Validating Weak Form Market Efficiency in American Stock Markets with Trend Deterministic Price Data and Machine Learning

Samuel Showalter, Jeffrey Gropp

Department of Economics, DePauw University, Greencastle, IN 46135, USA

Abstract

The Efficient Market Hypothesis has been a staple of economics research for decades. In particular, weak form market efficiency – the notion that past prices cannot predict future performance – is strongly supported by econometric evidence. In contrast, machine learning algorithms implemented to predict stock price have been touted, to varying degrees, as successful. Moreover, some data scientists boast the ability to garner above-market returns using price data alone. This study endeavors to connect existing econometric research on weak-form efficient markets with data science innovations in algorithmic trading. First, a traditional exploration of stationarity in stock index prices over the past decade is conducted with Augmented Dickey-Fuller and Variance Ratio tests. Then, an algorithmic trading platform is implemented with the use of five machine learning algorithms. Econometric findings identify potential stationarity, hinting technical evaluation may be possible, though algorithmic trading results find little predictive power in any machine learning model, even when using trend-specific metrics. Accounting for transaction costs and risk, no system achieved above-market returns consistently. Our findings reinforce the validity of weak-form market efficiency.

Keywords: Efficient Market Hypothesis, Support Vector Machine, Random Forest, Bayesian Theory, K-Nearest Neighbors, Logistic Regression, Moving Average, Dickey-Fuller Test, Stationarity, Unit Root, Variance Ratio

1. Introduction

Stock market efficiency has been proposed, tested, and debated for nearly a century. Three versions of the Efficient Market Hypothesis have arisen from an immense body of economic and econometric research. For a market to be deemed efficient, prices must reflect currently available information [18]. In its weakest form, available information is defined to be historical prices. More stringent theories like the semi-strong EMH include all publicly available information, while the strongest form adds to that information that is not publicly known [18].

While efficient market research gained wide-spread attention in the 1960s and 1970s, these theories originated nearly a century prior. In his book *Theory of Speculation*, Bachelier [1] first proposed the notion of efficient markets as the concept of "fair game" economics. The expected return of the speculator is zero. [1] In fact, his characterization of stock price movements as a drunkard's errant walk coined the term "Random Walk", used to name the Random Walk Theory [12].

Today, many econometric and statistical techniques are employed to validate efficient market behavior. In particular, unit root tests discern whether or not time series data is stationary [14]. Stationarity characterizes a time-series whose mean, variance, autocorrelation, and other traits remain constant over time, implying the presence of trending and potential exploitation opportunities. Conversely, a unit root represents a time series that demonstrates systematic randomness, and can be one cause for not seeing stationarity in data [40]. Methods for testing for a unit root include, but are not limited to, [9, 40, 14] Dickey-Fuller, Phillips-Perron, and Kwiatkowski-Phillips-Schmidt-Shin (KPSS) tests. At the same time, alternative tests of the weak form EMH include the Variance Ratio, Serial Correlation, and Runs tests [3].

Evidence supporting efficient market hypotheses from the 1960s went relatively undisputed for decades. Though the EMH[1] has been tested and supported by many economists, Eugene Fama is considered the pioneer of efficient market research. In his 1965 dissertation [17], Fama extensively examines the characteristics of stock price movement, ultimately publishing what he and many defer to as "strong and voluminous evidence in favor of the random-walk hypothesis." His work identifying the nature of speculation and price movement continues well into the 21st century, often in collaboration with Kenneth French [22].

Researchers aligned with efficient market theories experiment with different methods, but often converge to the same conclusion: one cannot generate above-market returns through speculation or "timing the market" when attributes such as risk, transaction costs, and expenses are considered [34]. Non-stationarity in stock data implies – but does not prove – an inability to predict future price movement and generate above-market returns through speculation. One example of a historical trending anomaly proven to not be consistently predictive is the January effect, where stock prices consistently rise in January every year [34]. The Small Firm Effect similarly asserts that

[1] Unless otherwise specified, EMH in this study refers to weak-form market efficiency.

smaller firms outperform larger companies, though little evidence verifies its validity. Other, more far-fetched occurrences include the Hemline Effect, where the popular hemline length of women's clothing indicates the returns of the market [34].

Dominance of EMH theory peaked in the 1970s [45], as research opposing the Efficient Market Hypothesis gained traction around the 1980s. At this time, evidence countering market efficiency grew, but through different lines of reasoning. For example, Modigliani et. al. (1979) regard asset prices as chronically undervalued [33] due to inflation, while others attribute market inefficiency to irrationality on behalf of the economic participant [12]. Groundbreaking research on the psychology of choice finds people inherently and consistently act irrationally when influenced by certain environments [48]. Even so, some maintain that consistently achieving above-market returns is still unlikely when returns are adjusted for risk and fees, despite evidence of market trending [2] . In this regard, stock market inefficiency and speculative arbitrage are separate matters entirely. Establishing markets are inefficient does not inherently imply that above average returns can be gleaned through arbitrage.

Today, yet another field is influencing the validity of the weak form EMH. Innovations in technology have given rise to artificially intelligent systems. In particular, the influence of machine learning – a discipline dedicated to algorithms that "learn" to classify or pattern match – is altering the nature of business and academia alike [50]. One field particularly affected is financial engineering. The use of algorithmic trading platforms seeking arbitrage opportunities has proliferated, as have publications detailing methods of attracting "abnormal returns" [11, 26]. Some analyses assert their trading algorithms are able to outperform the market outright by training models on price data alone [38].

2. Contribution

This paper seeks to merge independent silos of research in economics and machine learning, ultimately to garner a holistic perspective on the validity of the weak form Efficient Market Hypothesis. While debate on efficient markets persists in economics, few contest the weak form EMH due to its conservative nature [20]. In general, economist sentiment in academia affirms the weak form Efficient Market Hypothesis, and many make such an assumption when conducting research unrelated to market efficiency [46].

By contrast, machine learning research has found little consensus in its findings, partially due to the wide variety of implementations used to predict stock market movement. Nevertheless, there is a consistent trend of deriving above average returns from publicly available data [42, 16, 11, 38]. The majority of implementations make use of information that may not be considered by the weak from EMH. Even so, studies completed by [11] and [38] purport to have built a system that can beat the market on price data alone.

Perhaps machine learning solutions are correct. The value of AI is embedded in its ability to perform as well or better than humans at pattern detection [35]. Moreover, a growing population of economists are not finding financial markets around the world to be weak-form efficient [51, 37, 7, 41]. Granted, some of these markets are considered to be "emerging" and not fully developed.[2].

Thus, to control for as many extraneous factors as possible and also test the weak form EMH, we first implement two econometric tests for market efficiency. Augmented Dickey-Fuller (ADF) and Lo-Mackinlay [31] variance ratio tests are commonly used to support and refute the Random Walk Hypothesis, respectively. While each does not explicitly indicate whether above-market returns are possible through speculation or arbitrage, they provide a framework from which further exploration can be conducted. There may be opportunities to exploit market trends to gain a profit if econometric studies indicate as much.

To logically transition between stationarity and market returns, an algorithmic trading platform is built and implemented. Daily stock index price and volume data from developed American financial markets – IXIC, GSPC, RUT, and DJI[3]– is collected. Then, 10 technical indicators identical to those collected by [38] are derived in the same manner. Trend-specific metrics that convert ratios into their buy sell intuition are also generated[4]

Stock data is fed into a machine learning platform that predicts the following day's stock price momentum (down, or up[5]) using the technical ratios from the previous day. After each prediction is made, the de-facto outcome of the market is harnessed to re-train the system such that it constantly has current information. Prediction results feed a trading platform that tracks performance (%-return, Alpha, Beta, Volatility, Sharpe, Sortino) using $100,000 of principal to trade a specific security. Risk free and buy-and-hold returns are used as benchmarks.

Stationarity, prediction, and trading tests together provide a general perspective of weak form market efficiency's validity. While this is but one approach to garnering above-market returns, it provides strong evidence for characterizing the weak form EMH. Independent verification is also conducted with 100 randomly selected stocks traded as a part of the S&P 500.

3. Stock and Stock Index Data Sources

The algorithmic trading platform only uses adjusted close prices and trades according to the black demarcations in Figure 1. The blue timeline represents market open and close on a given day, and the trading platform's training data depends on the adjusted close the previous day.

[2] A wide body of research hints that emerging markets and the economies of countries in socio-political turmoil are not efficient.

[3] These tickers represent the Nasdaq, S&P 500, Russell 2000, and Dow Jones Industrial Average.

[4] These are determined by modeling the same indicators as [38].

[5] A designation of "up" also includes no change in price, as it represents a break even point or better for the investor.

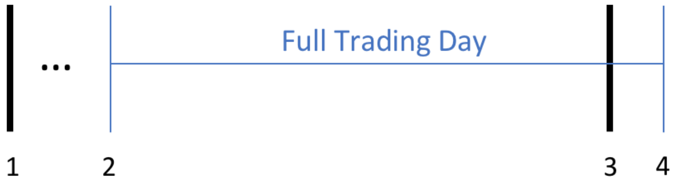


Figure 1: Visualization of regular trading day compared to trade interval for platform. 2 and 4 represent market open and market close. 1 and 3 represent the moment milliseconds before market close over a two day consecutive window.

However, we assume the algorithm can not trade after hours. The algorithm generates a prediction and trades using the price found during the final milliseconds before the trading day ends[6]. These trades stem from predictions on the market activity the following day. A maximum of one trade occurs daily, and mistakes do not alter trading decisions.

Stock index data is acquired over Yahoo Finance, while the stock price data for individual securities is procured through the Quandl WIKI API. These data sources are reputable, but prices were programmatically checked for blatant errors and missing data before any experimentation was conducted. Both of these data sources define adjusted close as the closing stock price adjusted for stock splits, reverse splits, and dividends.

4. Quantitatively Characterizing Efficient Markets

4.1. Econometric Examination of Stochastic Price Trends

Beating the market is a multi-faceted concept. Not only does it require that price movements exhibit predictability, but that variations must have the potential to be systematically exploited [22]. Most commonly, value investors believe that markets under- and over-value stocks based on public sentiment. Frequent exploitation of market efficiency inherently and paradoxically increases efficiency, asserted by [6] with high frequency trading. Therefore, markets will only maintain their inefficiency if exploited by a select few.

To simplify matters, economists often objectively test for randomness in stock data and use their findings as a proxy for both market inefficiency and the potential to generate above-market returns. Unit root tests have evolved into many different forms, partially because ”most aggregate economic time series contain a unit root” [27]. That is, most time-series in raw form appear non-stationary. However, some series can be converted into stationary systems through de-trending, differencing, and other methods [5]. Data coerced to stationarity in this manner is considered difference-stationary. If a unit root test implies the presence of stationarity, then there may exist securities with non-random price movements that can be exploited.

Variance Ratio tests rely on the assumption that the ratio of variance for two different time frames should be approximately proportional to the ratio of those lengths of time. The expansion of variance over time in non-stationary data is considered a linear process [8]. Additionally, Runs tests [49] examine the length of ”runs”, where stock price repeatedly moves in the same direction[7]. In particular, it discerns whether or not price movements are mean reverting, one of the most-commonly proposed hypotheses of Random Walk opponents [12].

4.2. Automated Trading and Binary Classification

Algorithmically, price movements are considered to have either up or down momentum. Two potential outcomes predicate two possible predictions, and converts stock price prediction into a binary classification problem. As displayed in Table 1, four potential outcomes arise from a prediction-result pair.

		Actual Price Δ	
Prediction	Price Up	True Positive (TP)	Price Up Positive (FP)
	Price Down	False Negative (FN)	True Negative (TN)
		Price Up	Price Down

Table 1: Confusion matrix for daily stock price momentum prediction.

Many assessment metrics are derived from Table 1. Listed in equations 1, 2, 3, and 4 are the eight basic indicators derived from confusion matrices: true and false positive and negative rates (TPR, FNR, TNR, FPR), positive and negative predictive value (PPV, NPV), false discovery rate (FDR), and false omission rate (FOR). Visually, all equations in the left column focus on successful classification, while the right column focuses on failure rates.

$$TPR = \frac{TP}{TP + FP} \qquad FNR = \frac{FN}{FN + TN} \tag{1}$$

$$TNR = \frac{TN}{TN + FN} \qquad FPR = \frac{FP}{FP + TP} \tag{2}$$

$$PPV = \frac{TP}{TP + FN} \qquad FDR = \frac{FN}{TP + FN} \tag{3}$$

$$NPV = \frac{TN}{TN + FP} \qquad FOR = \frac{FP}{TN + FP} \tag{4}$$

Conceptually, TPR and TNR determine how many price movements were appropriately classified. Both of these metrics are direct evaluations of model performance by class, and ask the question: *how many price movements were classified correctly?* Conversely, PPV and NPV can be described as a *prediction centric* evaluation of the confusion matrix, conditionally examining results given a specific prediction. That is,

[6] It is assumed that adjusted close stock prices are negligibly different from the price milliseconds before close. Adjusted close serves as a substitute for this price.

[7] In this analysis, stock movement is reduced to up or down momentum.

for all days predicted to have price increase, how many actually did?

Many higher-order measurements combine the ratios above into a holistic indicator of performance. Two particularly important for metrics are Accuracy and the F_1 Score [38]. Accuracy is a common form of evaluation that highlights the number of days that were classified correctly. By contrast, the F_1 Score places excess emphasis on upward price movement.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

$$F_1\ Score = \frac{2TP}{2TP + FP + FN} \tag{6}$$

Both measurements characterize the performance of different price trends, but tell little of overall trading efficacy. Therefore, portfolios are tracked with a variety of features, including the number of trades a system makes, portfolio returns, Alpha, Beta, Volatility, Sharpe, and Sortino. Defined and contextualized in *Research Methods*, portfolio evaluation allows investors to understand the basis of returns as a function of different factors like risk (σ) and correlation with the market benchmark (β). Modern Portfolio Theory (MPT) [32] involves gaining the highest return for a given level of risk. Optimally, an algorithmic trading system would abide by these tenets and exist on an efficient frontier.

5. Efficient Market Research: Findings and Challenges

Definitive conclusions on the weak-form Efficient Market hypothesis are difficult to ascertain. Evidence that stock prices follow a Random Walk does not support the notion that prices reflect all available information. Furthermore, asserting that stocks generally reflect all information is distinct from *always* reflecting available information immediately. Modern technology [6] allows for arbitrage to execute in microseconds, necessitating instantaneous price corrections.

When Fama first proposed the efficient market hypothesis [17], such technology did not exist. Over time, his continued research has transitioned from confident affirmation of EMH theories [18] to addressing skepticisms, and finally conceding that valid studies have "uncover[ed] empirical regularities" in price data [19] that provide a compelling argument for the contrapositive. Strong arguments maintain that financial markets are at least temporarily inefficient.

Leading critics of efficient market theories in academia are Lo and Mackinlay [30] and Richard Thaler [12], and the central notion of their research is that economic agents participating in financial markets do not universally act rationally [12]. Be it the Tulip Mania craze of the 1600s [34] or the 2008 financial crisis, there are many examples of what appears to be irrational speculation en-masse[8]. Similarly, some assert empirical evidence of randomness in stock price movements is also evidence of proper asset valuation. Opponents like [30] are quick to deem this non-sequitural, fallacious logic.

Complicating matters further, making money off of stock market trends[9] is often considered a separate debate from simply determining that stock prices do not follow a random walk. Many machine learning studies tout successful attempts to gain abnormal returns, albeit many do not include factors such as risk, transaction costs, and the like.

Additionally, studies by [11], [26], and others make use of sentiment data, company fundamentals, or other information that may not be considered when testing the weak-form efficient market hypothesis. However, [38] and [25] both train their models with technical ratios alone. In Indian markets, [38] achieve accuracy in excess of 85% for four securities as well as more than double the returns of their benchmarks. However, the limited nature of four results invites random anomalies or security picking by the authors. No information is provided on how these four securities were chosen. Regardless, our analysis will provide evidence that refutes these findings as they pertain to U.S. markets.

6. Research Methods

6.1. Econometric Analysis of Random Walk Theory

To garner a general understanding of market efficiency, baseline tests were run on 100 randomly[10] selected S&P 500 securities over the past ten years[11]. More specifically, both tests are run on the daily log returns[12]. Augmented Dickey Fuller (ADF) tests, one of the commonly cited tests of efficient market proponents, and Variance Ratio tests were both conducted. Lo and Mackinlay are staunch opponents of Random Walk theory [30] and created a specific implementation of the Variance Ratio test to verify their hypotheses [31]. Taken together, these independent analyses will both frame the ongoing random walk debate as well as the algorithmic experiments to come.

Augmented Dickey-Fuller (ADF) Tests: ADF tests are a form of unit root analysis that incorporates lags to account for serial correlation [9]. Over a long time period, data may appear to follow patterns when, in actuality, these movements are the product of randomness or drift[13] [29]. ADF is statistically defined as examining the null hypothesis of "an (autoregressive integrated moving average) ARIMA(p, 1,0) process against the stationary ARIMA(p + 1, 0, 0) alternative" [9]. If x_t is a time series, then ADF regression takes the form

$$\Delta x_t = \mu + \gamma t + \alpha x_{t-1} + \sum_{j=1}^{k-1} \beta_j \Delta x_{t-j} + \mu_t \tag{7}$$

[8] With that said, it can be argued that it is entirely rational to invest in something so long as buyers will pay a higher price. From this lens, agents in the Tulip Bulb mania were rational investors.

[9] It is assumed that abnormal returns are made off of markets that do possess trending.

[10] All S&P 500 stocks were shuffled using Numpy's (Python) pseudo-random number generator. Only stocks with at least 20 years of trading on a public exchange were chosen. This opens up the study to survivorship bias.

[11] The ten years spanned from Jan 1. 2008 to Jan. 1. 2018

[12] The natural logarithm of the first differences in the data.

