14.5484
Annualised Sharpe Ratio	0.5122	0.5091	0.5421	0.6441
Annualised Sortino Ratio	0.8247	0.8198	0.8734	1.0421
Max Drawdown (%)	-31.4349	-31.4929	-29.1421	-28.0612
Max Drawdown Period (days)	250.8053	254.6374	152.2901	124.1794
MAR Ratio	0.2390	0.2389	0.2710	0.3361
Max DD/Vol	2.1092	2.1166	2.0779	1.9268
Annualised Downside Deviation	9.3066	9.3058	8.7569	9.0233
Beta	-0.0811	-0.0764	-0.1137	-0.0653
	(ii) Gradient Boosting Machine			
	GBM	GBM (put call spread)	AEC GBM	Bi-objective GBM
Annualised Return (%)	8.2256	8.0872	10.5628	9.5228
Annualised Std	15.2176	15.1763	15.4006	14.9089
Annualised Sharpe Ratio	0.5809	0.5742	0.7162	0.6711
Annualised Sortino Ratio	0.9105	0.9066	1.1127	1.0519
Max Drawdown (%)	-33.8870	-32.6567	-31.2007	-32.6957
Max Drawdown Period (days)	423.9924	476.9046	204.2634	358.6679
MAR Ratio	0.2613	0.2659	0.3628	0.3101
Max DD/Vol	2.2299	2.1527	2.0191	2.1919
Annualised Downside Deviation	9.7889	9.6990	9.9716	9.5591
Beta	0.3334	0.3043	0.4910	0.3738

Table 11: 10-day Ahead Long/Short Investment Strategy Backtest Results
The table reports investment performance statistics for the long/short investment strategy backtest that uses 10-day ahead predictions from the logistic regression and gradient boosting models. The acronym LR stands for logistic regression. The acronym GBM stands for gradient boosting machine.

	(i) Logisitic Regression			
	LR	LR (put call spread)	AEC LR	Bi-objective LR
Annualised Return (%)	1.7399	1.2479	4.4511	8.9664
Annualised Std	14.5608	14.5489	12.6107	14.2428
Annualised Sharpe Ratio	0.1874	0.1565	0.3982	0.6605
Annualised Sortino Ratio	0.2878	0.2428	0.5958	0.9916
Max Drawdown (%)	-32.5856	-32.9302	-28.2925	-31.5601
Max Drawdown Period (days)	235.6756	205.8397	203.0840	73.9771
MAR Ratio	0.0603	0.0472	0.1664	0.3267
Max DD/Vol	2.2335	2.2599	2.2384	2.2031
Annualised Downside Deviation	9.7005	9.7143	8.5217	9.6365
Beta	0.0739	0.0770	0.1989	0.2052
	(ii) Gradient Boosting Machine			
	GBM	GBM (put call spread)	AEC GBM	Bi-objective GBM
Annualised Return (%)	1.2366	2.8427	7.8442	7.6551
Annualised Std	14.4883	13.8423	15.4558	14.7814
Annualised Sharpe Ratio	0.1546	0.2618	0.5484	0.5573
Annualised Sortino Ratio	0.2302	0.3906	0.7857	0.8145
Max Drawdown (%)	-39.7291	-36.9987	-35.5431	-32.8688
Max Drawdown Period (days)	763.7137	647.5000	123.2176	206.5763
MAR Ratio	0.0448	0.1029	0.2305	0.2705
Max DD/Vol	2.7306	2.6793	2.2994	2.2063
Annualised Downside Deviation	10.0550	9.5142	10.8274	10.2418
Beta	0.4013	0.4037	0.6653	0.5790

It is clear from the tables that cost-sensitive models outperform their respective benchmarks for both the one-month and 10-day ahead forecasts. Among the logistic regression models, the bi-objective models earn the largest annualized returns and have the highest Sharpe ratio. Among the GBM models, both the AEC and bi-objective GBM models earn higher annualized returns and have substantially higher Sharpe ratios than their benchmarks. At the one-month ahead horizon the AEC GBM model marginally outperforms the bi-objective model. The opposite is true at the 10-day ahead horizon. These results demonstrate that combining log-loss and AEC produces machine learning models that generate superior risk-adjusted returns.

Long/short strategies that use forecasts from our cost-sensitive models have lower downside risk. First, they have a higher Sortino ratio, underpinned by higher annualized returns and smaller annualized downside deviations. Second, the cost-sensitive models have a maximum draw-down that is almost always smaller than for the benchmark models, a maximum drawdown period that is significantly shorter than for the benchmarks, and a smaller maximum-drawdown per unit of risk. Finally, our AEC and bi-objective models have a higher MAR ratio. In summary, both logistic regressions and GBM models which incorporate cost-sensitive learning generate forecasts that reduce downside risk in a long/short investment strategy, relative to strategies that use forecasts from a traditional log-loss machine learning model.

This result that our AEC and bi-objective model strategies have a lower downside risk is particularly interesting given that these models achieved lower F1 (0) scores than their benchmarks. These lower F1 (0) scores were almost entirely attributable to a lower specificity score. To reconcile these results and to further understand the dynamics of our AEC and bi-objective models, Table 12 reports the recall scores for the subset negative returns that are larger than one standard deviation and for the subset of negative returns that are smaller than one standard deviation. When compared to benchmark specificity scores, the AEC and bi-objective models have relatively smaller specificity scores for the subset of negative returns that are less than or equal to one standard deviation in absolute value. That is, our cost-sensitive models tend to focus more on correctly classifying large negative returns rather than small negative returns. Our investment strategy results show that this negative return classification behaviour is not economically detrimental and can reduce investors' exposure to downside risk.

