Applying Machine Learning to Trading Strategies: Using Logistic Regression to Build Momentum-based Trading Strategies [*]

Patrick Beaudan[†] and Shuoyuan He[‡]

Current draft: January 2019

ABSTRACT: This paper proposes a machine learning approach to building investment strategies that addresses several drawbacks of a classic approach. To demonstrate our approach, we use a logistic regression algorithm to build a time-series dual momentum trading strategy on the S&P 500 Index. Our algorithm outperforms both buy-and-hold and several base-case dual momentum strategies, significantly increasing returns and reducing risk. Applying the algorithm to other U.S. and international large capitalization equity indices generally yields improvements in risk-adjusted performance.

***JEL Classifications*:** C10, G12, G14, G17

***Data*:** From public sources identified in the text

***Keywords*:** Trading strategy; Active investing; Dual momentum; Machine learning; Logistic regression; Trend-following

[*] We thank Étienne de Bazelaire for his assistance with this research.

[†] Patrick Beaudan is the CEO of Belvedere Advisors, LLC / Emotomy. Tel: (415) 839-5239 and email: pb@emotomy.com.

[‡] Shuoyuan He is at A. B. Freeman School of Business, Tulane University. Tel: (504) 247-1292 and email: shhe@tulane.edu.

1. Introduction

Price momentum is the empirical finding that the price of an asset that has been rising in the recent past will continue to rise in the near future (Jegadeesh and Titman 1993, Fama and French 2008, Moskowitz et al. 2012, Asness et al. 2013). In practice, the classic approach to momentum investing usually consists of evaluating the performance of an asset over one or multiple historical time windows, typically of less than a year, to decide whether and how much to allocate to that asset. The evaluation of historical performance is repeated at regular intervals ranging from a few weeks to a few months depending on factors such as tax implications, trading costs and the expected impact on the strategy's performance.

There are three major drawbacks to this classic approach. First, it usually relies on a back-testing trial-and-error process that eliminates features once and for all that are not deemed to be valuable. While the number of driving features that are eventually selected may seem small[1], the trial-and-error process may have examined and eliminated many combinations of features before selecting a final few to manage a strategy. The process may also disregard features altogether perhaps on the basis of experience without considering the multiple ways in which a feature could be computed. Momentum, for instance, could be calculated using many techniques around date intervals and smoothing of the price history.

The second problem is the ease with which historical data can be overfit, and the lack of a mechanism within the classic approach that checks for and minimizes overfitting. Given a price time series and other historical data, strategy parameters can always be adjusted to maximize target

[1] The driving features selected can be a handful of indicators such as historical momenta over various time-windows, a minimum profitability target for dual-momentum approaches, a trailing stop loss threshold and a timing rule for rebalancing portfolios.

performance metrics. When such adjustments result in overfitting, the strategy is unlikely to perform as expected going forward.

The third problem is the under-performance of classic momentum strategies at market turning points. This problem arises from the inherently backward-looking nature of momentum trading strategies. Consider a strategy investing in a single asset based on an assessment of price momentum being up or down. When prices rise continuously over a long-enough period, the strategy is invested into that asset and gains in value. When prices fall continuously over a long-enough period, the strategy may be in cash or short the asset, respectively maintaining its value or appreciating. Turning points, however, are where the strategy loses value and gives up some of its cumulative gains since it takes time to conclude that momentum has turned negative after a period of rising prices, or has turned positive after a period of falling prices[2].

Our intent in this paper is to outline how machine learning techniques can be used to mitigate the issues discussed above and improve the risk-return trade-off in momentum strategies beyond what is possible using classic approaches. Machine learning strategies have the inherent ability to learn from and adapt to changing market conditions using more numerous and more complex combinations of features than is practically feasible in a classic approach. Polynomial combinations of various features, for example, can be created in an attempt to identify non-obvious, complex non-linear patterns. While the total number of features used in machine learning may

[2] Over time horizons measured in decades that range across multiple market cycles, the cumulative impact of the temporary losses in these transition turning periods is offset by the gains made in favorable periods. That is because the universe of investable assets that are selected for an investment strategy will of course have a positive future expected rate of return over the long run, a characteristic that momentum approaches are very good at capturing. From a practical point of view, however, investors have been known to display limited patience and they may experience sudden spikes in personal risk aversion just as losses accumulate or money sits in cash while markets rise. Increasing the confidence of investors in an investment approach when performance suffers is a key objective of any asset management business. This can be partially accomplished through a better, demonstrably more robust design of investment strategies as suggested in this paper.

seem high relative to a classic approach, techniques can be used to check for and minimize overfitting. Consequently, machine learning strategies can be designed to use far more features than their classic counterparts without necessarily overfitting data.

To demonstrate our approach, we focus in this paper on applying logistic regression techniques to the simplest form of time-series dual-momentum investing, namely using the price history of a single asset to assess whether to invest in that asset or move to cash at the end of each trading day for our long-only strategy, or establish a short position instead of cash in the case of the long-short strategy. To highlight the key decision points in designing this type of strategy, we do not use any non-price data nor derivative indicators such as futures or options data, but solely the daily adjusted closing price of the asset. The features we selected in our algorithm are the same types of performance metrics used in classic approaches that may combine price momenta and drawdowns over various historical time intervals.

The rest of this paper is organized as follows. Section 2 presents the securities price data. Section 3 describes our approach in building the time-series dual momentum trading strategy. Section 4 describes the polynomial features, their corresponding learned parameters, and the relationship between features and investment performance. Section 5 documents the long-only strategy's investment performance when applied to the S&P 500 Index (SPX). Section 6 compares the performance of the long-only and long-short strategies applied to the S&P 500 Index, and documents the long-only strategy's performance when applied to equity indices other than SPX. We also discuss the impact on investment performance of periodically retraining the algorithm's parameters. Section 7 concludes the paper.

2. Data

The data used in this study is obtained from the Bloomberg database. Our main analysis is carried out on the SPX between December 30, 1927 and December 12, 2018, a sample that consists of 22,846 daily prices spanning over 90 years. In addition, we analyze the following seven equity indices: S&P Small Cap 600 Index (SML), S&P Mid Cap 400 Index (MID), FTSE 100 Index (UKX), FTSEurofirst 300 Index (E300), Tokyo Stock Exchange Price Index (TPX), Dow Jones Industrial Average Index (INDU), and Dow Jones Transportation Average Index (TRAN). Table 1 summarizes the time periods analyzed in this paper for each of these indices.

The performance metrics used in this paper are described in Table 2. For simplicity, all Sharpe ratios are calculated using a constant one percent annual risk-free rate. All Sortino ratios are calculated using a constant one percent Minimum Acceptable Return. All computations are based on the convention of 252 business days per calendar year and 21 business days per calendar month. Table 3 summarizes the key performance indicators of our sample indices over their entire respective sample periods.

3. Methodology

In this section, we describe two base cases, the buy-and-hold approach and an implementation of a classic time-series dual-momentum strategy. That is followed by a description of our machine learning approach that recasts momentum-investing into a classification problem, namely a problem where the decision of whether to invest or move to cash must be made daily based on the binary prediction of whether investing will be profitable or not over a target future time horizon.

3.1 Base Cases

3.1.1 Buy-and-hold Approach

In the buy-and-hold approach, the investment decision is to invest on a specific date and hold the position irrespective of market conditions.

3.1.2 Classic Time-series Dual-Momentum Strategy

In a classic time-series dual-momentum approach, the investment decision each day is whether to invest in an asset or move to a safer asset that can be cash or treasury bonds for instance. Whilst there are multiple ways in which this type of strategy can be implemented and refined, we keep our base case simple to provide a clear comparison with the machine learning approach. We construct the base case in the following way. The momentum, i.e. the percentage price change of a security, is calculated over a historical time horizon of twelve months, skipping the most recent month (Asness et al. 2013)[3]. If momentum is higher than a minimum annual profitability threshold δ, which we set in this paper at 5 percent[4], the decision is to invest in the asset. If momentum is lower than the threshold, the portfolio is moved to cash in the long-only strategy, or moved to a short position in the long-short strategy. This investment decision is revisited at regular intervals

[3] We skip the most recent month to avoid 1-month stock return reversal documented in the finance literature (Jegadeesh 1990, and Grinblatt and Moskowitz 2004).

[4] We select a minimum profitability threshold of 5 percent in this study for simplicity. Other values could be selected based on experience with the underlying asset being invested in. Intuitively, this means that we are requesting the algorithm to predict that any annual price rise of less than 5 percent would be a loss. This may be viewed as an attempt to increase the precision of the results, since the algorithm will only predict a gain when a price is predicted to rise well above its current level rather than just to remain flat. However, whether precision really is increased will depend on the volatility of the asset and the length of the future return horizon H. If the horizon is far away in the future, a 5 percent annual gain may not be particularly meaningful, while for short horizons it may indicate a rapid rise that could have a good chance to continue in the near term. If volatility suddenly spikes however, a 5 percent gain, even if achieved over long time horizons, can be reversed in less than one trading day.

regardless of market conditions. In this paper we select a rebalancing frequency of one month for simplicity.

3.2 Machine Learning Approach

3.2.1 Logistic Regression Algorithm

In the machine learning approach, the investment decision of whether to invest in an asset or move to cash is made daily in the following way. We build a logistic regression model that makes a binary daily prediction of whether the return over a certain future time horizon H[5] is likely to be positive. To incorporate the approach of dual-momentum investing, we define a profitability vector Y that the algorithm will attempt to predict as

$$Y = \begin{pmatrix} y_1 \\ \vdots \\ \vdots \\ y_m \end{pmatrix}$$

where each daily value y_i is either one or zero depending on whether future profitability p_i[6] exceeds a minimum annual profitability threshold δ, namely:

$$y_i = \begin{cases} 1 \; if \; p_i \geq \delta \\ 0 \; if \; p_i < \delta \end{cases}$$

We seed our model with two sets of basic features, momenta and drawdowns, that can be readily calculated from the historical prices of the asset. Since momentum is an auto-correlation problem, where we essentially try to predict the performance of an asset using that asset's historical

[5] In practice, most classic momentum strategies will re-evaluate a portfolio somewhere between each trading day and once a month – or 21 business days. For tax reasons, which are not a consideration in this paper, portfolio rebalancing may be limited to once per quarter or even slower. We limit our discussion of logistic regression to predictions of performance over future time horizons that range from one to 21 business days, with the expectation that the predictive power of the algorithm will fade as the time horizon gets longer.

[6] Future profitability p_i is measured as the annualized percentage change in price over a time horizon H, specifically: $p_i = \left(\frac{Price_{i+H}}{Price_i}\right)^{\frac{252}{H}} - 1$

prices, other financial metrics that are purely functions of price such as the Sharpe or Sortino ratio, or upside and downside volatility, could also be incorporated in our list of features.

The belief implied by our selection of features is that observing the change in the shape of the price history using multiple historical time windows for momenta and drawdowns is more pertinent than considering other metrics to predict short-term profitability. Using momentum over multiple time frames dovetails the classic approach of using cross-over momentum for investment decisions, while adding drawdown information is conceptually equivalent to enabling the use of stop losses in a classic approach where trailing stop losses are typically triggered after a drawdown threshold is breached. The intent therefore is that by providing momentum and drawdown information over multiple time frames, the algorithm will learn which of these features are relatively more important than others as market conditions evolve and weigh each accordingly as it tries to predict future performance.

The momentum features we use are based on the following numbers of business days: 30, 60, 90, 120, 180, 270, 300 and 360. Drawdowns are evaluated on 15, 60, 90 and 120-day time windows. These time-frames are based on experience and broadly match what would be expected to be pertinent historical horizons on which to evaluate short and long-term momenta, losses and stop-loss triggers. Our objective is not to maximally optimize our set of initial features but rather to demonstrate what a logistic regression algorithm can deliver using a set of features that should seem intuitively reasonable to experienced asset managers. Note that the profitability vector Y as well as all features are dimensionless numbers. Features are normalized to value between -1 and +1 with zero mean prior to training the algorithm.

To enable the algorithm to identify non-linear patterns in a simple way, we let it create new features automatically from the basic features it was seeded with. This is done by adding as new features the polynomial combinations of the basic features up to a specified polynomial degree. For instance, suppose that we let the algorithm create quadratic combinations of our initial features, and that feature x_1 represents the six-month momentum on a particular day while feature x_2 represents the drawdown over a two-month window. The algorithm will then create new features as x_1x_2, x_1^2 and x_2^2 and use these in addition to the original features x_1 and x_2 to try to predict performance. The degree of the polynomial that the algorithm is allowed to use in creating new features is a parameter in logistic regression that we discuss further in Section 3.2.4.

Denoting the value of the features on each day with time index i as a vector X_i with elements $x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)}$, and the weights assigned to these features as a vector θ with elements $\theta_1, \theta_2, \ldots, \theta_n$, our hypothesis is that whether y_i is positive or not can be predicted as follows:

$$h_\theta(x) = g(\theta^T x)$$

where g is the sigmoid function

$$g(z) = \frac{1}{1 + e^{-z}}$$

and where we predict $h_\theta(x) = 1$, when $\theta^T x > 0$, and $h_\theta(x) = 0$, when $\theta^T x \leq 0$.

To minimize the possibility of overfitting the available data, we split our data samples into two distinct sets referred to as training sets and test sets[7]. The daily weights θ_i of each feature x_i are found by minimizing the prediction error between actual future performance and the performance predicted by the function h_θ over training sets of historical data, while investment

[7] We omit the cross-validation set in this study due to limited data. To show the generalizability of the model, we test our strategy on other U.S. and international equity indices (see Section 6.1).

performance reported in this study is calculated solely on the test sets. Since we let the algorithm create new features through polynomial combinations of the seed parameters, the total number of features can rise quickly as the specified degree of acceptable polynomials increases. This in turn could lead to overfitting the training set data and result in poor strategy performance when the model is applied to test sets. In order to address this issue we use regularization, which enables the algorithm to minimize the relative influence of some features without a-priori modifying the set of features or changing the form of our hypothesis. Incorporating a regularization parameter λ to reduce potential overfitting, we minimize the prediction error on our training set data and derive the corresponding weight vectors θ by minimizing the following cost function:

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left\{y^{(i)}\,log\left(h_\theta\left(x^{(i)}\right)\right) + \left(1-y^{(i)}\right)\,log\left(1-h_\theta\left(x^{(i)}\right)\right)\right\} + \frac{\lambda}{2\,m}\sum_{j=1}^{n}{\theta_j}^2$$

3.2.2 Error Metrics

The design of our logistic regression algorithm requires a number of inputs that we posited in Section 3.2.1 based on experience. These include the selected basic features as well as a minimum annualized performance threshold used when assessing future profitability. The algorithm also requires inputs that we have mentioned but not quantified, such as the size of the training set, the value of the regularization parameter λ and the degree of polynomials used to build non-linear features.

To verify that the values selected for these parameters do not lead to overfitting our data, we will measure the accuracy of the algorithm on each training and test set using precision, recall and F-Scores as defined in Table 2. For all days predicted to have positive future performance, precision measures which fraction actually had positive performance. In contrast, for all days for which future performance is actually positive, recall measures which fraction that was correctly predicted as having positive future performance. The F-Score is a combination of these two metrics such that higher F-Scores are more desirable and very low precision or recall yields a relatively low F-Score.

3.2.3 Periodic Retraining of the Algorithm

When computing the investment strategy with our machine learning approach, model weights are trained on a subset of the available data, the training set, that is distinct from the data used to compute that strategy. This leads to two design questions, namely how large should the training set be, and how often should the training set be refreshed in order to update model weights with recent data as the strategy computation progresses in time?

Regarding the size of the training set, larger training sets are helpful in preventing overfitting but result in a smaller range of dates available for computing the investment strategy. When designing a strategy focused on equities, for which market cycles lasting 7 to 10 years are defining characteristics and primary drivers of returns, it is sensible to train the algorithm over a number of these market cycles. In the case of SPX where data is available over 90 years between 1927 and 2018, using 40 percent of the data for the training set would allocate 36 years to the initial training of the algorithm, equivalent to three or more typical equity market cycles. The strategy could then be computed from the mid 1960's to 2018, or about 50 years that include multiple market cycles and dramatic changes in all aspects of the financial markets. The results discussed in this paper are based on a 40 percent data allocation to training sets, although we found that allocating anywhere over 30 percent for training, namely three or more typical market cycles, yields qualitatively similar results.

Figure 1 displays the costs and error metrics of training sets containing 30 percent to 60 percent of all available dates in a randomized order, the remaining randomized dates being equally split between two test sets. In all cases, the values in the test sets are comparable to those of the corresponding training set, suggesting that overfitting is not likely an issue. The figure also shows that the values of the error metrics do not vary significantly with the size of the training set, suggesting that training set sizes in that range are not likely to materially impact the behavior of our model.

We now turn to a discussion of the frequency at which we retrain the parameters of our algorithm. Despite the significant size of a training set containing 40 percent of available data, it is unlikely that an algorithm trained on data from 50 years ago could be relied upon to make investment decisions today. The solution to this issue is to periodically retrain the algorithm at the same time as the strategy is computed marching forward in time. This can be done by sliding the time window of the training set forward while the strategy advances in time, retraining the parameters, i.e. recomputing the weights θ over the new training set, and using these new feature weights to carry on computing the strategy.

This introduces a new parameter in the algorithmic design, the frequency at which the training set should be changed. Since the algorithm is trained on approximately three market cycles, intuitive values for the retraining frequency could be between a half or a full market cycle, i.e. between five and ten years. While this seems reasonable for a strategy allocating to a single asset class for which all components can be expected to fall within broadly concurrent market cycles, it is not clear that a fixed retraining frequency specified a-priori would be sensible for multi asset class investment strategies. The length and timing of market cycles in bonds, equities and commodities are vastly different, while currencies and other assets might be driven by the changing economic policies of central banks and governments over time.

To improve the likelihood that our algorithm adapts to and performs with different assets, we developed an approach that enables it to learn autonomously when the time has come to retrain itself, as follows. The algorithm first trains its parameters on the initial training set, in our case 40 percent of the available data. The strategy computation is initiated on the first date of the test set (after the initial training set) using these parameters, and the algorithm checks the convergence of the cost function $J(\theta)$ on the increasing set of dates of the test set as the strategy is computed. Convergence is deemed to be achieved when the annualized rate of decay of the cost function falls below a specified tolerance level which we set at 0.01 percent for simplicity. Once convergence is achieved, the training set is slid forward so that it ends at the date before retraining occurs, and the strategy continues to be computed with the retrained parameters. Over time, the retraining frequency is averaged across all training sets to smoothly incorporate the algorithm's experience across different market regimes. To minimize computation time, convergence is checked at regular time intervals instead of at every time step. We have used an interval of 50 trading days, corresponding to a calendar period slightly larger than two months.

Applying this approach to SPX using 40 percent of the data for training sets, cubic polynomials for our hypothesis function and a regularization parameter λ of one, the algorithm creates seven different training and test sets with an average retraining frequency in the range of 1,900 business days, or slightly less than calendar 8 years. The retraining frequencies found while computing the SPX trading strategy on each of the seven test sets are shown in Figure 2.

To clarify what the test sets correspond to, consider that the logistic regression builds the investment strategy by walking forward in time and retrains its parameters when necessary. Each time a retraining event occurs, a new training set of data is created by sliding the training data window forward, the feature weights are recalculated so as to minimize the cost function J(θ), and the strategy thereafter uses these new weights to predict future profitability on a daily basis and execute its investment decision.

Therefore, while the strategy is computed by walking forward over a continuous set of dates from August 3, 1964 to December 12, 2018, the date record is conceptually split into different "test sets" that each corresponds to the range of dates when the weights calculated from a particular training set are used. For instance, the first test set corresponds to the period of days over which the investment strategy is computed using the feature weights calculated in the first training set, and so on.

The complete layout of these training sets, test sets and their error metrics is shown in Table 4. Cost and errors in the test sets are uniformly within a 10% range of the respective metrics in the training sets. None of these metrics change significantly across training or test sets, despite the underlying data representing vastly different market regimes over a 90-year period. This suggests that overfitting is not likely an issue in our model.