[13] Just as subsets of the decimals of π appear to follow patterns, so too can random stock data.

where "Δ is the difference operator and μ_t, is a white-noise innovation" [9]. ADF tests in this study were automated using the *urca* package of R. The number of lags included in the analysis was automatically determined using the Akaike Information Criteria (AIC), and the maximum number of lags p_{max} considered was based on the paradigm of [43] outlined in equation 8.

$$p_{max} = 12\left(\frac{T}{100}\right)^{\frac{1}{4}} \tag{8}$$

T represents the total size of the sample (in this case, number of days). ADF tests return a test statistic t that lies on a t-distribution with $df = T - 1$. The resulting p-value of this analysis determines whether to reject the null hypothesis in favor of the alternative (equation 9),

$$H_0 : \rho = 1 \qquad H_a : \rho < 1 \tag{9}$$

where ρ represents the root. As implied by the name, $\rho = 1$ signifies a unit root, while $\rho < 1$ indicates stationarity. Conversely, if one were testing for an explosive root, $H_a : \rho > 1$ would be used.

Lo-Mackinlay (1988) Variance Ratio (VR) Tests: Variance ratio tests, conceptually outlined in *Econometric Examination of Stochastic Price Trends*, are implemented by first finding the variance ratio over k-periods [8] with the calculation of

$$Var(k) = \frac{Var(x_t + x_{t1} + \; + x_{tk+1})/k}{Var(x_t)} \tag{10}$$

$$= 1 + 2\sum_{i=1}^{k-1}(\frac{k-i}{k})\rho_i \tag{11}$$

"where ρ_i is the i-th lag autocorrelation coefficient of x_t" [8]. Using this logic for individual variance ratio calculation, the test statistic is calculated under the assumption that individual metrics are not *i.i.d*. The robust test statistic M_2 is denoted in equation 12.

$$M_2(k) = \frac{VR(x;k) - 1}{\phi^* k^{(\frac{1}{2})}} \tag{12}$$

The coefficient ϕ^* represents the asymptotic variance under heteroskedastic conditions and is examined in the two formulas below [8] under the assumption $V(k) = 1$.

$$\begin{aligned}\phi^*(k) &= \sum_{j=1}^{k-1}\left[\frac{2(k-j)}{k}\right]^2 \delta(j) \\ \delta(j) &= \left\{\sum_{t=j+1}^{T}(x_t-\hat{\mu})^2(x_{t-j}-\hat{\mu})^2\right\} \div \left\{\left[\sum_{t=1}^{T}(x_t-\hat{\mu})^2\right]^2\right\}\end{aligned}$$

In all tests conducted, we calculate $M_2(k)$ for daily intervals with the estimated mean $\hat{\mu}$. Daily intervals were chosen such that variance ratios corresponded to daily trading intervals. Test statistics and p-values are stored for all securities, the null hypothesis of Variance Ratio tests is equivalent to equation 9.

6.2. Automating Fraud Detection Experiments

To examine a variety of securities and models, a modular testing platform was created. Encapsulated in *Test* objects, a specified stock-data sample and model run predict price movement for a specific security, as shown in Figure 1. Results are stored after each testing cycle in a *Logger* object as a result log before the process repeats for a new security. If a trader is specified, stock predictions will be sent to an Automated Trader object which will execute and create a trade log documenting all portfolio activity. At the end of the experiment, a master log of all execution parameters, average results, and meta-data is sent to the *Logger*.

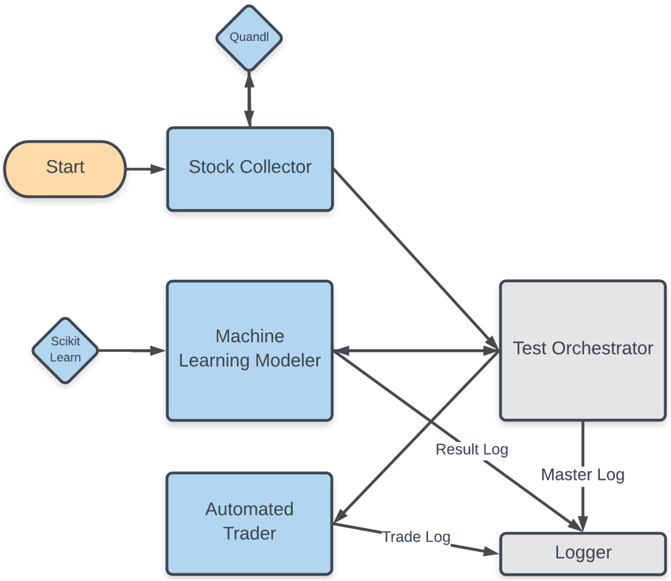


Figure 2: Scalable automation platform for testing fraud detection algorithms

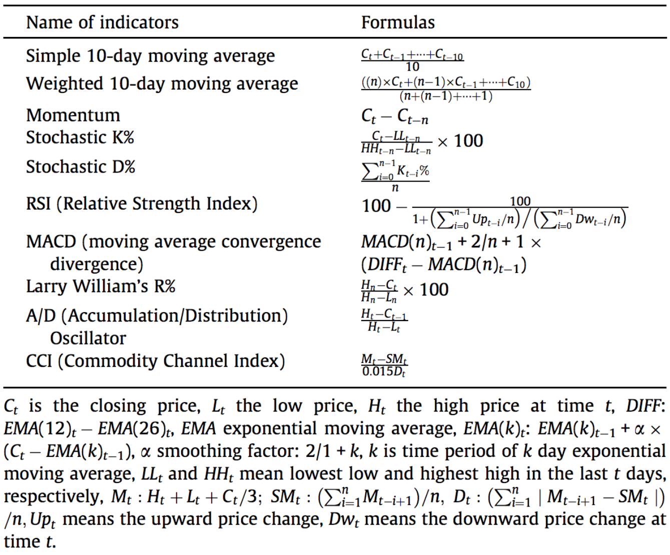
Stock Data: All executions begin with a command sent to the sampling engine that includes either a file name or stock ticker and a range of dates. The Stock Collector object will then either import stock prices from a flat file or download them via the Quandl API. Data first undergoes a quality assurance process that checks both for missing and ill-formatted data using outlier detection. If no issues are found, ten technical ratios are derived via the formulas in Figure 3.

Moving averages and other indicators necessitate a loss of data[14]. The number of empty rows is counted and stored before records are removed. Daily price momentum[15] is also shifted to the previous day. As mentioned in *Stock and Stock Index Data Sources*, technical ratios are utilized to predict price movement the following day. Therefore, by shifting price momentum back, a single row includes the technical indicators of that day as well as the price momentum[16] of the following. Finally, the refined stock dataset is passed to the Machine Learning Modeler engine.

[14] One cannot generate a 10 day moving average without have 9 days pass.

[15] For this study, upward price momentum is denoted with a 1, and downward a 0.

[16] As the class label, price momentum is central to training supervised machine learning algoirthms.



Name of indicators	Formulas
Simple 10-day moving average	$\frac{C_t+C_{t-1}+\cdots+C_{t-10}}{10}$
Weighted 10-day moving average	$\frac{((n)\times C_t+(n-1)\times C_{t-1}+\cdots+C_{10})}{(n+(n-1)+\cdots+1)}$
Momentum	$C_t - C_{t-n}$
Stochastic K%	$\frac{C_t-LL_{t-n}}{HH_{t-n}-LL_{t-n}} \times 100$
Stochastic D%	$\frac{\sum_{i=0}^{n-1} K_{t-i}\%}{n}$
RSI (Relative Strength Index)	$100 - \frac{100}{1+\left(\sum_{i=0}^{n-1} Up_{t-i}/n\right)/\left(\sum_{i=0}^{n-1} Dw_{t-i}/n\right)}$
MACD (moving average convergence divergence)	$MACD(n)_{t-1} + 2/n + 1 \times (DIFF_t - MACD(n)_{t-1})$
Larry William's R%	$\frac{H_n-C_t}{H_n-L_n} \times 100$
A/D (Accumulation/Distribution) Oscillator	$\frac{H_t-C_{t-1}}{H_t-L_t}$
CCI (Commodity Channel Index)	$\frac{M_t-SM_t}{0.015D_t}$

C_t is the closing price, L_t the low price, H_t the high price at time t, $DIFF$: $EMA(12)_t - EMA(26)_t$, EMA exponential moving average, $EMA(k)_t$: $EMA(k)_{t-1} + \alpha \times (C_t - EMA(k)_{t-1})$, α smoothing factor: $2/1 + k$, k is time period of k day exponential moving average, LL_t and HH_t mean lowest low and highest high in the last t days, respectively, $M_t : H_t + L_t + C_t/3;\ SM_t : (\sum_{i=1}^{n} M_{t-i+1})/n,\ D_t : (\sum_{i=1}^{n} \mid M_{t-i+1} - SM_t \mid)/n, Up_t$ means the upward price change, Dw_t means the downward price change at time t.

Figure 3: Technical indicators as provided by [25]

Modeling: Provided a training sample, the Machine Learning Modeler begins predicting price movement. Just as a hedge fund will consistently re-train its system as new data arose, the algorithm implemented for this study tests a single day, then incorporates its prediction and results back into the model. This process repeats daily so all price information is considered. Implemented with Sci-kit Learn, the modeler tests Logistic Regression, Support Vector Machines, Random Forest, K-Nearest Neighbors, and Gaussian Naive Bayes classifiers. Each model is trained and tested on the same range of data[17], with performance metrics logged ad-hoc.

All models are tested from 2008 to 2018 (2520 days) to align findings with those of [38] and [25]. They are also recursive. With the initial training dataset, the platform will train the model but only predict the first record. The results if this execution are then fed back into the training dataset, and the modeler sends all of the stock data, now with added predictions, to an Automated Trader object.

Trading: The Automated Trader determines buy, sell, and hold actions for a security based on a combination of momentum prediction and liquidity. If the trader is in a state of liquidity[18] and receives an upward prediction, it will issue a buy command and purchase as much of the security in question as possible[19]. Now that the trader is not in a state of liquidity, any further buy orders will be converted to holds until a sell order is executed. The same concept applies to repeated sell predictions.

After buy, sell, and hold signals are assigned to each day, portfolio returns are calculated across the whole period. Transaction costs of $4.95 are applied to all trades, aligning with the costs of Fidelity Brokerage, and factored in to total returns. Excess cash is invested at the current risk free rate, which has been discounted to daily returns. These two sums – excess cash and equity assets – sum to the total portfolio value and determine portfolio return.

Portfolio return R_p is employed to calculate several ancillary metrics, shown in Table 2. R_m represents the return of the market benchmark, and R_f is the risk free return of U.S. Treasury Bills. Regarding Table 2, σ_p denotes portfolio volatility while σ_d denotes downside deviation[20].

Table 2: Portfolio Performance Metrics

Metric	Formula
Alpha (α)	$R_p - [R_f + (R_m - R_F)\beta]$
Beta (β)	$\frac{Cov(R_p,R_m)}{Var(R_m)}$
Volatility (σ)	$\sqrt{\frac{\sum_{i=1}^{N} \lvert R_i - \bar{R_p}\rvert^2}{N}}$
Sharpe	$\frac{\bar{R_p} - R_f}{\sigma_p}$
Sortino	$\frac{\bar{R_p} - R_f}{\sigma_d}$

6.3. *Implementing Stock Price Momentum Classifiers*

Five supervised learning models were chosen for predicting stock price momentum, and are outlined in the sections below. These were all chosen for their precedence in being applied to generate above-market returns through price movement predictions[21]. Meta-parameters for each classifier are outlined in Table 3.

Logistic Regression (LOG): One of the most commonly implemented binary classification analyses [23], Logistic Regression is a discriminant Bayesian model that approaches binary classification through direct calculation of Bayesian posterior $P(y|x)$ of the joint probability distribution $P(x, y)$ [36]. Logistic regressions assume little correlation (if any) exists between predictors in the feature space, as well as that few outliers and minimal skew are present. Despite this, many studies still implement Logistic Regression to predict price movements, and purport to have found success [15]. Prevailing research also finds discriminative calculation – even without computational or data quality considerations – to be generally preferred over generative models [36]. The empirical findings of this model will be compared with Gaussian Naive Bayes predictors to test this hypothesis.

[17] For all index experiments, 30 years of data was passed to the modeler with a request to test on the last 10 years.

[18] Traders always begin liquid with $100,000 of principal.

[19] It is assumed that the trader is unable to purchase fractional shares of a stock.

[20] This is the standard devation of all returns with below a minimum acceptable return (MAR) of the risk free return R_f for that day.

[21] All classifiers have been used to predict stock prices in literature, though some make use of non-price data.

Table 3: Machine Learning Parameters by Model

	Parameters						
Model Name	Kernel	Penalty	C	γ (rbf)	*degree* (d) ($poly$)	Trees (#)	K (#)
SVM	rbf, poly		0.5, 1, 5	auto, 1, 4	1, 2, 3		
LOG		l1, l2	0.01, 1, 5, 10, 50, 100				
RF						20, 40 ... 100	
KNN							20, 40 ... 140

Logistic regressions are tuned in this study by manipulating the regularization function and the regularization strength C.

$$L1 = argmin_w \sum_{i=1}^{n} \left[y_i - \sum_{j=0}^{m} w_j x_{ij} \right]^2 + \lambda \sum_{j=0}^{m} |w_j| \tag{13}$$

$$L2 = argmin_w \sum_{i=1}^{n} \left[y_i - \sum_{j=0}^{m} w_j x_{ij} \right]^2 + \lambda \sum_{j=0}^{m} w_j^2 \tag{14}$$

In machine learning, $L1$ loss is also known as Least Absolute Deviations (LAD) and included as part of regularization functions that classifiers seek to minimize. $L2$ is similar to $L1$ loss with the exception that it seeks to minimize the Least Squares Error (LSE).

Both of these regularization functions are categorized for weights w, output label y, and prediction x in equations 13 and 14. C, another parameter tuned for Logistic Regression, is a term for inverse regularization strength $\left(\frac{1}{\lambda}\right)$. This characterizes how harshly a model's complexity should be penalized during training. Classifiers will add terms to better fit to a dataset, but too much tuning causes over-fitting in the model and prevents it from generalizing trends to a population.

Support Vector Machine (SVM): Vapnik (1995) [10] first proposed Support Vector Machines (SVM) for pattern recognition by categorizing the problem as fitting an optimal separating hyperplane in $\mathbb{R}^n$ feature space, where n is the number of features. By treating observations as *support vectors*, Vapnik characterizes the classifier as a Lagrangian optimization of the form:

$$y(x) = sign\left[\sum_{i=1}^{N} \alpha_k y_k \Psi(\vec{x}, \vec{x}_k) + b \right] \tag{15}$$

"for a training set of N data points $\{y_k, \vec{x}_k\}_{k=1}^{N}$ where $\vec{x}_k \in \mathbb{R}^n$ is the k^{th} input pattern and $y_k \in \mathbb{R}$ is the k^{th} output pattern." [47]. α_k are positive, real constants, as is b. Notably, SVM makes use of kernel function Ψ, which transforms data for optimal for hyperplane separation. Since Vapnik discovered Support Vector Machine's efficacy at pattern recognition, variant classifiers have become common due to the flexibility afforded by kernel functions. For example, Suykens (1999) [47] determined a popular method for implementing SVM with least squares optimization. Kernel research continues to derive new applications for different disciplines.

Indeed, Support Vector Machines implement a variety of different kernels depending on the application. This study examines two of the most popular kernels, the radial basis function (rbf) and the polynomial (poly) kernel.

$$\textbf{rbf}: \quad \Psi(\vec{x}, \vec{x}_k) = e^{-\gamma ||\vec{x} - \vec{x}_k||^2}$$
$$\textbf{poly}: \quad \Psi(\vec{x}, \vec{x}_k) = (\vec{x}^T \vec{x}_k + r)^d$$

For RBF functions, γ is a free parameter that tunes the significance of a single training sample [39]. Additionally, C trades off misclassification of training examples against simplicity of the decision surface (not shown in equations). On the other hand, d denotes the degree of the polynomial and r is a constant that trades off the weighting of higher and lower order terms in the polynomial [44]. r is not manipulated and left as the default of zero. Other kernels exist, but RBF and POLY were implemented by [38] and [25]. Since these studies each examined foreign[22] financial markets, we wish to compare our performance in American markets while controlling for as many extraneous factors as possible.

Random Forests (RF): Breiman (2001) [4] defines Random Forests as "a combination of tree predictors such that each tree depends on the values of a random vector." An ensemble algorithm, Random Forest classifiers derive their efficiency from a sufficiently large number of decision trees. Each decision tree in a group may suffer from high generalization error and overfitting. However, if taken together as a random forest voting system, the ensemble has been proven to produce a "limiting ... generalization error" [4].

Gaussian Naive Bayes (GNB): As opposed to the discriminant Bayesian implementation employed by Logistic Regression [36], Gaussian Naive Bayes classification empirically generates[23] the joint probability distribution $P(x, y)$ and then uses Bayes Theorem to calculate $P(y|x)$. Favored for their simple and efficient implementation, Bayesian models consistently perform well across a variety of applications [52]. Outlined by [28], Gaussian Naive Bayesian models assume features are independent and model a normal distribution.

[22] Foreign markets are defined as non-U.S. markets.

[23] Distribution created with sample mean $\hat{\mu}$ and variance $\hat{\sigma}^2$.

$$f(x \mid \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{16}$$

By using an empirically discovered $\hat{\mu}$ and $\hat{\sigma}$ for each feature, a Gaussian probability distribution can be derived, as shown in equation 16. This information is then be fed into record classification in equation 17, where $P(A|B)$ is equal to the product of the probabilities of each feature B_i belonging to the normal distribution of that feature $N(\mu_{B_i}, \sigma_{B_i})$ derived from training data.

$$P(A|B) = P(A) \prod_{i=1}^{n} P(B_i|A) \tag{17}$$

K-Nearest Neighbors (KNN): The control classifier, K-Nearest Neighbors (KNN) is a rudimentary machine learning classifier. Though highly dependent on the dataset and application, KNN can be an effective means of prediction. A test prediction is generated by finding the K most similar records in the training data, and returning the most common class label in the set. Similarity d of a training record x relative to test record y is often determined using the Euclidean distance (18) function, defined for a dataset with n features as

$$d(x) = \sum_{i=1}^{n} \sqrt{(x_i - y_i)^2} \tag{18}$$

where x_i is the i^{th}feature of the training record. KNN has shown to be an effective algorithm for some classification problems, but is known [24] to struggle in many applications. Since predictions are derived purely from records seen in the training dataset, KNN struggles with economic problems involving sparse feature spaces (e.g. continuous price data).