Table 12: Specificity scores for large and small negative returns
The table reports specificity scores for two subsets of the out-of-sample forecast period. The first subset is negative returns that are larger than one standard deviation ($r_t^{(-)} < std(\mathbf{r})$) in absolute value. The second subset is negative returns that are less than or equal to one standard deviation ($r_t^{(-)} \geq std(\mathbf{r})$) in absolute value.

	h=30		h=10	
	$r_t^{(-)} < std(\mathbf{r})$	$r_t^{(-)} \geq std(\mathbf{r})$	$r_t^{(-)} < std(\mathbf{r})$	$r_t^{(-)} \geq std(\mathbf{r})$
LR	0.5434	0.5087	0.4363	0.4876
LR (Put Call Spread)	0.5370	0.5116	0.4251	0.4904
AEC LR	0.5471	0.4629	0.4153	0.4223
Bi-objective LR	0.5267	0.4161	0.4463	0.3396
GBM	0.4674	0.3565	0.3819	0.4645
GBM (Put Call Spread)	0.4625	0.3512	0.3865	0.4102
AEC GBM	0.3271	0.2147	0.1562	0.1733
Bi-objective GBM	0.3884	0.2683	0.2118	0.2207

Figures 1 and 2 plot the average cumulative portfolio value for long/short strategies using the one-month and 10-day ahead predictions, respectively. After initially under-performing between 2008 and 2012, the long/short strategy that uses one-month ahead predictions from the bi-objective LR and GBM models outperforms from 2014 onward. At the 10-day horizon both the bi-objective LR model and bi-objective

GBM model comprehensively outperform benchmark models. Hence, an investor could have realized material economic gains by incorporating cost-sensitive machine learning into return direction forecasts for U.S. equity market indices. Interestingly, all 10-day ahead long/short strategies suffer significant drawdowns during the COVID-19 crisis while strategies that use the one-month ahead predictions do not.

We find that the strategies using GBM model forecasts earn higher returns and have higher Sharpe ratios than the strategies using LR model forecasts at the one-month horizon. This is despite not being able to comprehensively outperform logistic regression in terms of the classification performance discussed in Section 4. That is, one-month ahead forecasts from non-linear ensemble machine learning algorithms appear to translate into better investment performance than forecasts made by linear models. This finding is consistent with our conclusion that GBM algorithm tends to extract a larger gain from a cost-sensitive learning structure when compared to the elastic-net LR algorithm. Moreover, at the one-month ahead horizon the GBM models appear to perform better over the COVID-19 crisis period than the LR models, which make a substantial profit during the initial drawdown but give back most of the gains in the subsequent market rally. This finding is broadly consistent with that of Bali et al. (2023) who also find that gradient boosting models exhibited superior investment performance over the COVID-19 crisis.

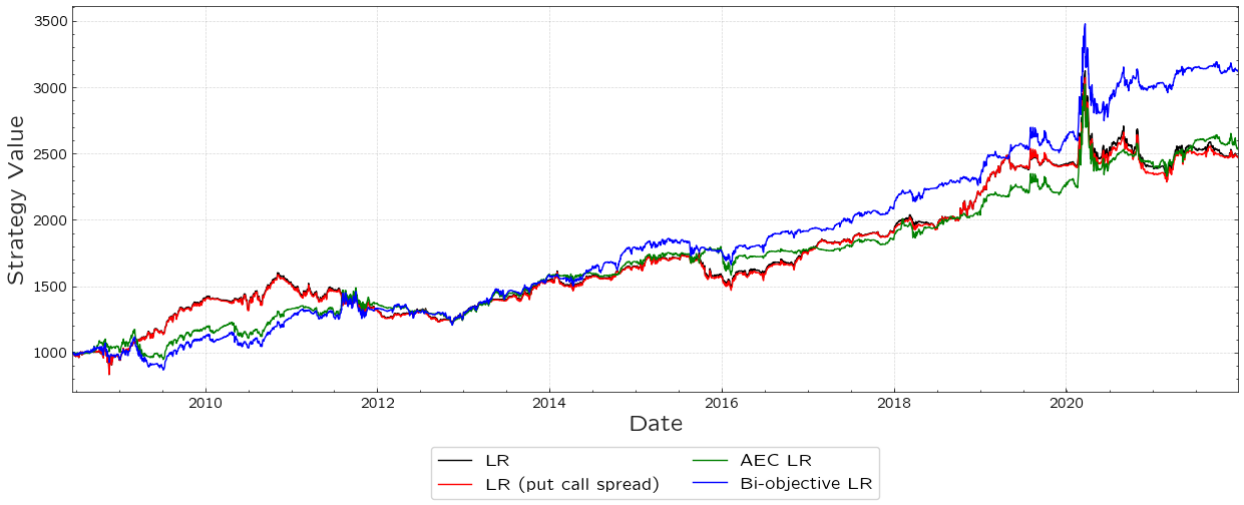
The LR models have a substantially smaller beta to the equal-weighted buy & hold strategy, potentially offering diversification benefits in multi-strategy investment settings. All GBM models have a positive beta. When comparing strategies over different forecast horizons we find that one-month ahead investment strategies earn higher returns and tend to have higher Sharpe ratios than 10-day ahead. Finally, since institutional investors such as pension funds tend to be predominately long-only investors, we demonstrate in Appendix B that our conclusions about relative model performance are mostly robust to using a long-only investment strategy where short positions are replaced with equivalent long positions in the Vanguard U.S. total bond market ETF.