3.2.4 The Degree of Polynomial Features and the Prediction Time Horizon

Our logistic regression algorithm predicts whether profitability over a certain future time horizon H is likely to be above a specified minimum threshold. To predict profitability, which we anticipate will not be a simple linear function of the basic momentum and drawdown features, we allow the algorithm to build polynomial combinations of these basic features. Conceptually, should future performance be a complex non-linear function that involves the basic features and perhaps even some other market factors that were not incorporated in our model, the new polynomial features will correspond to some of the lower-level polynomial terms in the Taylor series expansion of that function. Incorporating these terms in the model should thus help capture some aspects of the complex non-linear relationship between future performance and basic features, which linear models such as the classic momentum approaches are unable to do.

This leads to two other design questions, namely what degree of polynomials should be used, and how far into the future should we attempt to predict performance, i.e. how many days should H represent?

With respect to the degree of polynomials, Figure 3 shows the maximum performance improvement over buying and holding SPX across all prediction time horizons ranging from one day to one month that are achieved when using a long-only approach with polynomials of degrees 2, 3 and 4. Quadratic polynomials improve annualized returns by about 10 percent over buy-and-hold SPX, while cubic and fourth-order polynomials respectively improve returns by 31 percent and 13 percent. Table 5 shows the performance of the investment strategy using different degrees of polynomials[8]. Letting the algorithm create new polynomial features significantly improves the balance of risk versus return of the investment strategy relative to SPX as measured by maximum drawdowns, average drawdowns and Sharpe ratios. While using quadratic polynomials does not produce higher absolute returns than SPX, the risk-adjusted returns as measured by Sharpe ratios are slightly higher than those of SPX. The power of logistic regression when using higher degree polynomials is visible in the results of cubic and higher order polynomials. The algorithm employs a total of 455 features with cubic polynomials, and 1,820 features with fourth-order polynomials.

[8] Note that Figure 3 and Figure 5 are based on different prediction horizons. While Figure 3 shows the maximum performance across all prediction horizons, Figure 5 shows the performance based on a 3-day prediction horizon.

In both of these cases the resulting optimization problem, namely calculating the weights that minimize the cost function across that number of features for approximately 10,000 individual dates in each training set, is easily solved on a modern laptop computer within a few minutes.

To select our time horizon H, we compare the strategy's results with H ranging from 1 to 21 days using both cubic and fourth order polynomials. Figure 4 compares the annual return, Sharpe ratio, maximum and average daily drawdowns of the investment strategy using cubic polynomials over the time horizons ranging from 1 to 21 days. As shown, the predictive power of the algorithm eventually diminishes as the time horizon increases. Predicting performance three business days into the future seems to be optimal across all of the strategy's key performance metrics. Interestingly the worst performance corresponds to trying to predict the next day's performance. This is best understood when considering that the probability distribution of SPX daily returns is only slightly positive, on the order of a few basis points, and looks normally distributed across positive and negative returns. That means that daily returns are almost randomly positive or negative, and trying to predict that outcome using only price itself is unlikely to be successful. Over longer time horizons, the batting average in favor of positive performance must shift for the index to eventually return positive performance. However, why three days turns out to be an optimal horizon instead of four or more days is likely unknowable and not a result we anticipate would hold for other types of equities, such as single stocks or small capitalization indices, or other asset classes.

The use of fourth-order polynomials yields improvements over SPX in either return or risk for all time horizons greater than one day as shown in Figure 5. The level of these improvements is broadly stable between 2 and 10 days, although results achieved using cubic polynomials are more attractive from an investment point of view and involve less complexity in its features.

It is possible that using higher than fourth-order polynomials and experimenting with different time horizons and basic features would result in greater improvements in investment performance over SPX than those shown in Table 5. Our objective is to use intuitive momentum-based features that we believe from experience are well suited to capturing momentum in large capitalization stocks. Just as we did not attempt to optimize the classic base case in this paper, we are not attempting to optimize our logistic regression approach for SPX. For simplicity of comparison of these approaches, we select the cubic polynomial approach for our discussion of logistic regression through the remainder of this paper. This is the first order of polynomials that we expect will capture meaningful non-linear relationships between future performance, momenta and drawdowns.

Figure 6 summarizes the improvement over SPX using logistic regression with cubic polynomials and a three-day prediction time horizon. The algorithm improves annualized returns by 31 percent while reducing average daily drawdowns by 30 percent from -11.2 percent for SPX to -7.8 percent for the strategy. This in turns lowers volatility and improves the Sharpe ratio by 56 percent from 0.35 for SPX to 0.54 for logistic regression.

3.2.5 Regularization Parameter

The purpose of the regularization parameter λ is to minimize the tendency of an algorithm to overfit data. It adds a penalty to each feature in the cost function in the form of $\lambda\times\theta^2$ that forces θ to be small. Since the magnitude of the feature weights θ in the training sets ranges principally between -1 and +1 as will be shown in Section 4.1 below, the penalty will only be significant for weights with absolute value of magnitude close to 1. Features with weights closer to zero will not be much affected by λ since the penalty term θ^2 will be much smaller than θ.

For features whose weights approximately equal ±1, a value of λ=1 will add a penalty of the same order of magnitude as the existing term for θ in the cost equation. A value of λ=10 would add a penalty about ten times larger, and so on. Therefore, we anticipate that the regularization parameter will start having an impact on limiting overfitting with a value close to 1. Selecting a λ of higher order of magnitude will significantly reduce the weight of these features the algorithm deemed the most meaningful in the first place, giving them smaller values that closer to those of less important features.

To take advantage of the benefits of regularization with regards to overfitting while maintaining a meaningful weight gap between the most and the least significant features within each training set, we select a value of λ=1 for all investment strategy results presented in this paper.

4. Polynomial Features

We now discuss the feature weights θ computed by the algorithm with a view to building some intuition into how logistic regression manages the input parameters, in our case consisting of historical momenta and drawdowns, and their polynomial combinations.

4.1 Features and Their Weights across Training Sets

In our current analysis of SPX with the basic features described in section 3, using cubic polynomials results in a total of 455 features to try and fit each training set data comprising 9,138 dates, or 40 percent of the total available. Since θ is a vector of feature weights, each θ vector contains as many elements as there are features. Figure 7 shows the ratio between the maximum and minimum value of the weights in each θ vector calculated for each of the 7 training sets in the strategy. That ratio is a measure of the relative importance of the features within each training set.

Its order of magnitude ranges from about 1,000 to 10,000, which means that some features will be up to 10,000 more important than others in some of the test sets.

A more detailed summary of the range of the weights within each training set is shown in Figure 8, including their highest and lowest values, their average, and values above and below one standard deviation as indicated by the top and bottom edges of the rectangular box across each vertical line. Despite the changing relative importance amongst the weights demonstrated in Figure 7, the range of the weights is stable across the entire time history from the year 1927 through 2018.

Figure 9 presents a different view of the model weights found for each feature across all training sets over time from 1927 to 2018. The reference number of each feature on the horizontal axis refers to the order in which that feature was created by the algorithm, which could be arbitrarily changed by changing the order in which basic feature are specified. Therefore, the chart is not to be viewed as a continuous curve. Rather, for each feature on the horizontal axis, the vertical axis displays the range of weight values that all the training sets have assigned to that feature. Note that the same feature can contribute positively or negatively over time to the prediction of profitability. The magnitude of the weights across features is fairly stable with most weights falling between -1 and +1.

A detailed picture of the changing importance of different features over time is presented in Figure 10. The two inset figures show respectively the ten most and least important features across the 7 training sets. The vertical axis represents the reference number of each feature, ordered so that the numbers below the horizontal blue line represent the basic seed features we have specified, the set of features between the blue and red lines represents those constructed from quadratic polynomials, and cubic order features are above the red line. The salient characteristic of these charts is the variation in the features that each training set has assessed as most or least important amongst the 455 being analyzed. None of the features are uniformly important or disregarded over time, and the variability extends across the range of linear, quadratic and cubic features. This validates the choice to present results in this paper derived from higher than quadratic polynomials to enable the logistic regression to capture some aspects of the expected non-linear relationship between performance, momentum and drawdowns.

Figure 11 displays an analysis of the feature weights calculated in the first training set. The values in Figure 11 essentially expose the details within the leftmost vertical bar in Figure 8. Different features across all polynomial orders contribute positively and negatively to the prediction of performance, and a significant number of features are mostly disregarded with weights either close to zero or much smaller than the largest weights. Figure 11-d displays the distribution of the absolute value of features weights, which is the true measure of the contribution of each feature, whether positive or negative, in predicting future performance.

Figure 12 shows that a majority of features are assigned low and different relative weights within each training set, which means that the investment strategy will be driven by a changing subset of features as time progresses and the algorithm retrains its weight parameters. Overall, these results indicate that none of the features dominate the predictive power of the algorithm.

4.2 Decision Boundaries: Visualizing the Relationship Between Features and Future Performance

To develop further insight in the behavior of the algorithm, this section presents a visualization of the relationship between a sample of the model's features and the three-day future performance which the algorithm is designed to predict. Recall that future performance is represented by the vector denoted as Y in our algorithm. The algorithm does not actually predict the level of future performance, but only whether performance is likely to exceed a minimum target threshold of δ. If at a given date in a training set for which future performance is known, that performance is above the threshold, the value of Y on that date is 1. If the performance falls below the threshold, Y is set to zero.

Figures in this section display the scatter of Y relative to various combination of features, with green representing dates where three-day performance is above the threshold, and red where it falls short. Were some combinations of features to represent perfect ways to predict positive versus negative performance, the corresponding scatter plots would neatly outline decision boundaries, namely regions containing mostly green data separate and apart from regions dominated by red data. In contrast to a classic momentum approach that uses a handful of technical indicators, machine learning approaches can repeatedly be trained over large numbers of combinations of complex features to try and identify non-linear patterns, or decision boundaries, which delineate market conditions where future performance is expected to meet the threshold target.

Figure 13 shows one example of the relationship between future performance and two of the features initially selected for the algorithm: the short-term historical momentum of 30 days, somewhat longer than one calendar month, and a longer-term momentum of 120 days corresponding to about 6 months. These types of features that mix short and long historical time horizons are commonly used in technical trading algorithms based on cross-over momentum. It would be tempting to conclude from Figure 13 that when short-term momentum is positive or even slightly negative, the three-day future performance would seem to be dominated by green data, indicating that the strategy is likely to meet the target profitability threshold. Whether this is statistically meaningful or even pertinent in relation to other available combinations of features, is represented by the weights that the training set finally assigns to each feature when minimizing the cost function J(θ).

One subtlety of these two-dimensional scatter plots is that the color plotted last dominates the visual effect even in areas where green and red data may be equally concentrated. To demonstrate this, Figure 13 shows the same scatter relationships, with the positive performance test (green data) plotted last on the left chart, and plotted first on the right chart. The correct interpretation of these charts requires a three-dimensional examination of these relationships, or a careful review of these twin sets of two-dimensional graphs.

For simplicity, we display in this section the two-dimensional views of decision boundaries with the understanding that most of the differentiation between areas dominated by positive or negative performance actually lies on the edges of these scatter charts. Figures 14 to Figure 16 show the relationship in the last training set of SPX between three-day future performance and a sample of linear, quadratic and cubic polynomial features respectively. In the case of quadratic and cubic features, we simply label the features using their reference number from 0 to 454. The point is not as much to identify the exact polynomial combination displayed, as it is to observe the vast complexity of relationships the logistic regression algorithm is able to use to try and predict performance.

To summarize, the logistic regression algorithm attempts to capture the relationships between features and future performance in each training set, examples of which are provided in Figures 13 through Figure 16, reflect those in the weights ascribed to each feature, and then predict on each date in the test set whether performance is likely or not to exceed the threshold. This daily binary prediction point in turn drives the investment decision for the single asset the strategy allocates to.

5. Long-Only Investment Performance

This section presents the investment results of the long-only logistic regression algorithm applied to SPX. The algorithm is designed in the following manner that summarizes our presentation to this point. The algorithm uses cubic polynomials to build non-linear features from the set of basic features specified before the computation starts. These input features include drawdowns and momenta over various historical time windows, normalized to zero-mean over time within a range of -1 to +1 to prevent unintended bias in the resulting weights. The algorithm uses a regularization parameter λ equal to one to lower the chance of overfitting data. Its design is intended to predict each day whether the performance of SPX is likely to exceed a minimum threshold δ of 5 percent annually over the following three business days. If the algorithm predicts a profitability lower than this threshold, the long-only strategy moves to cash at day end while the long-short strategy establishes a short position in SPX at day end. Otherwise the strategy is or remains invested into SPX at market close. Slippage, trading costs, borrowing costs and tax implications are ignored.

The available historical price data of SPX is partially allocated to an initial training set containing 40 percent of the data from December 30, 1927 to June 30, 1964, and the algorithm learns autonomously as the strategy walks forward in time when to slide that training window to retrain the weights θ it ascribes to each feature.

As noted in Section 3.2.4, using higher order polynomial features yields better investment results than either linear or quadratic approaches. Our objective in this paper is not to optimize a momentum trading strategy for SPX, but to demonstrate that a machine learning approach that is straightforward to implement and inexpensive to compute can deliver improvements in investment results not only for SPX over a buy-and-hold approach and the classic base case described in section 3.1, but also when applied without modification to other assets within the same asset class.

Table 5 summarizes the investment performance of the logistic regression long-only strategy. It generates an 8.6 percent annual return with a Sharpe ratio of 0.54, a maximum historical drawdown of 45 percent and an average daily drawdown of 8 percent. In comparison, Table 6 shows the investment performance using the base-case approaches with various dual-momentum parameters. As shown, none of the strategies under the classic approach produce investment results as attractive as the logistic regression algorithm from a risk-return point of view.

Figure 17 displays the evolution of several key performance metrics over time for the logistic regression long-only approach. The growth of the strategy is compared to the growth of SPX on which are superimposed in red the time periods when the strategy is in cash. The comparisons of volatility and daily drawdowns between the strategy and SPX confirm that the algorithm seems able to fairly consistently reduce volatility and ongoing losses in vastly different market environments.

The trading frequency chart is a visual indicator of when the strategy moves from its position on the lower horizontal axis (where it is in cash) to the top horizontal axis where it is invested. Moving from cash to investment is shown as a green vertical bar, while moving from the investment to cash is shown in red. The turnover of the strategy is an easier metric to visualize and is shown annualized at each date in the record. A roundtrip from being invested to cash and back to an invested position generates a turnover of 2, since there is only one security in the portfolio and the decision is to be 100 percent in cash or in the asset. Therefore, the high average annual turnover (19.8) of the strategy is not to be viewed as a barrier to implementing practical client portfolios but rather as a by-product of how the approach to investing in one asset was designed for the purpose of this research.

Figure 18 focuses on the performance of the logistic regression long-only strategy in the worst performance periods of SPX. Figure 18-a displays results in the nine periods between August 3, 1964 and December 12, 2018 when SPX lost 15 percent or more. The start and end dates identify the period when the respective SPX drawdown started and ended, so that the performance of SPX across that period is zero by construction and the maximum drawdown reached within the period is greater than the 15 percent threshold. Figure 18-b shows a similar analysis in cases where SPX reaches at least a 25 percent drawdown.

In the majority of these extreme performance environments, logistic regression was able to generate positive returns ranging between +4 percent and +48 percent. In the periods when the algorithm experienced a loss, the maximum drawdown was similar to that of SPX (see Figure 19), and losses were confined to single digit percentage points. This suggests that the logistic regression strategy is able to consistently deliver higher returns with maximum downside risk that is generally lower than that of SPX, and with average daily drawdowns that can be significantly lower than the buy-and-hold strategy. It is able to do so over long periods of time, so that improvements are not concentrated in one or a few test sets.

Table 7 compares the performance of the strategy to SPX over the period corresponding to each test set. The results show that the algorithm perform better than SPX across all test sets except the fifth one from July 1997 to April 2005. Over time the strategy is able to deliver excess performance by managing risk in periods of heavy losses for the index. For instance, while the algorithm was underperforming the index from July 1997 to mid-2008, it recovered above the index when strong downside volatility continued to drag the equity markets down from September to October 2008. It is worth bearing in mind that the logistic regression strategy cannot overperform SPX other than by reducing risk, i.e. by being in cash in periods of prolonged downturns, since it only invests in SPX or cash. The over-performance displayed in various test sets is a demonstration of risk control, borne out of the algorithm's ability to repeatedly predict short-term performance with enough accuracy to capture upside while controlling losses one day at a time.

Figure 20 compares the growth of the strategy to that of SPX over each test set, rescaling the price data to 1,000 on the first date of each test set. Figure 21 compares the daily drawdowns over test sets assuming that both our strategy and SPX start at a new highwater mark on the first day of each test. This allows a comparison of the evolution of losses and recoveries after each retraining of the strategy. Finally, annualized volatility over each test set is shown in Figure 22.

Across all test sets, the logistic regression long-only algorithm has resulted in lower volatility as well as average daily drawdowns generally far below SPX, with the largest reduction in losses of 59% in the second test set.

6. Additional Analysis

In this section, we address three additional topics: (1) Long-short investing applied to the S&P 500 Index, (2) the applicability of our approach to indices other than SPX, and (3) the impact of the retraining frequency on the performance of the trading strategy.

6.1 Long-Short Investing

Our discussion up to this point has centered on long-only investments where the portfolio is moved 100 percent to cash if profitability predictions fall below the target minimum threshold.

In the long-short approach by contrast, the portfolio is moved 100 percent to a short SPX position instead of cash. Borrowing costs are neglected, and since our SPX data excludes dividend reinvestments no adjustment to our price data is necessary for this approach.

Table 8 compares the investment results of SPX to those of long only and long-short strategies using both a classic dual-momentum approach and our logistic regression algorithm. The classic approach is computed with a 21-day rebalancing frequency and 252-day look-back window to compute momentum.

Results of the classic dual-momentum strategy are far worse when using a long-short approach than with a long-only approach. This is due to the year-long look-back period the strategy uses to assess whether momentum exceeds the target threshold. When the strategy is short the index, it will lose accumulated profits over a potentially long period of time once the index starts trending up following its point of maximum drawdown. On average, the long-short approach loses more in this fashion than it makes being short while the index moves down, resulting in lower returns (4.1% versus 4.9% for the long-only approach) and worse risk metrics across the board. For instance, the average daily drawdown of the long-short approach is -25 percent versus -7 percent for the long-only approach. Both the long-only and long-short approaches for the classic dual-momentum strategy yield worst risk-return results than the simple buy-and-hold approach.

By contrast, both long-only and long-short logistic regression strategies yield higher returns at significantly lower risk than SPX. With a 10.0 percent annualized return, the long-short approach is also more profitable than the long-only approach with an 8.6 percent return. The long-short strategy's risk metrics – maximum drawdowns, average daily drawdowns and volatility – are 10 percent to 20 percent worse than those of the long-only strategy, but that increase in risk is broadly offset by the higher return so that the Sharpe ratio of both approaches is materially identical at 0.55. We have run a number of comparisons between long-short and long-only approaches to investing into SPX with logistic regression, varying the minimum profitability threshold, the target time horizon and retraining frequencies. In the quasi-totality of cases, we found results qualitatively the same as those described above. The long-short approach yields higher returns that the long-only approach, albeit with somewhat worse risk metrics and similar Sharpe ratios.

6.2 Applying the Strategy to Other Equity Indices

So far, our tests have been focusing on a single equity index – the S&P 500 Index. Although care was taken not to overfit the data, some decisions were made based on the analysis of the price data of that index, for instance the selection of third-order polynomials, the three-day time horizon that drives the prediction of future performance, as well as the magnitude of the regularization parameter. Moreover, features themselves were selected from experience to reflect typical financial metrics used by momentum-based trading models applied to large capitalization equities, and it is possible that the authors' experience could itself be biased towards SPX which is one of the most studied and important indices in global finance. Large capitalizations have their own market dynamics driven by usually deeper equity research coverage than mid or small capitalizations, as well as a broader range of tracking indices and investable funds such as exchange traded funds, mutual funds, unit investment trusts, and UCITS. Additionally, U.S. equity markets are historically the largest and most liquid. Consequently, the algorithm developed above for a U.S. equity index may not perform well when applied to non-U.S. markets, different assets classes or stocks of different market capitalization.