7. Empirical Findings

7.1. Econometric Findings Indicate Possible Predictability

Test statistics from Augmented Dickey-Fuller tests unanimously reject the null hypothesis of a unit root in favor of stationarity. The Akaike Information Criteria assigned between 25 and 28 lags for each test, and p-values were well below 1% significance. If the maximum lag is allowed to double from what was previously determined by [43], all test statistics translate to higher values, but even then the majority of tests still rest below a 1% significance value. These results, shown in Figure 4, present compelling evidence that price movements for 100 stocks from 1998 to 2008 are not entirely random.

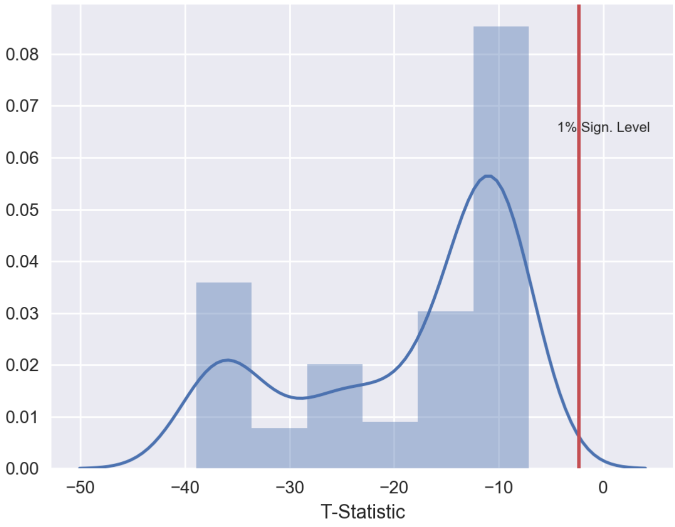


Figure 4: Relative frequency histogram of T-statistics for Augmented Dickey-Fuller tests of 100 randomly selected S&P 500 securities over 10 years.

With that said, maximum lag levels are difficult to set appropriately. Adding lags until statistic significance no longer exists may not necessarily be an appropriate approach [14]. It also can be conducted in different ways, including the Akaike and Bayesian Information Criteria. Regardless, setting lags is not the purpose of this analysis, and it will be assumed that the recommendations of previous literature [43] are sound in this application.

To better gauge the validity of Augmented Dickey Fuller findings, Variance Ratio tests for the same securities were conducted over the same time period. VR test results depict a similar – though less stark – verification of ADF findings. Shown in Figure 5, 93% and 70% of p-values rest below significance levels of 5% and 1%, respectively. Repeated testing with different securities did not yield a unanimous verdict on the Random Walk Hypothesis, but does reinforce the notion that some level of price predictability existed in the past 20 years.

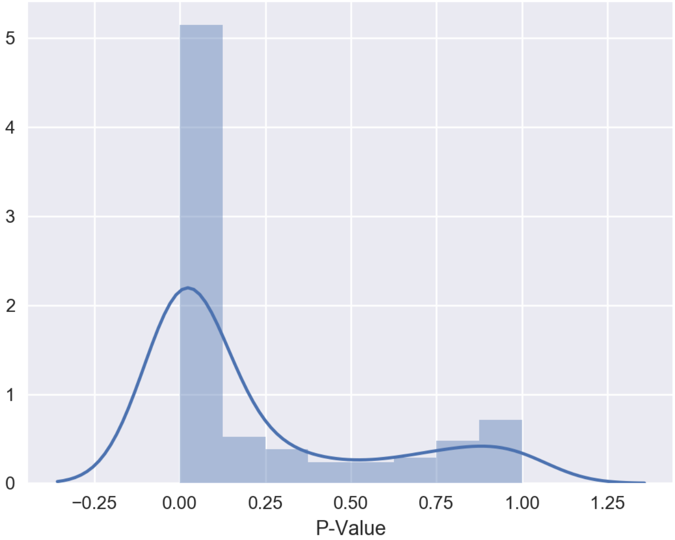


Figure 5: Relative frequency histogram of P-values for Lo-Mackinlay Variance Ratio tests of 100 randomly selected S&P 500 securities over 10 years.

Unfortunately, Augmented Dickey-Fuller and Variance Ratio tests do nothing more than identify stationarity in general. No deductions can be made about the characterization or duration of trending in financial markets. Conclusions drawn by [30] articulate that a smaller variance ratio than expected can be seen as evidence of mean reversion[24]. Their deduction may indeed be correct, but variance ratios can not be definitively categorized as evidence for mean-reversion without further information. Rejecting a null hypothesis is not equivalent to accepting an alternative. It simply proffers the opportunity for further analysis.

Indeed, the following sections expound on the trends witnessed when machine learning algorithms are applied to predict price movements and inform trading decisions. Though these experiments do not touch on mean reversion directly, they test the ability of supervised learning algorithms to discern price movement trends.

7.2. Price Predictions Align with Random Chance

Across all machine learning models tested[25], prediction accuracy did not statistically deviate from approximately random chance. Shown in the horizontal histogram of Figure 6, predictive accuracy observations were approximately normally distributed about a mean of 52% with a slight left skew. Average accuracy may rest above 50% because upward momentum is slightly more likely long term, though this hypothesis is unverified. Trading performance (Alpha) above the risk free return R_f possessed a similar distribution centered on returns of zero.

Empirical distributions found for accuracy and alpha both align with the weak form Efficient Market Hypothesis. The

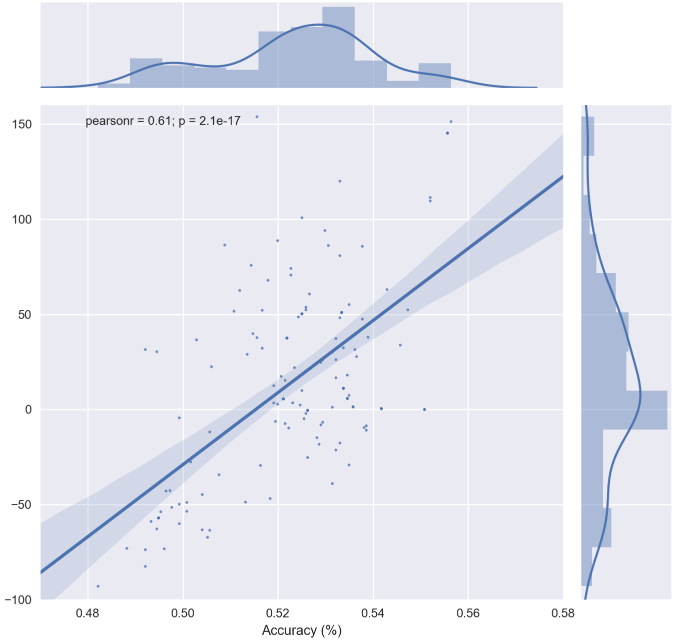


Figure 6: Trend specific trading returns (Alpha) for all models, mapped as a function of accuracy. Alpha is represented as percent return over 10 years. Hue represents a 95% confidence interval.

active return gained on an investment, Alpha represents the returns acquired above and beyond the market benchmark. Weak form market efficiency implies that beating the market via the exploitation of price trends is impossible. Indeed, with a plurality of models only able to achieve the market return or worse, trading results thus far seem to strongly affirm efficient market theories.

Notably, accuracy has a fairly strong, positive relationship with Alpha. It is expected that an algorithm effective at predicting price movement would generate a higher return. Even so, witnessing this trend empirically[26] verifies the notion that algorithmic trading platforms may be capable of success. The steep slope of returns against accuracy implies that a model with even 65% accuracy over a long period of time could significantly outperform a market benchmark with our strategies[27].

The positive relationship between Accuracy and Alpha was derived from less than 100 independent measurements. Due to the structure of the algorithmic trading model, all tests aside from RF[28] are static in their performance[29]. While the relationship between Alpha and Accuracy–as well as performance overall–aligns with efficient market theories and previous finan-

[24] Intuitively, this means that variance does not continue to expand over time, and may be drawn back to a central or trending point.

[25] Includes all of the parameters tested for each model, trending, and non-trending data

[26] Though accuracy rarely exceeds 55% over ten years, abnormal returns seem highly possible if a model were able to sustain an accuracy of 65% or greater.

[27] However, it is important to note that these strategies do take on significantly more risk. Risk should always benchmark returns, as discussed later in the article.

[28] As a stochastic ensemble of decision trees, Random Forest performance naturally varies even when parameters are not changed.

[29] Generally, machine learning algorithms have a stochastic element to performance as the result of varied training data. It is illogical to create a trading dataset from random days of market activity, so results in this study are not stochastic.

cial research, more testing is necessary to ascertain if our assertions are sound.

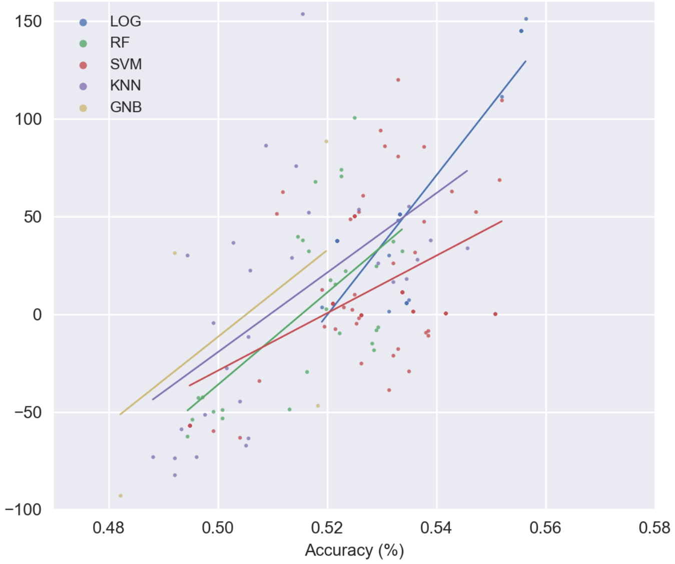


Figure 7: Trend specific trading returns (Alpha) by model, mapped as a function of accuracy. Alpha is represented as percent return over 10 years.

Accuracy overall appeared to revert to a mean of 50%, but Figure 7 indicates some models consistently outperform others. Denoted by a specific color, each model's accuracy and Alpha performance data was regressed. Across all four stock indices, no notable differences can be ascertained for the majority of models. However, the Logistic regression's average performance exceeded its peer algorithms in both accuracy and portfolio returns.

The cause of Logistic Regression's abnormal returns is unknown, and may be a manifestation of random chance. Yet, when utilized to generate predictions for trading, it outperformed or matched the performance of the market for all four stock indices. Furthermore, it was the best performing algorithm in over 75% of trade simulations run[30]. These findings also mirror the findings of [36], who asserted the superiority of discriminative models for many classification problems.

Consistently outperforming its market benchmark, Logistic Regression may have identified subtle market trends. Despite its inability to predict price movements with more than 56% accuracy, Logistic Regression above-market returns skyrocketed to a maximum of 150% over ten years. However, the validity and consistency with which Logistic Regression can achieve abnormal returns is questionable.

One interpretation of Logistic Regression trading is that it discerns strong price movements and correctly trades to exploit them. Such a strategy could easily gain abnormal returns without accurately predicting a majority of price movements. Another equally valid perspective is that the returns witnessed were random and anomalous, implying Logistic Regression is incapable of consistently outperforming the market if applied to other indices and securities. Independent testing, conducted and articulated in the following sections, supports the latter.

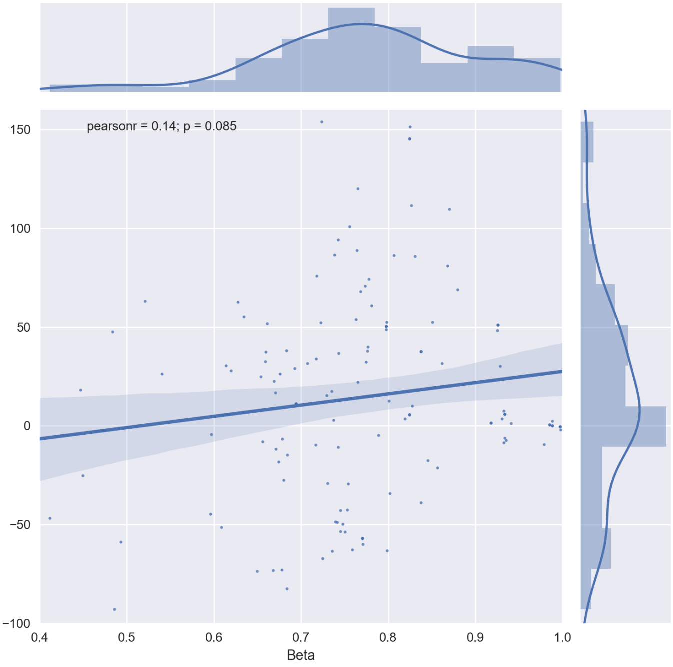


Figure 8: Trend specific trading returns (Alpha) for all models, mapped as a function of Beta. Alpha is represented as percent return over 10 years. Hue represents a 95% confidence interval.

Trading performance bore little relationship with the Beta of the trading strategy (Figure 8). The empirical findings of [22] also find Beta to be uncorrelated with overall returns, though their analysis utilizes Beta as a proxy for the appeal and expected returns of a company's stock. A slight positive trend between Beta and Alpha is observed, but the lack of correlation increases the likelihood that the trend is the product of randomness. We maintain that any trend between Beta and Alpha would dissipate as additional experimental observations are added.

Additionally, Beta coefficients for all trading models are never negative and never exceed 1. Contextually, this means that changes in portfolio return are constrained to either mirror the movements of the market or move unrelated to it entirely. For many fund managers, a relatively low Beta is desired as it implies returns can be gained regardless of market volatility[31]. Positive Beta values also indicate trading algorithms generally do not move against the market benchmark. Both of these constraints in Beta are largely due to the structure of the algorithmic trading system.

Short selling is not programmed into the trading systems of this study. Therefore, the only states of a portfolio are vested in the market or liquid and receiving risk free returns. Even if the trading algorithm invested and sold at inopportune times, losses

[30] Simulations here refers to average performance across all parameters for each model, and represents composite performance as opposed to a single trading run.

[31] A low Beta is appealing only if it accompanies results that are as good or better than market returns.

would not achieve a Beta lower than zero. Restrictions on Beta values above one are also attributed to the same design patterns.

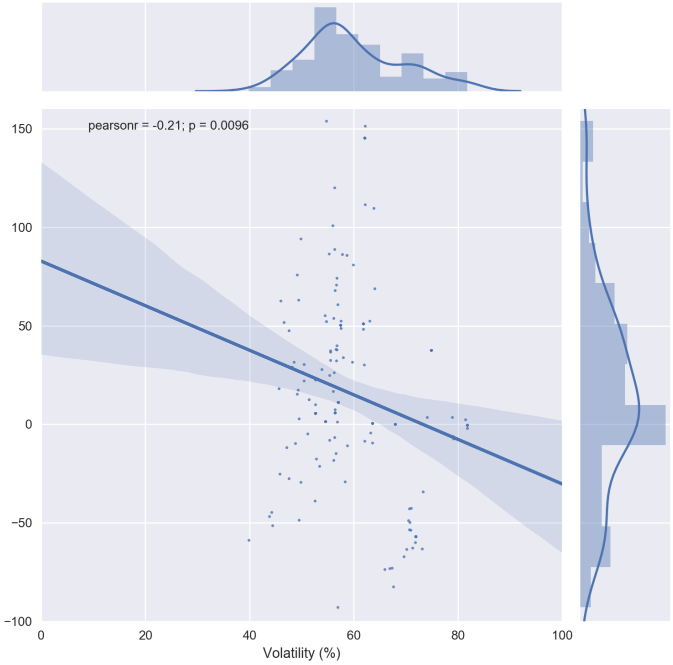


Figure 9: Trend specific trading returns (Alpha) for all models, mapped as a function of Volatility. Alpha is represented as percent return over 10 years. Hue represents a 95% confidence interval.

Little information on the relationship between Alpha and Volatility can be gleaned from this study as little correlation exists. Similar to the trending seen in Figure 8, Figure 9 illustrates a slight negative relationship between Volatility and Alpha. High variation in data causes any conclusions on this relationship to be dubious. A collection of experiments separated from general findings[32] all possess consistently poor performance and high volatility. If these outliers are removed, the negative relationship disappears. Additional experimentation is necessary to distinguish a Volatility-Alpha relationship for this algorithmic trading system.

7.3. Trading Systems Generally Unable Outperform Market

Aggregate trends outlined in the previous section are visualized in Figures 10 - 17 and grouped by index. Each index outlines trading performance for all models using either trending or non-trending data. The effects of non-trending data on performance are left for a later section, and focus here will be placed exclusively on trending data (Figures 10, 12, 14, and 16). Trading platform structure also induces trading performance trends to mirror the benchmark, and was expected. An algorithm that correctly differentiates between up and down price movements over time would mirror market trends while simultaneously exceeding market performance.

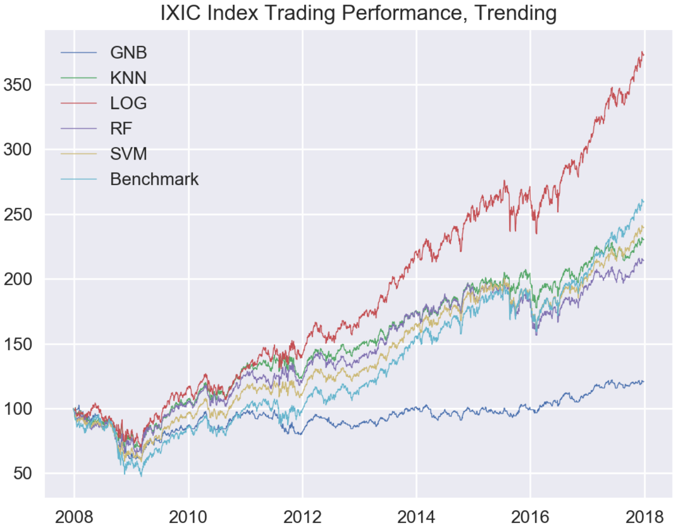


Figure 10: IXIC 10 year trading performance for non-trending algorithms by model. Returns represented in USD - thousands. Benchmark represents buy-and-hold strategy.

