Machine Learning, Investor Sentiment and the Financial Market

Iles Acly Ayati

Católica School of Business and Economics,
BI Norwegian Business School

Contents

1 Introduction

This paper examines the relationships between a selected set of investor sentiment measures with a range of U.S. stock market portfolios, categorized by size and book-to-market ratio. The selected investor sentiment features are closed-end fund discounts, the change in the VIX Index, and a set of the CFTC Commitment of Traders Report data. While it has been documented that the VIX Index and closed-end fund discounts are good predictors of stock market returns (Lee et al., 1991; Bekaert and Hoerova, 2014), there is very little literature analyzing the Commitment of Traders Report and its use for stock market predictions. Still, and especially by some foreign exchange speculators, this data is considered a market signalling tool. Furthermore, there exists even less literature analyzing the causality of investor sentiment measures on the stock market and vice versa. If there existed evidence that these changes in these features affect the following market movements, they could potentially be tools that should be considered for further research and investment analysis. For this reason, this paper investigates the contemporaneous as well as the causal relationships that might exist between a selected set of sentiment features and stock market portfolios. To capture as much information about the relationships, both contemporaneous and causal, a wide set of models are deployed. Some of these are linear machine learning algorithms that are popular and frequently used in finance and economics. Lastly, a Random Forest Regression is deployed to try and capture non-linear relationships between features and portfolios. The results largely confirm the existing literature: Closed-end fund discounts and changes in the VIX Index are indeed significant predictors of stock market movements. Notably, changes in the VIX Index is a powerful feature to explain stock market movements, even when used as a lone feature. What is new here is the evidence that trader positioning can indeed explain price movements, especially that of stocks with high book-to-market ratios. Moreover, the results show that the causal relationships are not as evident. There are cases where forecast errors are considerably improved, especially for high book-to-market ratio stocks, but due to some important violations of the Classical Linear Regression Model assumptions, any steadfast conclusions are at risk of being spurious.

2 Related Literature

Empirical studies on stock price predictions are among the most popular in financial market research. Economists are divided when it comes to modelling price movements. On one side, you have the proponents of randomness in the prices movements - they don't follow any measurable patterns, and are thus impossible to fully predict. Samuelson (1965) set to prove that stock prices follow a random walk if rational competitive investors require a fixed rate of return. *Efficient capital markets: A review of theory and empirical work* (Malkiel and Fama, 1970) is arguably the most cited paper in the history in financial research after its demonstration that stock prices are indeed close to a random walk.

The opposing view that market movements are rooted in human behavior and our expectations - which are not always rational, has also been well documented, though usually on a more anecdotal basis. For instance, the early 60's were characterized by a high demand for small, newly issued growth stocks (Malkiel, 1999). In pages 55-57, Malkiel mentions a 'new-issue mania' that later led to a decline in growth stocks much stronger than the rest of the market. Investors seemed to have ignored the uncertainty of newly issued electronics stocks in favor of promised overnight returns. On the other hand, De Long et al. (1990) takes a more theoretical approach, and argues that changes in investor sentiment are not fully countered by arbitrageurs and therefore affect security returns. Arbitrageurs are "fully rational" investors who trade to exploit that the price of a security may sometimes deviate from that of its perfect substitute - a portfolio of other securities with the same risk and return profile. For example, if the price of a security is below the price of the substitute portfolio, arbitrageurs will sell the portfolio and buy the security until the prices are back to equal. When the substitute is indeed perfect, this arbitrage is completely riskless. Thus, arbitrageurs have perfectly elastic demand for the security at the price of its substitute portfolio. Such riskless arbitrage is very effective for derivative securities such as futures and options, but for individual stocks and bonds, perfect substitutes are usually not available. Instead, *close* substitutes are traded, which implies that the arbitrageurs' demand is no longer perfectly elastic as there will exist some degree of risk-return differences between the security and its close substitute. In that regard, De Long et al. (1990) argues that arbitrageurs will not be able to completely offset investor sentiment. This "noise trader approach" was further developed upon by other economists. Lee et al. (1991)

examined closed-end funds and proposed that their discount fluctuations are driven by investor sentiment. One thing that both schools of thought agree upon is the difficulty in predicting asset price movements. Many studies aim to provide a rational model directly proving that prices are indeed predictable (De Bondt and Thaler, 1985; Jegadeesh, 1990; Lo and MacKinlay, 1990). However, profiled economists seem to share the opinion that they don't really add more to what we already have in form of the Capital Asset Pricing Model (CAPM). [1]