To check the broader applicability of our algorithm beyond SPX, we apply it to the other indices listed in Table 1. Table 9 summarizes the resulting key performance metrics when investing in these indices using the same long-only logistic regression algorithm that was used for SPX.

Figure 23 to Figure 25 display in percentage terms the excess performance over a buy-and-hold investment into SPX of the logistic regression algorithm. Figure 23 shows that when applied to the U.S. mid-capitalization index, the approach adds no significant value. Returns are about 3 percent higher and the Sharpe ratio is 9 percent better than the buy-and-hold index, but the volatility, average and maximum drawdowns do not improve materially. These small improvements in returns are not sufficient in practice to offset considerations of model risk, slippage, trading costs and management fees. Therefore, the algorithm does not appear to be practically applicable as-is to mid-capitalization equities.

The algorithm applied to U.S. small capitalizations results in a degradation in return on the order of 35 percent, and a modest decrease in investment risk. This is consistent with the expectation that the market drivers of small capitalization stocks are different than those of larger stocks.

We next apply the algorithm to three non-U.S. equity indices: the FTSE 100 Index, the Tokyo Stock Exchange Price Index (TPX) and the FTSEurofirst 300 Index (E300). While the average capitalization of the stocks in these indices is typically smaller than SPX, the underlying securities are large capitalizations in their home markets. Thus, we expect that some aspects of investor behavior that drive the supply and demand for these stocks are similar to those driving SPX. Results for these market indices show improvements in return and risk metrics far larger than for SPX, although in aggregate the SPX strategy is the most appealing one from the point of view of risk-adjusted return as well as the absolute level of returns that it generates. Results for these three non-U.S. equity indices are shown in Figure 24.

The FTSE 100 Index (UKX) showed improvements across all key performance metrics, in particular an 87 percent increase in annual return over the period of December 19, 1997 to December 12, 2018, accompanied by a reduction in average daily drawdowns from -16.3 percent to -11.8 percent. The Sharpe ratio of the algorithm rose over 310 percent from 0.03 to 0.11.

The Tokyo Stock Exchange Price Index (TPX) similarly showed improvements across all key performance metrics over the period of May 16, 1977 to December 12, 2018. Return increased 149 percent from 3.54 percent to 8.81 percent, while average daily drawdowns were reduced from 38 percent to 8 percent.

Similar improvements were seen when investing in the largest 300 stocks in the FTSE Developed Europe Index (E300) from March 8, 1999 to December 12, 2018. Returns rose 309 percent from 0.58 percent to 2.37 percent, while average drawdowns fell 56 percent from 26 percent to 11 percent.

Figure 25 shows the results of the algorithm applied to two other large capitalization U.S. equity indices: the Dow Jones Industrial and the Dow Jones Transportation Average. These indices were selected because of the length of the available price history which goes back to the 1920's as shown in Table 1.

The Dow Jones Industrial Average is the unique case in this analysis where the algorithm modestly reduces returns but proportionately reduces risk even further, resulting in a trading strategy with a more attractive risk profile albeit with lower return. The loss of profitability, from 6.4 percent to 6.1 percent annualized, may seem small but would translate into a large monetary differential when compounded 60 years from October 26, 1959 to December 12, 2018. On the other hand, the reduction in maximum drawdowns from 54 percent to 37 percent, and in average daily drawdowns from 10 percent to 6 percent are material enough that within the context of a multi-security portfolio, the algorithm could be expected to perform without changes.

The Dow Jones Transportation Average in contrast shows improvements more closely aligned with those of SPX: a 14 percent increase in return with a 44 percent reduction in average daily drawdowns and a 31 percent drop in the maximum drawdown between October 26, 1959 and December 12, 2018.

In sum, it seems that our logistic regression algorithm can perform adequately when applied to other equity indices that fall in the same asset class as SPX, namely large capitalization stocks.

6.3 Impact of the Retraining Frequency on the Performance of the Strategy

While the trading strategy discussed in this study was designed to learn autonomously when it becomes necessary to retrain the weights of the features, we also examined the performance of the strategy in cases where the retraining frequency is fixed and specified a priori instead of learnt. Figure 26 displays the performance of the strategy when the weights of the features are retrained at fixed frequencies ranging from 250 business days (i.e., about one calendar year) to 3,000 business days (i.e., about 12 years).

It is interesting to note that while pre-determined training frequencies of up to 2,000 days, or 8 years, can add significant value over a buy-and-hold SPX strategy in the form of added returns or significantly lower risk or both, letting the algorithm learn when to retrain itself yields better investment performance than any of the fixed retraining frequency approaches.

In our algorithm, the retraining decision is triggered by the convergence of the test set cost function as the strategy walks forward in time. This convergence of the test set cost function to a value that is somewhat higher than, but close to, the corresponding training set cost, is what one expects of an algorithm that does not overfit data. An optimal trigger to force the retraining of model parameters would be an indicator that the weights calculated on the training set that ended several years in the past are no longer effective in predicting ongoing performance. Notwithstanding the success of the approach analyzed in this paper, whether the convergence of the cost function on a given test set can serve as a reasonable proxy for such an indicator requires additional research beyond the scope of this paper.

7. Conclusion and Future Research

In this study, we propose a machine learnings approach to build momentum-based trading strategies. Specifically, we design a logistic regression algorithm which takes as inputs momenta and drawdowns over various historical time periods, and tries to predict whether price will rise over a fixed threshold in the near future. The approach is kept as simple as possible by excluding all non-price economic or market data that could be also relevant in predicting market prices. In doing so, we constrained the machine learning approach to optimizing a price auto-correlation problem on a portfolio containing one security only.

We found that, when applied to SPX, the optimal value of the prediction time horizon is three business days, although using a time horizon of up to 10 days could add value either by increasing returns, decreasing risk or both. We also found that well performing investment strategies can be created with a range of non-linear features, whether these are built from quadratic, cubic or higher-order polynomials.

The algorithm yields a range of small to large improvements when applied to U.S. and international large capitalization indices. The combination of purposely generic momentum features, an autonomously learnt retraining frequency, a regularization parameter of the same order of magnitude as the model weights, and the use of cubic polynomial features to predict short-term future profitability, is able to capture the momentum effect in different geographies and across long periods covering different stock market regimes.

There are multiple opportunities to build on the ideas presented in this paper. First, while logistic regression offers a simple yet powerful framework within which to build machine learning investment strategies, there is no intuitive reason why an approach that uses polynomial combinations of features should be optimal. Future research may explore the performance of other machine learning techniques such as Support Vector Machines and neural networks.

Second, this study constrains the algorithm to features that are based solely on historical prices for simplicity, although there is no reason to believe this is the best approach. In fact, there are many economic and market indicators that are intuitively likely to be meaningful when predicting security prices. Future research may expand the set of features to include non-price data to try and improve the algorithm's predictive power.

Third, our strategy is designed as an investment into a single asset for clarity of presentation. Future research can extend to portfolios that invest in multiple securities across one or multiple asset classes. Moreover, there is scope to investigate the application of machine learning algorithms based on factors other than momentum, or that combine multiple factors, such as value, size or asset quality for instance, into one investment strategy.

References

Asness, C. S., Moskowitz, T. J., & Pedersen, L. H. (2013). Value and momentum everywhere. *The Journal of Finance*, *68*(3), 929–985.

Fama, E. F., & French, K. R. (2008). Dissecting anomalies. *The Journal of Finance*, *LXIII*(4).

Grinblatt, M., & Moskowitz, T. J. (2004). Predicting stock price movements from past returns: the role of consistency and tax-loss selling. *Journal of Financial Economics*, *71*(3), 541–579.

Jegadeesh, N., & Titman, S. (1993). Returns to buying winners and selling losers: implications for stock market efficiency. *The Journal of Finance*, *48*(1), 65.

Jegadeesh, N. (1989). Evidence of predictable behavior of security returns. *The Journal of Finance*, *45*(3), 881–898.

Moskowitz, T. J., Ooi, Y. H., & Pedersen, L. H. (2012). Time series momentum. *Journal of Financial Economics*, *104*(2), 228–250.

FIGURE 1

SPX Error Metrics for Different Training Set Sizes

These figures show comparisons of SPX error metrics and the cost function among training, and two test sets using different training set sizes.

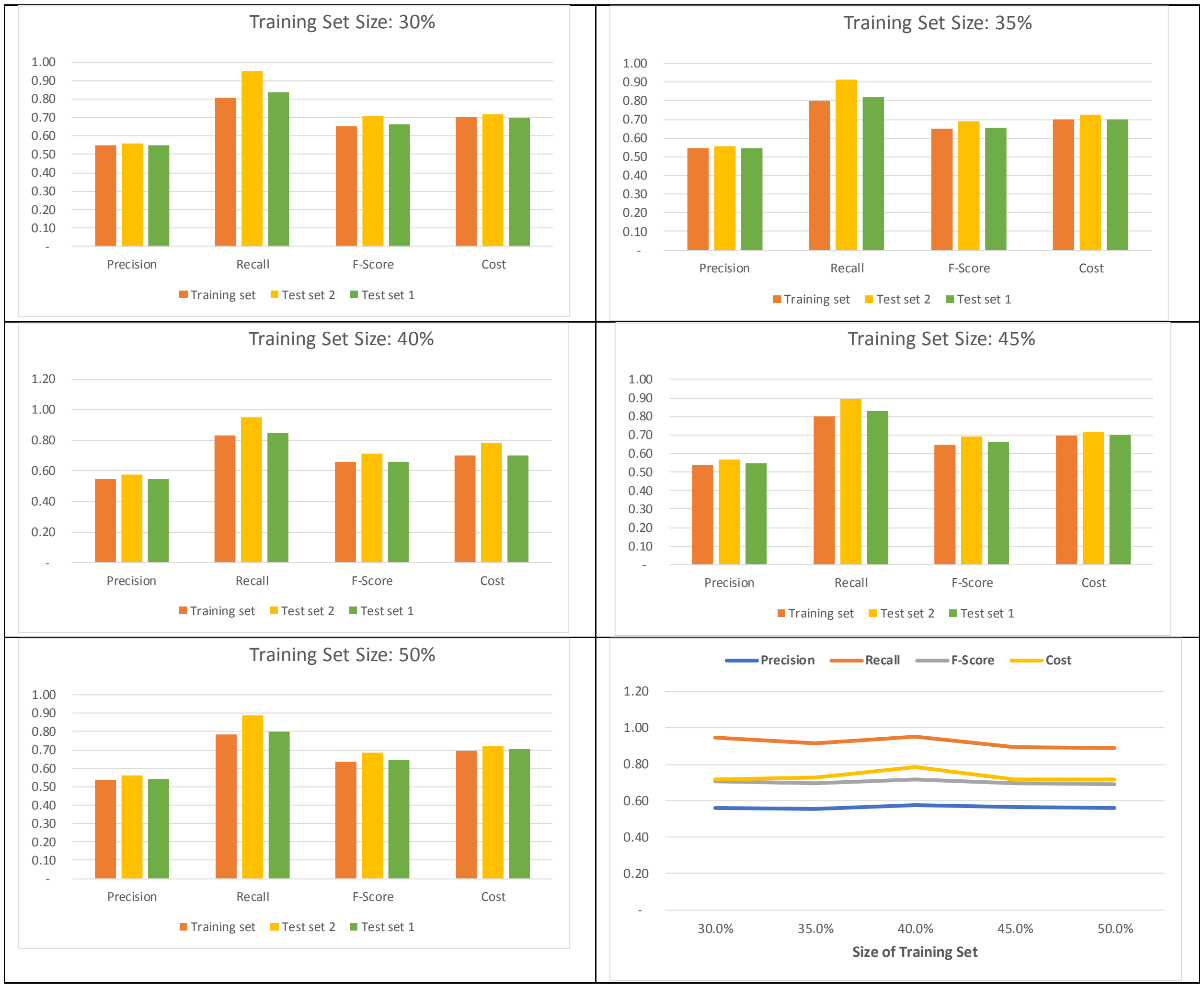


FIGURE 2

Learning the Best Retraining Frequencies Across Test Sets For SPX

These figures depict the costs calculated with the test sets. The x-axis represents the number of days in the test set. The y-axis represents the cost calculated with the test set.

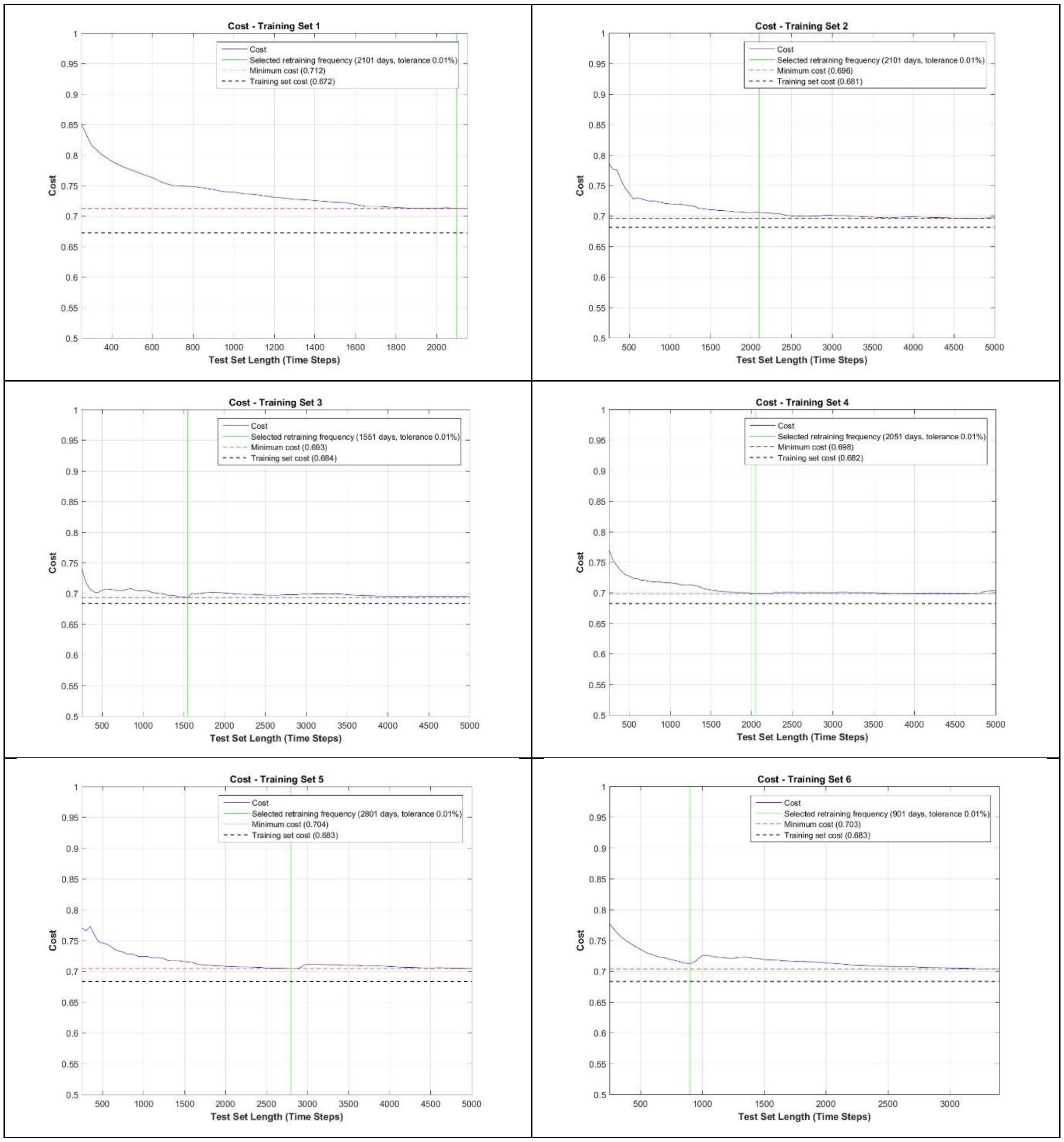


FIGURE 2 (Continued)

Learning the Best Retraining Frequencies Across Test Sets For SPX

These figures depict the costs calculated with the test sets. The x-axis represents the number of days in the test set. The y-axis represents the cost calculated with the test set.

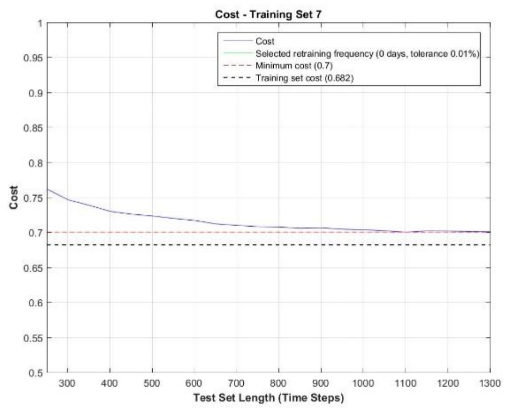
	

FIGURE 3

Impact of Polynomial Orders on the Excess Performance of Long-Only Logistic Regression Relative to SPX

These figures depict the excess performance of long-only logistic regression with different degrees of polynomials, relative to buying and holding SPX. The data is from August 3, 1964 to December 12, 2018.

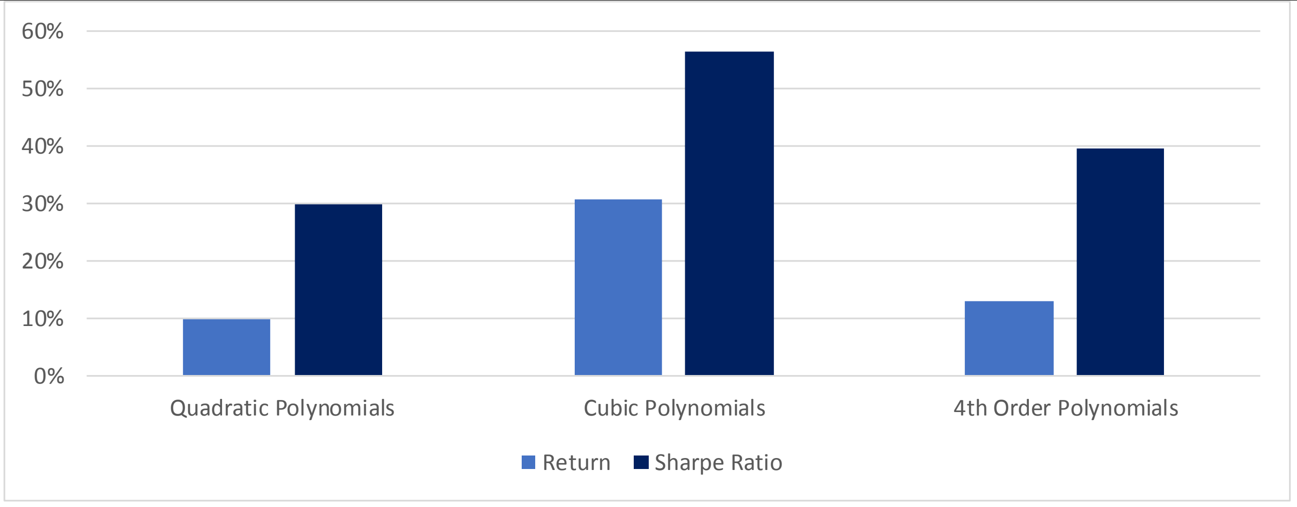
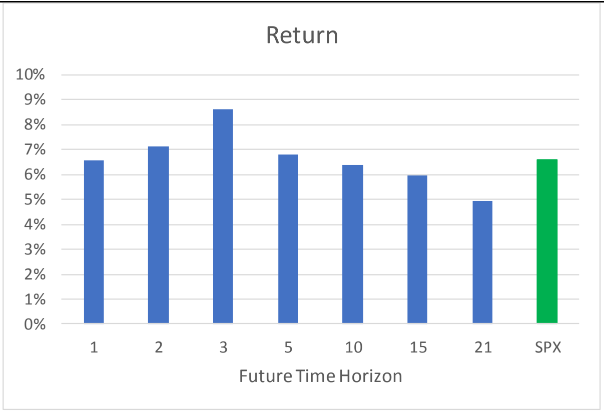


FIGURE 4

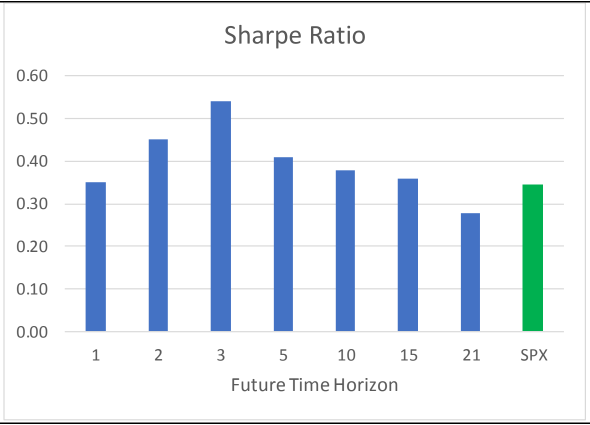
Impact of Investment Time Horizon on Investment Performance: Long-Only Logistic Regression with Cubic Polynomials

These figures depict investment performance for long-only logistic regression with cubic polynomials with different time horizons. The data is from August 3, 1964 to December 12, 2018.