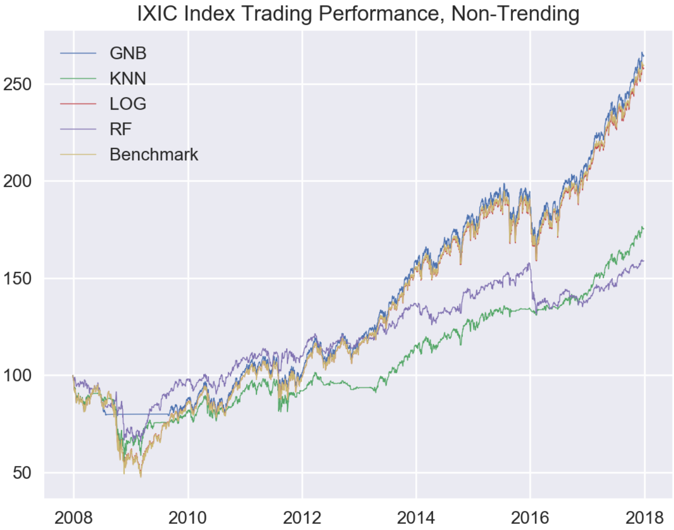


Figure 11: IXIC 10 year trading performance for non-trending algorithms by model. Returns represented in USD - thousands. Benchmark represents buy-and-hold strategy.

The performance of Logistic Regression in Figure 10 illustrates this point. Portfolio returns exceed the market at an increasing rate, and price shifts in the benchmark more profoundly affect Logistic Regression volatility. Other models do not exhibit these trends. Instead, transaction costs and a general inability to time market entry and exit progressively reduce principal value.

Logistic Regression completed an average of 360[33] trades over a ten year period, well under an average of 720 for RF and KNN, 600 for SVM, and 465 for GNB. Total transaction

[32] Observations are shown in the bottom right corner of Figure 9.

[33] These are approximate, average trade numbers across multiple tests. See Appendices A - D for details.

costs were much lower for Logistic Regression as a result, but likely not greatly influential on performance.

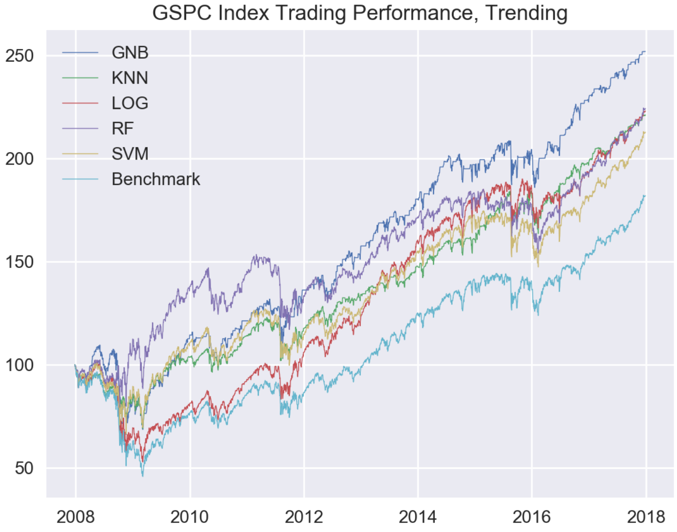


Figure 12: GSPC 10 year trading performance for trending algorithms by model. Returns represented in USD - thousands. Benchmark represents buy-and-hold strategy.

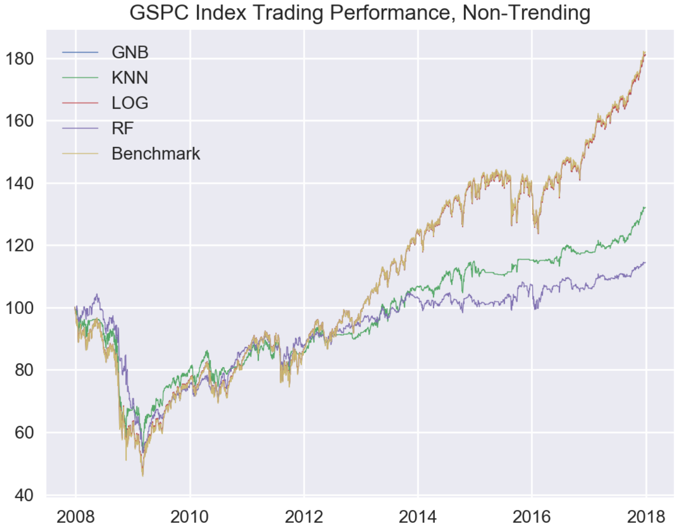


Figure 13: GSPC 10 year trading performance for non-trending algorithms by model. Returns represented in USD - thousands. Benchmark represents buy-and-hold strategy.

Conversely, LOG consistently garnered the highest accuracy of all models at 55.56%, which greatly altered its 10-year trading performance. Trade log review indicates a tendency for LOG to execute trades shortly before periods of high volatility. The opposite can be said for trade results of Gaussian Naive Bayes classifiers.

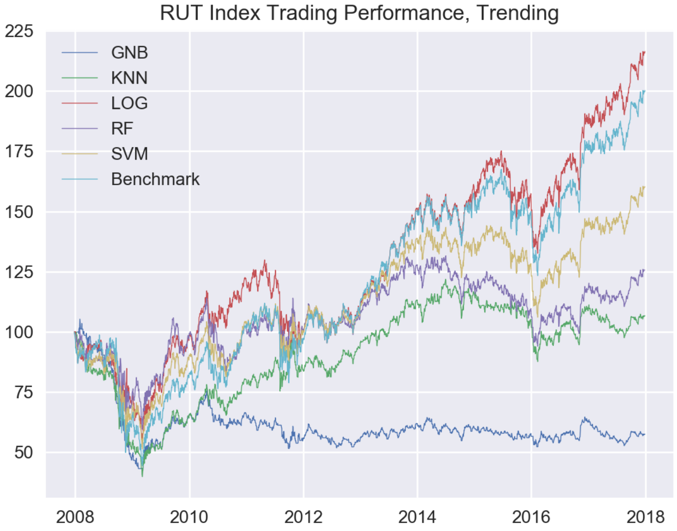


Figure 14: RUT 10 year trading performance for trending algorithms by model. Returns represented in USD - thousands. Benchmark represents buy-and-hold strategy.

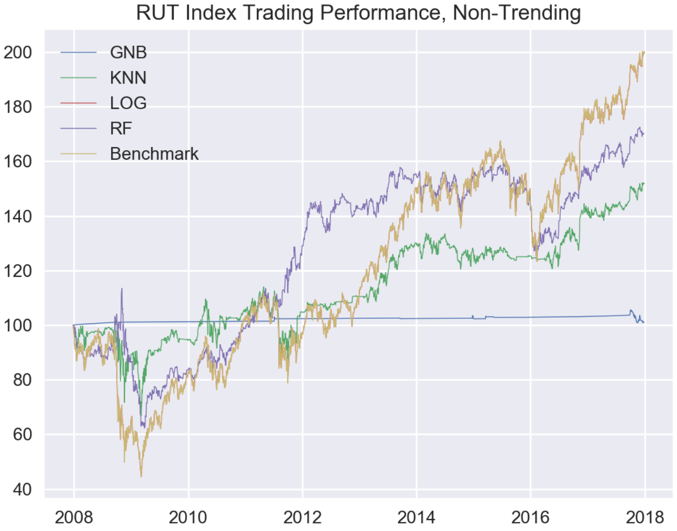


Figure 15: RUT 10 year trading performance for non-trending algorithms by model. Returns represented in USD - thousands. Benchmark represents buy-and-hold strategy.

Though most similar in statistical implementation and number of trades to LOG, GNB traded the worst out of any model for IXIC. Trades were often made sporadically, showing no relationship to price movement whatsoever. RF, KNN, and SVM models achieved near-market success. Holding periods were much shorter for these securities on average[34]. However, upon closer examination LOG and GNB often alternated frequent, near daily trades with long holding periods[35]. Other models tended to trade at more consistent intervals.

[34] Holding periods for KNN, SVM, and RF were approximately double LOG for all tests.

[35] An average of roughly two and a half weeks.

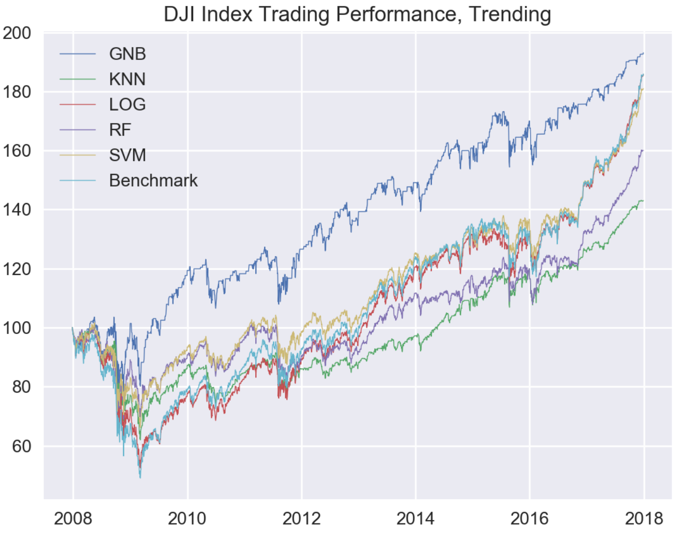


Figure 16: DJI 10 year trading performance for trending algorithms by model. Returns represented in USD - thousands. Benchmark represents buy-and-hold strategy.

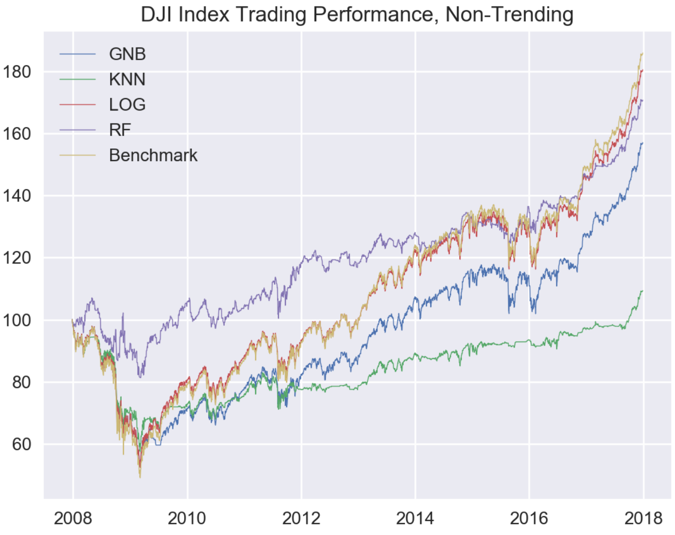


Figure 17: IXIC 10 year trading performance for non-trending algorithms by model. Returns represented in USD - thousands. Benchmark represents buy-and-hold strategy.

The cadence and number of trades remained consistent for each model as they were applied to different stock indices[36]. Surprisingly, Gaussian Naive Bayes classifiers garnered the highest return for GSPC and DJI but the lowest return for IXIC and RUT. Trading frequency and cadence is strikingly similar across all four indices. Trade logs found that in general, the distinction in performance was attributable to GNB trading IXIC and RUT into a losing position shortly before large price movements and holding the position until volatility dropped. By contrast, GNB consistently traded into and held winning positions in DJI and GSPC.

The origin of this simple but influential discrepancy in performance for GNB is not immediately clear. Variation in the consistently of GNB performance affirms the notion that these results are the product of random trading decisions. Generally, all models do not notably exhibit rules with which they use to trade, suggesting they have no predictive power in determining price changes.

Even if price momentum algorithms are picking up on trends, the volatility of the most successful models was significantly higher than the market benchmark. Logistic Regression's volatility over ten years was 65% on average, while RF and KNN maintained volatility of 55% and GNB 50%. Additional risk in finance correlates to additional reward [21]. Even if it could be proven that model predictions were identifying trends in the market, above-market returns can only refute efficient market theories if risk undertaken by the agent can be controlled for and minimized.

Hedging with the strategies utilized by this study may propose a more convincing argument against weak-form EMH. Assuming that a model can be found that consistently outperforms the market, one could undertake small counter-positions in a separate index to assuage the risk of the portfolio while maintaining above-market returns. Logistic Regression appears to be the only model that may be able to consistently receive excess returns, and is more thoroughly tested on 100 randomly selected S&P500 securities in the next section.

7.4. Independent Verification Implies Performance Random

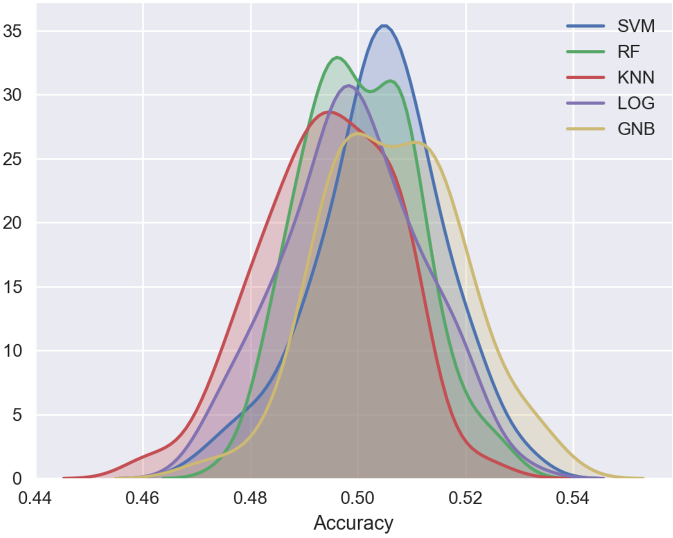


Figure 18: Stock movement accuracy histogram of 100 randomly selected S&P500 securities over 10 years by model.

Logistic Regression is applied to test the same 100 randomly selected S&P500 stocks utilized for econometric research of 20 years and tuned using optimal parameters found for all four stock indices [37]. However, to verify general performance accuracy on these securities, all models generate predictions. Each

[36] Logistic regression and Gaussian Naive Bayes were the only models that consistently varied trade cadence.

[37] Best parameters were chosen by averaging the alpha of all LOG models across four indices.

model's parameters are also chosen by finding the best average Alpha across all indices. These results are stored, but ultimately only Logistic Regression predictions were used for trading[38].

Accuracy distributions for all stocks surprisingly find Logistic Regression to be among the least accurate models, while SVM and GNB performed the best. Performance differences between models were minute. It is difficult to ascertain if accuracy distributions would converge on a mean of 50% if additional stocks were tested. We maintain it would. The variation in Figure 18 does not appear significant enough to have arisen from anything besides random chance[39].

Independent findings in Figure 18 diverge from the expected performance of Logistic Regression. Regardless of if accuracy converges about a normal distribution centered at 50%, it underscores the possibility that Logistic Regression's performance with stock indices can not be replicated. Yet, perhaps Logistic Regression has the ability to consistently detect and trade appropriately in moments before high volatility, as was the case for index performance.

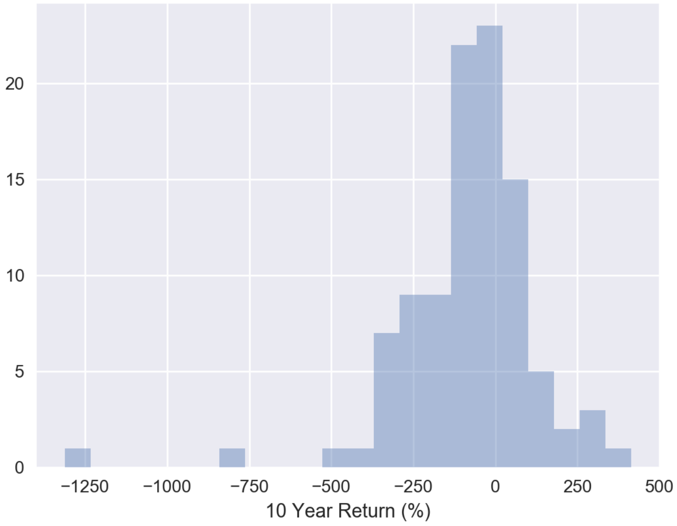


Figure 19: Trading performance (Alpha) histogram of 100 randomly selected S&P500 securities over 10 years for all models.

Trading performance, outlined in Figure 19, refutes such a claim. Across all 100 securities, Alpha distributed around returns of 0% with a slight left skew towards negative returns. The influence of transaction costs may generate left skew, but likely not to the extent seen in Figure 19. Returns depict strong evidence that Logistic Regression does not have a consistent means of predicting price movements or achieving above-market returns.

Considered together, the experiments of this study starkly contrast the work of [38] and [11] in Indian markets, [25] in Istanbul, and [13] in Brazil. With that said, all three of these markets host a small fraction of the trading volume and total market capitalization of United States markets. Therefore, we assert several hypotheses that may explain discrepancies between existing literature and our analysis.

First, perhaps the models utilized in this study were not sophisticated enough to garner above market returns. This potentiality is highly unlikely, as all implementation methods mirror those of [38]. One could argue that the time-frame analyzed was simply a more efficient time-period for markets, but this is ignored as it asserts market efficiency recently arose[40]. Conversely, the United States financial markets are more efficient than most foreign markets.

The majority of machine learning studies that only utilize price data were conducted in developing or developed foreign markets much smaller than those of the United States. This factor could indeed be confounding findings, and should be further assessed by completing weak-form analysis in other large developed markets[41]. Third, previously published studies may have selectively disclosed data from their analysis such that it aligns with their claims. Infidelity is of particular concern for [38], who tests only four Indian securities with no independent verification. Nevertheless, the volume of studies with similar weak-form findings implies that results are at least partially valid for certain markets.

7.5. Trend-specific Data Superior for Algorithm Training

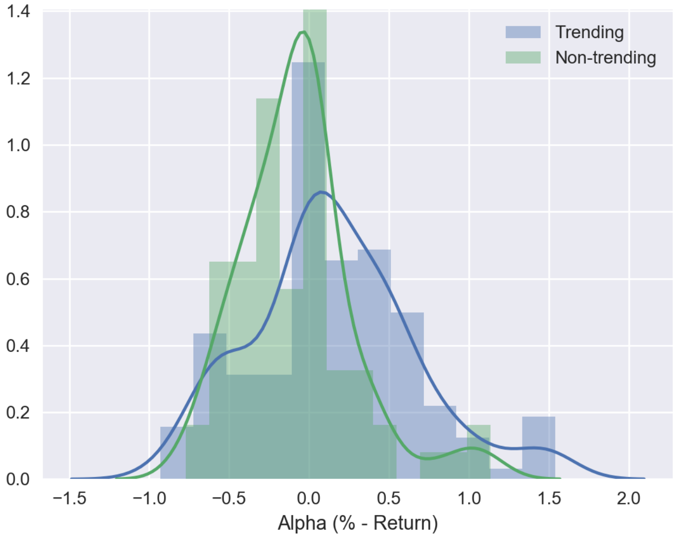


Figure 20: Relative frequency histogram of trading returns (Alpha) for all trending and non-trending models. Alpha is given as percent return over 10 years.