The Latin statement *'Cum hoc ergo propter hoc'* is a reminder about the logical fallacy we are faced with when drawing conclusions that are exclusively based on observations of two events occurring simultaneously. In English, this is known as 'correlation is not causation'. Granger (1969) neatly provided the framework to test for causality between time series, and is widely recognized as *the* method for investigating causal relationships in econometrics. The basic premise of the paper relied heavily on the notion that cause cannot happen after effect. Consequently, it focused on 'lagged' causality in linear relationships. Granger's definition of causality is as follows: X_t is causing Y_t if we are better able to predict Y_t using all available information U than if the information apart from X_t had been used. Obviously, using 'all available information' is impossible, which in practice means that we instead have to select the relevant information for the time series Y_t. To test for instantaneous causality, we would also include the contemporaneous or new information in the universe U, while lagged causality will only contain information up to and including $t-1$. Due to its simplicity it gained a lot of popularity as well as criticism by economists. The paper was later extended to include frameworks for instantaneous causality as well as tackling some issues and limitations regarding non-linear and deterministic relationships. In an efficient market, the best price forecast of tomorrow's prices are today's prices, since the probability of an increase equals that of a decrease (Malkiel and Fama, 1970). By that reasoning, the presence of causality could be interpreted as evidence of market inefficiencies of the weak form (Niarchos and Alexakis, 1998). However, not all economists agree on this point. Two cointegrated time series as defined in Engle and Granger (1987) implies Granger causality in at least one direction, but has been shown that there is no general equivalence between the existence of arbitrage opportunities and cointegration (Dwyer Jr and Wallace, 1992). In finance, most studies on causality focus on the relationship

[1] Thaler, R.H. and Fama, E. F. (2014, October 18). Chicago Booth Review Interview: https://review.chicagobooth.edu/economics/2016/video/are-markets-efficient

between stock prices and their respective trading volume, where most test for instantaneous causality (Hiemstra and Jones, 1994). However, there are studies in this field relying exclusively on traditional, lagged Granger causality tests (Smirlock and Starks, 1988; Jain and Joh, 1988). Although such tests can be good in explaining linear causal relations, their usefulness in non-linear casual relations is debatable (Hiemstra and Jones, 1994). For this reason, traditional Granger causality tests might overlook significant non-linear patterns that machine learning algorithms may capture.

In recent years, machine learning has practically become a buzzword in finance and economics. However, its applications are to many, perhaps most, still unclear. Nevertheless, machine learning tools are changing the playing field as they are continuously implemented and further advanced by analytic departments across the industries. However, there are limitations to all tools, and the issues associated with financial time series predictions still prevail - the time series are inherently noisy, complex and chaotic (Kumar and Thenmozhi, 2006; Kara et al., 2011; Patel et al., 2015). But that has not stopped researchers from adopting new and experimental techniques in their pursuit. An example of innovative adaptations of machine learning tools in economics is Natural Language Processing (NLP). NLP is used to analyze the effects of web texts for stock market predictions (Das and Chen, 2007; Tetlock, 2007; Tetlock et al., 2008). Words from web articles are collected and transformed into measures of investor sentiment by their positive or negative charges, before they are fed to neural networks along with stock returns. Findings show that market events as reported in financial news articles can indeed be transformed into sentiment measures and used to predict prices. Another example is Patel et al. (2015), which compares four popular ensemble algorithms; Artificial Neural Network (ANN), Support Vector Machine (SVM), Random Forest and Naive Bayes. The study focuses on data pre-processing where continuous data is transformed into discrete data to increase prediction performances. The results show that Random Forest produces the best classification predictions. Tsai et al. (2011) also applies classifier ensemble methods to analyze stock returns. It compares prediction performances using hybrid methods of majority voting and bagging, as well as combining two types of classifier ensembles with single classifiers. The findings are somewhat conflicting, but they do conclude that on average, combining multiple classifiers (e.g. an ensemble of ANNs, random forest classifiers, and logistic regressions) provide better prediction accuracies than a homogenous ensemble of classifiers (e.g. ensemble of ANNs or

random forest classifiers only).

3 Data and Feature Definitions

The collected data sets consist of monthly figures from January 1994 to December 2018. The number of observations varies between 279 to 299, depending on the model and the number of lags used. When no lags are used, all 299 observations are included. All variables are percentage changes. Response variables are already prepared in the Kenneth French Data Library as monthly percentage changes, so no further manipulations are made. The sentiment measures, however, are a mix of index prices and level figures. I compute the log change of all the features' time series. This is to avoid any issues with percentage changes of negative numbers, as most of the features in the Commitment of Traders (COT) data set are net levels that fluctuate on both sides of zero. Now, let's conduct more comprehensive definitions of each variable.

3.1 Response Variables

Following the logic of Lee et al. (1991) I expect that the closed-end fund discount is better at explaining the variance in small stock returns than big stock returns. On the other hand, one could hypothesize that due the aggregate nature of the COT features, they are better at explaining big stock returns. The same logic applies to the CBOE Volatility Index, which is directly derived from the total S&P 500 and not its subsets. Therefore, all regression routines will be performed on a range of stock portfolios to better capture their performance on different subsets of the S&P 500.

The Kenneth French Data Library (KFDL) contains many portfolios formed on sets of characteristics such as size, operating income, investment levels etc. The chosen response variables in this paper are *6 Portfolios Formed on Size and Book-to-Market* from KFDL. These portfolios are constructed at the end of each June. They are the intersections of two portfolios formed on size of market equity (ME) and three portfolios formed on the ratio of book equity (BE) to market equity (BE/ME). The size breakpoint for year t is the median NYSE market equity at the end of June of year t. BE/ME for June of year t is the book equity for the last fiscal year

end in t-1 divided by ME for December of t-1. The BE/ME breakpoints are the 30th and 70th NYSE percentiles. To simplify the regression models, the six portfolios are denoted $responses_i$ for $i \in I$, where $I = \{1, ..., 6\}$.