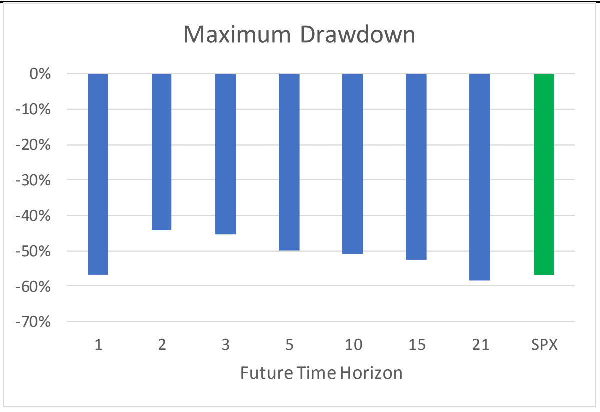
Annual Return



Sharpe Ratio



Maximum Drawdown



Average Daily Drawdown

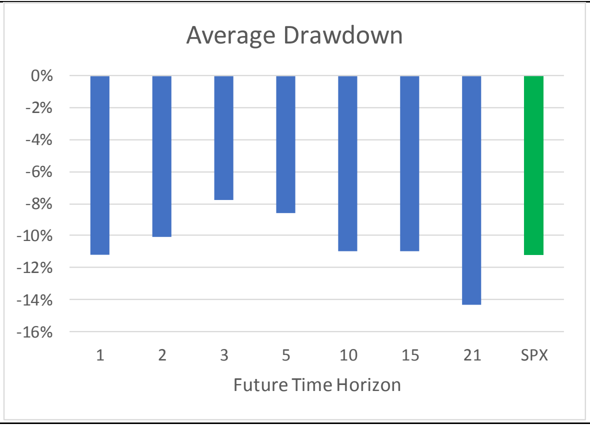
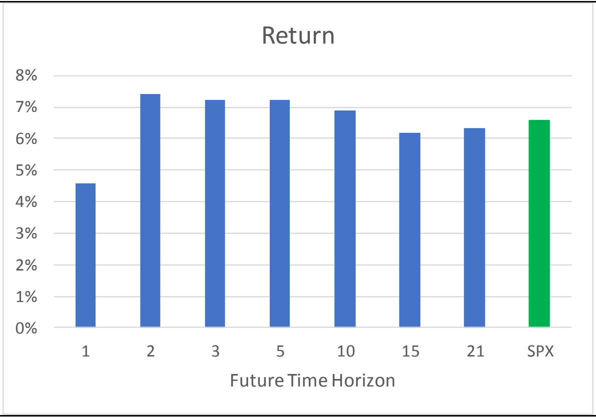


FIGURE 5

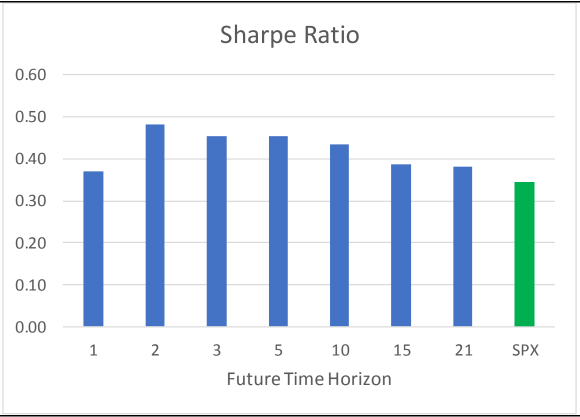
Impact of Investment Time Horizons on Investment Performance: Long-Only Logistic Regression with 4th Order Polynomials

These figures depict investment performance for long-only logistic regression applied to SPX with 4th order polynomials with different time horizons. The data is from August 3, 1964 to December 12, 2018.

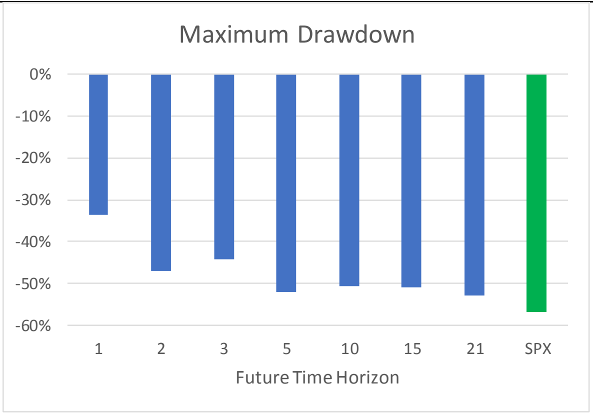
Annual Return



Sharpe Ratio



Maximum Drawdown



Average Daily Drawdown

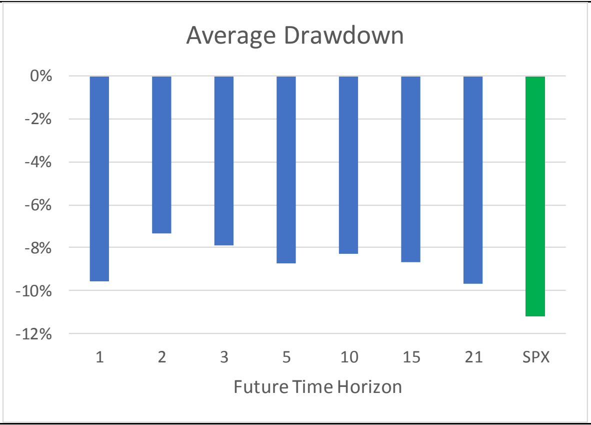


FIGURE 6

Improvement Over SPX Using Cubic Polynomials – 3-Day Investment Horizon

These figures depict the improvement in investment performance of cubic polynomials over SPX with a three-day investment horizon and long-only approach. Data is from August 3, 1964 to December 12, 2018.

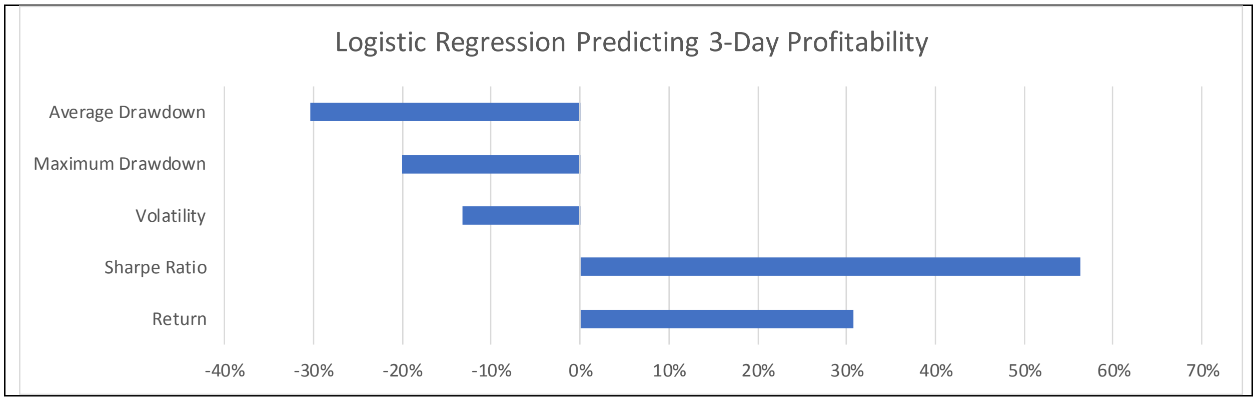


FIGURE 7

Relative Importance of Features within Each Training Set

This figure depicts relative scaling of θ elements or feature weights, across the seven training sets.

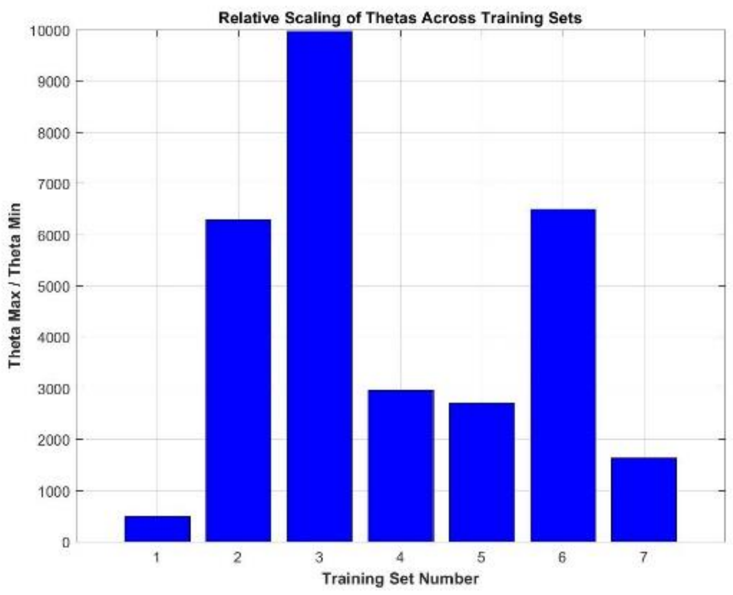


FIGURE 8

Range of All Feature Weights within Each Training Set

This figure depicts the range of all feature weights within each of the seven training sets.

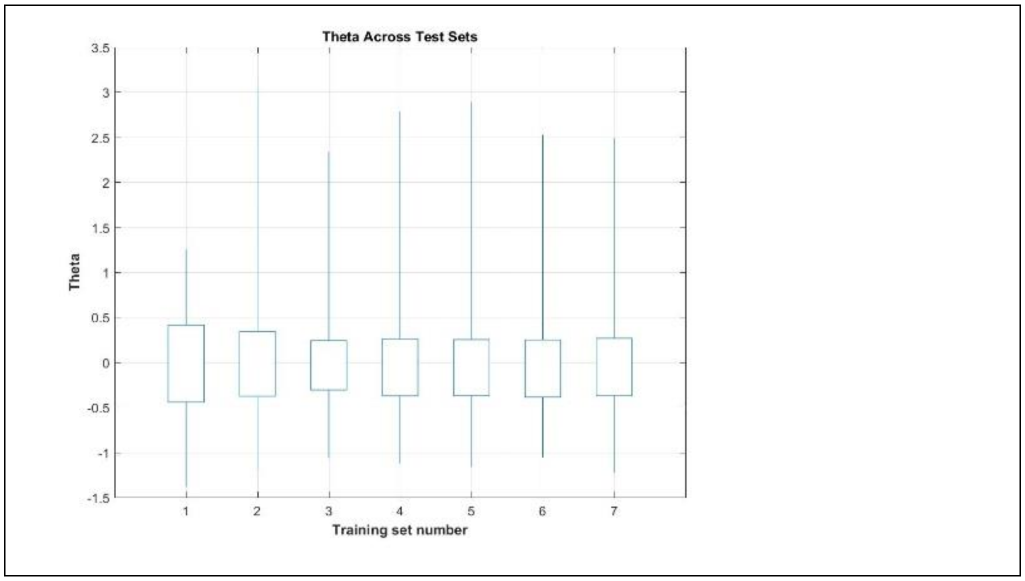


FIGURE 9

Weight Range of Each Feature across All Training Sets

This figure depicts the weight range of each feature across all seven training sets.

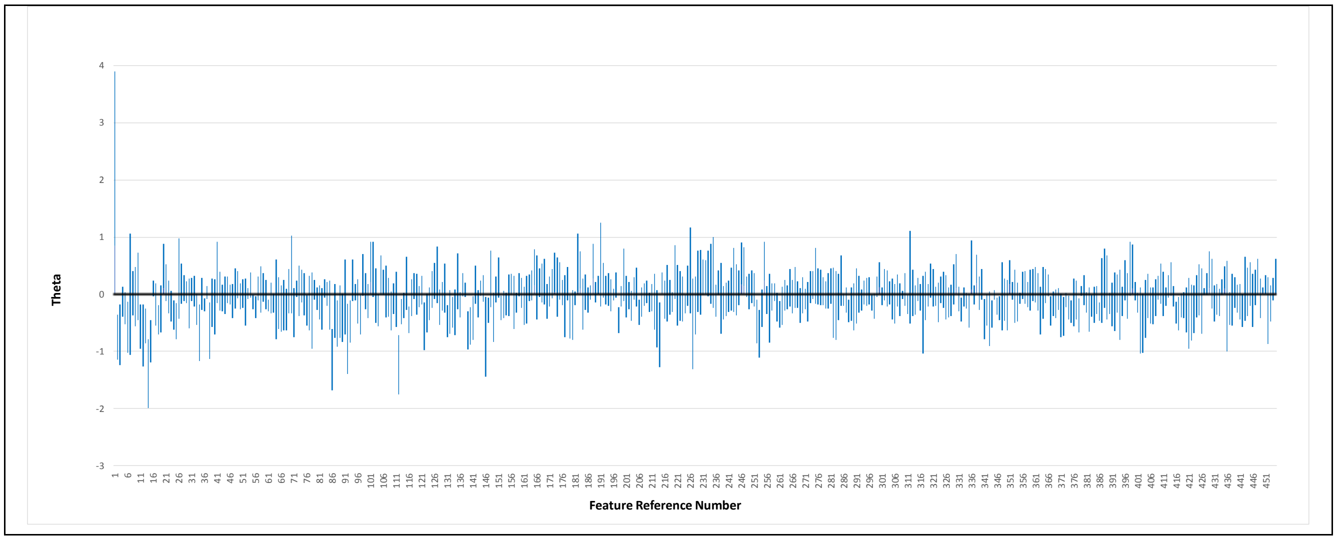


FIGURE 10

Ten Most/Least Important Features across Training Sets

These figures depict the ten most and ten least important features across the seven training sets.

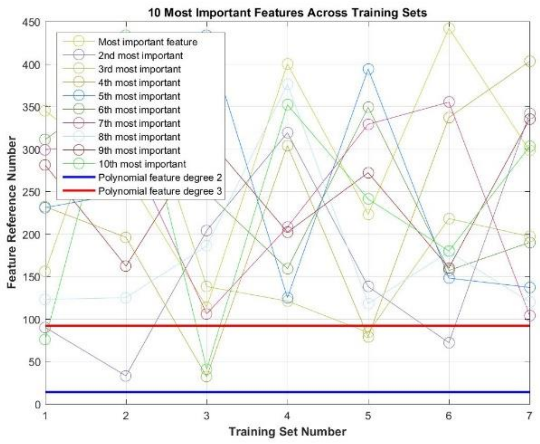
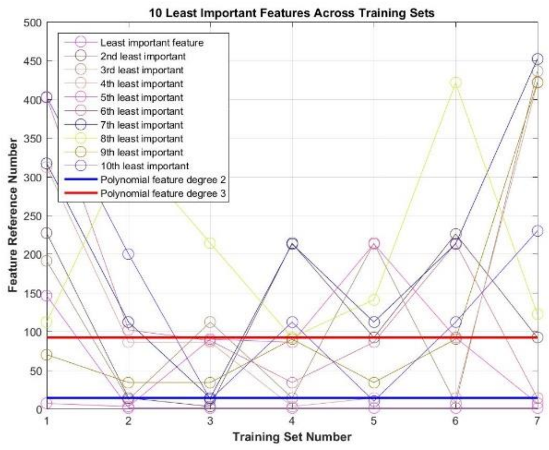
Ten Most Important Features Across Training Sets	**Ten Least Important Features Across Training Sets**
	

FIGURE 11

Feature Weights Calculated in Training Set #1

These figures describe the feature weights calculated in the first training set.

Figure 11-a: Weight Values	**Figure 11-b: Weight Z-Scores**

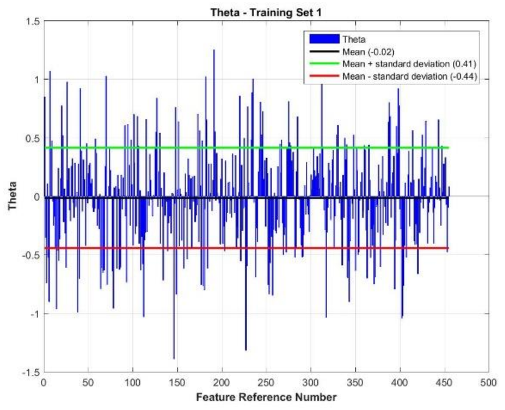
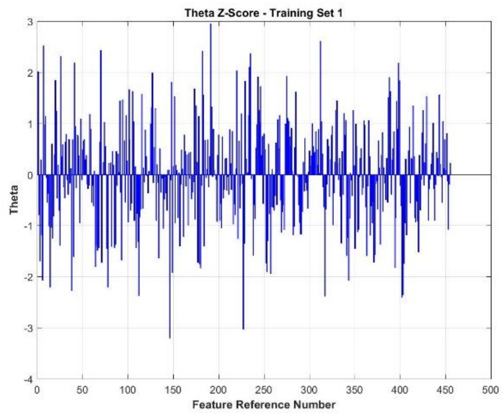

	

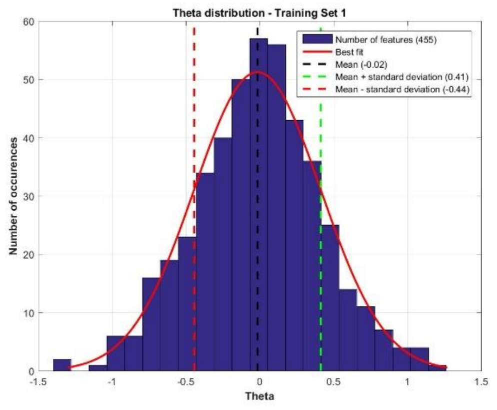
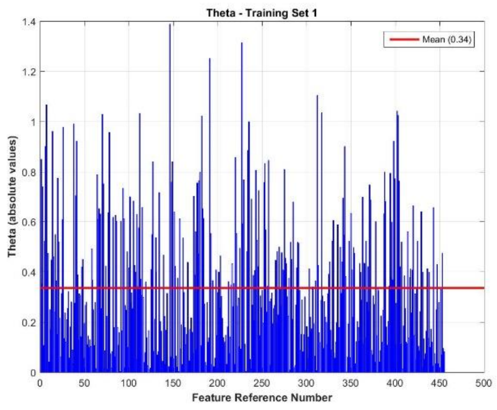
Figure 11-c: Weight Distribution	**Figure 11-d: Weight Absolute Values**
	

FIGURE 12

Feature Weights Calculated by Training Sets

These figures depict the SPX feature weights calculated by each training set.