Trending and non-trending performance in this study mirrored the findings of [38, 25, 13]. In Figures 11,13,15, and 17, non-trending data proved to be unable to properly train any models. Not a single algorithm consistently outperformed the market.[42], and many did not trade more than once. As shown,

[38] An excellent avenue for further research would be to extend trading simulations to other models.

[39] Hypothesis testing is required to verify this claim.

[40] The probability of this is extremely low, and would refute decades of replicated, controlled experiments.

[41] China, Japan, and Europe are all excellent candidates in this regard.

[42] This does not simply apply to averages. Every trading experiment aside from two GNB runs with non-trending data underperformed relative to the market.

the majority of algorithms appear to mirror the market exactly, taking and maintaining a buy-and-hold position. Closer inspection reveals that many models only outputted one trading decision over the entire ten years. Some, like GNB trading for RUT, spent nearly 10 years without trading at all.

With continuous ratio data, the intuition behind the metric may be difficult for the computer to identify pro-grammatically. In fact, it is our assumption that the abnormal trading (or lack thereof) of models can be attributable to an inability to properly tune to a large, sparse feature space. Some algorithms like RF overcame this sparsity more easily, likely because of its stochastic, ensemble based prediction methods. However, more analytical models like Logistic Regression were rendered useless by continuous data.

Additionally, leaving technical ratios in their raw form increased model training and testing time substantially. In backtesting, Random Forest regressions took four times as long to complete with non-trending data. Non-trending Support Vector Machine tests were unable to be collected, as a single experiment took eight days to complete on our servers. Particularly in investing, where timing is essential to success for any active fund manager, algorithms must be able to operate efficiently. If there is any potential for above-market returns by leveraging technical data or otherwise, distilling ratios and indicators into the message[43] they convey is the most effective method for preprocessing we have found thus far. Even so, we remain skeptical of methods that purport to generate above market returns by timing the market.

8. Discussion

The objective of this analysis was to jointly consider econometric and machine learning research as they pertain to weak-form Efficient Market theory. Economic and econometric research has transitioned from generally supporting of the validity of efficient markets to hotly debating its merits and claims. By contrast, machine learning research applied to predicting asset prices consistently finds success predicting market trends and garnering abnormal returns. Even today, when all EMH theories are debated in economics and computer science alike, the consistency with which publications achieve above-market returns is surprising.

Our study merges these fields of research in an effort to better understand the discrepancies in empirical findings. Econometric research is first conducted to determine if stock price movements appear random. Augmented Dickey-Fuller and Variance Ratio tests both seek to identify stationarity in data. Historically, ADF test results are consistently observed to support efficient market claims while VR tests more commonly refute them. Both of these tests are run on 100 randomly selected S&P500 securities. ADF test results run counter to traditional findings, as every test statistic rested below a 1% significance level. All hypothesis tests rejected the null in favor of stationarity. VR tests were more mixed, but still strongly implied stationarity with 70% of p-values below a 1% significance level.

Establishing the possibility for market trending and therefore predictive potential, the econometric findings were then supplemented by machine learning implementations. With a scalable, dynamic system, daily stock price moment predictions were created in a training-testing loop that refreshed daily. Predictions collected in this manner informed a trading system that made buy, sell, and hold decisions accordingly. Transaction costs as well as portfolio performance metrics were recorded.

Four stock indices were tested using the aforementioned framework. A highly diversified asset, stock indices are a composite of hundreds (or thousands) of stocks, mitigating many of the risks associated with bankruptcy or company failure. Since the trading system implemented for experimentation only trades one security at a time, the inherent diversification of stock indices makes them particularly suitable for controlled experimentation. Though verified with several individual S&P500 company stocks, empirical findings for stock indices are particularly insightful as they most directly align with the notion of generating above-market returns while holding risk constant.

Econometric findings did not align with machine learning prediction in American markets. No algorithm was able to garner predictive accuracy significantly above 50%, and while Logistic Regression garnered above-market returns for all stock indices[44], independent verification with 100 randomly selected S&P500 stocks was inconclusive. We maintain that the price-movement accuracy and trading performance indicate models trained on only price data do not have the predictive power necessary to generate abnormal returns.

With that said, our findings do not suggest that stock prices are random or represent all available information. The former claim was refuted by the results of our Augmented Dickey-Fuller and Variance Ratio tests, while the latter was not considered at all in this study. Random Walk theory and Efficient Market hypotheses are subtly distinct from one another, and findings for each must be carefully qualified. Regardless, the aggregate conclusions drawn from the results of our analysis compellingly affirm the validity of the weak-form Efficient Market Hypothesis.

Machine learning as a discipline is still evolving at what appears to be an accelerating rate. The capabilities of pattern-recognition software in the future are potentially limitless[45]. Therefore, research investigating both Random Walk and Efficient Market theories must continue. Moreover, this analysis exclusively examined weak-form efficient markets. The advent of sentiment analysis, natural language processing, and ancillary data that may inform price-movement occurrences should also be considered when testing semi-strong and strong form efficient markets. Fortunately, the insatiable urge for researchers and investors alike to increase the returns of their assets will further Efficient Market and Random Walk research in perpetuity.

[43] In this case, messages took the form of either a buy or sell signal.

[44] The final results, as discussed in this paragraph, solely refer to tests conducted on trend-specific data.

[45] With that said, even the best pattern-recognition technology fails when no patterns exist.

9. References

[1] Bachelier, L. *Théorie de la spéculation*. Gauthier-Villars, 1900.

[2] Banz, R. W. The relationship between return and market value of common stocks. *Journal of financial economics 9*, 1 (1981), 3–18.

[3] Borges, M. R. Efficient market hypothesis in european stock markets. *The European Journal of Finance 16*, 7 (2010), 711–726.

[4] Breiman, L. Random forests. *Machine Learning 45*, 1 (Oct 2001), 5–32.

[5] Breitung, J. The local power of some unit root tests for panel data. In *Nonstationary panels, panel cointegration, and dynamic panels*. Emerald Group Publishing Limited, 2001, pp. 161–177.

[6] Brogaard, J., Hendershott, T., and Riordan, R. High-frequency trading and price discovery. *The Review of Financial Studies 27*, 8 (2014), 2267–2306.

[7] Buguk, C., and Brorsen, B. W. Testing weak-form market efficiency: Evidence from the istanbul stock exchange. *International review of financial analysis 12*, 5 (2003), 579–590.

[8] Charles, A., and Darné, O. Variance-ratio tests of random walk: an overview. *Journal of Economic Surveys 23*, 3 (2009), 503–527.

[9] Cheung, Y.-W., and Lai, K. S. Lag order and critical values of the augmented dickey–fuller test. *Journal of Business & Economic Statistics 13*, 3 (1995), 277–280.

[10] Cortes, C., and Vapnik, V. Support-vector networks. *Machine Learning 20*, 3 (Sep 1995), 273–297.

[11] Dash, R., and Dash, P. K. A hybrid stock trading framework integrating technical analysis with machine learning techniques. *The Journal of Finance and Data Science 2*, 1 (2016), 42 – 57.

[12] De Bondt, W. F. M., and Thaler, R. H. Anomalies: A mean-reverting walk down wall street. *Journal of Economic Perspectives 3*, 1 (March 1989), 189–202.

[13] de Oliveira, F. A., Nobre, C. N., and Zárate, L. E. Applying artificial neural networks to prediction of stock price and improvement of the directional prediction index–case study of petr4, petrobras, brazil. *Expert Systems with Applications 40*, 18 (2013), 7596–7606.

[14] Dickey, D. A., and Fuller, W. A. Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American statistical association 74*, 366a (1979), 427–431.

[15] Dutta, A., Bandopadhyay, G., and Sengupta, S. Prediction of stock performance in indian stock market using logistic regression. *International Journal of Business and Information 7*, 1 (2015).

[16] Enke, D., and Thawornwong, S. The use of data mining and neural networks for forecasting stock market returns. *Expert Systems with applications 29*, 4 (2005), 927–940.

[17] Fama, E. F. The behavior of stock-market prices. *The Journal of Business 38*, 1 (1965), 34–105.

[18] Fama, E. F. Efficient capital markets: A review of theory and empirical work. *The Journal of Finance 25*, 2 (1970), 383–417.

[19] Fama, E. F. Efficient capital markets: Ii. *The journal of finance 46*, 5 (1991), 1575–1617.

[20] Fama, E. F. Market efficiency, long-term returns, and behavioral finance1the comments of brad barber, david hirshleifer, s.p. kothari, owen lamont, mark mitchell, hersh shefrin, robert shiller, rex sinquefield, richard thaler, theo vermaelen, robert vishny, ivo welch, and a referee have been helpful. kenneth french and jay ritter get special thanks.1. *Journal of Financial Economics 49*, 3 (1998), 283 – 306.

[21] FAMA, E. F., and FRENCH, K. R. The crosssection of expected stock returns. *The Journal of Finance 47*, 2, 427–465.

[22] Fama, E. F., and French, K. R. The capital asset pricing model: Theory and evidence. *Journal of Economic Perspectives 18*, 3 (September 2004), 25–46.

[23] Hosmer Jr, D. W., Lemeshow, S., and Sturdivant, R. X. *Applied logistic regression*, vol. 398. John Wiley & Sons, 2013.

[24] Imandoust, S. B., and Bolandraftar, M. Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background. *International Journal of Engineering Research and Applications 3*, 5 (2013), 605–610.

[25] Kara, Y., Boyacioglu, M. A., and mer Kaan Baykan. Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the istanbul stock exchange. *Expert Systems with Applications 38*, 5 (2011), 5311 – 5319.

[26] Kuo, R., Chen, C., and Hwang, Y. An intelligent stock trading decision support system through integration of genetic algorithm based fuzzy neural network and artificial neural network. *Fuzzy Sets and Systems 118*, 1 (2001), 21 – 45.

[27] Kwiatkowski, D., Phillips, P. C., Schmidt, P., and Shin, Y. Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root? *Journal of econometrics 54*, 1-3 (1992), 159–178.

[28] Lewis, D. D. Naive (bayes) at forty: The independence assumption in information retrieval. In *Machine Learning: ECML-98* (Berlin, Heidelberg, 1998), C. Nédellec and C. Rouveirol, Eds., Springer Berlin Heidelberg, pp. 4–15.

[29] Leybourne, S. J., Mills, T. C., and Newbold, P. Spurious rejections by dickey¢fuller tests in the presence of a break under the null. *Journal of Econometrics 87*, 1 (1998), 191 – 203.

[30] Lo, A. W., and Mackinlay, A. C.

[31] Lo, A. W., and MacKinlay, A. C. Stock market prices do not follow random walks: Evidence from a simple specification test. *The review of financial studies 1*, 1 (1988), 41–66.

[32] Markowitz, H. Portfolio selection*. *The Journal of Finance 7*, 1, 77–91.

[33] Modigliani, F., and Cohn, R. A. Inflation, rational valuation and the market. *Financial Analysts Journal 35*, 2 (1979), 24–44.

[34] Nalkiel, B. G., and Malkiel, B. G. Random walk down wall street: Including a life-cycle guide to personal investing, 1991.

[35] Nasrabadi, N. M. Pattern recognition and machine learning. *Journal of electronic imaging 16*, 4 (2007), 049901.

[36] Ng, A. Y., and Jordan, M. I. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in neural information processing systems* (2002), pp. 841–848.

[37] Nisar, S., and Fcma, M. H. Testing weak form of efficient market hypothesis: Empirical evidence from south-asia. *World Applied Sciences Journal 17*, 4 (Oct 2012), 414–427.

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

[39] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. Scikit-learn: Machine learning in python. *Journal of machine learning research 12*, Oct (2011), 2825–2830.

[40] Phillips, P. C., and Perron, P. Testing for a unit root in time series regression. *Biometrika 75*, 2 (1988), 335–346.

[41] Poshakwale, S. Evidence on weak form efficiency and day of the week effect in the indian stock market. *Finance India 10*, 3 (1996), 605–616.

[42] Schumaker, R. P., and Chen, H. Textual analysis of stock market prediction using breaking financial news: The azfin text system. *ACM Transactions on Information Systems (TOIS) 27*, 2 (2009), 12.

[43] Schwert, G. W. Tests for unit roots: A monte carlo investigation. *Journal of Business & Economic Statistics 7*, 2 (1989).

[44] Shashua, A. Introduction to machine learning: Class notes 67577. *arXiv preprint arXiv:0904.3664* (2009).

[45] Shiller, R. J. From efficient markets theory to behavioral finance. *Journal of economic perspectives 17*, 1 (2003), 83–104.

[46] Summers, L. H. Does the stock market rationally reflect fundamental values? *The Journal of Finance 41*, 3 (1986), 591–601.

[47] Suykens, J. A., and Vandewalle, J. Least squares support vector machine classifiers. *Neural processing letters 9*, 3 (1999), 293–300.

[48] Tversky, A., and Kahneman, D. The framing of decisions and the psychology of choice. *Science 211*, 4481 (1981), 453–458.

[49] Wald, A., and Wolfowitz, J. On a test whether two samples are from the same population. *The Annals of Mathematical Statistics 11*, 2 (1940), 147–162.

[50] Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[51] Worthington, A. C., and Higgs, H. Weak-form market efficiency in european emerging and developed stock markets. discussion paper no. 159.

[52] Zhang, H. The optimality of naive bayes. *AA 1*, 2 (2004), 3.

Appendix A. IXIC Accuracy and Trading Performance

Table A.4: IXIC 10 Year Trading Performance with Non-Trending Data

			Trading Performance								
Model (*Parameters*)	Acc. (%)	No. of Trades	R_f (%)	R_m (%)	R_p (%)	Above Mkt.	σ (%)	β (%)	α (%)	Sharpe (%)	Sortino (%)
Random Forest											
(# *Trees*)			2.878	159.9							
20	48.53	1099			15.87	−144.00	43.66	0.409	12.99	12.99	1.081
40	49.92	1116			98.16	−61.71	42.95	0.395	95.28	95.28	3.868
60	49.64	1043			64.14	−95.73	47.32	0.480	61.26	61.26	2.723
80	49.48	1043			22.97	−136.90	48.31	0.501	20.09	20.09	1.336
100	49.09	1031			56.94	−102.93	45.57	0.445	54.07	54.07	2.522
120	49.17	1033			43.15	−116.72	45.96	0.452	40.27	40.27	2.060
140	50.08	1023			111.20	−48.67	46.93	0.471	108.33	108.33	3.870
K-Nearest Neighbors											
(# *Neighbors*)			2.878	159.9							
20	50.16	382			64.20	−95.67	49.37	0.522	−20.55	61.32	2.517
40	50.16	249			14.35	−145.52	51.14	0.563	−76.97	11.47	0.940
60	51.39	185			71.03	−88.85	54.90	0.645	−33.19	68.15	2.523
80	52.54	133			77.59	−82.28	56.07	0.676	−31.39	74.71	2.575
100	52.54	121			112.22	−47.65	58.16	0.723	−4.19	109.35	3.183
120	52.98	101			110.48	−49.39	59.14	0.753	−10.57	107.60	3.049
140	52.54	97			78.72	−81.16	61.04	0.800	−49.81	75.84	2.409
Logistic Regression											
(*Penalty*, *C*)			2.878	159.9							
All	55.08	1			158.14	−1.73	67.92	0.988	0.13	155.26	3.192
Gaussian Naive Bayes	54.33	21	2.878	159.9	164.64	4.77	50.92	0.552	75.15	161.76	4.048

Table A.5: IXIC 10 Year Trading Performance with Trend Specific Data

			Trading Performance								
Model (*Parameters*)	Acc. (%)	No. of Trades	R_f (%)	R_m (%)	R_p (%)	Above Mkt.	σ (%)	β (%)	α (%)	Sharpe (%)	Sortino (%)
Random Forest											
(# *Trees*)			2.878	159.9							
20	52.90	730			130.27	−29.60	55.32	0.654	24.79	127.40	3.639
40	52.90	722			97.80	−62.07	55.34	0.656	−8.04	94.93	3.042
60	52.82	726			95.50	−64.37	56.52	0.684	−14.83	92.62	2.940
80	52.86	744			90.48	−69.39	56.14	0.674	−18.26	87.61	2.856
100	53.37	718			138.78	−21.09	55.49	0.659	32.48	135.90	3.746
120	52.94	723			102.88	−56.99	56.28	0.679	−6.53	100.00	3.075
140	53.21	739			143.83	−16.05	55.51	0.659	37.41	140.95	3.855
K-Nearest Neighbors											
(# *Neighbors*)			2.878	159.9							
20	53.45	941			91.12	−68.75	45.60	0.447	18.06	88.25	3.306
40	52.94	861			113.96	−45.91	50.23	0.540	26.29	111.09	3.636
60	53.65	721			128.09	−31.79	53.82	0.619	27.96	125.21	3.662
80	53.49	677			157.69	−2.18	54.48	0.634	55.27	154.82	4.137
100	53.21	633			124.92	−34.95	55.99	0.671	16.74	122.04	3.489
120	53.89	631			148.25	−11.62	56.56	0.683	38.08	145.37	3.832
140	54.56	619			149.30	−10.58	57.92	0.717	33.81	146.42	3.702
Support Vector Machine											
(*Kernel*, *C*, γ *or Degree*)			2.878	159.9							
rbf, 0.5, 4	53.37	679			123.12	−36.75	56.95	0.694	11.30	120.24	3.383
rbf, 0.5, 1	53.37	679			123.12	−36.75	56.95	0.694	11.30	120.24	3.383
rbf, 0.5, auto	53.85	481			108.62	−51.25	58.78	0.742	−10.82	105.74	3.013
rbf, 1, 4	53.37	679			123.12	−36.75	56.95	0.694	11.30	120.24	3.383
rbf, 1, 1	53.37	679			123.12	−36.75	56.95	0.694	11.30	120.24	3.383
rbf, 1, auto	53.49	561			88.41	−71.46	58.31	0.730	−29.13	85.53	2.697
rbf, 5, 4	53.37	679			123.12	−36.75	56.95	0.694	11.30	120.24	3.383
rbf, 5, 1	53.37	679			123.12	−36.75	56.95	0.694	11.30	120.24	3.383
rbf, 5, auto	53.21	705			135.15	−24.73	56.21	0.676	26.18	132.27	3.662
poly, 0.5, 1	55.08	1			158.14	−1.73	67.92	0.988	0.13	155.26	3.192
poly, 0.5, 2	55.20	275			249.09	89.22	63.80	0.870	109.63	246.21	4.417
poly, 0.5, 3	54.29	393			147.63	−12.24	49.40	0.521	63.00	144.75	3.962
poly, 1, 1	55.08	1			158.14	−1.73	67.92	0.988	0.13	155.26	3.192
poly, 1, 2	55.16	283			209.97	50.10	64.00	0.880	68.98	207.09	4.002
poly, 1, 3	53.77	455			126.37	−33.50	47.52	0.484	47.55	123.49	3.745
poly, 5, 1	55.08	1			158.14	−1.73	67.92	0.988	0.13	155.26	3.192
poly, 5, 2	54.72	319			188.77	28.90	62.99	0.850	52.39	185.90	3.863
poly, 5, 3	52.62	603			48.23	−111.64	45.79	0.450	−25.23	45.35	2.027
Logistic Regression											
(*Penalty*, *C*)			2.878	159.9							
l2, 0.01	55.20	343			244.11	84.24	62.15	0.827	111.45	241.23	4.501
l2, 1	55.63	363			283.67	123.80	62.12	0.825	151.31	280.79	4.849
l2, 5	55.56	365			277.55	117.68	62.10	0.824	145.25	274.67	4.795
l2, 10	55.56	365			277.55	117.68	62.10	0.824	145.25	274.67	4.795
l2, 50	55.56	365			277.55	117.68	62.10	0.824	145.25	274.67	4.795
l2, 100	55.56	365			277.55	117.68	62.10	0.824	145.25	274.67	4.795
Gaussian Naive Bayes	51.83	465	2.878	159.9	20.80	−139.07	43.76	0.412	−46.69	17.92	1.154