3.2 Sentiment Features

Closed-End Fund Discount: The difference between a publicly traded fund's price and the net asset value of this fund. This should in theory be close to zero, yet because these funds are traded only secondhand, meaning no new issues or buybacks. The supply of shares is constant, while the demand for these funds is subject to investor sentiment which leads to price-to-nav deviations.

$$cefd_t = \frac{NAV_t - Trading\ Price_t}{NAV_t}, \quad where \quad NAV_t = \frac{Assets_t - Liabilities_t}{Number\ of\ outstanding\ shares_t}.$$

CBOE Volatility Index: A market index that represents the market's expectation of 30-day forward-looking volatility. It is derived from the price inputs of the S&P 500 index options and provides a measure of market risk and investors' sentiment. Implied volatility is calculated by taking the market price of an option, entering it into the Black-Scholes formula, and back-solving for the value of the volatility.

$$vixret_t = \log\left(\frac{VIX_t}{VIX_{t-1}}\right).$$

Historical CFTC Commitment of Traders Report for Financial Futures:
The COT reports provide a breakdown of each Tuesday's open interest for futures and options on futures markets in which 20 or more traders hold positions equal to or above the reporting levels established by the Commodity Futures Trading Commission (CFTC).

The COT reports are based on position data supplied by reporting firms (e.g. FCMs, clearing members, foreign brokers and exchanges). While the position data is supplied by reporting firms, the actual trader category or classification is based on the predominant business purpose self-reported by traders on the CFTC Form 401 and is subject to review by CFTC staff for reasonableness. CFTC then aggregates the data and publishes it every Thursday. Thus, all the figures are of aggregate nature, which simplifies matters for researchers.

The selected features from this data are related to, and inspired by that of Sanders et al. (2004). The names are directly borrowed from that paper, as the intuition behind their construction remains somewhat similar: I want to transform trader positioning into some measure of 'net pressure'. However, their actual definitions are not the same here, as I use the long-to-short ratio as a measure of pressure, whereas the cited paper uses spreads or differences as their measure.

Without further ado, here are the feature definitions for this paper:

OI: Open Interest is the total of all S&P500 futures contracts entered into and not yet offset by a transaction, delivery, or by exercise. The aggregate of all long open interest is equal to the aggregate of all short open interest. Here, I use the log-change in Open Interest as an investor sentiment measure.

$$OI_t = \log\left(\frac{Open\ Interest_t}{Open\ Interest_{t-1}}\right).$$

CPNL: Commercials' Pressure Net Long. Here defined as the log-change in Commercials' long-to-short ratio (LS) on S&P500 futures. Commercial traders are defined by the CFTC as producers, merchants, processors and users of the physical commodity who manage their business risks with use of futures or option markets.

$$CPNL_t = \log\left(\frac{Commercial\ LS_t}{Commercial\ LS_{t-1}}\right).$$

NCPNL: Non-Commercials' Pressure Net Long. Here defined as the log-change in non-commercials' long-to-short ratio on S&P500 futures. Non-commercial traders are defined by the CFTC as professional money managers (e.g. CTAs, CPOs, and hedge funds) as well as a wide array of other Non-commercial (speculative) traders.

$$NCPNL_t = \log\left(\frac{Non\text{-}commercial\ LS_t}{Non\text{-}commercial\ LS_{t-1}}\right).$$

TOTPNL: Total Reportables' Pressure Net Long. Here defined as the log-change in total reportables' long-to-short ratio on S&P500 futures: Total Reportables refers to the sum of futures contracts held by commercial and non-commercial traders registered by CFTC.

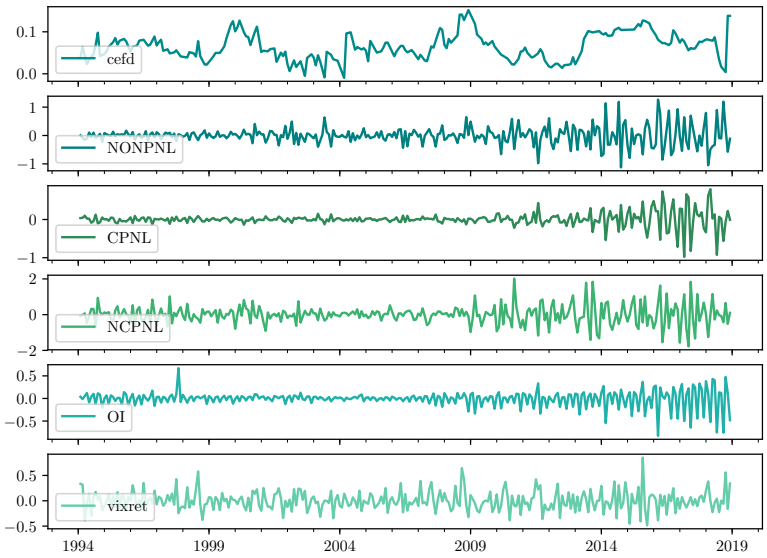
$$TOTPNL_t = \log\left(\frac{Total\ reportables\ LS_t}{Total\ reportables\ LS_{t-1}}\right).$$

NONPNL: Non-Reportables' Pressure Net Long. Here defined as the log-change in non-reportables' long-to-short ratio on S&P500 futures: The sum of non-reportable contracts should always offset the sum total reportables. Because of this duality, only NONPNL is used in the regressions.

$$NONPNL_t = \log\left(\frac{Non\text{-}reportables\ LS_t}{Non\text{-}reportables\ LS_{t-1}}\right).$$

Figure 1 shows the full time series of each sentiment measure. Looking at this, the variance in the COT features seems to be increasing in the recent years. One may also suspect that stationarity in the closed-end fund discount is lacking and may pose an issue. This is addressed in the methodology.