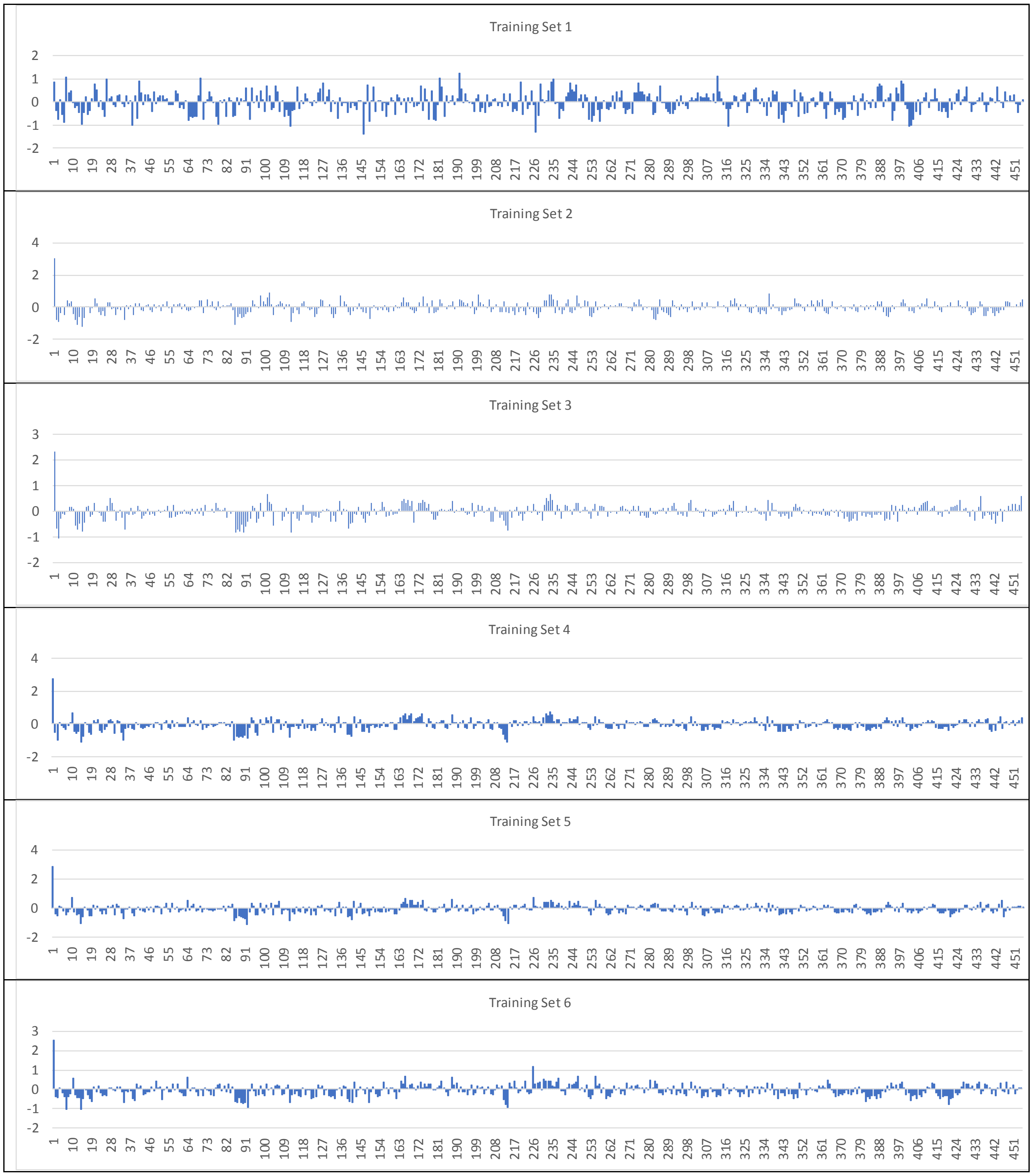


FIGURE 12 (Continued)

Feature Weights Calculated by Training Sets

These figures depict the feature weights calculated by training sets.

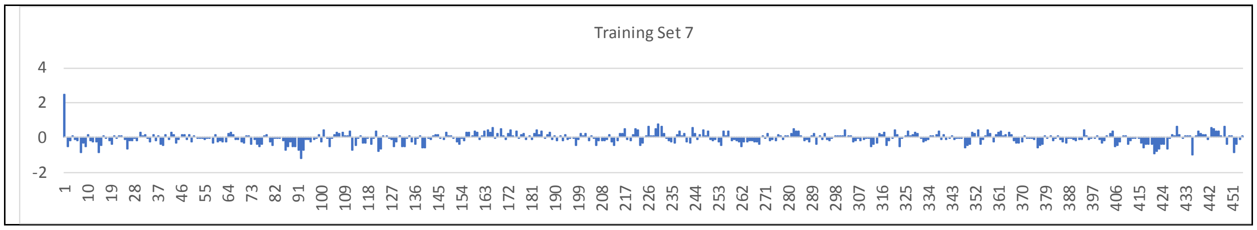


FIGURE 13

The Relationship between 30-Day and 120-Day Historical Momenta and Future Performance in Training Set #1

These figures depict the scatter of future performance relative to 30-day and 120-day momenta in training set 1. Scatter is plotted green when future performance Y is above the target threshold δ. Scatter is plotted red when future performance Y is below the target threshold δ. Data is from December 30, 1927 to June 30, 1964.

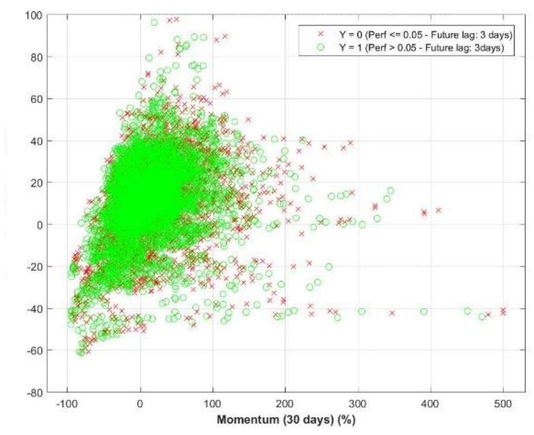
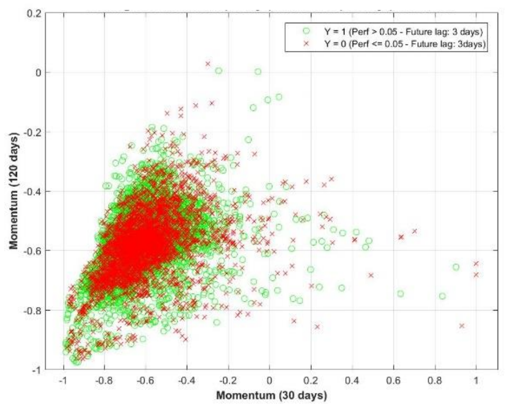
Y = 1 (Green) Plotted Last	**Y = 0 (Red) Plotted Last**
	

FIGURE 14

Sample SPX 3-Day Future Performance versus Linear Features in Training Set #7

These figures depict the scatter of future SPX 3-day performance relative to linear features in training set 7. Scatter is plotted green when future performance Y is above the target threshold δ. Scatter is plotted red when future performance Y is below the target threshold δ.

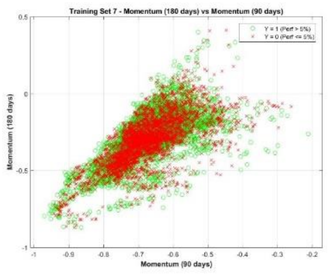
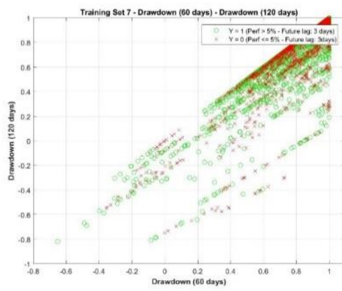
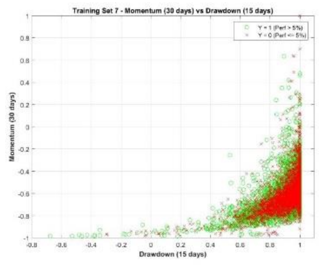
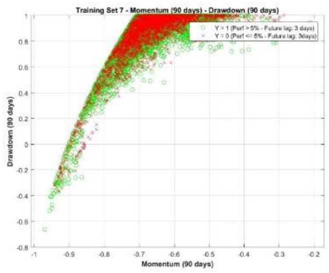
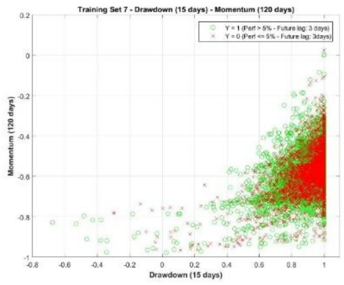
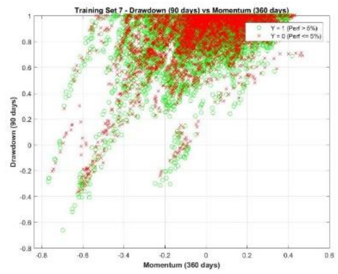
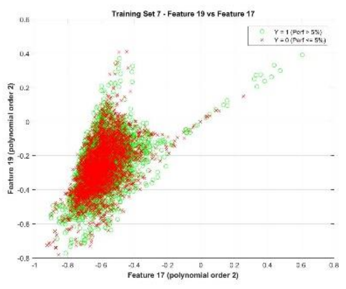
90-Day versus 180-Day Momenta	**60-Day versus 120-Day Drawdowns**	**15-Day Drawdowns versus 30-Day Momentum**
		
90-Day Momentum versus 90-Day Drawdowns	**15-Day Drawdowns versus 120-Day Momentum**	**360-Day Momentum versus 90-Day Drawdowns**
		

FIGURE 15

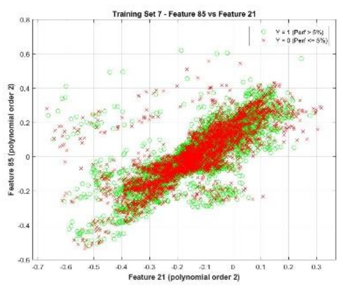
Sample SPX 3-Day Future Performance versus Quadratic Features in Training Set #7

These figures depict the scatter of future SPX 3-day performance relative to quadratic features in training set 7. Scatter is plotted green when future performance Y is above the target threshold δ. Scatter is plotted red when future performance Y is below the target threshold δ.

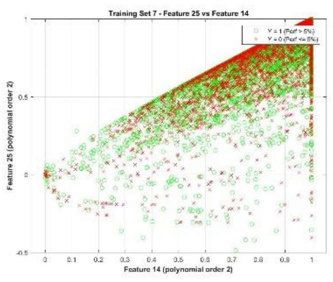
Feature #17 versus #19



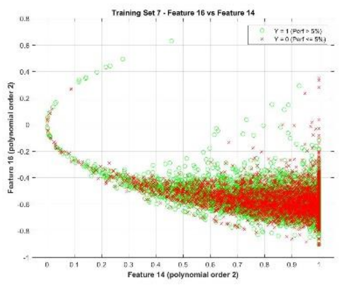
Feature #21 versus #85



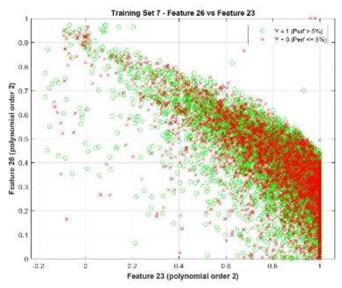
Feature #14 vs #25



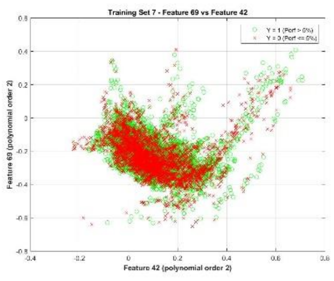
Feature #14 versus #16



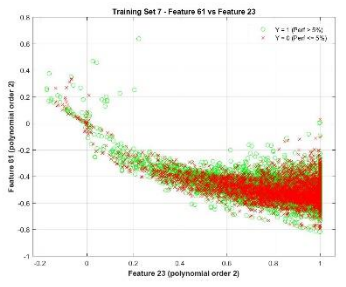
Feature #23 versus #26



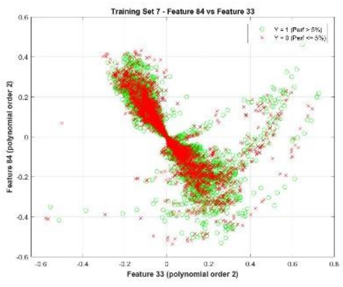
Feature #42 versus #69



Feature #23 versus #61



Feature #33 versus #84



Feature #26 versus #67

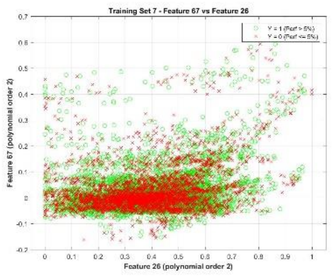


FIGURE 15 (Continued)

Sample SPX 3-Day Future Performance versus Quadratic Features in Training Set #7

These figures depict the scatter of future SPX 3-day performance relative to quadratic features in training set 7. Scatter is plotted green when future performance Y is above the target threshold δ. Scatter is plotted red when future performance Y is below the target threshold δ.

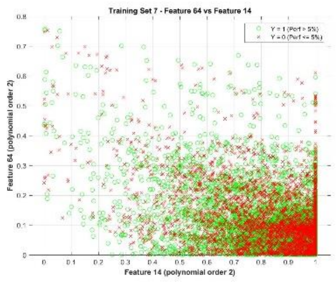
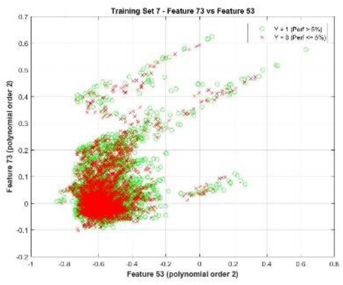
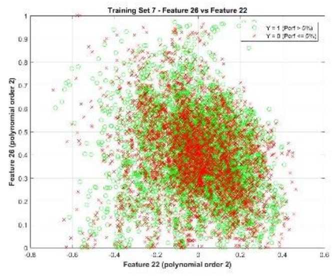
Feature #14 versus #64	**Feature #53 versus #73**	**Feature #22 versus #26**
		

FIGURE 16

Sample SPX 3-Day Future Performance versus Cubic Features in Training Set #7

These figures depict the scatter of future SPX 3-day performance relative to cubic features in training set 7. Scatter is plotted green when future performance Y is above the target threshold δ. Scatter is plotted red when future performance Y is below the target threshold δ.

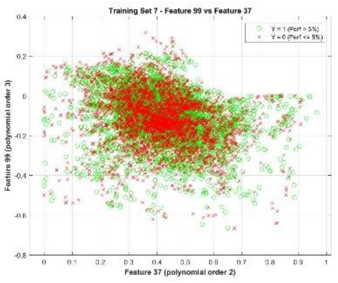
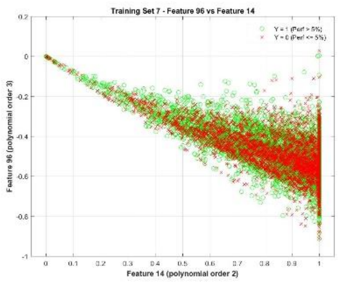
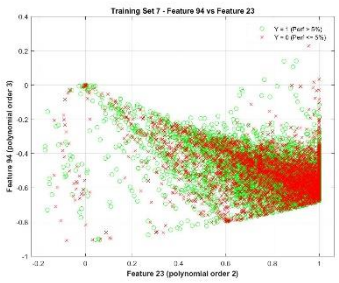
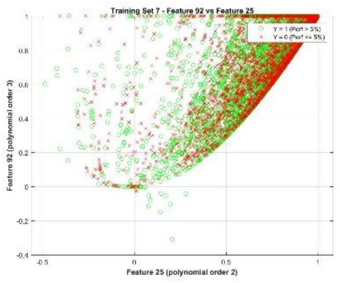
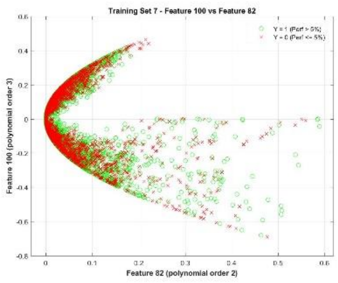
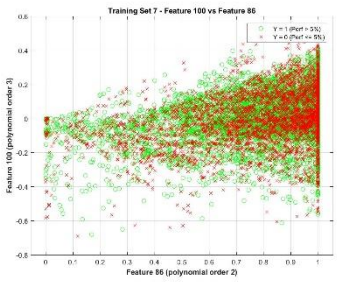
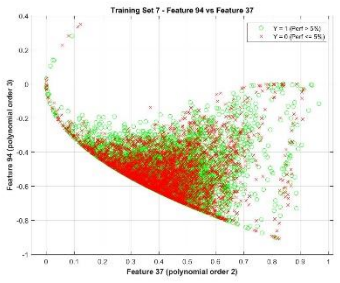
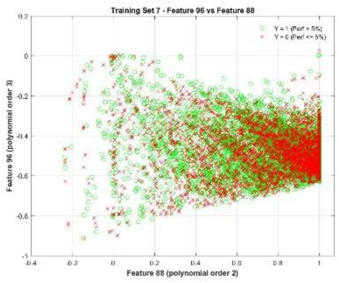
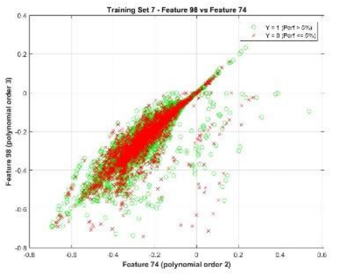
Feature #37 versus #99	**Feature #14 versus #96**	**Feature #23 versus #94**
		
Feature #25 versus #92	**Feature #82 versus #100**	**Feature #86 versus #100**
		
Feature #47 versus #94	**Feature #88 versus #96**	**Feature #74 versus #98**
		

FIGURE 16 (Continued)

Sample SPX 3-Day Future Performance versus Cubic Features in Training Set #7

These figures depict the scatter of future SPX 3-day performance relative to cubic features in training set 7. Scatter is plotted green when future performance Y is above the target threshold δ. Scatter is plotted red when future performance Y is below the target threshold δ.

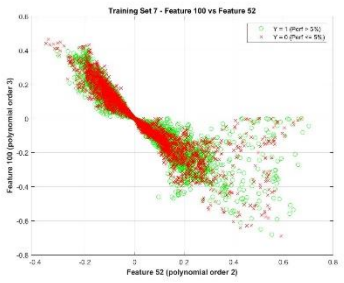
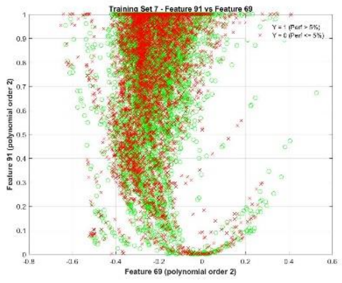
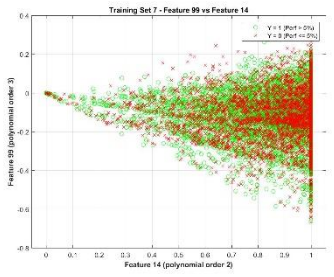
Feature #52 versus #100	**Feature #69 versus #91**	**Feature #14 versus #99**
		

FIGURE 17

SPX Long-Only Logistic Regression Performance

These figures depict the performance of the logistic regression long-only strategy. Data is from August 3, 1964 to December 12, 2018.