Table A.6: IXIC 10 Year Model Performance with Non-Trending Data

	Model Performance											
Model	TP	FN	TN	FP	TPR	TNR	PPV	NPV	Recall	Prec.	Acc.	F_1
($Parameters$)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	$Score$
Random Forest												
(# $Trees$)												
20	25.16	29.92	23.37	21.55	45.68	52.03	53.87	43.86	48.85	48.86	48.53	49.43
40	27.82	27.26	22.10	22.82	50.50	49.20	54.94	44.77	49.85	49.86	49.92	52.63
60	28.97	26.11	20.67	24.25	52.59	46.02	54.44	44.19	49.31	49.31	49.64	53.50
80	28.85	26.23	20.63	24.29	52.38	45.94	54.29	44.03	49.16	49.16	49.48	53.32
100	28.77	26.31	20.32	24.60	52.23	45.23	53.90	43.57	48.73	48.74	49.09	53.06
120	29.05	26.03	20.12	24.80	52.74	44.79	53.94	43.59	48.76	48.77	49.17	53.33
140	29.56	25.52	20.52	24.40	53.67	45.67	54.78	44.57	49.67	49.67	50.08	54.22
K-Nearest Neighbors												
(# $Neighbors$)												
20	28.25	26.83	21.90	23.02	51.30	48.76	55.11	44.95	50.03	50.03	50.16	53.13
40	32.42	22.66	17.74	27.18	58.86	39.49	54.39	43.91	49.17	49.15	50.16	56.54
60	35.08	20.00	16.31	28.61	63.69	36.31	55.08	44.92	50.00	50.00	51.39	59.07
80	37.22	17.86	15.32	29.60	67.58	34.10	55.70	46.17	50.84	50.94	52.54	61.07
100	38.73	16.35	13.81	31.11	70.32	30.74	55.45	45.79	50.53	50.62	52.54	62.01
120	40.08	15.00	12.90	32.02	72.77	28.71	55.59	46.23	50.74	50.91	52.98	63.03
140	41.94	13.13	10.60	34.33	76.15	23.59	54.99	44.65	49.87	49.82	52.54	63.87
Logistic Regression												
($Penalty$, C)												
All	55.08	0.00	0.00	44.92	100.00	0.00	55.08	0.00	50.00	27.54	55.08	71.03
Gaussian Naive Bayes	48.93	6.15	5.40	39.52	88.83	12.01	55.32	46.74	50.42	51.03	54.33	68.18

Table A.7: IXIC 10 Year Model Performance with Trend Specific Data

	Model Performance											
Model	TP	FN	TN	FP	TPR	TNR	PPV	NPV	Recall	Prec.	Acc.	F_1
(*Parameters*)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	$Score$
Random Forest												
(# *Trees*)												
20	43.06	12.02	9.84	35.08	78.17	21.91	55.10	45.01	50.04	50.06	52.90	64.64
40	43.41	11.67	9.48	35.44	78.82	21.11	55.06	44.84	49.97	49.95	52.90	64.83
60	43.29	11.79	9.52	35.40	78.60	21.20	55.02	44.69	49.90	49.86	52.82	64.73
80	43.25	11.83	9.60	35.32	78.53	21.38	55.05	44.81	49.95	49.93	52.86	64.73
100	43.65	11.43	9.72	35.20	79.25	21.64	55.36	45.97	50.45	50.66	53.37	65.19
120	43.45	11.63	9.48	35.44	78.89	21.11	55.08	44.92	50.00	50.00	52.94	64.87
140	43.17	11.90	10.04	34.88	78.39	22.35	55.31	45.75	50.37	50.53	53.21	64.86
K-Nearest Neighbors												
(# *Neighbors*)												
20	34.76	20.32	18.69	26.23	63.11	41.61	56.99	47.91	52.36	52.45	53.45	59.90
40	37.10	17.98	15.83	29.09	67.36	35.25	56.06	46.83	51.31	51.44	52.94	61.19
60	41.63	13.45	12.02	32.90	75.58	26.77	55.86	47.20	51.17	51.53	53.65	64.24
80	43.13	11.94	10.36	34.56	78.31	23.06	55.52	46.44	50.69	50.98	53.49	64.97
100	43.57	11.51	9.64	35.28	79.11	21.47	55.26	45.59	50.29	50.43	53.21	65.07
120	44.09	10.99	9.80	35.12	80.04	21.82	55.66	47.14	50.93	51.40	53.89	65.66
140	45.67	9.40	8.89	36.03	82.93	19.79	55.90	48.59	51.36	52.25	54.56	66.78
Support Vector Machine												
(*Kernel*, *C*, γ *or Degree*)												
rbf, 0.5, 1	44.60	10.48	8.77	36.15	80.98	19.52	55.23	45.57	50.25	50.40	53.37	65.67
rbf, 0.5, 1	44.60	10.48	8.77	36.15	80.98	19.52	55.23	45.57	50.25	50.40	53.37	65.67
rbf, 0.5, auto	48.13	6.94	5.71	39.21	87.39	12.72	55.11	45.14	50.06	50.13	53.37	67.60
rbf, 1, 4	44.60	10.48	8.77	36.15	80.98	19.52	55.23	0.00	50.25	27.62	53.85	65.67
rbf, 1, 1	44.60	10.48	8.77	36.15	80.98	19.52	55.23	45.57	50.25	50.40	53.37	65.67
rbf, 1, auto	46.63	8.45	6.87	38.06	84.65	15.28	55.06	44.82	49.97	49.94	53.37	66.72
rbf, 5, 4	44.60	10.48	8.77	36.15	80.98	19.52	55.23	45.57	50.25	50.40	53.49	65.67
rbf, 5, 1	44.60	10.48	8.77	36.15	80.98	19.52	55.23	45.57	50.25	50.40	53.37	65.67
rbf, 5, auto	44.37	10.71	8.85	36.07	80.55	19.70	55.16	45.23	50.12	50.19	53.37	65.48
poly, 0.5, 1	55.08	0.00	0.00	44.92	100.00	0.00	55.08	0.00	50.00	27.54	53.21	71.03
poly, 0.5, 2	52.18	2.90	3.02	41.90	94.74	6.71	55.46	0.00	50.73	27.73	55.08	69.97
poly, 0.5, 3	44.21	10.87	10.08	34.84	80.26	22.44	55.92	48.11	51.35	52.01	55.20	65.92
poly, 1, 1	55.08	0.00	0.00	44.92	100.00	0.00	55.08	0.00	50.00	27.54	54.29	71.03
poly, 1, 2	52.02	3.06	3.13	41.79	94.45	6.98	55.46	0.00	50.72	27.73	55.08	69.88
poly, 1, 3	42.58	12.50	11.19	33.73	77.31	24.91	55.80	47.24	51.11	51.52	55.16	64.81
poly, 5, 1	55.08	0.00	0.00	44.92	100.00	0.00	55.08	0.00	50.00	27.54	53.77	71.03
poly, 5, 2	51.31	3.77	3.41	41.51	93.16	7.60	55.28	0.00	50.38	27.64	55.08	69.39
poly, 5, 3	38.25	16.83	14.37	30.56	69.45	31.98	55.59	46.06	50.72	50.82	54.72	61.76
Logistic Regression												
(*Penalty*, *C*)												
l2, 0.01	50.63	4.44	4.56	40.36	91.93	10.16	55.65	50.66	51.04	53.15	55.20	69.33
l2, 1	50.63	4.44	5.00	39.92	91.93	11.13	55.92	52.94	51.53	54.43	55.63	69.54
l2, 5	50.56	4.52	5.00	39.92	91.79	11.13	55.88	52.50	51.46	54.19	55.56	69.47
l2, 10	50.56	4.52	5.00	39.92	91.79	11.13	55.88	52.50	51.46	54.19	55.56	69.47
l2, 50	50.56	4.52	5.00	39.92	91.79	11.13	55.88	52.50	51.46	54.19	55.56	69.47
l2, 100	50.56	4.52	5.00	39.92	91.79	11.13	55.88	52.50	51.46	54.19	55.56	69.47
Gaussian Naive Bayes	36.11	18.97	15.71	29.21	65.56	34.98	55.29	45.31	50.27	50.30	51.83	59.99

Appendix B. GSPC Accuracy and Trading Performance

Table B.8: GSPC 10 Year Trading Performance with Non-trending Data

			Trading Performance								
Model	Acc.	No. of	R_f	R_m	R_p	Above	σ	β	α	Sharpe	Sortino
(*Parameters*)	(%)	Trades	(%)	(%)	(%)	Mkt.	(%)	(%)	(%)	(%)	(%)
Random Forest											
(# *Trees*)			2.878	81.8							
20	49.09	1101			15.77	−66.01	42.05	0.427	12.89	12.89	1.048
40	48.77	1129			1.96	−79.81	42.31	0.433	−0.91	−0.91	0.410
60	49.60	1087			15.15	−66.63	44.04	0.468	12.27	12.27	1.003
80	48.85	1063			7.95	−73.83	41.35	0.414	5.07	5.07	0.680
100	48.65	1109			27.76	−54.02	42.73	0.441	24.88	24.88	1.554
120	48.97	1091			49.64	−32.14	44.31	0.474	46.76	46.76	2.337
140	47.90	1063			−16.61	−98.39	44.90	0.488	−19.49	−19.49	−0.502
K-Nearest Neighbors											
(# *Neighbors*)			2.878	81.8							
20	50.99	390			68.52	−13.26	46.76	0.529	23.91	65.64	2.727
40	50.56	251			41.34	−40.43	45.72	0.506	−1.46	38.46	1.890
60	51.27	197			14.14	−67.63	49.85	0.600	−36.09	11.27	0.929
80	51.63	149			27.10	−54.68	50.45	0.617	−24.45	24.22	1.360
100	50.87	131			17.83	−63.95	53.19	0.683	−38.94	14.95	1.041
120	50.83	121			14.65	−67.12	55.62	0.747	−47.19	11.77	0.966
140	51.98	85			41.48	−40.30	56.91	0.785	−23.31	38.60	1.682
Logistic Regression											
(*Penalty*, *C*)			2.878	81.8							
All	54.17	1			81.19	−0.59	63.58	0.985	0.578	78.31	2.283
Gaussian Naive Bayes	54.17	1	2.878	81.8	81.19	−0.59	63.58	0.985	0.578	78.31	2.283

Table B.9: GSPC 10 Year Trading Performance with Trend Specific Data

			Trading Performance								
Model (*Parameters*)	Acc. (%)	No. of Trades	R_f (%)	R_m (%)	R_p (%)	Above Mkt.	σ (%)	β (%)	α (%)	Sharpe (%)	Sortino (%)
Random Forest											
(# *Trees*)			2.878	81.8							
20	52.50	827			163.21	81.44	55.93	0.755	100.74	160.33	4.253
40	51.79	785			131.37	49.60	56.40	0.768	67.87	128.50	3.722
60	51.55	761			102.06	20.28	56.69	0.776	37.96	99.18	3.171
80	51.47	791			104.06	22.28	56.69	0.777	39.91	101.18	3.223
100	51.67	773			96.24	14.46	56.65	0.775	32.23	93.36	3.065
120	52.26	769			134.76	52.98	56.64	0.774	70.84	131.88	3.751
140	52.26	765			138.41	56.63	56.74	0.778	74.17	135.53	3.800
K-Nearest Neighbors											
(# *Neighbors*)			2.878	81.8							
20	49.44	1009			81.66	−0.11	50.39	0.614	30.37	78.78	3.150
40	50.60	959			78.29	−3.49	52.58	0.669	22.65	75.41	2.957
60	51.67	819			112.19	30.41	54.68	0.722	52.31	109.31	3.532
80	52.58	803			116.90	35.12	56.12	0.763	53.83	114.02	3.432
100	50.28	767			98.25	16.47	55.47	0.743	36.75	95.37	3.191
120	50.87	749			147.52	65.74	55.25	0.738	86.41	144.64	4.104
140	51.55	723			213.75	131.97	54.73	0.723	153.80	210.87	5.182
Support Vector Machine											
(*Kernel*, *C*, γ *or Degree*)			2.878	81.8							
rbf, 0.5, 4	52.42	697			114.46	32.68	57.48	0.798	48.65	111.58	3.322
rbf, 0.5, 1	52.58	689			118.32	36.55	57.49	0.798	52.47	115.44	3.384
rbf, 0.5, auto	53.29	457			152.18	70.40	59.87	0.868	80.82	149.30	3.694
rbf, 1, 4	52.50	699			116.23	34.45	57.47	0.798	50.42	113.35	3.353
rbf, 1, 1	52.50	699			116.23	34.45	57.47	0.798	50.42	113.35	3.353
rbf, 1, auto	53.06	553			152.65	70.87	57.79	0.806	86.14	149.77	3.884
rbf, 5, 4	52.50	699			116.23	34.45	57.47	0.798	50.42	113.35	3.353
rbf, 5, 1	52.50	699			116.23	34.45	57.47	0.798	50.42	113.35	3.353
rbf, 5, auto	52.66	649			125.30	43.52	56.88	0.781	60.81	122.42	3.533
poly, 0.5, 1	54.17	1			81.19	−0.59	63.58	0.985	0.58	78.31	2.283
poly, 0.5, 2	53.81	51			70.64	−11.14	63.55	0.979	−9.48	67.76	2.117
poly, 0.5, 3	54.17	1			81.19	−0.59	63.58	0.985	0.58	78.31	2.283
poly, 1, 1	54.17	1			81.19	−0.59	63.58	0.985	0.58	78.31	2.283
poly, 1, 2	53.85	125			67.93	−13.85	62.10	0.933	−8.54	65.05	2.074
poly, 1, 3	53.77	119			154.12	72.34	58.66	0.831	85.69	151.24	3.699
poly, 5, 1	54.17	1			81.19	−0.59	63.58	0.985	0.58	78.31	2.283
poly, 5, 2	53.61	201			102.51	20.73	59.70	0.862	31.65	99.63	2.783
poly, 5, 3	53.29	207			183.40	101.63	56.28	0.765	120.15	180.52	4.286
Logistic Regression											
(*Penalty*, *C*)			2.878	81.8							
l2, 0.01	53.13	451			106.39	24.62	62.00	0.928	30.26	103.51	2.864
l2, 1	53.29	439			124.09	42.31	61.80	0.925	48.22	121.21	3.122
l2, 5	53.33	437			127.06	45.29	61.83	0.926	51.15	124.18	3.162
l2, 10	53.33	437			127.06	45.29	61.83	0.926	51.15	124.18	3.162
l2, 50	53.33	437			127.06	45.29	61.83	0.926	51.15	124.18	3.162
l2, 100	53.33	437			127.06	45.29	61.83	0.926	51.15	124.18	3.162
Gaussian Naive Bayes	51.98	424	2.878	81.8	151.93	70.15	56.27	0.764	88.77	149.05	3.980

Table B.10: GSPC 10 Year Model Performance with Non-Trending Data

	Model Performance											
Model	TP	FN	TN	FP	TPR	TNR	PPV	NPV	Recall	Prec.	Acc.	F_1
(*Parameters*)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	$Score$
Random Forest												
(# *Trees*)												
20	24.60	29.56	24.48	21.35	45.42	53.42	53.54	45.30	49.42	49.42	49.09	49.15
40	25.99	28.17	22.78	23.06	47.99	49.70	52.99	44.70	48.84	48.85	48.77	50.37
60	26.63	27.54	22.98	22.86	49.16	50.13	53.81	45.48	49.64	49.65	49.60	51.38
80	26.51	27.66	22.34	23.49	48.94	48.74	53.02	44.68	48.84	48.85	48.85	50.90
100	27.02	27.14	21.63	24.21	49.89	47.19	52.75	44.34	48.54	48.55	48.65	51.28
120	27.50	26.67	21.47	24.37	50.77	46.84	53.02	44.60	48.80	48.81	48.97	51.87
140	27.02	27.14	20.87	24.96	49.89	45.54	51.98	43.47	47.72	47.73	47.90	50.92
K-Nearest Neighbors												
(# *Neighbors*)												
20	26.07	28.10	24.92	20.91	48.13	54.37	55.49	47.01	51.25	51.25	50.99	51.55
40	30.16	24.01	20.40	25.44	55.68	44.50	54.25	45.93	50.09	50.09	50.56	54.95
60	31.90	22.26	19.37	26.47	58.90	42.25	54.66	46.52	50.58	50.59	51.27	56.70
80	33.41	20.75	18.21	27.62	61.68	39.74	54.75	46.74	50.71	50.74	51.63	58.01
100	35.28	18.89	15.60	30.24	65.13	34.03	53.85	45.22	49.58	49.54	50.87	58.95
120	36.11	18.06	14.72	31.11	66.67	32.12	53.72	44.92	49.39	49.32	50.83	59.50
140	38.89	15.28	13.10	32.74	71.79	28.57	54.29	46.15	50.18	50.22	51.98	61.83
Logistic Regression												
(*Penalty*, *C*)												
All	54.17	0.00	0.00	45.83	100.00	0.00	54.17	0.00	50.00	27.08	54.17	70.27
Gaussian Naive Bayes	54.17	0.00	0.00	45.83	100.00	0.00	54.17	0.00	50.00	27.08	54.17	70.27