Figure 1: Time series of all sentiment features



4 Methodology

In this paper I want to investigate the relationship between a selected basket of investor sentiment measures and a small range of stock market portfolios. More specifically, I want to test the hypotheses that these features may have explanatory power as well as test whether there exists a causal relationship between the features and portfolios. Moreover, I want to test *what* feature is the best in doing this.

The question regarding investors' rationality is, as previously mentioned, a hard one to answer. For instance, Warren Buffet states that the stock markets are rational most of the time, but he acknowledges that it has its occasional 'seizures' [2]. Investor sentiment is often confused with the irrationality usually associated with those periods of 'seizure' in the sense that it is hard to measure it and it is deeply rooted in human behavior. Consequentially, one may think that sentiment has an insignificant role in explaining price movements over time, and that it should certainly not be able to forecast them. However, recall that an efficient market implies that prices reflect all current and past relevant information, which would include investor sentiment as well. Just because it is hard to measure does not mean it no longer qualifies as information that prices reflect. For this reason, it is important to have a clear definition of what is actually being tested. I am not testing or trying to prove the inefficiency of the market. Instead, I simply look for evidence that investor sentiment can both explain and cause price movements.

To do this, I first have to clear some definitions. The term 'prediction' is used a lot in statistical inference, but it does not have a universal definition. For this thesis, the term prediction will be a general reference to any predictions made from any model. 'Forecasts', on the other hand, will be used when referring to predictions made from lagged time series regressors exclusively. For example, a prediction from a vector autoregressive model would fall under the term 'forecast', since it uses lags of one time series to predict another, but a contemporaneous prediction from an ordinary least square model would not. This distinction is important to avoid any confusion when the models are compared.

The approach can be divided into 3 stages:
In Stage 1 Generalized Least Squares models are deployed to test the features' and their princi-

[2]Buffett, W. E. (2018, February 23). Letter to Shareholders of Berkshire Hathaway Inc.

pal components' explanatory power. Stage 2 tests for causal relationships between features and responses. Stage 3 tests for causal relationships specifically in the feature → portfolio direction, only this time using machine learning algorithms to test for causality - using both features and their principal components as causing variables.

4.1 Univariate Generalized Least Squares

First, let's analyze how the features perform individually. To do this, I first perform a sequence of univariate Generalized Least Squares (GLS) regressions for each feature on each of the six portfolios. By doing this I also obtain information about the covariance structure of the residuals. Below, the portfolios are for simplicity called $response_{i \in I}$, where $I = \{1, 2, ..., 6\}$. Since these regressions are repeated for all six features on all six portfolios, a total of 36 regressions are made.

$$\begin{aligned} response_{it} &= \alpha_i + \beta_1 cefd_t + \epsilon_{it} \quad , \\ response_{it} &= \alpha_i + \beta_1 NONPNL_t + \epsilon_{it} \quad , \\ &\vdots \\ response_{it} &= \alpha_i + \beta_1 vixret_t + \epsilon_{it} \quad . \end{aligned} \tag{1}$$

In traditional univariate OLS regressions we use the observed data $\{y_t, x_t\}_{t=1,2,...,T}$ and minimize the sum of squared residuals of $Y = X\beta + \varepsilon$. Then, so long as the Classical Linear Regression Model (CLRM) assumptions hold, $\mathrm{E}[\varepsilon \mid X] = 0$, and $\mathrm{Cov}[\varepsilon \mid X] = \Omega = \sigma^2 I_n$. An important note is that the constant variance assumption of CLRM implies that the covariance matrix Ω is diagonal and constant $\sigma^2 I$. The estimated coefficients are then be unbiased, consistent and efficient. However, taking a quick glance at the features over time, some of them seem to have time-varying volatility. which can insinuate heteroscedastic variance in the residuals. While the coefficients will remain unbiased and consistent (i.e. the estimated coefficients reflect the true coefficients), they will not be efficient - their standard errors will likely be underestimated. Recall that the main reason for doing these univariate regressions is to paint a realistic picture of the explanatory power of these features. A quick glance at Figure 1 gives reason to suspect heteroscedastic errors, and hence, I perform a White's test to confirm whether this is true or not.