6 Conclusion

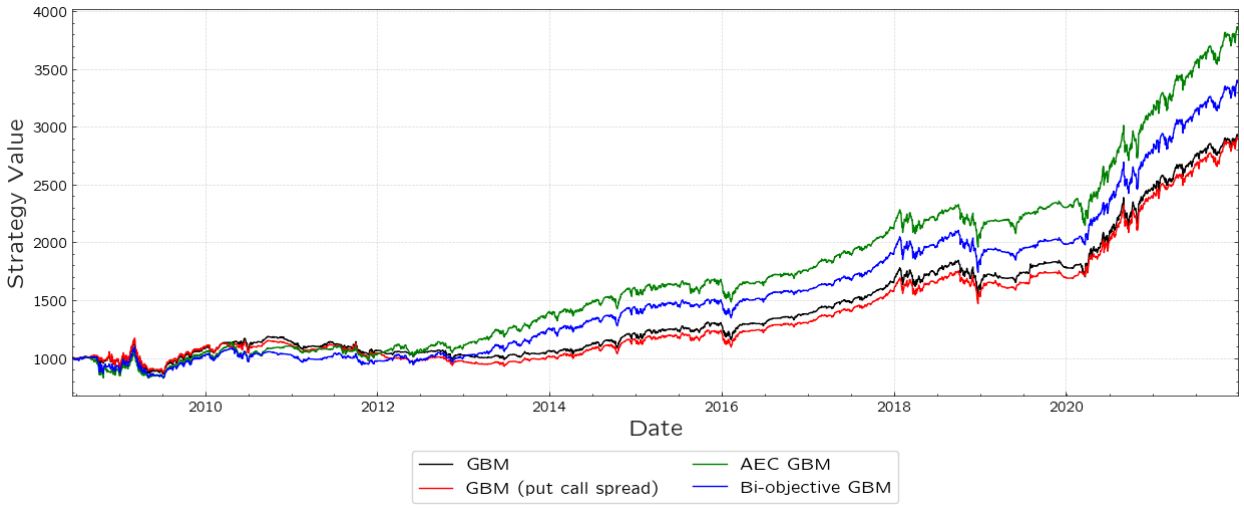
Existing applications of machine learning models to forecast the direction of equity market returns assume that misclassification costs are constant and equal. In this study, we design example dependent cost-sensitive objective functions for this task. These objective functions incorporate time-varying and asymmetric costs of false-positive and false-negative classification errors. In particular, we study a novel bi-objective function that combines the average expected cost with the log-loss objective function. Hence, our models incorporate the idea that the cost of making different classification errors changes over the business cycle.

As measures of the misclassification cost we use at-the-money put and call option prices. These prices are naturally forward-looking and incorporate the investor's expectations and preferences about future risk and return. We train logistic regressions and gradient-boosting machines using the cost-sensitive objective functions and we show that these models improve the classification performance of positive future returns. In particular, our models always correctly classify a larger fraction of future positive returns at both a one-month and 10-day ahead forecast horizon for both equity market indices and individual stocks. A long-short strategy that trades an equal-weighted portfolio of three U.S. equity indices based upon the return direction forecasts of the cost-sensitive models earns superior risk-adjusted returns. Moreover, strategies that use cost-sensitivity have a lower downside risk. Investors could have earned superior returns by using our cost-sensitive objective functions when traininng financial machine learning models.

This paper contributes to the growing literature that applies machine learning to forecast the direction of equity market returns (Fischer and Krauss, 2018; Iworiso and Vrontos, 2020; Mascio et al., 2021). We show that augmenting the objective functions in these models to better suit the dynamics of financial markets improves classification performance. The paper is also related to the literature studying the informational relationships between stock and option markets (see, e.g., Pan and Poteshman, 2006; Johnson and So,

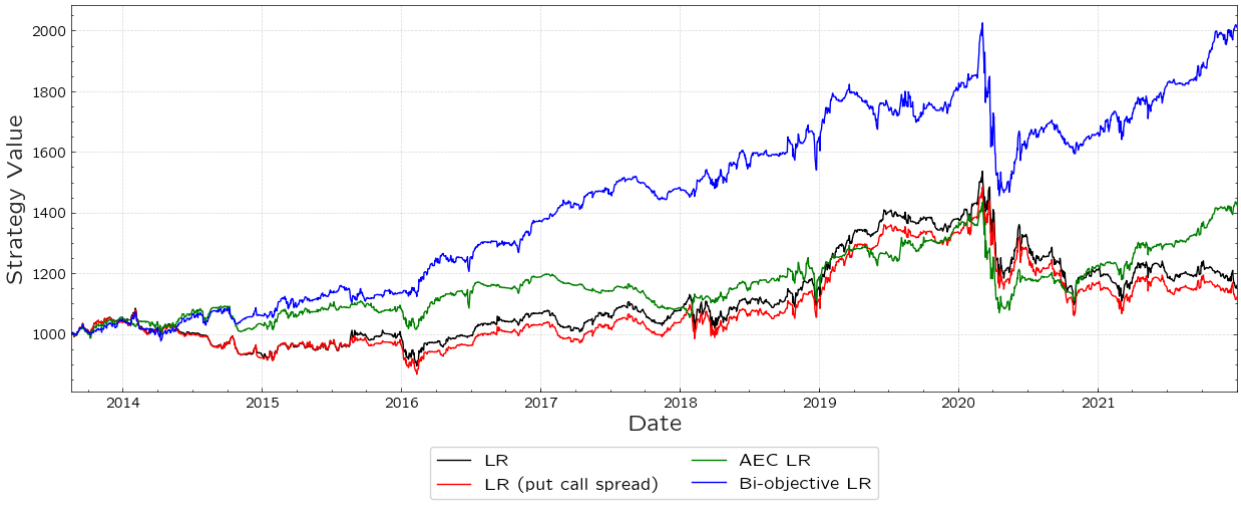


(a) Logistic Regression Long/Short Backtest Portfolio Values

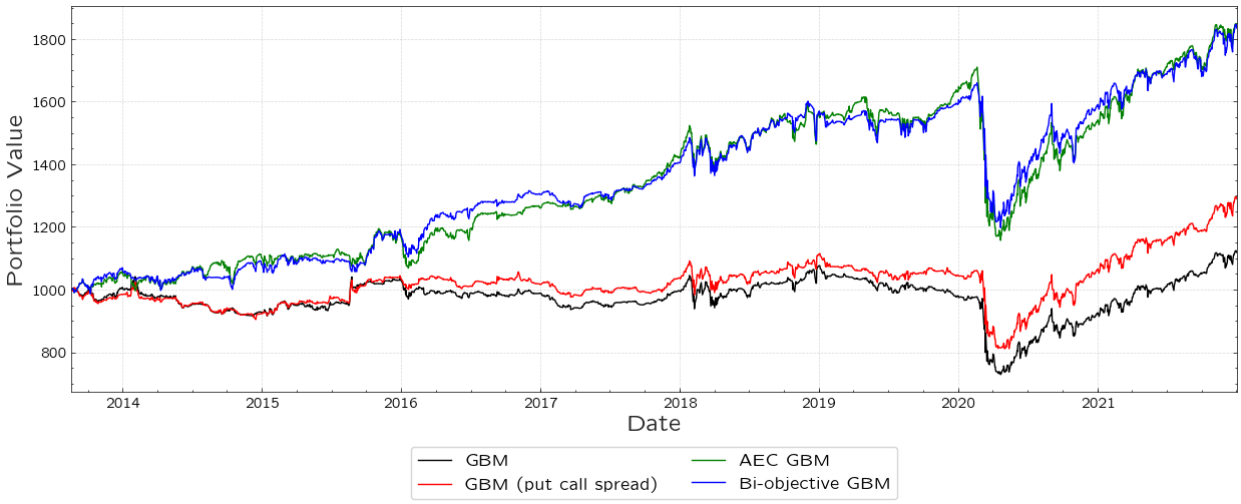


(b) GBM Long/Short Backtest Portfolio Values

Figure 1: The Figure plots the average portfolio value for the long/short investment strategy backtests that use 30-day ahead predictions from the logistic regression (LR) and GBM models.