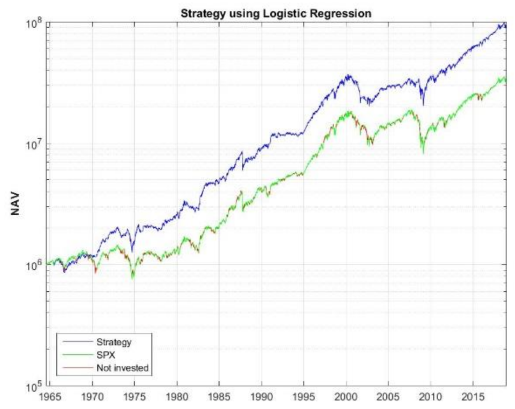
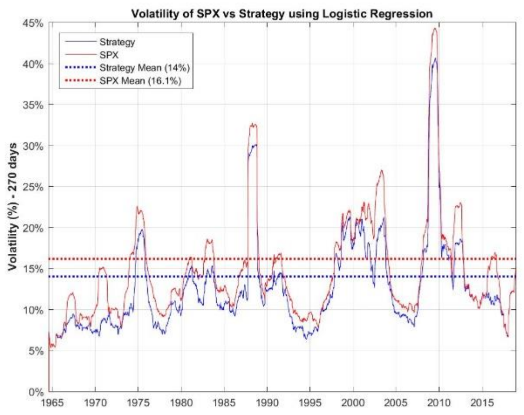
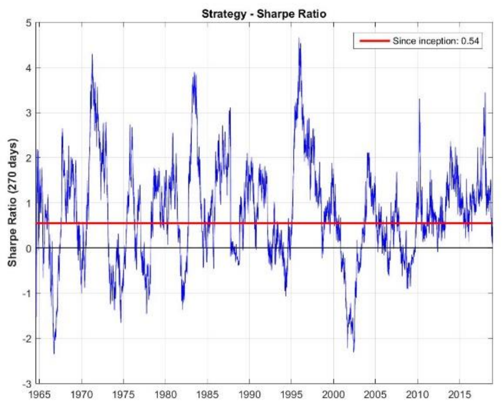
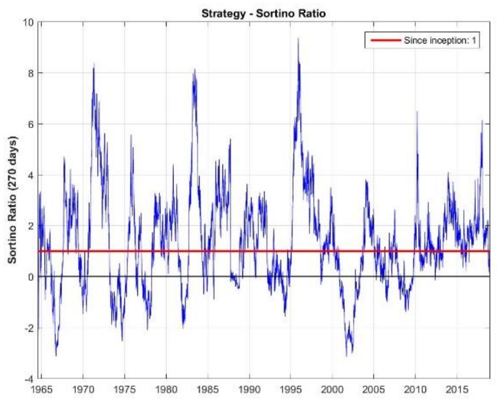
Growth of Strategy Starting at 1,000,000	**Annualized Volatility**
	
Rolling Annual Sharpe Ratio	**Rolling Annual Sortino Ratio**
	

FIGURE 17 (Continued)

Long-Only Logistic Regression Performance

These figures depict the performance of the logistic regression long-only strategy. Data is from August 3, 1964 to December 12, 2018.

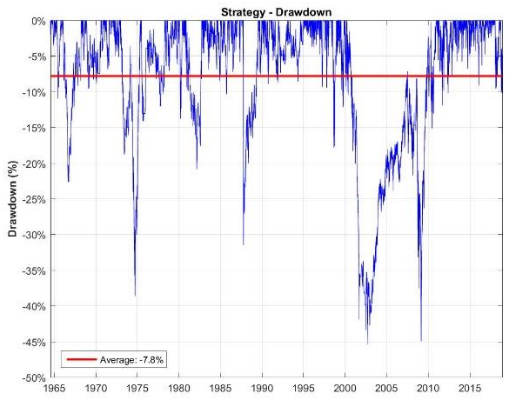
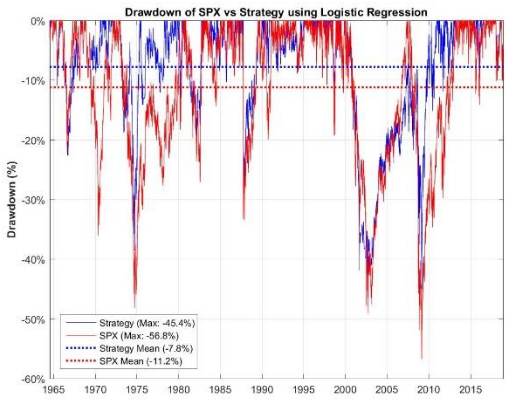
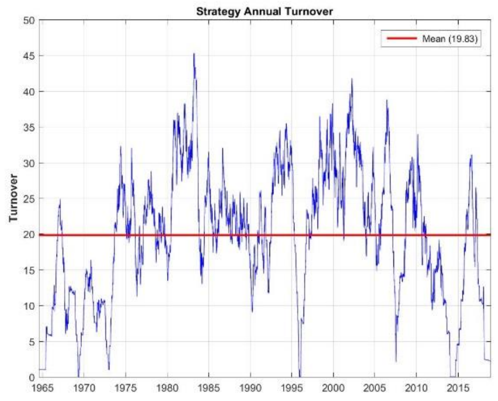
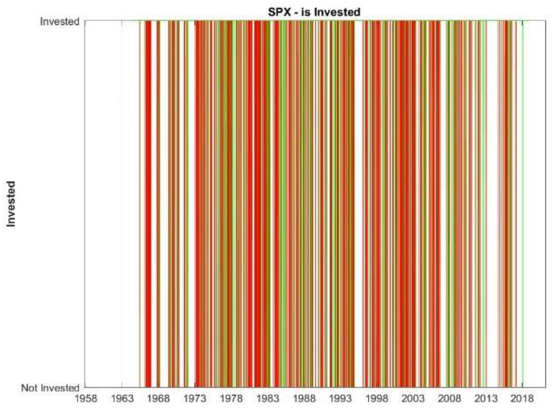
Daily Drawdowns	Daily Drawdowns Compared to SPX
	
Annual Turnover	**Trading Dates**
	

FIGURE 17 (Continued)

SPX Long-Only Logistic Regression Performance

These figures depict the performance of the logistic regression long-only strategy. Data is from August 3, 1964 to December 12, 2018.

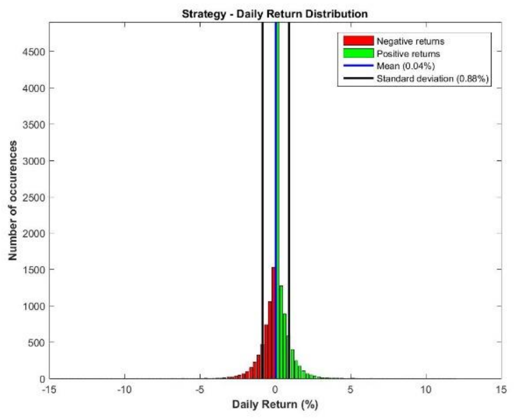
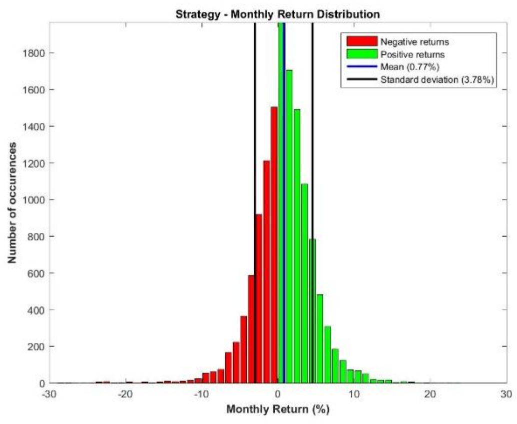
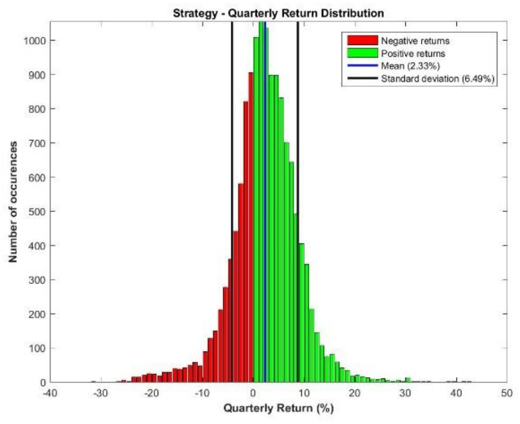
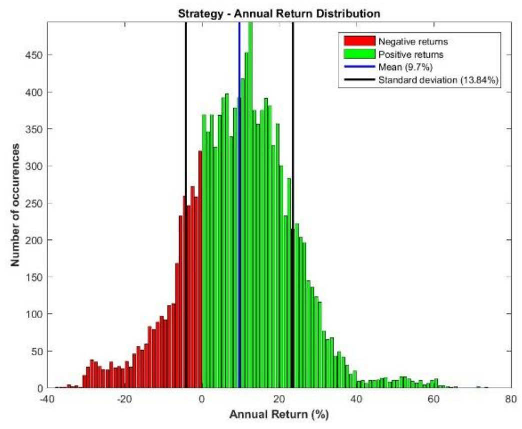
Distribution of Daily Returns	**Distribution of Monthly Returns**
	
Distribution of Quarterly Returns	**Distribution of Annual Returns**
	

FIGURE 18

Long-Only Logistic Regression Performance in Periods of 15% and 25% SPX Drawdowns

These figures depict the performance of the long-only logistic regression in periods of 15% and 25% SPX drawdowns.

Figure 18-a: Logistic Regression Performance in Periods of 15% SPX Drawdown

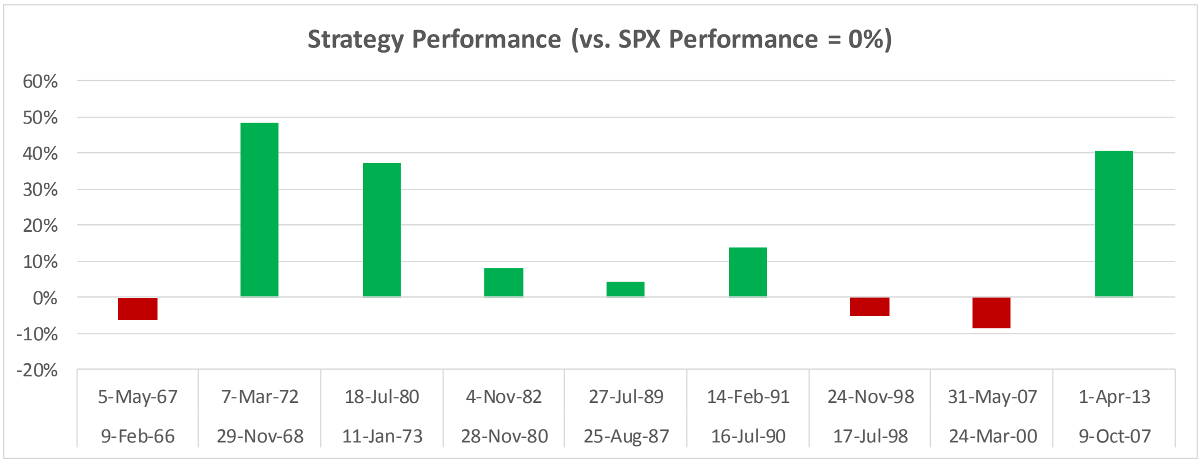


Figure 18-b: Logistic Regression Performance in Periods of 25% SPX Drawdown

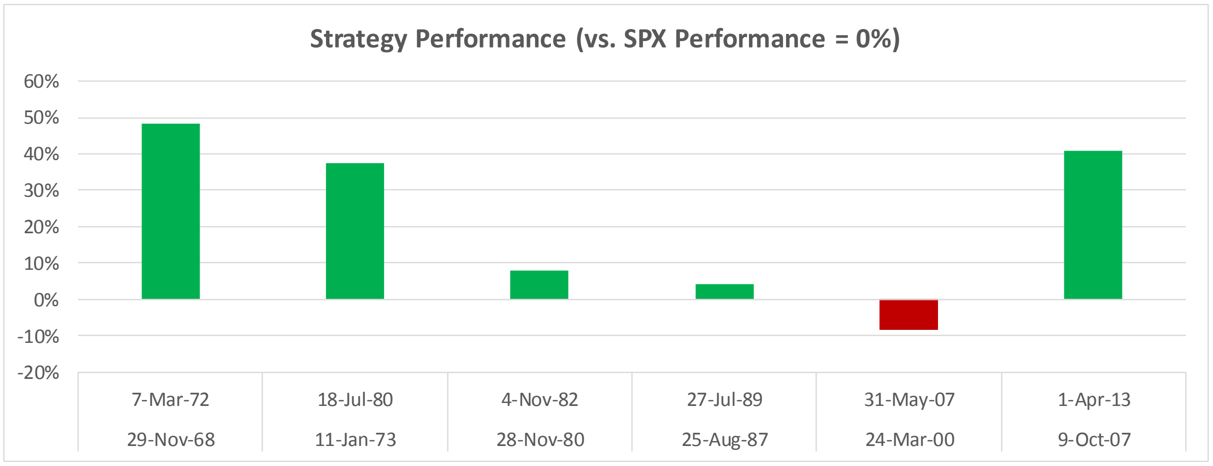


FIGURE 19

Drawdowns of SPX and Long-Only Logistic Regression in Periods When SPX Reaches 15% and 25% Drawdown

These figures depict the drawdowns of SPX and long-only logistic regression in periods when SPX reaches 15% and 25% SPX drawdowns.

Figure 19-a: Drawdowns of SPX and Logistic Regression in Periods When SPX Reaches A 15% Drawdown

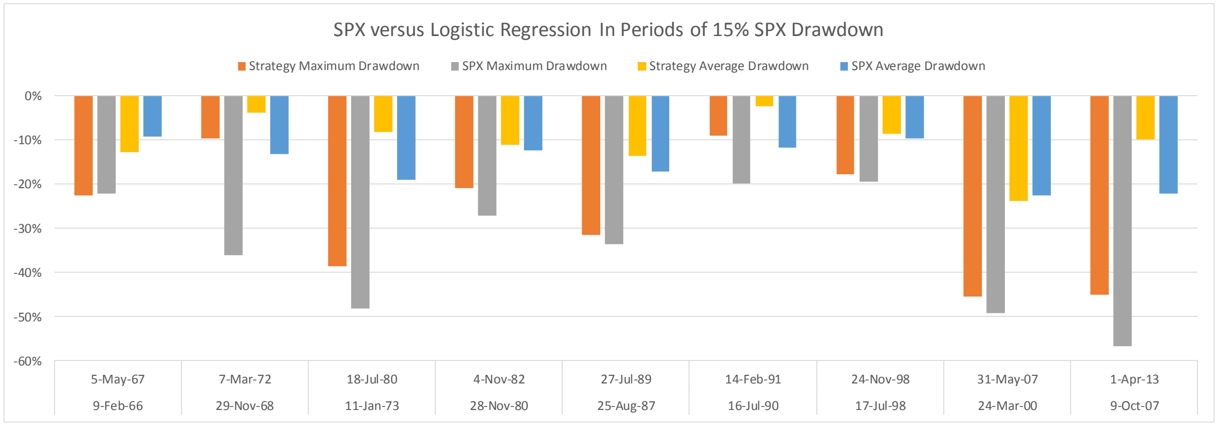


Figure 19-b: Drawdowns of SPX and Logistic Regression in Periods When SPX Reaches A 25% Drawdown

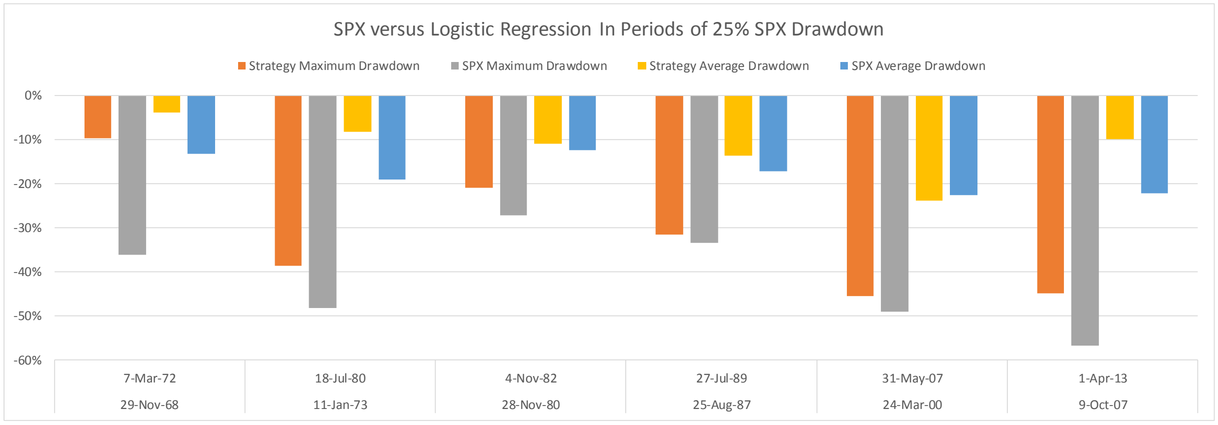


FIGURE 20

Long-Only Logistic Regression and SPX Returns by Test Set

These figures compare the returns of long-only logistic regression and buy-and-hold SPX by test sets. Logistic regression is depicted in **Blue**, and SPX is depicted in **Red**. Data is from August 3, 1964 to December 12, 2018.

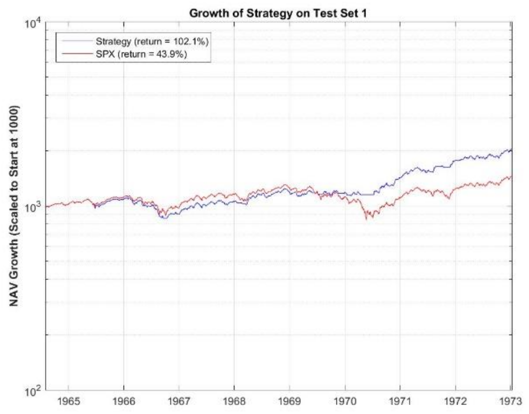
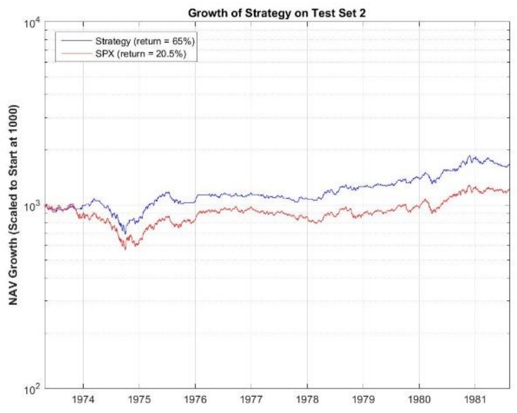
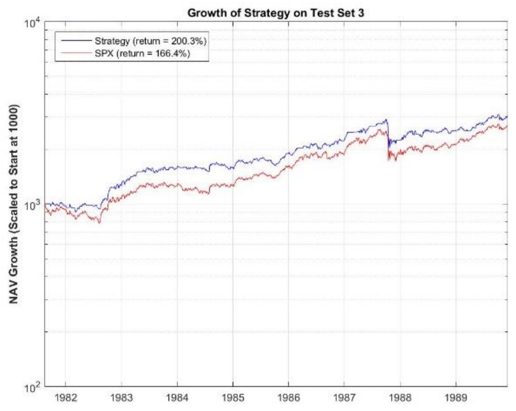
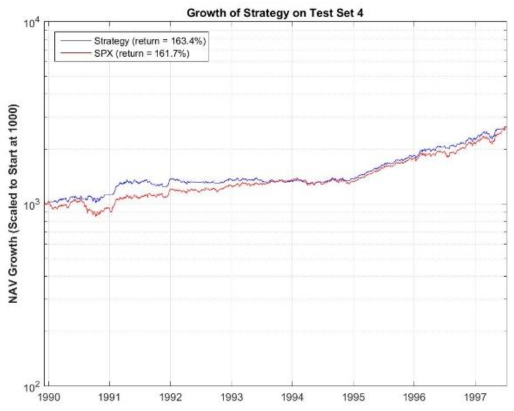
Test Set #1	**Test Set #2**
	
Test Set #3	**Test Set #4**
	

FIGURE 20 (Continued)

Long-Only Logistic Regression and SPX Returns by Test Set

These figures compare the returns of long-only logistic regression and buy-and-hold SPX by test sets. Logistic regression is depicted in **Blue**, and SPX is depicted in **Red**. Data is from August 3, 1964 to December 12, 2018.