I proceed to use a Generalized Least Squares (GLS) model to better capture the true significance of each feature. The optimization problem is:

$$\hat{\beta} = \underset{b}{\operatorname{argmin}} \, (Y - Xb)^{\intercal} \, \Omega^{-1} (Y - Xb)$$

The procedure begins by factoring the non-diagonal covariance matrix Ω such that $\Omega = CC^{\intercal}$, where $C = \Omega^{-\frac{1}{2}}$. Then the transformation $C^{-1}Y = C^{-1}X\beta + C^{-1}\varepsilon$ will assure constant variance in the residuals. If we let $\varepsilon^* = C^{-1}\varepsilon$ then it can be shown that $\operatorname{Var}[\varepsilon^* \mid X] = C^{-1}\Omega \left(C^{-1}\right)^{\intercal} = I$. The coefficients will remain the same, but will have slightly different standard errors. They are estimated by $\hat{\beta} = \left(X^{\intercal}\Omega^{-1}X\right)^{-1} X^{\intercal}\Omega^{-1}Y$.
This shows that a GLS regression is equivalent to performing an OLS regression on a linearly transformed version of the data, which means that the interpretation of the results will remain the same.

4.2 Multivariate Generalized Least Squares

To examine how the features perform together, I proceed with multivariate GLS regressions. However, this immediately poses an issue: The features are correlated, which means that CLRM's linear independence assumption that $\Pr\big[\, \operatorname{rank}(X) = k_{\text{features}}\big] = 1$ is violated. Measuring joint significance is primarily for comparison with the machine learning approach later and the comparison will not make a lot of sense if these results are inconsistent. There are a few solutions to this multicollinearity issue, but most of them involve making changes to the data, i.e. either cutting out some features or changing the number of observations. Cutting out features would defeat the purpose of our feature selection goal, and increasing the number of observations is not feasible as collecting more data would require new channels and permissions that are not available. Thus, we can either ignore it, use a subset of the data, or transform the feature vectors so that they are orthogonal or linearly independent. This can be done by extracting their principal components and use them as features. Doing that will retain most of the information of the features and simultaneously assure linear independence. Since the primary purpose of these multivariate regressions is to compare with the machine learning approach later, where I will use principal components anyway, I use the principal components here as well. However, I first ignore the collinearity issue and simply bundle all the features together to see how they perform jointly, and then repeat the multivariate regression using the

first three principal components. This will also give an idea of how much the collinearity issue increases the estimated coefficients' standard errors.

This GLS regression routine is also performed on all $i \in I$ portfolios. Note that this routine consists of 12 GLS regressions:

$$response_{it} = \alpha_i + \beta_1 cefd_t + \beta_2 NONPNL_t + \beta_3 CPNL_t + \beta_4 NCPNL_t + \beta_5 OI_t + \beta_6 vixret_t + \epsilon_{it} \tag{2}$$

$$response_{it} = \alpha_i + \beta_1 PC1_t + \beta_2 PC2_t + \beta_3 PC3_t + \epsilon_{it}$$

4.2.1 Principal Components Analysis

Principal Component Analysis (PCA) is a method that is often used to reduce the dimensionality of large data sets. It transforms a set of variables into a smaller one such that it still contains most of the information of the large set. In python, both the Statsmodels module and the Sci-kit Learn (SKlearn) library can be used to perform PCA. For the multivariate regressions, using the statsmodels.PCA function is straightforward and easy: I simply set it to standardize the variables and use the Singular Value Decomposition (SVD) method to obtain the principal components. The SKlearn approach is slightly different, though only because of the data splitting. The formal procedure stays the same: First, the data is standardized such that all vectors have a zero mean and standard deviation equal to one. This is done to assure that the relative scaling of the vectors is uniform across all variables so no variables weigh more than others when the principal components are obtained. Then, SVD is applied to transform the scaled data into principal components and their weights. The mathematical representation of the SVD procedure is as follows:

Let $\Sigma_{n\times p}$ be a diagonal matrix of positive numbers σ_k called the singular values of the feature matrix X. Singular values are equal to the square root of the eigenvalues λ_k of $X^{\intercal}X$. Now, let $U_{n\times n}$ and $W_{p\times p}$ be matrices where all columns are orthogonal vectors. The columns of U are called the left singular vectors of X, while the columns of W are the right singular vectors. U and W are obtained by computing the eigenvectors of $XX^{\intercal}$ and $X^{\intercal}X$, respectively. Then, it can be shown that $X = U\Sigma W^{\intercal}$. After SVD is performed, the principal components are obtained as the vectors of the score matrix $T = XW$.

Again, the only difference between Statsmodels' and Sklearn's approach is that in the SKlearn approach, X will be restricted to a subset of the entire sample of observations called the train set. Thus, the principal components obtained from the SVD will be slightly different from those of the Statsmodels approach, but in both cases, they will contain most of the variance in the features.