(a) Logistic Regression Long/Short Backtest Portfolio Values



(b) GBM Long/Short Backtest Portfolio Values

Figure 2: The Figure plots the average portfolio value for the long/short investment strategy backtests that use 10-day ahead predictions from the logistic regression (LR) and GBM models.

2012; An et al., 2014). We explore new ways of using options information in financial machine learning by incorporating put and call prices into the objective function in a principled manner. Finally, our study is also related to an emerging literature that compares the utility of linear and non-linear machine learning models in financial modeling and forecasting tasks (see, e.g., Gu et al., 2020; Rasekhschaffe and Jones, 2019; Christensen et al., 2021).

Promising directions for future research include using in-the-money or out-of-the-money option prices to provides further incremental improvements in predictive power (see, e.g., Shirvani et al., 2019) and studying the effect of different weighting schemes when combining the log-loss and average expected cost.

A Predictor Variable Descriptions

This appendix provides a description of the predictor variables used in our LR and GBM models. We also state the frequency at which the variable is available and the data source used to construct each variable.

1. **ADS**: The Aruoba-Diebold-Scotti Business Conditions Index (see, e.g., Aruoba et al., 2009). This variable is available at a daily frequency. Data is obtained from the Philadelphia Federal Reserve.
2. **US_SCDI**: The U.S. state coincident diffusion index constructed by the Federal Reserve Bank of Philadelphia. This variable is the number of U.S state coincident indices posting a positive month-on-month change. This variable is available at a monthly frequency and is obtained from Federal Reserve Bank of Philadelphia.
3. **OECD_ACWI_breadth**: The fraction of OECD composite leading indicators, consumer confidence indicators and business confidence indicators with positive month-on-month changes for all OECD member countries in the MSCI ACWI index. This variable is available at a monthly frequency. OECD indicator data is obtained from the OECD Main Economic Indicators database.
4. **OECD_LCI**: The Conference Board's leading credit index for the U.S. This variable is available at a monthly frequency. Data is obtained from Macrobond.
5. **FIBER_IP** The Foundation for International Business and Economic Research U.S. industrial production index. This variable is available at a monthly frequency. Data is obtained from Macrobond.
6. **BBKMCOIX**: The Brave Butters and Kelly coincident index for the U.S. This variable is available at a monthly frequency. Data is obtained from FRED.
7. **3yTnote_yield**: The 26-week change in the three-year treasury note yield. This variable is available at a daily frequency. Data is obtained from FRED.
8. **ISM_US_PMI**: The ISM U.S. manufacturing PMI index. This variable is available at a monthly frequency. Data is obtained from Bloomberg.
9. **Default_spread**: The difference between the Moody's Baa corporate bond yield and the Moody's Aaa corporate bond yield. This variable is computed at a daily frequency. Data is obtained from FRED.
10. **term_spread**: The 10-year U.S. treasury bond yield minus the 3-month U.S. treasury bill yield. This variable is computed at a daily frequency. Yield data is obtained from FRED.
11. **MOVE**: The Merrill Lynch Option Volatility Estimate index. This variable is available at a monthly frequency. Data is obtained from Bloomberg.
12. **Baa_yield**: The 26-week change in the Moody's Baa corporate bond yield. This variable is available at a daily frequency. Data is obtained from FRED.
13. **CBLI_diffusion**: The diffusion index from the Conference Board's U.S. leading economic index. This variable is available at a monthly frequency. Data is obtained from Macrobond.
14. **lag_return**: We compute 22 lags of daily returns (**lag_return1**, ..., **lag_return22**). Price data used to compute this variable is obtained from Compustat.
15. **RV**: The daily range volatility estimator of Garman and Klass (1980). We compute the average of this daily volatility estimator over the past 1-trading day (**RV1**), 5-trading days (**RV5**) and 22-trading days (**RV22**). This variable is computed at a daily frequency. Price data used to compute this variable is obtained from Compustat.

16. **realized_kurtosis**: The realized kurtosis of daily returns over the past 22-trading days (**realized_kurtosis22**), 63-trading days (**realized_kurtosis63**) and 128-trading days (**realized_kurtosis128**).This variable is computed at a daily frequency. Price data used to compute this variable is obtained from Compustat.

17. **realized_skew**: The realized skewness of daily returns over the past 22-trading days (**realized_skew22**), 63-trading days (**realized_skew63**) and 128-trading days (**realized_skew128**).This variable is computed at a daily frequency. Price data used to compute this variable is obtained from Compustat.

18. **realized_semivariance**: The realized semi-variance of daily returns over the past 22-trading days (**realized_semivariance22**), 63-trading days (**realized_semivariance63**) and 128-trading days (**realized_semivariance128**).This variable is computed at a daily frequency. Price data used to compute this variable is obtained from Compustat.

19. **MA**: The moving average (MA) return. We compute the 5-trading day MA (**MA5**), 22-trading day MA (**MA2**), 63-trading day MA (**MA63**) and the 128-trading day MA (**MA128**). This variable is computed at a daily frequency. Price data used to compute this variable is obtained from Compustat.

20. **MACD_histogram**: The difference between the 12-day EWMA exponentially weighted moving average (EWMA) minus the 26-day EWMA and the 9-day moving average of the 12-day EWMA minus the 26-day EWMA. This variable is computed at a daily frequency. Price data used to compute this variable is obtained from Compustat.