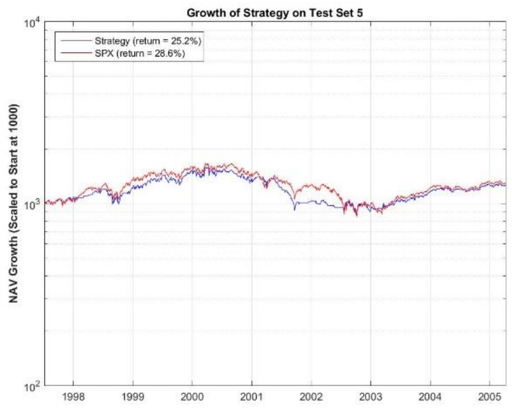
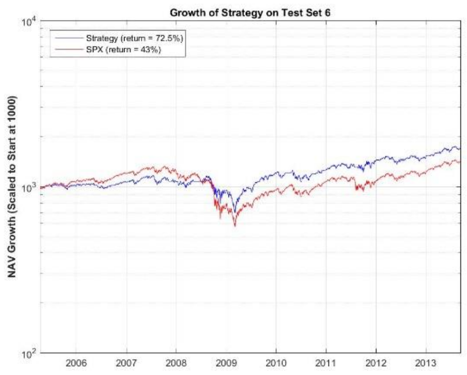
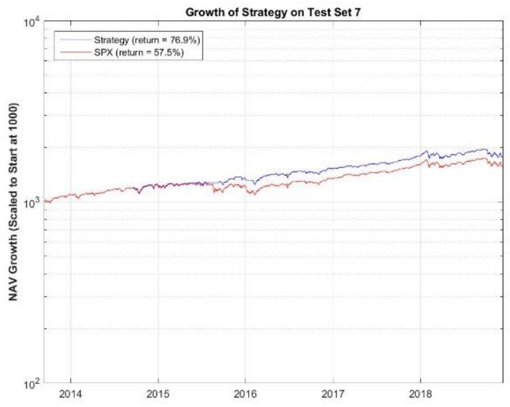
Test Set #5	**Test Set #6**
	
Test Set #7	
	

FIGURE 21

Long-Only Logistic Regression and SPX Drawdowns by Test Set

These figures compare the drawdowns of long-only logistic regression and buy-and-hold SPX by test sets. Logistic regression is depicted in **Blue**, and SPX is depicted in **Red**. Data is from August 3, 1964 to December 12, 2018.

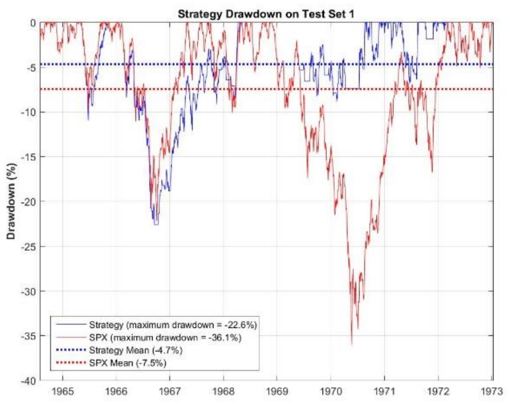
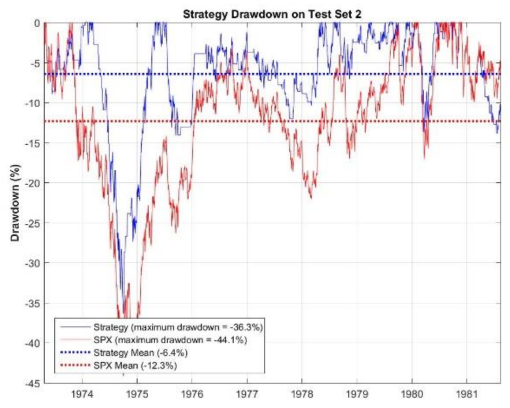
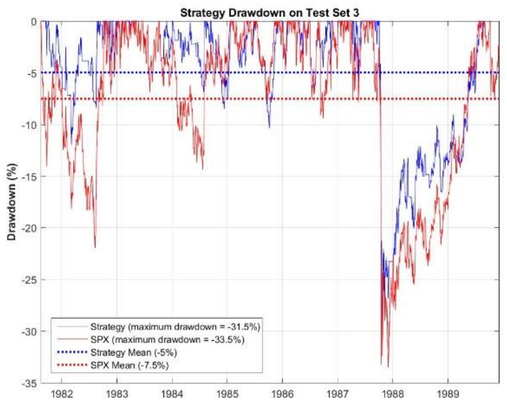
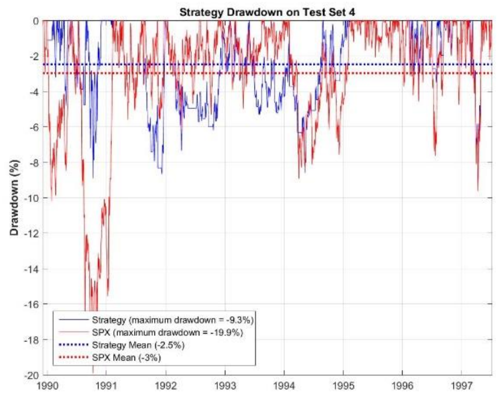
Test Set #1	**Test Set #2**
	
Test Set #3	**Test Set #4**
	

FIGURE 21 (Continued)

Long-Only Logistic Regression and SPX Drawdowns by Test Set

These figures compare the drawdowns of long-only logistic regression and buy-and-hold SPX by test sets. Logistic regression is depicted in **Blue**, and SPX is depicted in **Red**. Data is from August 3, 1964 to December 12, 2018.

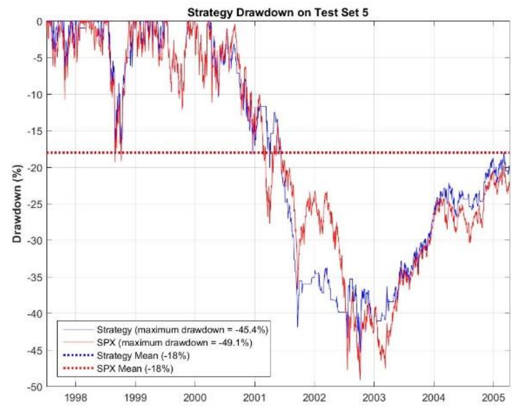
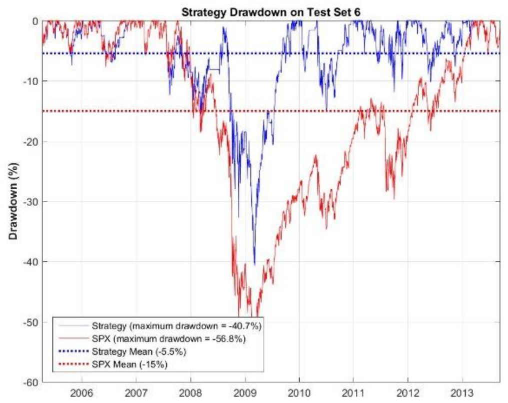
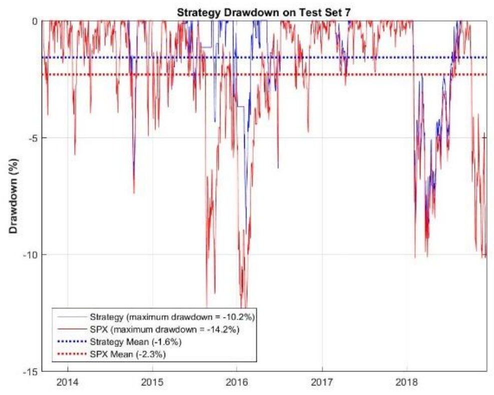
Test Set #5	**Test Set #6**
	
Test Set #7	
	

FIGURE 22

Long-Only Logistic Regression and SPX Annualized Volatility by Test Set

These figures compare the annualized volatility of long-only logistic regression and buy-and-hold SPX by test sets. Logistic regression is depicted in **Blue**, and SPX is depicted in **Red**. Data is from August 3, 1964 to December 12, 2018.

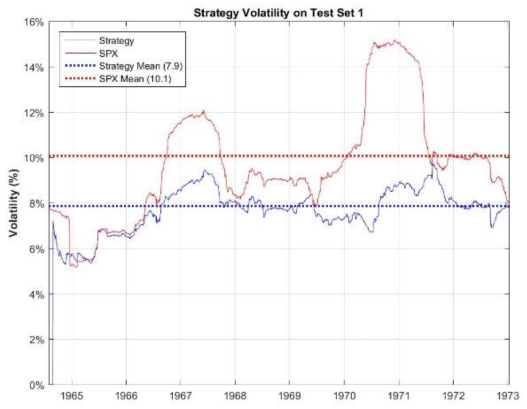
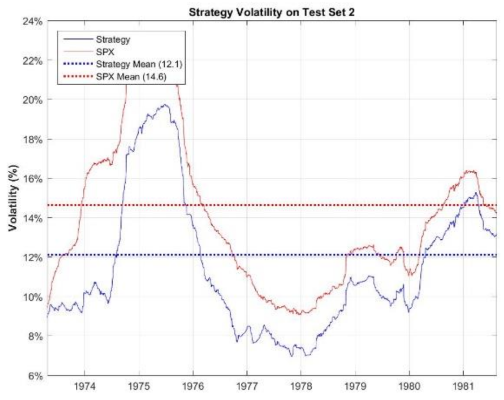
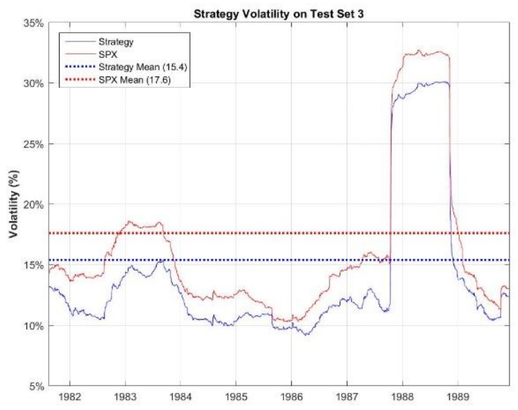
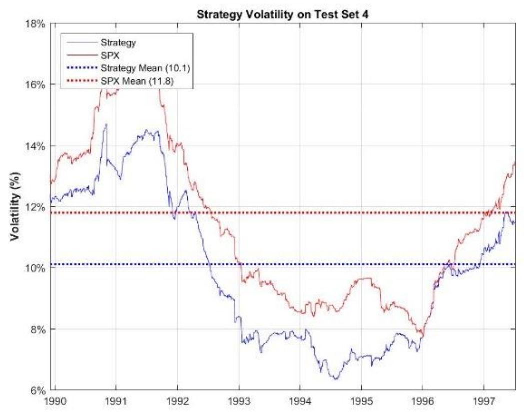
Test Set #1	**Test Set #2**
	
Test Set #3	**Test Set #4**
	

FIGURE 22 (Continued)

Long-Only Logistic Regression and SPX Annualized Volatility by Test Set

These figures compare the annualized volatility of long-only logistic regression and buy-and-hold SPX by test sets. Logistic regression is depicted in **Blue**, and SPX is depicted in **Red**. Data is from August 3, 1964 to December 12, 2018.

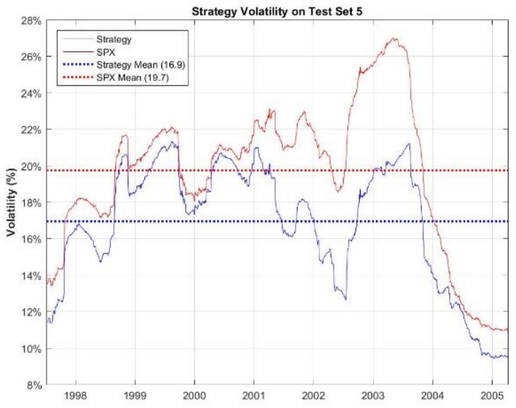
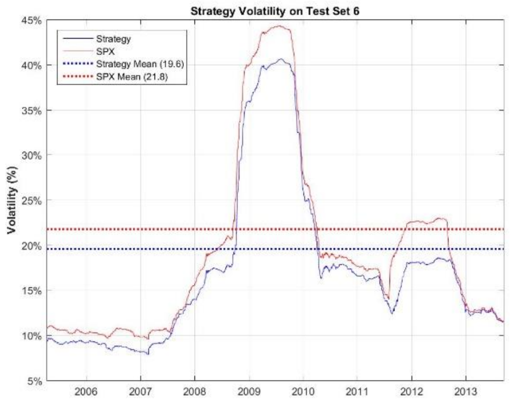
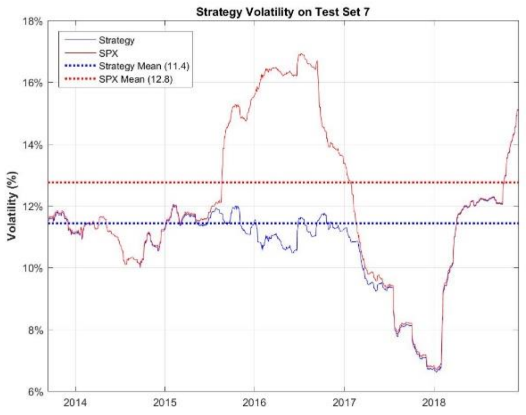
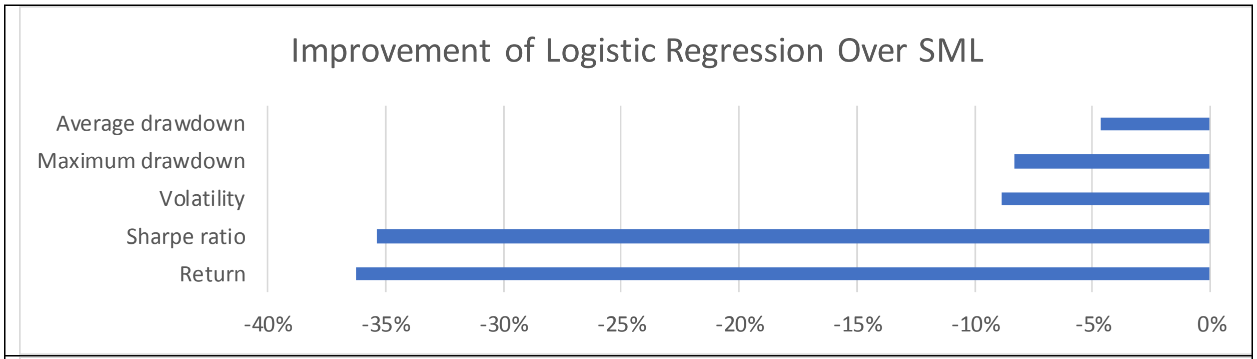
Test Set #5	**Test Set #6**
	
Test Set #7	
	

FIGURE 23

Relative Performance of Long-Only Logistic Regression Applied to Small and Mid-Cap U.S. Indices

These figures depict the relative excess performance of long-only logistic regression when applied to small and mid-cap U.S. indices.



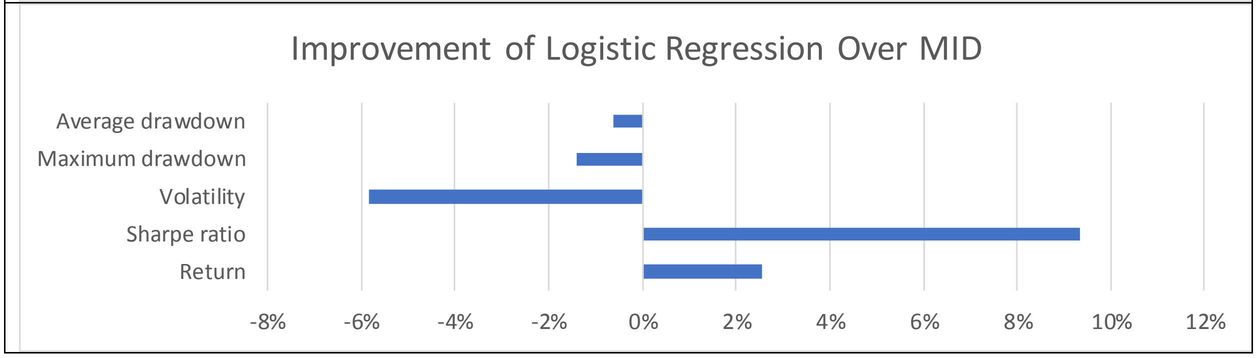
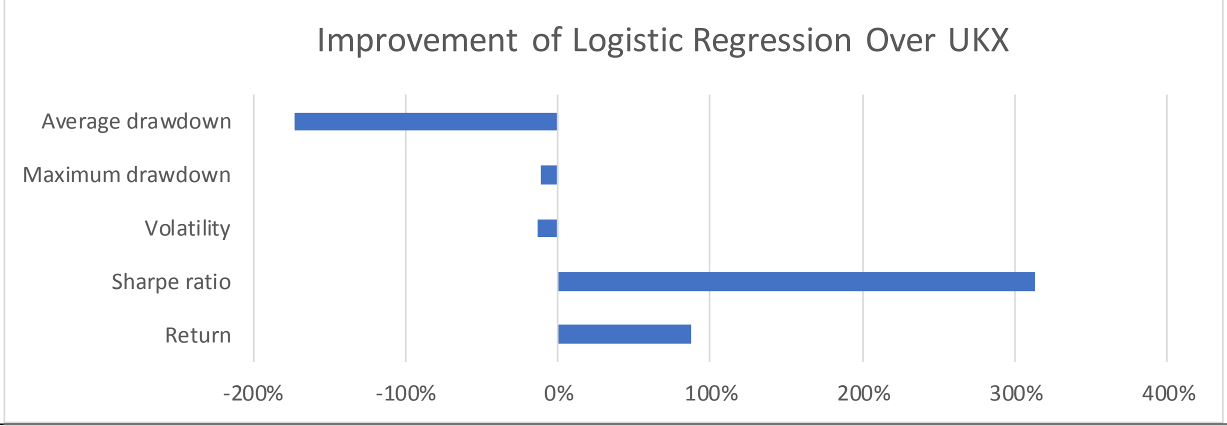
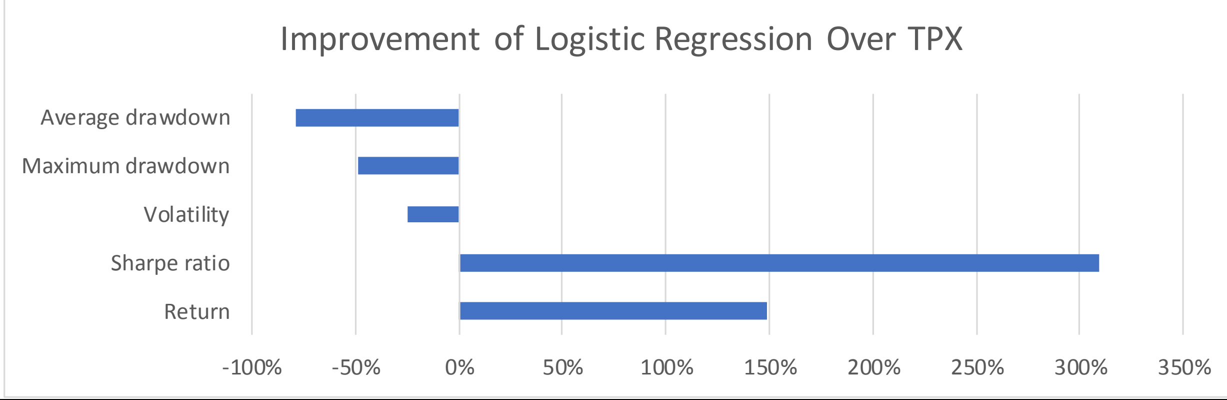


FIGURE 24

Relative Performance of Long-Only Logistic Regression Applied to Non-U.S. Equity Indices

These figures depict the relative excess performance of long-only logistic regression when applied to non-U.S. equity indices.