4.3 Granger Causality

The previous regressions test the explanatory power of the features where the time indexing is contemporary. Naturally, the data points in the previous regressions were denoted $\{y_t, x_{i,t}\}_{i\in I, t=1,...,299}$, which means that I tested if and how the features *correlate* with the portfolios month by month, rather than test whether they *cause* the variation. Recall that the main argument for using investor sentiment measures as features is the hypothesis that they might affect next month's portfolio returns, so it makes sense to test this with a Granger causality test. The model used to test this is a vector autoregressive model of order p, a VAR(p). The order refers to how many lag variables are used, and is selected by the Akaike Information Criterion (AIC). AIC is a model selection estimator that maximizes the model fit evaluated at the maximum likelihood estimates of the parameters, but penalizes for the number of lags to be used in the model (Akaike, 1974). Rather than selecting the number of lags to use in the model arbitrarily, it select the number of lags while still capturing most of the variation in our data set. If we let $\hat{L}$ be the maximum value of the likelihood function for the model, then the information criterion is defined by

$$\text{AIC} \;=\; 2k - 2\ln(\hat{L}).$$

For this model, I denote the responses as $(y_i)_{i\in I}$ and features $(x_j)_{j\in F}$, where $F = \{1, 2, ..., 6\}$. The estimated VAR(p) model is then,

$$y_t = c + A_1 y_{t-1} + A_2 y_{t-2} + \cdots + A_p y_{t-p} + B_1 x_{t-1} + B_2 x_{t-2} + \cdots + B_p x_{t-p} + e_t, \qquad (3)$$

where each y_i is a column vector with k observations and each A_i is a $k \times k$ matrix. The full

matrix notation is then

$$\begin{bmatrix} y_{1,t} \\ x_{2,t} \\ \vdots \\ x_{k,t} \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_k \end{bmatrix} + \begin{bmatrix} a^1_{1,1} & a^1_{1,2} & \cdots & a^1_{1,k} \\ a^1_{2,1} & a^1_{2,2} & \cdots & a^1_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ a^1_{k,1} & a^1_{k,2} & \cdots & a^1_{k,k} \end{bmatrix} \begin{bmatrix} y_{1,t-1} \\ x_{2,t-1} \\ \vdots \\ x_{k,t-1} \end{bmatrix} + \cdots + \begin{bmatrix} a^p_{1,1} & a^p_{1,2} & \cdots & a^p_{1,k} \\ a^p_{2,1} & a^p_{2,2} & \cdots & a^p_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ a^p_{k,1} & a^p_{k,2} & \cdots & a^p_{k,k} \end{bmatrix} \begin{bmatrix} y_{1,t-p} \\ x_{2,t-p} \\ \vdots \\ x_{k,t-p} \end{bmatrix} + \begin{bmatrix} e_{1,t} \\ e_{2,t} \\ \vdots \\ e_{k,t} \end{bmatrix}.$$

Executing this routine remains somewhat similar; I define our regression data set and use the VAR model provided in the Statsmodels module. This module is very neat and comes with a lot of outputs such as summary statistics and autocorrelation plots. More importantly, I can be confident that the standard errors are robust; It is set up to test for heteroscedasticity and re-run the regressions so that the errors are $\sim N(0, \sigma_k^2)$. Looping through all features successively, then for each iteration, it will compute the optimal lag order according to AIC and run a new VAR(p). However, recall from the definition of Granger causality that the universe of information should contain as much relevant information as possible. CAPM shows that the market risk premium mrp contains most of the relevant information, which justifies including it in our model. Lastly, the VAR module in Statsmodels is equipped with a Granger causality test submodule, which is fairly easy to implement. I simply run the regressions and use their outputs as inputs for the causality test submodule, along with our instructions to test for the isolated effects of adding the sentiment features. It will then perform a traditional F-test to conclude whether the coefficients of the feature lags are jointly significant or not. Table 5 shows the p-values of these tests. Table 6 shows the corresponding results of the causality tests in the opposite direction. That is, whether the portfolio returns affect the sentiment features. In cases where $(x_j)_{j\in F} \to (y_i)_{i\in I}$ AND $(y_i)_{i\in I} \to (x_j)_{j\in F}$, there exists a feedback system, implying that both variables mutually affect each other. In theory, this would indicate a more complex relationship between feature and portfolio, and it would make it difficult to draw any steadfast conclusions.

5 Machine Learning Approach

In python, Statsmodels is generally the most frequently used module for traditional statistical analysis in finance, as it provides many outputs for in-depth analysis, such as summaries, T-stats, log-likelihood and so on. The GLS and VAR regressions earlier were all performed using

Statsmodels classes, as the main goal was to illustrate the relationship between the features and portfolios. However, the downside of this approach is that results may be subject to sampling bias. In other words, the results were all in retrospect and one cannot be confident that the features would perform similarly if they had been analyzed in real-time. Alternatively, one can use SKlearn, which contains a wide array of both supervised and unsupervised learning modules. The major advantage of using machine learning tools is that we can fit the regressions on a subset of the data and test them on a different, 'unseen', subset. This partitioning approach is known as train-test splitting. Moreover, we can validate the models by cross-validation, which will re-split the data in new partitions and conduct the same regressions again with tuned hyperparameters. This way, I can be more confident in how generalized the models really are (Bishop, 2006).

The selected models for this paper are OLS, Ridge, Lasso and Random Forest Regression (RFR). OLS, Ridge and Lasso are linear regression models, similar to GLS and VAR, while Random Forest is an ensemble learning algorithm that is non-parametric and also captures non-linear relationships. Unfortunately, there are no GLS or VAR model algorithms in the SKLearn package (Pedregosa et al., 2011), and constructing them from scratch is beyond the aim of this thesis. For that reason, the existing OLS, Ridge and Lasso regression algorithms are chosen to be compared with the VAR regressions - recall that the main reasons for the machine learning approach here is; 1) To see how the features perform out-of-sample, and 2) investigate which features that contribute the most to the model.