21. **Williams_R**: The Williams R technical indicator. This variable is computed at a daily frequency. Price data used to compute this variable is obtained from Compustat.

22. **ATR**: The Average True Range technical indicator. This variable is computed at a daily frequency. Price data used to compute this variable is obtained from Compustat.

23. **RSI**: The RSI technical indicator. This variable is computed at a daily frequency. Price data used to compute this variable is obtained from Compustat.

B Long-Only Investment Strategy Results

In this appendix we report the results from an equal-weighted long-only market timing strategy that uses the one-month and 10-day ahead return direction predictions for the three equity market indices. This strategy takes a long position in the equity market index when a model predicts the class label 1, and a long position in the Vanguard Total Bond market ETF (BND) when the model predicts the class label 0. We use the same transaction cost assumptions and the same method to mitigate path dependence as described in Section 5.

Tables 13 and 14 present measures of investment performance for the long-only investment strategy based upon the one-month and 10-day ahead return direction predictions. We find that our AEC and bi-objective models still earn higher annualized returns and Sharpe ratios than benchmark models. However, the size of this difference is smaller at the one-month ahead forecast horizon. For a long-only investment strategy all models tend to have similar maximum drawdown statistics. However, our AEC and bi-objective models have substantially smaller maximum drawdown periods at the 10-day forecast horizon. Moreover, the cost-sensitive models always have higher Sortino and MAR ratios. Hence, even when using a long-only investment strategy, our AEC and bi-objective LR and GBM models outperform benchmark models by generating superior risk adjusted investment performance and lower downside risk.

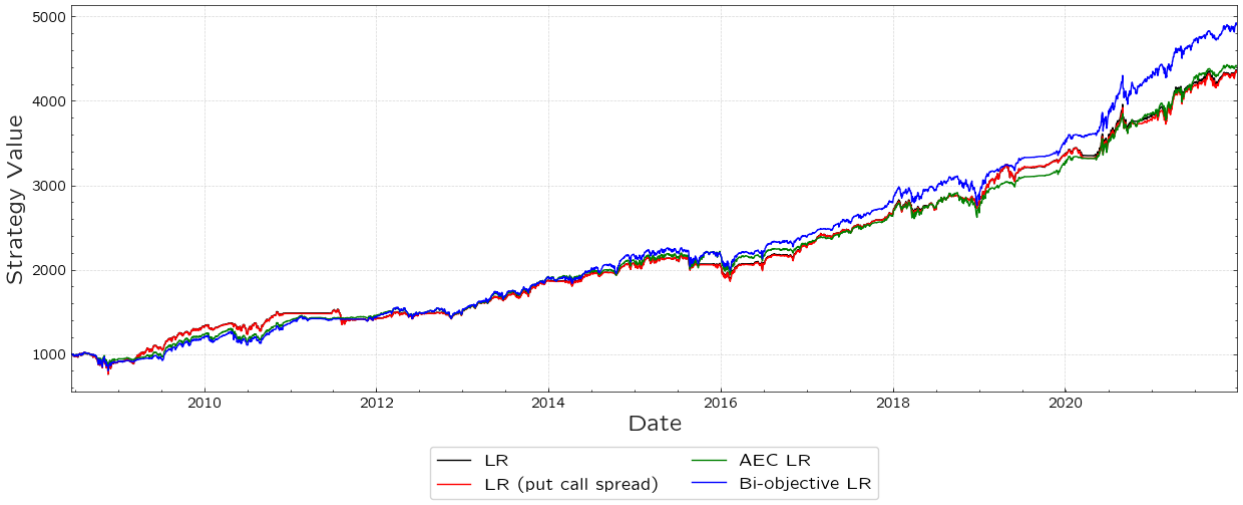
Figures 3 and 4 plot the average cumulative portfolio value for long-only strategies using the one-month and 10-day ahead predictions, respectively. In all cases our AEC and bi-objective models outperform benchmark models by achieving a higher terminal portfolio value by the end of the backtest. Consistent with the results for the long/short investment strategy, we find that bi-objective models outperform the AEC models in three out of four cases. Moreover, we again find that the GBM models tend to earn higher returns than the LR models, and that a long-only investment strategy that uses the one-month ahead forecasts earns higher returns than a long-only investment strategy based upon the 10-day ahead forecasts.

Table 13: 30-day ahead Long-Only Investment Strategy Backtest Results
The table reports the investment performance statistics for the long-only investment strategy backtest that uses 30-day ahead predictions from the logistic regression and gradient boosting models. The acronym LR stands for logistic regression. The acronym GBM stands for gradient boosting machine.