Table B.11: GSPC 10 Year Model Performance with Trend Specific Data

	Model Performance											
Model	TP	FN	TN	FP	TPR	TNR	PPV	NPV	Recall	Prec.	Acc.	F_1
(*Parameters*)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	$Score$
Random Forest												
(# *Trees*)												
20	40.91	13.25	11.59	34.25	75.53	25.28	54.44	46.65	50.41	50.54	52.50	63.27
40	41.03	13.13	10.75	35.08	75.75	23.46	53.91	45.02	49.61	49.46	51.79	62.99
60	41.59	12.58	9.96	35.87	76.78	21.73	53.69	44.19	49.25	48.94	51.55	63.19
80	41.03	13.13	10.44	35.40	75.75	22.77	53.69	44.28	49.26	48.98	51.47	62.84
100	41.43	12.74	10.24	35.60	76.48	22.34	53.79	44.56	49.41	49.17	51.67	63.16
120	41.71	12.46	10.56	35.28	77.00	23.03	54.18	45.86	50.01	50.02	52.26	63.60
140	41.90	12.26	10.36	35.48	77.36	22.60	54.15	45.79	49.98	49.97	52.26	63.71
K-Nearest Neighbors												
(# *Neighbors*)												
20	27.70	26.47	21.75	24.09	51.14	47.45	53.49	45.10	49.29	49.29	49.44	52.28
40	30.00	24.17	20.60	25.24	55.38	44.94	54.31	46.01	50.16	50.16	50.60	54.84
60	32.34	21.83	19.33	26.51	59.71	42.16	54.96	46.96	50.94	50.96	51.67	57.23
80	31.90	22.26	20.67	25.16	58.90	45.11	55.91	48.15	52.00	52.03	52.58	57.37
100	28.41	25.75	21.87	23.97	52.45	47.71	54.24	45.92	50.08	50.08	50.28	53.33
120	29.56	24.60	21.31	24.52	54.58	46.49	54.66	46.41	50.54	50.54	50.87	54.62
140	30.24	23.93	21.31	24.52	55.82	46.49	55.22	47.11	51.16	51.16	51.55	55.52
Support Vector Machine												
(*Kernel*, *C*, γ *or Degree*)												
rbf, 0.5, 4	43.29	10.87	9.13	36.71	79.93	19.91	54.12	0.00	49.92	27.06	52.62	64.54
rbf, 0.5, 1	43.57	10.60	9.01	36.83	80.44	19.65	54.20	45.95	50.05	50.07	52.42	64.76
rbf, 0.5, auto	47.34	6.83	5.95	39.88	87.40	12.99	54.28	46.58	50.19	50.43	52.58	66.97
rbf, 1, 4	43.29	10.87	9.21	36.63	79.93	20.09	54.17	0.00	50.01	27.09	53.29	64.58
rbf, 1, 1	43.29	10.87	9.21	36.63	79.93	20.09	54.17	45.85	50.01	50.01	52.50	64.58
rbf, 1, auto	45.60	8.57	7.46	38.37	84.18	16.28	54.30	46.53	50.23	50.42	52.50	66.02
rbf, 5, 4	43.29	10.87	9.21	36.63	79.93	20.09	54.17	0.00	50.01	27.09	53.06	64.58
rbf, 5, 1	43.29	10.87	9.21	36.63	79.93	20.09	54.17	45.85	50.01	50.01	52.50	64.58
rbf, 5, auto	43.49	10.67	9.17	36.67	80.29	20.00	54.26	46.20	50.15	50.23	52.50	64.76
poly, 0.5, 1	54.17	0.00	0.00	45.83	100.00	0.00	54.17	0.00	50.00	27.08	52.66	70.27
poly, 0.5, 2	53.45	0.71	0.36	45.48	98.68	0.78	54.03	0.00	49.73	27.02	54.17	69.83
poly, 0.5, 3	54.17	0.00	0.00	45.83	100.00	0.00	54.17	0.00	50.00	27.08	53.81	70.27
poly, 1, 1	54.17	0.00	0.00	45.83	100.00	0.00	54.17	0.00	50.00	27.08	54.17	70.27
poly, 1, 2	52.70	1.47	1.15	44.68	97.29	2.51	54.12	0.00	49.90	27.06	54.17	69.55
poly, 1, 3	50.67	3.49	3.10	42.74	93.55	6.75	54.25	46.99	50.15	50.62	53.85	68.67
poly, 5, 1	54.17	0.00	0.00	45.83	100.00	0.00	54.17	0.00	50.00	27.08	53.77	70.27
poly, 5, 2	51.63	2.54	1.98	43.85	95.31	4.33	54.07	0.00	49.82	27.04	54.17	69.00
poly, 5, 3	48.25	5.91	5.04	40.79	89.08	11.00	54.19	46.01	50.04	50.10	53.61	67.39
Logistic Regression												
(*Penalty*, *C*)												
l2, 0.01	46.59	7.58	6.55	39.29	86.01	14.29	54.25	46.35	50.15	50.30	53.13	66.53
l2, 1	47.14	7.02	6.15	39.68	87.03	13.42	54.30	46.69	50.23	50.49	53.29	66.87
l2, 5	47.18	6.98	6.15	39.68	87.11	13.42	54.32	46.83	50.26	50.57	53.33	66.91
l2, 10	47.18	6.98	6.15	39.68	87.11	13.42	54.32	46.83	50.26	50.57	53.33	66.91
l2, 50	47.18	6.98	6.15	39.68	87.11	13.42	54.32	46.83	50.26	50.57	53.33	66.91
l2, 100	47.18	6.98	6.15	39.68	87.11	13.42	54.32	46.83	50.26	50.57	53.33	66.91
Gaussian Naive Bayes	28.21	25.95	23.77	22.06	52.09	51.86	56.12	47.81	51.97	51.96	51.98	54.03

Appendix C. RUT Accuracy and Trading Performance

Table C.12: RUT 10 Year Trading Performance with Non-trending Data

			Trading Performance								
Model	Acc.	No. of	R_f	R_m	R_p	Above	σ	β	α	Sharpe	Sortino
(*Parameters*)	(%)	Trades	(%)	(%)	(%)	Mkt.	(%)	(%)	(%)	(%)	(%)
Random Forest											
(# *Trees*)			2.878	100.7							
20	49.33	1112			20.41	−80.29	53.92	0.434	17.53	17.53	1.264
40	50.12	1078			65.96	−34.75	51.12	0.390	63.08	63.08	2.714
60	49.37	1066			59.63	−41.07	53.10	0.421	56.76	56.76	2.502
80	50.08	1028			151.45	50.75	55.92	0.467	148.57	148.57	4.351
100	49.48	1024			23.23	−77.47	54.80	0.448	20.35	20.35	1.350
120	49.92	1048			146.47	45.77	56.74	0.480	143.60	143.60	4.189
140	49.25	1038			23.26	−77.44	56.09	0.470	20.39	20.39	1.323
K-Nearest Neighbors											
(# *Neighbors*)			2.878	100.7							
20	49.33	462			85.63	−15.07	57.43	0.492	34.60	82.75	2.964
40	50.60	310			55.27	−45.43	61.74	0.569	−3.27	52.39	2.070
60	51.39	183			113.95	13.25	66.33	0.657	46.83	111.07	3.081
80	49.96	145			26.51	−74.19	66.73	0.666	−41.53	23.63	1.365
100	50.00	117			29.42	−71.28	68.20	0.695	−41.42	26.54	1.433
120	49.48	107			16.97	−83.73	66.98	0.671	−51.51	14.09	1.122
140	50.32	107			36.91	−63.79	67.03	0.671	−31.63	34.03	1.611
Logistic Regression											
(*Penalty*, *C*)			2.878	100.7							
All	52.62	1			100.04	−0.67	81.74	0.997	−0.40	97.16	2.368
Gaussian Naive Bayes	46.87	36	2.878	100.7	1.09	−99.62	4.33	0.003	−2.07	−1.79	−0.924

Table C.13: RUT 10 Year Trading Performance with Trend Specific Data

			Trading Performance								
Model	Acc.	No. of	R_f	R_m	R_p	Above	σ	β	α	Sharpe	Sortino
(*Parameters*)	(%)	Trades	(%)	(%)	(%)	Mkt.	(%)	(%)	(%)	(%)	(%)
Random Forest											
(# *Trees*)			2.878	100.7							
20	50.08	722			26.57	−74.13	70.45	0.741	−48.82	23.69	1.356
40	49.72	718			33.99	−66.71	71.00	0.753	−42.52	31.11	1.534
60	50.08	736			22.42	−78.28	70.63	0.745	−53.35	19.54	1.260
80	49.44	724			14.52	−86.18	71.28	0.759	−62.59	11.64	1.069
100	49.92	738			26.27	−74.44	70.74	0.748	−49.74	23.39	1.361
120	49.52	728			22.59	−78.12	70.89	0.750	−53.68	19.71	1.268
140	49.64	724			32.98	−67.72	70.62	0.745	−42.77	30.10	1.518
K-Nearest Neighbors											
(# *Neighbors*)			2.878	100.7							
20	49.92	1006			56.90	−43.80	63.23	0.597	−4.38	54.02	2.170
40	49.60	932			−4.81	−105.51	66.86	0.668	−73.04	−7.69	0.467
60	49.21	895			−7.05	−107.75	65.93	0.650	−73.49	−9.93	0.376
80	48.81	867			−3.68	−104.38	67.34	0.678	−72.87	−6.56	0.512
100	49.21	823			−12.54	−113.24	67.60	0.684	−82.31	−15.42	0.209
120	50.52	771			6.62	−94.09	69.61	0.725	−67.17	3.74	0.839
140	50.56	737			11.48	−89.22	70.10	0.736	−63.35	8.60	0.971
Support Vector Machine											
(*Kernel*, *C*, γ *or Degree*)			2.878	100.7							
rbf, 0.5, 4	49.48	666			21.45	−79.25	71.83	0.770	−56.79	18.57	1.242
rbf, 0.5, 1	49.48	666			21.45	−79.25	71.83	0.770	−56.79	18.57	1.242
rbf, 0.5, auto	50.75	377			47.21	−53.49	73.23	0.802	−34.10	44.33	1.740
rbf, 1, 4	49.48	666			21.45	−79.25	71.83	0.770	−56.79	18.57	1.242
rbf, 1, 1	49.48	666			21.45	−79.25	71.83	0.770	−56.79	18.57	1.242
rbf, 1, auto	50.40	499			17.81	−82.89	73.11	0.799	−63.20	14.93	1.130
rbf, 5, 4	49.48	666			21.45	−79.25	71.83	0.770	−56.79	18.57	1.242
rbf, 5, 1	49.48	666			21.45	−79.25	71.83	0.770	−56.79	18.57	1.242
rbf, 5, auto	49.92	639			18.50	−82.20	71.82	0.771	−59.79	15.63	1.161
poly, 0.5, 1	52.62	1			100.04	−0.67	81.74	0.997	−0.40	97.16	2.368
poly, 0.5, 2	52.14	193			86.97	−13.73	79.10	0.935	−7.40	84.10	2.290
poly, 0.5, 3	52.62	1			100.04	−0.67	81.74	0.997	−0.40	97.16	2.368
poly, 1, 1	52.62	1			100.04	−0.67	81.74	0.997	−0.40	97.16	2.368
poly, 1, 2	51.94	213			88.03	−12.67	79.04	0.934	−6.17	85.15	2.311
poly, 1, 3	52.58	3			98.46	−2.24	81.77	0.998	−2.00	95.58	2.349
poly, 5, 1	52.62	1			100.04	−0.67	81.74	0.997	−0.40	97.16	2.368
poly, 5, 2	52.30	229			97.48	−3.22	78.89	0.930	3.59	94.61	2.437
poly, 5, 3	52.46	47			102.05	1.35	81.38	0.988	2.50	99.17	2.409
Logistic Regression											
(*Penalty*, *C*)			2.878	100.7							
l2, 0.01	51.90	323			86.71	−14.00	74.07	0.819	3.68	83.83	2.357
l2, 1	52.18	321			122.35	21.65	74.88	0.838	37.52	119.47	2.827
l2, 5	52.18	321			122.35	21.65	74.88	0.838	37.52	119.47	2.827
l2, 10	52.18	321			122.35	21.65	74.88	0.838	37.52	119.47	2.827
l2, 50	52.18	321			122.35	21.65	74.88	0.838	37.52	119.47	2.827
l2, 100	52.18	321			122.35	21.65	74.88	0.838	37.52	119.47	2.827
Gaussian Naive Bayes	48.21	615	2.878	100.7	−42.49	−143.19	56.93	0.486	−92.87	−45.37	−1.505

Table C.14: RUT 10 Year Model Performance with Non-Trending Data

	Model Performance											
Model	TP	FN	TN	FP	TPR	TNR	PPV	NPV	Recall	Prec.	Acc.	F_1
($Parameters$)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	$Score$
Random Forest												
(# $Trees$)												
20	22.70	29.92	26.63	20.75	43.14	56.20	52.24	47.09	49.67	49.66	49.33	47.25
40	24.48	28.13	25.63	21.75	46.53	54.10	52.96	47.68	50.32	50.32	50.12	49.54
60	24.96	27.66	24.40	22.98	47.44	51.51	52.07	46.88	49.47	49.47	49.37	49.64
80	25.79	26.83	24.29	23.10	49.02	51.26	52.76	47.52	50.14	50.14	50.08	50.82
100	25.36	27.26	24.13	23.25	48.19	50.92	52.16	46.95	49.56	49.56	49.48	50.10
120	25.91	26.71	24.01	23.37	49.25	50.67	52.58	47.34	49.96	49.96	49.92	50.86
140	25.44	27.18	23.81	23.57	48.34	50.25	51.90	46.69	49.30	49.30	49.25	50.06
K-Nearest Neighbors												
(# $Neighbors$)												
20	23.02	29.60	26.31	21.07	43.74	55.53	52.21	47.05	49.63	49.63	49.33	47.60
40	26.15	26.47	24.44	22.94	49.70	51.59	53.27	48.01	50.64	50.64	50.60	51.42
60	27.66	24.96	23.73	23.65	52.56	50.08	53.91	48.74	51.32	51.32	51.39	53.23
80	27.42	25.20	22.54	24.84	52.11	47.57	52.47	47.22	49.84	49.84	49.96	52.29
100	29.05	23.57	20.95	26.43	55.20	44.22	52.36	47.06	49.71	49.71	50.00	53.74
120	29.09	23.53	20.40	26.98	55.28	43.05	51.88	46.43	49.16	49.15	49.48	53.52
140	31.90	20.71	18.41	28.97	60.63	38.86	52.41	47.06	49.75	49.74	50.32	56.22
Logistic Regression												
($Penalty$, C)												
All	52.62	0.00	0.00	47.38	100.00	0.00	52.62	0.00	50.00	26.31	52.62	68.95
Gaussian Naive Bayes	0.79	51.83	46.07	1.31	1.51	97.24	37.74	47.06	49.37	42.40	46.87	2.90