To isolate the effects of adding the features to our model, our strategy will use nested models. That is, I will compare a restricted model with an unrestricted model. For both models, the design matrices only contain lagged time series. Similar to the VAR model, I include as much relevant information as possible and then test whether adding the selected features to that universe will improve the predictions. The market risk premium will thus be a variable in both models. Response matrix Y is defined such that it only contains the temporal series (not lagged) of the respective responses. For the nested model, I restrict it so that it contains the lagged portfolio returns and the lagged market risk premium only. Then, the unrestricted model will also include investor sentiment features along with the lagged responses and market risk premium. Thus, I am effectively performing two Autoregressive Distributed Lag models

(ADL) of order p. Similar to VAR, the lag order is computed by AIC. Lastly, I compare the prediction error variances with and without the features. A more rigorous way to explain this procedure is this:

Let Y be the set containing the real-time portfolio returns so that the partitions $(Y_i)_{i\in I} = \{response_{i,t}\}$ represent each time series. Let $\bar{U}$ be the universe of relevant and available information. Note that the bar indicates that the set only includes past or lagged information. Now let $\bar{X}'$ and $\bar{X}$ be subsets of that universe such that $\bar{X}' \subset \bar{X} \subseteq \bar{U}$. The partitions $(\bar{X}'_i)_{i\in I} \subset \bar{X}'$ only contain lags of the market risk premium as well as lags of the respective responses. More precisely, for any $i \in I$ and including all p lagged series chosen by AIC, we have that $\bar{X}'_i = \{response_{i,t-q}, mrp_{t-q}\}_{q=1}^{p}$. Now, let $\bar{X}_{i,j} \subset \bar{X}$ be a superset of $\bar{X}'_i$ that also contains the lagged time series of our sentiment feature. For simplicity, let's call each one of them $\{feature_j\}_1^6$. Then, including all p lags once again, $\bar{X}_{i,j} = \{response_{i,t-q}, mrp_{t-q}, feature_{j,t-q}\}_{q=1}^{p}$. Finally, I iterate through the portfolios and features as before and regress:

$$Y_i = \bar{X}_i\beta + \epsilon_i \tag{4}$$

$$Y_{i,j} = \bar{X}_{i,j}\beta + \epsilon_{i,j} \tag{5}$$

This will be the setup for all of our machine learning linear regression models. Some important issues should therefore be addressed: First, how do we compare the fit of the models? There are many predefined scoring metrics in SKlearn that are easy to use, such as mean squared error and R^2. Traditionally, Granger causality tests use F-statistics to determine causality and R^2 to measure the fit of the model. However, this ADL approach is not a direct translation of the VAR framework, so we should consider a few important things: R^2 automatically increases with the number of regressors, and since I am comparing two models with different number of regressors, it may provide a poor measure for comparison. For this reason, adjusted R^2 would be better for measuring the causal effects. However, another issue with this approach has to do with the residuals of the regressions. If the expected prediction errors are not $IID(0, \sigma^2)$, the estimated coefficients will be biased (Keele and Kelly, 2006). While Table **??** from the tests shows that the residuals are not significantly correlated, heteroscedasticity seems to be a persisting issue. Thus, the regression coefficients may be inefficient. Since R^2 is only a good measure when all the CLRM assumptions hold, and I cannot be sure that they do, AIC is instead used as the

measure of how well the forecasts are. Similar to Adjusted R^2, AIC measures the goodness of fit while also penalizing for the number of estimated parameters.

5.1 OLS, Ridge, Lasso and Random Forest

SKLearn's LinearRegression algorithm corresponds to the traditional **OLS regression**. It takes no hyperparameters and simply fits a standard linear regression with coefficients $w = (w_1, ..., w_p)$ to minimize the residual sum of squares between the observed targets in the data set, and the targets predicted by the linear approximation. The mathematical representation of the optimizing problem is:

$$\min_{w} \quad ||Xw - y||^2. \tag{6}$$

This method is identical to the methodology in (**??**), except that here β is substituted for w to denote the coefficients in order to make a distinction between the traditional and the machine learning approach. It is worth to mention that the LinearRegression algorithm does not provide any summary statistics except the residual sum of squares. This makes it difficult to perform a Cholesky Decomposition of the covariance matrix, and consequentially I have to accept that some CLRM assumptions will be violated.