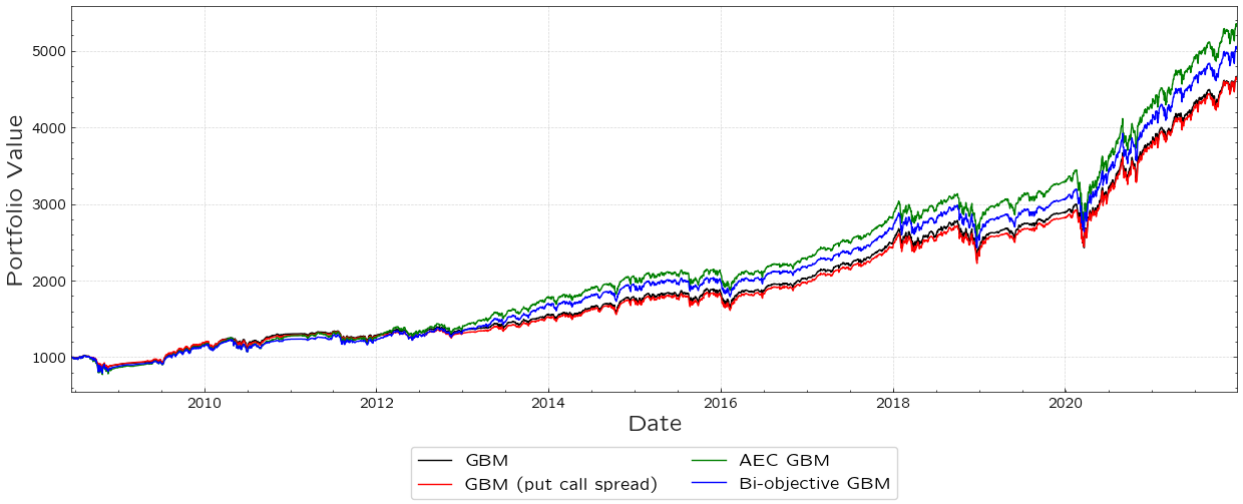
	(i) Logisitic Regression			
	LR	LR (put call spread)	AEC LR	Bi-objective LR
Annualised Return (%)	13.138446	13.114573	13.593156	14.438978
Annualised Std	11.799031	11.821157	11.009595	11.661907
Annualised Sharpe Ratio	1.091048	1.087995	1.181359	1.186636
Annualised Sortino Ratio	1.701953	1.697402	1.857297	1.870738
Max Drawdown (%)	-21.24132	-21.357174	-16.75813	-18.346208
Max Drawdown Period (days)	85.725191	91.251908	68.736641	72.179389
MAR Ratio	0.738531	0.730719	0.882177	0.88244
Max DD/Vol	1.755288	1.762789	1.505025	1.547568
Annualised Downside Deviation	7.602151	7.616665	7.01355	7.409411
Beta	0.429683	0.432062	0.41288	0.438231
	(ii) Gradient Boosting Machine			
	GBM	GBM (put call spread)	AEC GBM	Bi-objective GBM
Annualised Return (%)	13.544344	13.521493	14.318291	14.119873
Annualised Std	13.83034	13.674085	14.885795	14.103368
Annualised Sharpe Ratio	0.965659	0.975043	0.955211	0.984737
Annualised Sortino Ratio	1.488028	1.509821	1.480223	1.531325
Max Drawdown (%)	-25.7031	-24.279863	-27.00178	-25.192751
Max Drawdown Period (days)	48.183206	38.442748	42.530534	47.870229
MAR Ratio	0.573001	0.611103	0.588048	0.607738
Max DD/Vol	1.838044	1.75331	1.79078	1.768485
Annualised Downside Deviation	9.01478	8.873676	9.637017	9.09525
Beta	0.644486	0.630014	0.730442	0.667623

Table 14: 10-day ahead Long-Only Investment Strategy Backtest Results
The table reports the investment performance statistics for the long-Only investment strategy backtest that uses 10-day ahead predictions from the logistic regression and gradient boosting models. The acronym LR stands for logistic regression. The acronym GBM stands for gradient boosting machine.

	(i) Logisitic Regression			
	LR	LR (put call spread)	AEC LR	Bi-objective LR
Annualised Return (%)	9.6204	9.3903	10.8715	13.5226
Annualised Std	12.1273	12.1425	12.2533	12.7268
Annualised Sharpe Ratio	0.7985	0.7829	0.8784	1.0334
Annualised Sortino Ratio	1.1688	1.1447	1.2902	1.5326
Max Drawdown (%)	-20.0220	-20.5431	-21.0387	-21.1096
Max Drawdown Period (days)	13.4733	35.6031	17.6641	15.4695
MAR Ratio	0.5030	0.4780	0.5312	0.6824
Max DD/Vol	1.6403	1.6835	1.7110	1.6455
Annualised Downside Deviation	8.3292	8.3600	8.3647	8.6363
Beta	0.5288	0.5305	0.5902	0.5928
	(ii) Gradient Boosting Machine			
	GBM	GBM (put call spread)	AEC GBM	Bi-objective GBM
Annualised Return (%)	9.5120	10.0906	12.3127	12.4222
Annualised Std	13.8144	13.6658	15.4090	14.8237
Annualised Sharpe Ratio	0.7086	0.7500	0.8049	0.8390
Annualised Sortino Ratio	1.0168	1.0829	1.1730	1.2296
Max Drawdown (%)	-27.4180	-26.7558	-30.9441	-27.8431
Max Drawdown Period (days)	58.0305	73.0725	21.9084	23.1985
MAR Ratio	0.3662	0.3840	0.3978	0.4517
Max DD/Vol	1.9719	1.9542	2.0085	1.8764
Annualised Downside Deviation	9.6562	9.4876	10.5780	10.1339
Beta	0.6882	0.6916	0.8286	0.7848