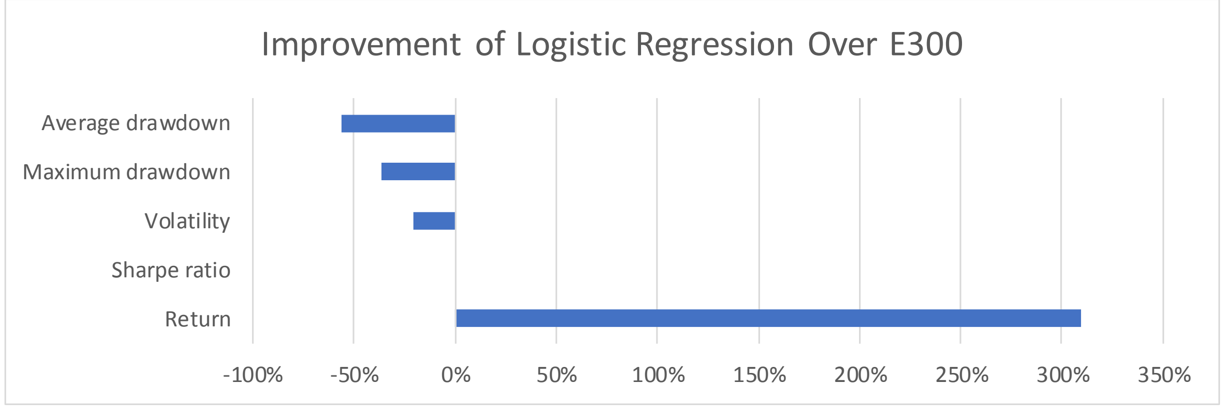
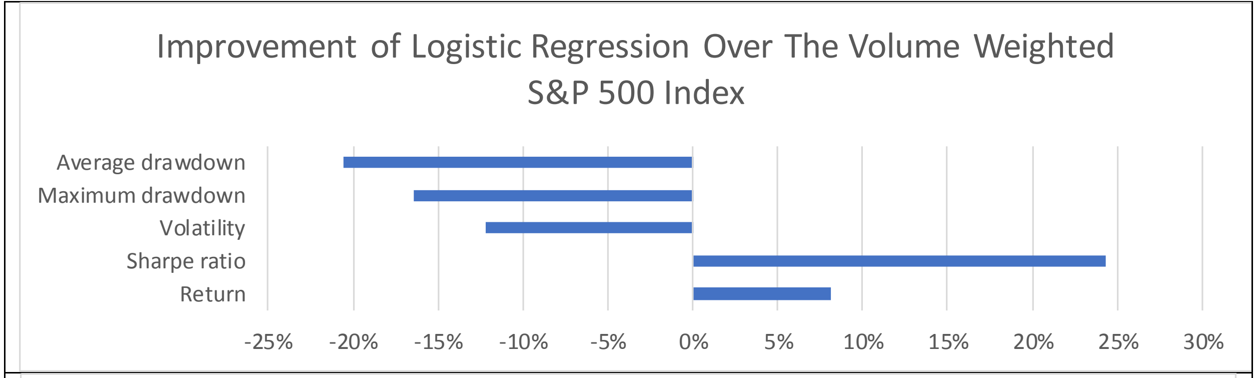
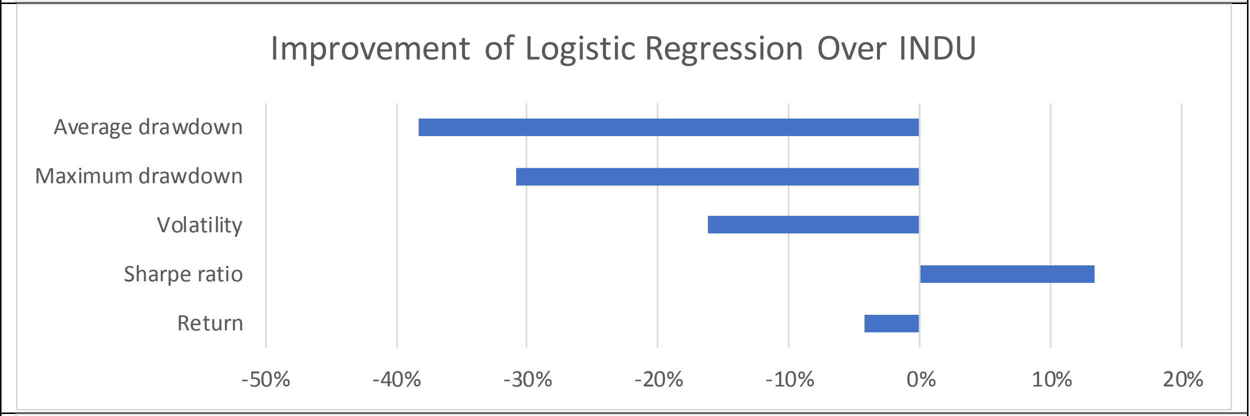


FIGURE 25

Relative Performance of Logistic Regression Applied to Other Large-Cap U.S. Equity Indices

These figures depict the relative excess performance of long-only logistic regression when applied to other large-cap U.S. equity indices.





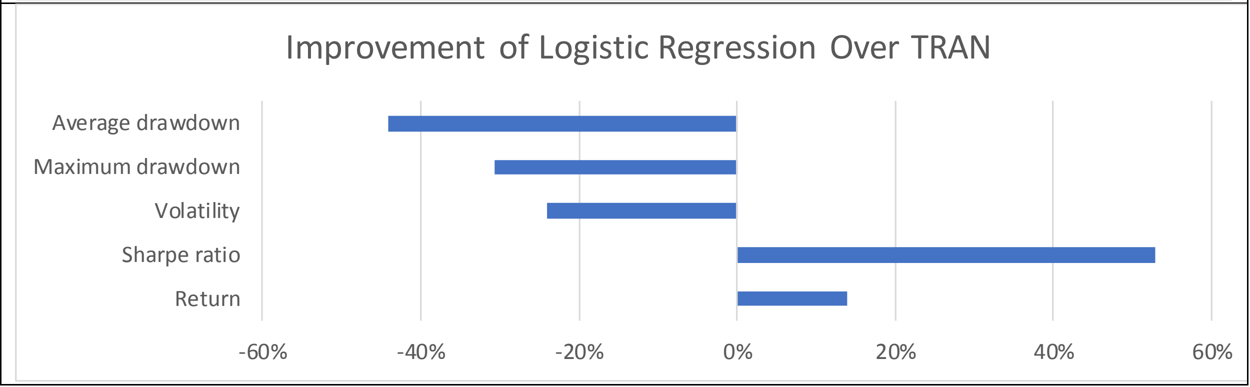


FIGURE 26

Impact of Retraining Frequency on the Performance of Logistical Regression Applied to SPX

These figures depict the impact of the retraining frequency on the performance of the SPX long-only strategy. The x-axis represents the number of days in the fixed retaining intervals. The y-axis represents different performance metrics. Data is from August 3, 1964 to December 12, 2018.

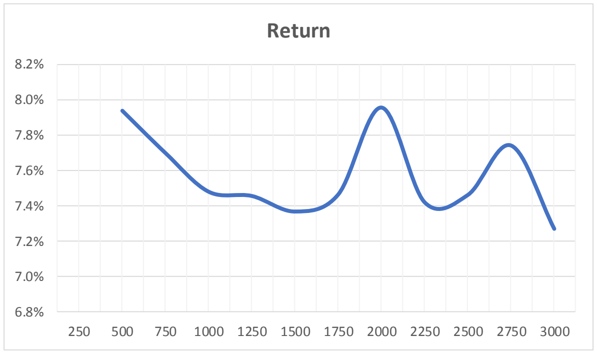
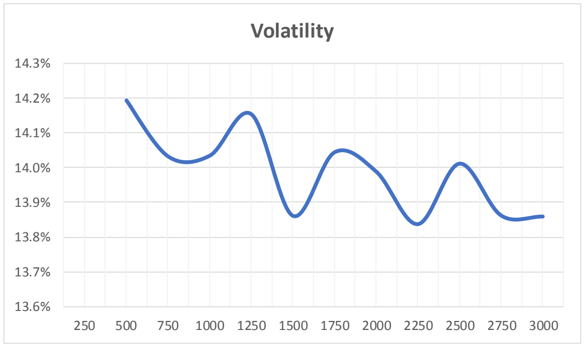
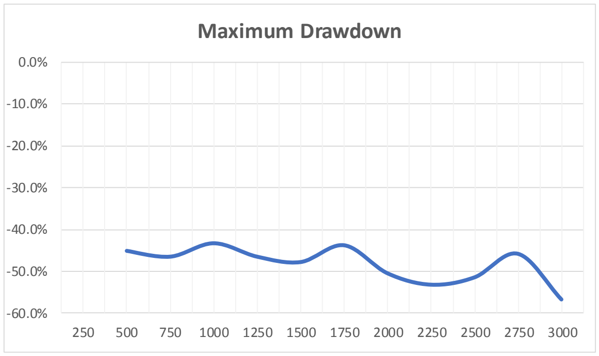
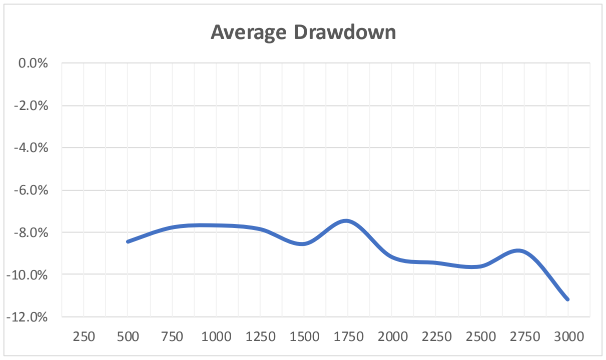
Annual Return	**Annual Volatility**
	
Maximum Drawdown	**Average Daily Drawdown**
	

TABLE 1

Sample Selection

This table summarizes the eight equity indices used in the study.

Name	Symbol	Start Date	End Date	Source
S&P 500 Index	SPX	December 30, 1927	December 12, 2018	Bloomberg
S&P Small Cap 600 Index	SML	December 31, 1993		
S&P Mid Cap 400 Index	MID	December 31, 1990		
FTSE 100 Index	UKX	December 19, 1997		
FTSEurofirst 300 Index	E300	December 31, 1985		
Tokyo Stock Exchange Price Index	TPX	December 19, 1997		
Dow Jones Industrial Average	INDU	January 2, 1920		
Dow Jones Transportation Average	TRAN	January 2, 1920		

TABLE 2

Definition of Performance Indicators

This table summarizes variable definitions.

Variables	**Descriptions**
Annual Return	Annual return is calculated as the ratio of the asset prices at the start and end of the time period under consideration.
Sharpe Ratio	Sharpe ratio is calculated as the average return minus the risk-free rate divided by the standard deviation of return on an investment.
Sortino Ratio	The Sortino ratio, a variation of the Sharpe ratio, differentiates harmful volatility from volatility in general by using a value for downside deviation. The Sortino ratio is the excess return over the risk-free rate divided by the downside semi-variance, and so it measures the return to "bad" volatility. (Volatility caused by negative returns is considered bad or undesirable by an investor, while volatility caused by positive returns is good or acceptable.) In this way, the Sortino ratio can help an investor assess risk in a better manner than simply looking at excess returns to total volatility, as such a measure does not consider how often returns are positive as opposed to how often they're negative.
Volatility	Volatility is calculated as the standard deviation of daily returns over a given period of time.
Maximum Drawdown	Maximum drawdown is calculated as the maximum value over a specified time interval of daily drawdowns, while the drawdown on a given day is the percentage difference between that day's asset value and the highest value of the asset prior to that day.
Average Daily Drawdown	Average daily drawdown is calculated as the average value over a specified time interval of daily drawdowns.
Precision	Precision is calculated as follows: $Precision = \frac{True\ positives}{True\ positives + False\ positives}$, while true positives are days which we predict to have positive performance and turn out to also have positive performance, and false positives are days which we predict to have positive performance and turn out to have negative performance
Recall	Recall is calculated as follows: $Recall = \frac{True\ positives}{True\ positives + False\ negatives}$, while false negatives are days which we predict to have negative performance and turn out to have positive performance
F Score	F score is calculated as follows: $F\ Score = 2\ \frac{Precision \times Recall}{Precision\ + Recall}$

TABLE 3

Index Performance Summary

This table summarizes the key performance indicators of our sample indices over the entire sample period.

	Annual Return	Sharpe Ratio	Volatility	Maximum Drawdown	Average Daily Drawdown
S&P 500 Index (SPX)	5.7%	0.25	19%	-86%	-22%
S&P Small Cap 600 Index (SML)	9.2%	0.40	21%	-59%	-9%
S&P Mid Cap 400 Index (MID)	10.8%	0.52	19%	-56%	-7%
FTSE 100 Index (UKX)	5.7%	0.27	17%	-53%	-13%
FTSEurofirst 300 Index (E300)	4.7%	0.21	18%	-61%	-19%
Tokyo Stock Exchange Price Index (TPX)	6.4%	0.31	17%	-76%	-29%
Dow Jones Industrial Average (INDU)	5.6%	0.26	18%	-89%	-21%
Dow Jones Transportation Average (TRAN)	5.0%	0.19	22%	-93%	-29%

TABLE 4

Learning SPX Best Retraining Time Windows – Error Metrics Across Data Sets

This table summarizes the error metrics across our seven training sets and seven test sets.

	Training Set 1	**Training Set 2**	**Training Set 3**	**Training Set 4**	**Training Set 5**	**Training Set 6**	**Training Set 7**
Start date	12/30/1927	09/10/1936	02/02/1945	07/07/1953	02/15/1961	12/24/1968	05/20/1977
End date	6/30/1964	03/21/1973	07/15/1981	11/02/1989	06/04/1997	03/08/2005	08/09/2013
Training sets:							
Cost ($J(\theta)$)	0.68	0.68	0.68	0.68	0.68	0.68	0.68
Precision	0.57	0.57	0.56	0.57	0.56	0.55	0.56
Recall	0.83	0.80	0.80	0.80	0.83	0.76	0.82
F-Score	0.68	0.66	0.66	0.66	0.67	0.64	0.67
Test sets:							
Cost ($J(\theta)$)	0.71	0.71	0.70	0.70	0.71	0.71	0.7
Precision	0.53	0.52	0.55	0.56	0.52	0.55	0.58
Recall	0.84	0.68	0.74	0.78	0.70	0.84	0.91
F-Score	0.65	0.59	0.64	0.65	0.60	0.66	0.71

TABLE 5

Investment Performance: SPX Versus Long-Only Logistic Regression

This table reports the investment performance of SPX versus long-only logistic regression. Data from August 3, 1964 to December 12, 2018.

	Annual return	**Sharpe Ratio**	**Volatility**	**Maximum drawdown**	**Average daily drawdown**
SPX	6.58%	0.35	16%	-57%	-11.2%
Logistic Regression with Initial Features only (Linear algorithm)	6.5%	0.36	15%	-50%	-9.8%
Logistic Regression with Quadratic Polynomials	6.1%	0.37	15%	-50%	-9.8%
Logistic Regression with Cubic Polynomials	8.6%	0.54	14%	-45%	-7.8%
Logistic Regression with 4th Order Polynomials	7.2%	0.45	14%	-44%	-7.9%

TABLE 6

SPX Investment Performance: Classic Approach Time-series Dual-Momentum Long-Only Strategy

This table reports the investment performance of a classic time-series dual-momentum long-only strategy with various driving parameters. Data is from August 3, 1964 to December 12, 2018.

	Rebalancing Frequency (days)	Momentum Lag (days)	Minimum Profitability Threshold	Annual Return	Sharpe Ratio	Volatility	Maximum Drawdown	Average Daily Drawdown
Case 1	21	252	1%	5.52%	0.38	12%	-33.2%	-7.4%
Case 2	21	252	0%	6.03%	0.41	12%	-33.5%	-7.1%
Case 3	21	252	5%	4.87%	0.34	11%	-33.2%	-7.4%
Case 4	21	189	0%	5.59%	0.38	12%	-33.2%	-7.6%
Case 5	21	189	1%	5.35%	0.37	12%	-33.2%	-8.0%
Case 6	21	189	5%	4.85%	0.34	11%	-33.2%	-8.5%
Case 7	1	252	5%	4.82%	0.31	12%	-51.2%	-12.1%
Case 8	1	252	0%	5.87%	0.37	13.0%	-52.2%	-11.6%

TABLE 7

Investment Performance: SPX Versus Long-Only Logistic Regression, by Test Sets

This table reports the investment performance of SPX versus long-only logistic regressions by test sets. Returns are cumulated over the entire test set period. Data is from August 3, 1964 to December 12, 2018.

	Return (not annualized)	**Sharpe Ratio**	**Volatility**	**Maximum Drawdown**	**Average Daily Drawdown**
Test Set #1 (Aug 3, 1964 – Apr 23, 1973)					
SPX	43.9%	0.34	10.1%	-36.1%	-7.5%
Logistic Regression	102 .1%	0.98	7.9%	-22.6%	-4.7%
Change	+133%	+189%	-22%	-37%	-37%
Test Set #2 (Apr 24, 1973 – Aug 14, 1981)					
SPX	20.5%	0.09	14.6%	-44.1%	-12.3%
Logistic Regression	65%	0.43	12.1%	-36.3%	-6.4%
Change	+218%	+398%	-17%	-18%	-48%
Test Set #3 (Aug 17, 1981 – Dec 5, 1989)					
SPX	166.4%	0.65	17.6%	-33.5%	-7.5%
Logistic Regression	200.3%	0.86	15.4%	-31.5%	-5%
Change	+20%	+31%	-13%	-6%	-17%
Test Set #4 (Dec 6, 1989 – Jul 7, 1997)					
SPX	161.7%	1.06	11.8%	-19.9%	-3%
Logistic Regression	163.4%	1.25	10.1%	-9.3%	-2.5%
Change	+1%	+18%	-14%	-8%	0%
Test Set #5 (Jul 8, 1997 – Apr 8, 2005)					
SPX	28.6%	0.12	19.7%	-49.1%	-18%
Logistic Regression	25.2%	0.11	16.9%	-45.4%	-18%
Change	-12%	-1%	-14%	-8%	0%
Test Set #6 (Apr 11, 2005 – Sept 11, 2013)					
SPX	43%	0.15	21.8%	-56.8%	-15%
Logistic Regression	72.5%	0.29	19.6%	-40.7%	-5.5%
Change	+69%	+90%	-10%	-28%	-64%
Test Set #7 (Sept 12, 2013 – Dec 12, 2018)					
SPX	57.5%	0.63	12.8%	-14.2%	-2.3%
Logistic Regression	76.9%	0.92	11.4%	-10.2%	-1.6%
Change	+34%	+45%	-10%	-28%	-32%

TABLE 8

Long-Only and Long-Short Investment Performance: Classic Time Series Dual Momentum and Logistic Regression Applied To SPX

This table compares the investment performance of long-only and long-short strategies applied to SPX: Classic time series dual-momentum and logistic regression. Data from August 3, 1964 to December 12, 2018.

	Annual return	**Sharpe Ratio**	**Volatility**	**Maximum drawdown**	**Average daily drawdown**
SPX	6.58%	0.35	16%	-57%	-11.2%
Classic Dual-Momentum / Long Only	4.87%	0.34	11%	-33%	-7.4%
Classic Dual-Momentum / Long-Short	4.07%	0.17	19%	-76%	-25.4%
Cubic Logistic Regression / Long Only	8.6%	0.54	14%	-45%	-7.8%
Cubic Logistic Regression / Long-Short	10.0%	0.55	16%	-51%	-10.2%

TABLE 9

Logistic Regression Long-Only Algorithm Applied to Other Equity Indices

This table reports the investment performance of the logistic regression long-only algorithm when applied to other equity indices.

	Annual return	**Sharpe Ratio**	**Volatility**	**Maximum drawdown**	**Average daily drawdown**
SPX	6.58%	0.35	16.1%	-56.8%	-11.2%
Logistic Regression	8.60%	0.54	14%	-45%	-7.8%
SML *(small caps)*	8.43%	0.33	22.5%	-59%	-9%
Logistic Regression	5.37%	0.21	20.5%	-54%	-8%
MID *(mid caps)*	7.46%	0.31	20.6%	-56%	-8%
Logistic Regression	7.65%	0.34	19.4%	-56%	-8%
UKX *(non U.S.)*	1.51%	0.03	18.7%	-53%	-16%
Logistic Regression	2.83%	0.11	16.2%	-47%	-12%
TPX *(non U.S.)*	3.54%	0.13	18.8%	-76%	-38%
Logistic Regression	8.81%	0.55	14.2%	-39%	-8%
INDU	6.36%	0.34	15.7%	-54%	-10%
Logistic Regression	6.09%	0.39	13.2%	-37%	-6%
TRAN	7.25%	0.32	19.4%	-61%	-15%
Logistic Regression	8.26%	0.49	14.8%	-42%	-9%
E300 *(non U.S.)*	0.58%	-0.02	19.4%	-61%	-26%
Logistic Regression	2.37%	0.09	15.4%	-39%	-11%