Table C.15: RUT 10 Year Model Performance with Trend Specific Data

	Model Performance											
Model (*Parameters*)	TP (%)	FN (%)	TN (%)	FP (%)	*TPR* (%)	*TNR* (%)	*PPV* (%)	*NPV* (%)	Recall (%)	Prec. (%)	Acc. (%)	F_1 *Score*
Random Forest												
(# *Trees*)												
20	40.56	12.06	9.52	37.86	77.07	20.10	51.72	44.12	48.59	47.92	50.08	61.90
40	40.52	12.10	9.21	38.17	77.00	19.43	51.49	43.20	48.21	47.35	49.72	61.71
60	40.60	12.02	9.48	37.90	77.15	20.02	51.72	44.10	48.58	47.91	50.08	61.92
80	40.24	12.38	9.21	38.17	76.47	19.43	51.32	42.65	47.95	46.98	49.44	61.42
100	40.56	12.06	9.37	38.02	77.07	19.77	51.62	43.70	48.42	47.66	49.92	61.83
120	40.36	12.26	9.17	38.21	76.70	19.35	51.36	42.78	48.02	47.07	49.52	61.52
140	40.32	12.30	9.33	38.06	76.62	19.68	51.44	43.12	48.15	47.28	49.64	61.56
K-Nearest Neighbors												
(# *Neighbors*)												
20	32.62	20.00	17.30	30.08	61.99	36.52	52.03	46.38	49.25	49.20	49.92	56.57
40	34.92	17.70	14.68	32.70	66.37	30.99	51.64	45.34	48.68	48.49	49.60	58.09
60	34.48	18.13	14.72	32.66	65.54	31.07	51.36	44.81	48.30	48.08	49.21	57.59
80	35.36	17.26	13.45	33.93	67.19	28.39	51.03	43.80	47.79	47.41	48.81	58.01
100	35.99	16.63	13.21	34.17	68.40	27.89	51.30	44.28	48.15	47.79	49.21	58.63
120	38.53	14.09	11.98	35.40	73.23	25.29	52.12	45.97	49.26	49.04	50.52	60.90
140	39.25	13.37	11.31	36.07	74.59	23.87	52.11	45.82	49.23	48.96	50.56	61.35
Support Vector Machine												
(*Kernel*, *C*, γ *or Degree*)												
rbf, 0.5, 4	41.51	11.11	7.98	39.40	78.88	16.83	51.30	41.79	47.86	46.54	53.29	62.17
rbf, 0.5, 1	41.51	11.11	7.98	39.40	78.88	16.83	51.30	41.79	47.86	46.54	49.48	62.17
rbf, 0.5, auto	46.55	6.07	4.21	43.17	88.46	8.88	51.88	40.93	48.67	46.40	49.48	65.40
rbf, 1, 4	41.51	11.11	7.98	39.40	78.88	16.83	51.30	41.79	47.86	46.54	50.75	62.17
rbf, 1, 1	41.51	11.11	7.98	39.40	78.88	16.83	51.30	41.79	47.86	46.54	49.48	62.17
rbf, 1, auto	44.88	7.74	5.52	41.87	85.29	11.64	51.74	41.62	48.47	46.68	49.48	64.41
rbf, 5, 4	41.51	11.11	7.98	39.40	78.88	16.83	51.30	41.79	47.86	46.54	50.40	62.17
rbf, 5, 1	41.51	11.11	7.98	39.40	78.88	16.83	51.30	41.79	47.86	46.54	49.48	62.17
rbf, 5, auto	42.18	10.44	7.74	39.64	80.17	16.33	51.55	42.58	48.25	47.06	49.48	62.75
poly, 0.5, 1	52.62	0.00	0.00	47.38	100.00	0.00	52.62	0.00	50.00	26.31	49.92	68.95
poly, 0.5, 2	50.40	2.22	1.75	45.63	95.78	3.69	52.48	0.00	49.73	26.24	52.62	67.81
poly, 0.5, 3	52.62	0.00	0.00	47.38	100.00	0.00	52.62	0.00	50.00	26.31	52.14	68.95
poly, 1, 1	52.62	0.00	0.00	47.38	100.00	0.00	52.62	0.00	50.00	26.31	52.62	68.95
poly, 1, 2	50.04	2.58	1.90	45.48	95.10	4.02	52.39	0.00	49.56	26.19	52.62	67.56
poly, 1, 3	52.58	0.04	0.00	47.38	99.92	0.00	52.60	0.00	49.96	26.30	51.94	68.92
poly, 5, 1	52.62	0.00	0.00	47.38	100.00	0.00	52.62	0.00	50.00	26.31	52.58	68.95
poly, 5, 2	49.96	2.66	2.34	45.04	94.95	4.94	52.59	0.00	49.94	26.29	52.62	67.69
poly, 5, 3	52.02	0.60	0.44	46.94	98.87	0.92	52.57	42.31	49.90	47.44	52.30	68.64
Logistic Regression												
(*Penalty*, *C*)												
l2, 0.01	48.17	4.44	3.73	43.65	91.55	7.87	52.46	45.63	49.71	49.05	51.90	66.70
l2, 1	48.41	4.21	3.77	43.61	92.01	7.96	52.61	47.26	49.98	49.94	52.18	66.94
l2, 5	48.41	4.21	3.77	43.61	92.01	7.96	52.61	47.26	49.98	49.94	52.18	66.94
l2, 10	48.41	4.21	3.77	43.61	92.01	7.96	52.61	47.26	49.98	49.94	52.18	66.94
l2, 50	48.41	4.21	3.77	43.61	92.01	7.96	52.61	47.26	49.98	49.94	52.18	66.94
l2, 100	48.41	4.21	3.77	43.61	92.01	7.96	52.61	47.26	49.98	49.94	52.18	66.94
Gaussian Naive Bayes	32.78	19.84	15.44	31.94	62.29	32.58	50.64	43.76	47.44	47.20	48.21	55.87

Appendix D. DJI Accuracy and Trading Performance

Table D.16: DJI 10 Year Trading Performance with Non-trending Data

			Trading Performance									
Model	Acc.	No. of	R_f	R_m	R_p	Above	σ	β	α	Sharpe	Sortino
(*Parameters*)	(%)	Trades	(%)	(%)	(%)	Mkt.	(%)	(%)	(%)	(%)	(%)
Random Forest											
(# *Trees*)			2.878	85.8							
20	49.52	1039			38.50	−47.33	34.22	0.353	35.62	35.62	2.306
40	50.00	1051			44.81	−41.02	38.37	0.439	41.93	41.93	2.436
60	49.88	1004			26.05	−59.78	38.90	0.463	23.17	23.17	1.605
80	50.24	1013			84.91	−0.91	40.39	0.483	82.04	82.04	3.794
100	51.31	981			156.11	70.29	40.39	0.481	153.24	153.24	5.769
120	51.15	1049			77.61	−8.22	37.92	0.430	74.73	74.73	3.570
140	50.71	1013			67.24	−18.59	40.51	0.486	64.36	64.36	3.183
K-Nearest Neighbors											
(# *Neighbors*)			2.878	85.8							
20	49.52	1039			38.50	−47.33	34.22	0.353	35.62	35.62	2.306
40	50.00	1051			44.81	−41.02	38.37	0.439	41.93	41.93	2.436
60	49.88	1004			26.05	−59.78	38.90	0.463	23.17	23.17	1.605
80	50.24	1013			84.91	−0.91	40.39	0.483	82.04	82.04	3.794
100	51.31	981			156.11	70.29	40.39	0.481	153.24	153.24	5.769
120	51.15	1049			77.61	−8.22	37.92	0.430	74.73	74.73	3.570
140	50.71	1013			67.24	−18.59	40.51	0.486	64.36	64.36	3.183
Logistic Regression											
(*Penalty*, *C*)			2.878	85.8							
All	53.57	1			80.53	−5.30	54.53	0.918	1.53	77.65	2.505
Gaussian Naive Bayes	53.17	21	2.878	85.8	57.05	−28.78	53.34	0.888	−19.50	54.17	2.044

Table D.17: DJI 10 Year Trading Performance with Trend Specific Data

			Trading Performance								
Model (*Parameters*)	Acc. (%)	No. of Trades	R_f (%)	R_m (%)	R_p (%)	Above Mkt.	σ (%)	β (%)	α (%)	Sharpe (%)	Sortino (%)
Random Forest											
(# *Trees*)			2.878	85.8							
20	51.98	820			66.79	−19.04	49.49	0.737	2.77	63.91	2.612
40	52.34	788			88.40	2.57	50.45	0.765	22.04	85.52	3.072
60	52.06	798			81.43	−4.40	49.26	0.735	17.57	78.55	2.983
80	52.14	778			78.67	−7.16	49.07	0.730	15.27	75.79	2.853
100	52.22	778			52.78	−33.04	48.79	0.717	−9.55	49.91	2.190
120	51.63	768			36.15	−49.67	49.80	0.754	−29.28	33.28	1.701
140	51.31	750			15.79	−70.04	49.43	0.739	−48.42	12.91	1.001
K-Nearest Neighbors											
(# *Neighbors*)			2.878	85.8							
20	49.33	1105			−14.92	−100.75	39.86	0.493	−58.69	−17.80	−0.587
40	49.76	908			2.13	−83.70	44.42	0.609	−51.25	−0.75	0.431
60	50.16	904			31.75	−54.08	47.58	0.680	−27.53	28.87	1.675
80	50.40	934			7.67	−78.16	44.25	0.596	−44.63	4.79	0.681
100	50.56	829			46.93	−38.90	47.11	0.671	−11.61	44.05	2.097
120	51.35	810			89.45	3.62	48.15	0.693	29.10	86.57	3.276
140	51.43	771			138.21	52.38	49.10	0.718	75.81	135.33	4.388
Support Vector Machine											
(*Kernel*, *C*, γ *or Degree*)			2.878	85.8							
rbf, 0.5, 4	52.10	674			76.89	−8.94	52.63	0.824	5.64	74.01	2.711
rbf, 0.5, 1	52.10	674			76.89	−8.94	52.63	0.824	5.64	74.01	2.711
rbf, 0.5, auto	52.50	503			81.68	−4.15	52.58	0.827	10.17	78.80	2.793
rbf, 1, 4	52.10	674			76.89	−8.94	52.63	0.824	5.64	74.01	2.711
rbf, 1, 1	52.10	674			76.89	−8.94	52.63	0.824	5.64	74.01	2.711
rbf, 1, auto	52.54	649			63.63	−22.19	51.11	0.789	−4.66	60.76	2.426
rbf, 5, 4	52.10	674			76.89	−8.94	52.63	0.824	5.64	74.01	2.711
rbf, 5, 1	52.10	674			76.89	−8.94	52.63	0.824	5.64	74.01	2.711
rbf, 5, auto	51.90	684			81.84	−3.99	51.38	0.801	12.55	78.96	2.871
poly, 0.5, 1	53.57	1			80.53	−5.30	54.53	0.918	1.52	77.65	2.505
poly, 0.5, 2	53.21	237			52.74	−33.09	53.41	0.856	−21.15	49.86	1.987
poly, 0.5, 3	52.98	273			158.68	72.85	49.78	0.742	94.23	155.80	4.359
poly, 1, 1	53.57	1			80.53	−5.30	54.53	0.918	1.52	77.65	2.505
poly, 1, 2	53.29	245			55.43	−30.39	52.77	0.845	−17.55	52.56	2.062
poly, 1, 3	51.07	463			109.35	23.52	46.62	0.661	51.64	106.47	3.818
poly, 5, 1	53.57	1			80.53	−5.30	54.53	0.918	1.52	77.65	2.505
poly, 5, 2	53.13	275			33.57	−52.26	52.49	0.837	−38.76	30.69	1.498
poly, 5, 3	51.19	637			117.50	31.67	45.94	0.628	62.57	114.62	4.148
Logistic Regression											
(*Penalty*, *C*)			2.878	85.8							
l2, 0.01	53.13	225			82.28	−3.54	56.77	0.941	1.37	79.41	2.545
l2, 1	53.49	259			87.69	1.86	56.35	0.933	7.46	84.81	2.663
l2, 5	53.45	261			86.08	0.25	56.41	0.933	5.78	83.20	2.631
l2, 10	53.45	261			86.08	0.25	56.41	0.933	5.78	83.20	2.631
l2, 50	53.45	261			86.08	0.25	56.41	0.933	5.78	83.20	2.631
l2, 100	53.45	261			86.08	0.25	56.41	0.933	5.78	83.20	2.631
Gaussian Naive Bayes	49.21	379	2.878	85.8	93.08	7.25	48.54	0.707	31.55	90.20	3.402

Table D.18: DJI 10 Year Model Performance with Non-Trending Data

	Model Performance											
Model	TP	FN	TN	FP	TPR	TNR	PPV	NPV	Recall	Prec.	Acc.	F_1
($Parameters$)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	$Score$
Random Forest												
(# $Trees$)												
20	24.37	29.21	25.16	21.27	45.48	54.19	53.39	46.28	49.83	49.83	49.52	49.12
40	25.63	27.94	24.37	22.06	47.85	52.48	53.74	46.59	50.17	50.16	50.00	50.63
60	26.27	27.30	23.61	22.82	49.04	50.85	53.52	46.38	49.95	49.95	49.88	51.18
80	27.30	26.27	22.94	23.49	50.96	49.40	53.75	46.61	50.18	50.18	50.24	52.32
100	27.78	25.79	23.53	22.90	51.85	50.68	54.82	47.71	51.27	51.26	51.31	53.29
120	27.42	26.15	23.73	22.70	51.19	51.11	54.71	47.57	51.15	51.14	51.15	52.89
140	27.70	25.87	23.02	23.41	51.70	49.57	54.19	47.08	50.64	50.64	50.71	52.92
K-Nearest Neighbors												
(# $Neighbors$)												
20	26.51	27.06	23.77	22.66	49.48	51.20	53.91	46.76	50.34	50.34	50.28	51.60
40	29.37	24.21	21.19	25.24	54.81	45.64	53.78	46.68	50.23	50.23	50.56	54.29
60	29.88	23.69	20.04	26.39	55.78	43.16	53.10	45.83	49.47	49.46	49.92	54.41
80	30.91	22.66	20.08	26.35	57.70	43.25	53.98	46.98	50.48	50.48	50.99	55.78
100	32.26	21.31	18.25	28.17	60.22	39.32	53.38	46.14	49.77	49.76	50.52	56.60
120	35.40	18.17	15.24	31.19	66.07	32.82	53.16	45.61	49.45	49.38	50.63	58.92
140	37.46	16.11	14.05	32.38	69.93	30.26	53.64	46.58	50.09	50.11	51.51	60.71
Logistic Regression												
($Penalty$, C)												
All	53.57	0.00	0.00	46.43	100.00	0.00	53.57	0.00	50.00	26.79	53.57	69.77
Gaussian Naive Bayes	52.62	0.95	0.56	45.87	98.22	1.20	53.42	36.84	49.71	45.13	53.17	69.21

Table D.19: DJI 10 Year Model Performance with Trend Specific Data

	Model Performance											
Model (*Parameters*)	TP (%)	FN (%)	TN (%)	FP (%)	*TPR* (%)	*TNR* (%)	*PPV* (%)	*NPV* (%)	Recall (%)	Prec. (%)	Acc. (%)	F_1 *Score*
Random Forest (# *Trees*)												
20	41.87	11.71	10.12	36.31	78.15	21.79	53.55	46.36	49.97	49.96	51.98	63.55
40	42.42	11.15	9.92	36.51	79.19	21.37	53.75	47.08	50.28	50.41	52.34	64.03
60	42.22	11.35	9.84	36.59	78.81	21.20	53.58	46.44	50.01	50.01	52.06	63.79
80	42.42	11.15	9.72	36.71	79.19	20.94	53.61	46.58	50.06	50.09	52.14	63.94
100	42.38	11.19	9.84	36.59	79.11	21.20	53.67	46.79	50.15	50.23	52.22	63.95
120	42.22	11.35	9.40	37.02	78.81	20.26	53.28	45.32	49.54	49.30	51.63	63.58
140	42.22	11.35	9.09	37.34	78.81	19.57	53.07	44.47	49.19	48.77	51.31	63.43
K-Nearest Neighbors (# *Neighbors*)												
20	26.67	26.90	22.66	23.77	49.78	48.80	52.87	45.72	49.29	49.29	49.33	51.28
40	30.75	22.82	19.01	27.42	57.41	40.94	52.86	45.45	49.17	49.16	49.76	55.04
60	32.34	21.23	17.82	28.61	60.37	38.38	53.06	45.63	49.37	49.34	50.16	56.48
80	31.59	21.98	18.81	27.62	58.96	40.51	53.35	46.11	49.74	49.73	50.40	56.02
100	31.47	22.10	19.09	27.34	58.74	41.11	53.51	46.34	49.93	49.92	50.56	56.00
120	32.78	20.79	18.57	27.86	61.19	40.00	54.06	47.18	50.59	50.62	51.35	57.40
140	33.81	19.76	17.62	28.81	63.11	37.95	53.99	47.13	50.53	50.56	51.43	58.20
Support Vector Machine (*Kernel*, *C*, γ *or Degree*)												
rbf, 0.5, 4	43.89	9.68	8.21	38.21	81.93	17.69	53.46	45.90	49.81	49.68	52.46	64.70
rbf, 0.5, 1	43.89	9.68	8.21	38.21	81.93	17.69	53.46	45.90	49.81	49.68	52.10	64.70
rbf, 0.5, auto	46.07	7.50	6.43	40.00	86.00	13.85	53.53	46.15	49.92	49.84	52.10	65.98
rbf, 1, 4	43.89	9.68	8.21	38.21	81.93	17.69	53.46	45.90	49.81	49.68	52.50	64.70
rbf, 1, 1	43.89	9.68	8.21	38.21	81.93	17.69	53.46	45.90	49.81	49.68	52.10	64.70
rbf, 1, auto	44.25	9.33	8.29	38.13	82.59	17.86	53.71	47.07	50.23	50.39	52.10	65.09
rbf, 5, 4	43.89	9.68	8.21	38.21	81.93	17.69	53.46	45.90	49.81	49.68	52.54	64.70
rbf, 5, 1	43.89	9.68	8.21	38.21	81.93	17.69	53.46	45.90	49.81	49.68	52.10	64.70
rbf, 5, auto	43.37	10.20	8.53	37.90	80.96	18.38	53.37	45.55	49.67	49.46	52.10	64.33
poly, 0.5, 1	53.57	0.00	0.00	46.43	100.00	0.00	53.57	0.00	50.00	26.79	51.90	69.77
poly, 0.5, 2	50.67	2.90	2.54	43.89	94.59	5.47	53.59	0.00	50.03	26.79	53.57	68.42
poly, 0.5, 3	46.11	7.46	6.87	39.56	86.07	14.79	53.82	47.92	50.43	50.87	53.21	66.23
poly, 1, 1	53.57	0.00	0.00	46.43	100.00	0.00	53.57	0.00	50.00	26.79	52.98	69.77
poly, 1, 2	50.56	3.02	2.74	43.69	94.37	5.90	53.64	0.00	50.13	26.82	53.57	68.40
poly, 1, 3	39.33	14.25	11.75	34.68	73.41	25.30	53.14	45.19	49.35	49.16	53.29	61.65
poly, 5, 1	53.57	0.00	0.00	46.43	100.00	0.00	53.57	0.00	50.00	26.79	51.07	69.77
poly, 5, 2	50.16	3.41	2.98	43.45	93.63	6.41	53.58	0.00	50.02	26.79	53.57	68.16
poly, 5, 3	34.72	18.85	16.47	29.96	64.81	35.47	53.68	46.63	50.14	50.16	53.13	58.72
Logistic Regression (*Penalty*, *C*)												
l2, 0.01	50.67	2.90	2.46	43.97	94.59	5.30	53.54	45.93	49.95	49.73	53.13	68.38
l2, 1	50.48	3.10	3.02	43.41	94.22	6.50	53.76	49.35	50.36	51.56	53.49	68.46
l2, 5	50.44	3.13	3.02	43.41	94.15	6.50	53.74	49.03	50.32	51.39	53.45	68.43
l2, 10	50.44	3.13	3.02	43.41	94.15	6.50	53.74	49.03	50.32	51.39	53.45	68.43
l2, 50	50.44	3.13	3.02	43.41	94.15	6.50	53.74	49.03	50.32	51.39	53.45	68.43
l2, 100	50.44	3.13	3.02	43.41	94.15	6.50	53.74	49.03	50.32	51.39	53.45	68.43
Gaussian Naive Bayes	27.10	26.47	22.10	24.33	50.59	47.61	52.70	45.51	49.10	49.10	49.21	51.63