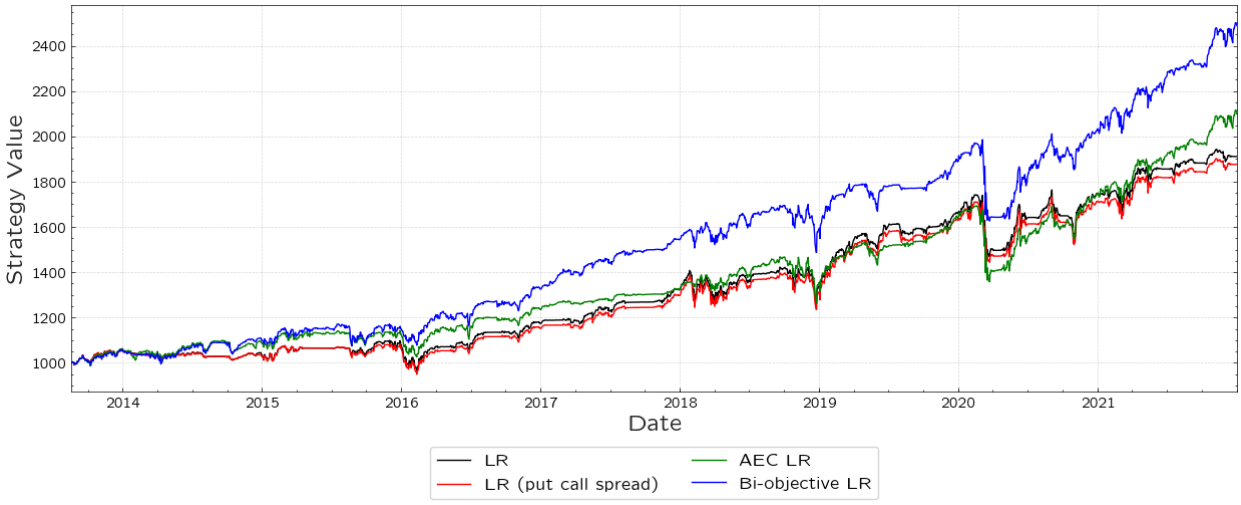


(a) Logistic Regression Long-only Backtest Portfolio Values

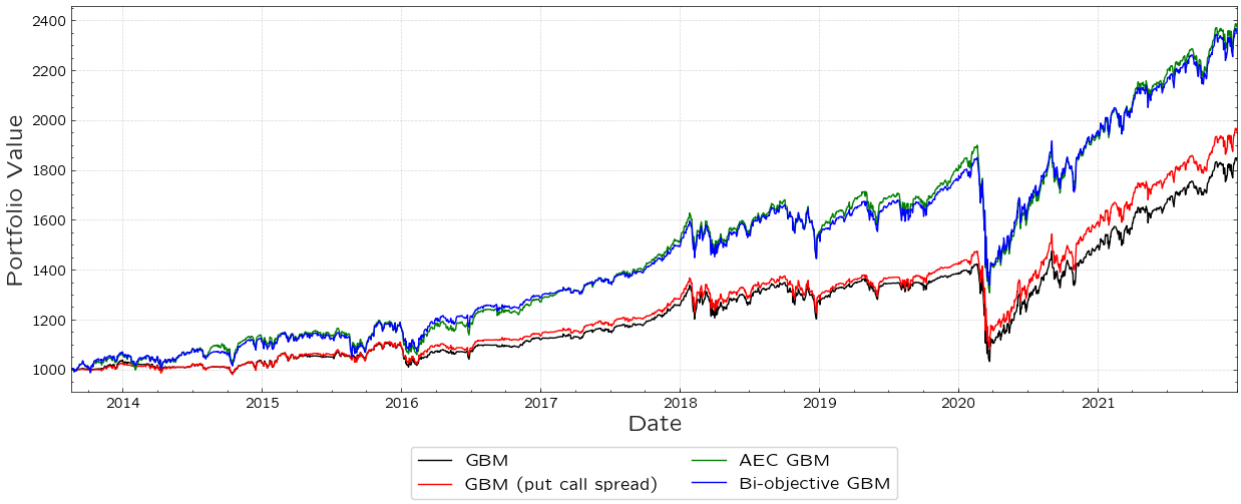


(b) GBM Long-only Backtest Portfolio Values

Figure 3: The figure plots the average portfolio value for the long-only investment strategy backtests that use 30-day ahead predictions from the logistic regression (LR) and GBM models.



(a) Logistic Regression Long-only Backtest Portfolio Values



(b) GBM Long-only Backtest Portfolio Values

Figure 4: The figure plots the average portfolio value for the long-only investment strategy backtests that use 10-day ahead predictions from the logistic regression (LR) and GBM models.

References

Akiba, T., S. Sano, T. Yanase, T. Ohta, and M. Koyama (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2623–2631.

An, B.-J., A. Ang, T. G. Bali, and N. Cakici (2014). The joint cross section of stocks and options. *The Journal of Finance 69*(5), 2279–2337.

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

Aruoba, S. B., F. X. Diebold, and C. Scotti (2009). Real-time measurement of business conditions. *Journal of Business & Economic Statistics 27*(4), 417–427.

Atmaz, A. and S. Basak (2019). Option prices and costly short-selling. *Journal of Financial Economics 134*(1), 1–28.

Augustin, P. and M. G. Subrahmanyam (2020). Informed options trading before corporate events. *Annual Review of Financial Economics 12*, 327–355.

Baetje, F. and L. Menkhoff (2016). Equity premium prediction: Are economic and technical indicators unstable? *International Journal of Forecasting 32*(4), 1193–1207.

Bahnsen, A. C., D. Aouada, and B. Ottersten (2015). Example-dependent cost-sensitive decision trees. *Expert Systems with Applications 42*(19), 6609–6619.

Bali, T. G., H. Beckmeyer, M. Mörke, and F. Weigert (2023). Option Return Predictability with Machine Learning and Big Data. *The Review of Financial Studies*.

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

Basak, S., S. Kar, S. Saha, L. Khaidem, and S. R. Dey (2019). Predicting the direction of stock market prices using tree-based classifiers. *The North American Journal of Economics and Finance 47*, 552–567.

Bergstra, J., R. Bardenet, Y. Bengio, and B. Kégl (2011). Algorithms for hyper-parameter optimization. *Advances in neural information processing systems 24*.

Bianchi, D., M. Büchner, and A. Tamoni (2021). Bond risk premiums with machine learning. *The Review of Financial Studies 34*(2), 1046–1089.

Biau, G. and E. Scornet (2016). A random forest guided tour. *Test 25*(2), 197–227.

Bollerslev, T., G. Tauchen, and H. Zhou (2009). Expected stock returns and variance risk premia. *The Review of Financial Studies 22*(11), 4463–4492.

Brogaard, J. and A. Zareei (2022). Machine learning and the stock market. *Journal of Financial and Quantitative Analysis*, 1–42.

Chen, L., M. Pelger, and J. Zhu (2023). Deep learning in asset pricing. *Management Science*.

Christensen, K., M. Siggaard, and B. Veliyev (2021). A machine learning approach to volatility forecasting. *Available at SSRN*.

Cremers, M., R. Goyenko, P. Schultz, and S. Szaura (2019). Do option-based measures of stock mispricing find investment opportunities or market frictions? *Available at SSRN 3347194*.

De Prado, M. L. (2018). *Advances in nancial machine learning*. John Wiley & Sons.

DeMiguel, V., L. Garlappi, and R. Uppal (2009). Optimal versus naive diversification: How inefficient is the 1/n portfolio strategy? *The Review of Financial studies 22*(5), 1915–1953.

Elkan, C. (2001). The foundations of cost-sensitive learning. In *International joint conference on arti cial intelligence*, Volume 17, pp. 973–978. Lawrence Erlbaum Associates Ltd.

Fawcett, T. (2006). An introduction to roc analysis. *Pattern Recognition Letters 27*(8), 861–874.

Fischer, T. and C. Krauss (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research 270*(2), 654–669.

Freyberger, J., A. Neuhierl, and M. Weber (2020). Dissecting characteristics nonparametrically. *The Review of Financial Studies 33*(5), 2326–2377.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189–1232.

Garman, M. B. and M. J. Klass (1980). On the estimation of security price volatilities from historical data. *Journal of Business*, 67–78.

Giacomini, R. and H. White (2006). Tests of conditional predictive ability. *Econometrica 74*(6), 1545–1578.

Goncalves-Pinto, L., B. D. Grundy, A. Hameed, T. van der Heijden, and Y. Zhu (2020). Why do option prices predict stock returns? the role of price pressure in the stock market. *Management Science 66*(9), 3903–3926.

Goyenko, R. and C. Zhang (2020). The joint cross section of option and stock returns predictability with big data and machine learning. *Available at SSRN 3747238*.

Gu, S., B. Kelly, and D. Xiu (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies 33*(5), 2223–2273.

Hand, D. J., C. Whitrow, N. M. Adams, P. Juszczak, and D. Weston (2008). Performance criteria for plastic card fraud detection tools. *Journal of the Operational Research Society 59*(7), 956–962.

Höppner, S., B. Baesens, W. Verbeke, and T. Verdonck (2022). Instance-dependent cost-sensitive learning for detecting transfer fraud. *European Journal of Operational Research 297*(1), 291–300.

Iworiso, J. and S. Vrontos (2020). On the directional predictability of equity premium using machine learning techniques. *Journal of Forecasting 39*(3), 449–469.

James, G., D. Witten, T. Hastie, and R. Tibshirani (2013). *An introduction to statistical learning*, Volume 112. Springer.

James, R., H. Leung, J. W. Y. Leung, and A. Prokhorov (2023). Forecasting tail risk measures for financial time series: An extreme value approach with covariates. *Journal of Empirical Finance*.

Jensen, T. I., B. T. Kelly, S. Malamud, and L. H. Pedersen (2022). Machine learning and the implementable efficient frontier. *Available at SSRN 4187217*.

Johnson, T. L. and E. C. So (2012). The option to stock volume ratio and future returns. *Journal of Financial Economics 106* (2), 262–286.

Jones, C. S., H. Mo, and T. Wang (2018). Do option prices forecast aggregate stock returns? *Available at SSRN 3009490*.

Kacperczyk, M. and E. S. Pagnotta (2019). Chasing private information. *The Review of Financial Studies 32* (12), 4997–5047.

Ke, G., Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems 30*.

Kelly, B. T., S. Malamud, and K. Zhou (2022). The virtue of complexity in return prediction. Technical report, National Bureau of Economic Research.

Kliesen, K. L., M. T. Owyang, E. K. Vermann, et al. (2012). Disentangling diverse measures: A survey of financial stress indexes. *Federal Reserve Bank of St. Louis Review 94* (5), 369–397.

Krauss, C., X. A. Do, and N. Huck (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the s&p 500. *European Journal of Operational Research 259* (2), 689–702.

Leippold, M., Q. Wang, and W. Zhou (2022). Machine learning in the chinese stock market. *Journal of Financial Economics 145* (2), 64–82.

Li, W.-X., J. J. French, and C. C.-S. Chen (2017). Informed trading in s&p index options? evidence from the 2008 financial crisis. *Journal of Empirical Finance 42*, 40–65.

Lin, T.-C. and X. Lu (2015). Why do options prices predict stock returns? evidence from analyst tipping. *Journal of Banking & Finance 52*, 17–28.

Long, H., A. Zaremba, W. Zhou, and E. Bouri (2022). Macroeconomics matter: Leading economic indicators and the cross-section of global stock returns. *Journal of Financial Markets*, 100736.

Mascio, D. A., F. J. Fabozzi, and J. K. Zumwalt (2021). Market timing using combined forecasts and machine learning. *Journal of Forecasting 40* (1), 1–16.

Mausser, H. (2006). Normalization and other topics in multi-objective optimization. In *Fields-MITACS Industrial Problems Workshop*, pp. 89. Citeseer.

Neely, C. J., D. E. Rapach, J. Tu, and G. Zhou (2014). Forecasting the equity risk premium: the role of technical indicators. *Management Science 60* (7), 1772–1791.

Pan, J. and A. M. Poteshman (2006). The information in option volume for future stock prices. *The Review of Financial Studies 19* (3), 871–908.

Petraki, A. (2020). The transaction costs manual: What is behind transaction cost figures and how to use them. Technical report, Schroder Investment Management Limited.

Rasekhschaffe, K. C. and R. C. Jones (2019). Machine learning for stock selection. *Financial Analysts Journal 75* (3), 70–88.

Roll, R., E. Schwartz, and A. Subrahmanyam (2010). O/s: The relative trading activity in options and stock. *Journal of Financial Economics 96* (1), 1–17.

Shi, H. (2007). *Best- rst decision tree learning*. Ph. D. thesis, The University of Waikato.

Shirvani, A., S. T. Rachev, and F. J. Fabozzi (2019). A rational finance explanation of the stock predictability puzzle. *arXiv preprint arXiv:1911.02194*.

Sundaram, R. K. and S. R. Das (2011). *Derivatives: principles and practice*. McGraw-Hill Irwin New York, NY.

Swade, A., S. Nolte, M. Shackleton, and H. Lohre (2023). Why do equally weighted portfolios beat value-weighted ones? *The Journal of Portfolio Management 49*(5), 167–187.

Timmermann, A. (2006). Forecast combinations. *Handbook of Economic Forecasting 1*, 135–196.

Timmermann, A. (2018). Forecasting methods in finance. *Annual Review of Financial Economics 10*, 449–479.

Varian, H. R. (2014). Big data: New tricks for econometrics. *Journal of Economic Perspectives 28*(2), 3–28.

Vinayak, R. K. and R. Gilad-Bachrach (2015). Dart: Dropouts meet multiple additive regression trees. In *Arti cial Intelligence and Statistics*, pp. 489–497. PMLR.

Wang, X., R. J. Hyndman, F. Li, and Y. Kang (2022). Forecast combinations: an over 50-year review. *International Journal of Forecasting*.