Recall that another recurring issue in machine learning algorithms is that fitting the regression to the training set and then predicting on unseen data runs the risk of *overfitting*. Ridge and Lasso regressions address this overfitting issue by imposing a penalty on the size of the coefficients. This is otherwise known as regularization. Instead, the coefficients are obtained by minimizing a *penalized* residual sum of squares. **Ridge regression** uses L2 regularization, and takes on the following optimization problem:

$$\min_{w} \quad ||Xw - y||_2^2 + \alpha||w||_2^2. \tag{7}$$

It resembles the traditional OLS optimization problem, but adds a regularization term. Here, the hyperparameter α is a non-negative number that reflects the strength of the regularization. If α is zero, the coefficients are identical to that of a traditional OLS solution. It can be shown that as α increases, the coefficients asymptotically converge towards zero, resulting in

a near-fully restricted model with horizontal predictions (Hastie et al., 2005, p. 73). **Lasso regression**, on the other hand, uses L1 regularization.

$$\min_{w} \quad \frac{1}{2n}||Xw - y||_2^2 + \alpha||w||_1. \tag{8}$$

Similar to Ridge, α is a hyperparameter, a non-negative number that pushes the optimal coefficients towards zero as it increases towards infinity. An important distinction from Ridge is that Lasso allows the coefficients to be zero. As a result, Lasso has feature selection embedded in its algorithm, whereas Ridge will never completely neglect any features, no matter how irrelevant they are. This follows from the type of norm they use. In 7 and 8 the subscripts denote what type of norm the respective regularization methods use. L2 regularization solves the optimization problem with respect to a *Euclidean distance*, while L1 regularization minimizes with respect to a *Manhattan distance*. The difference between these norms is mathematically summarized as follows:

$$\begin{aligned} ||w||_1 = &\quad ||w_1|| + ||w_2|| + \cdots + ||w_N||, \\ ||w||_2 = &\quad \sqrt{||w_1||^2 + ||w_2||^2 + \cdots + ||w_N||^2}. \end{aligned}$$

It can be shown that it is due to this difference in the norms that makes L1 regularization a tool for feature selection, whereas L2 regularization will only provide an indication for which features are less important (Hastie et al., 2005, p. 73).

Until now, all of the machine learning have been linear models. However, one might argue that there might exist non-linear relationships between these investor sentiment features and portfolio returns. *Bagging*, short for bootstrap aggregation, is a technique used to reduce the variance of an estimated prediction function. Bagging works especially well for procedures with high-variance and low-bias such as decision trees. For random forest regression, I simply fit the same regression tree many times to bootstrap-sampled versions of the training data, and average the result. Fully explaining the random forest algorithm is not the purpose here, as it assumes knowledge on terms such as decision trees, bagging and boosting, but the following is a general outline for the algorithm as provided in (Hastie et al., 2005, p. 588):

1. For decision trees $b = 1$ to B :
   (a) Pick a bootstrap sample $Z*$ of size N from the training data.

(b) Grow a random forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.

i. Select m variables at random from the p variables.
ii. Pick the best variable/split-point among the m.
iii. Split the node into two daughter nodes.

2. Output the ensemble of trees $\{T_b\}_1^B$

Here, I set $B = 500$ trees. This is the only hyperparameter that stays constant throughout the cross-validation. All other hyperparameters, e.g. minimum samples per split, mininum number of samples per leaf, or maximum number of features, are all fed to GridSearchCV in ranges of alternatives. It will then 'tune' the estimators such that it minimizes the squared errors. After training, all $\{T(x;\theta_b)\}_1^B$ are grown, the random forest regression predictor is:

$$\hat{f}(x*) = \frac{1}{B}\sum_{b=1}^{B} T(x*;\theta_b). \tag{9}$$

In short, the forecasts for unseen samples $x*$ are made by averaging the predictions from all the individual regression trees on the unseen test set. Note that θ_b characterizes the set of hyperparameters of random forest tree b. Since random forest regression is non-parametric, the feature selection mechanism is not determined by whether or not coefficients are zero. Instead, it outputs feature importances: At each split in each tree, the improvement in the split criterion, the mean squared error, is a measure of importance attributed to the splitting variable. This is then accumulated over all the trees in the forest for each variable and displayed as weights. For this reason, instead of iterating through each feature and get new outputs for feature importances, it makes more sense to simply feed all the features to the model along with the respective response to be forecasted. However, I want to stay somewhat consistent with the methodology of comparing the forecast errors of two models where one is nested within the other. Thus, I re-define the models slightly:

The nested model will have the same design as before, that is $\bar{X}' = \{response_{i,t-q}, mrp_{t-q}\}_{q=1}^{p}$.
The unrestricted model is simplified so that $\bar{X} = \{response_{i,t-q}, mrp_{t-q}, cefd_{t-q}, ..., vixret_{t-q}\}_{q=1}^{p}$.

Then, I perform the random forest 'equivalent' of the regressions (4). In particular, the forecasts to be compared are:

$$\hat{f}(\bar{X}') = \frac{1}{B}\sum_{b=1}^{B} T(\bar{X}';\theta_b), \tag{10}$$

$$\hat{f}(\bar{X}) = \frac{1}{B}\sum_{b=1}^{B} T(\bar{X};\theta_b). \tag{11}$$

Since the random forest regression is non-parametric, i.e. there are no coefficients to be obtained and penalized for as in AIC, the forecasts are measured by the mean squared error. I tried to construct a pseudo-measure for AIC by penalizing on the size of the forest instead of the number of coefficients, but the lack of literature on the matter makes it hard to justify. Figure **??** compares the forecasts of the nested model and the full unrestricted model.

5.2 Training, Cross-validation and Testing

Now that we have an understanding of the models, how do we deploy them? How should we split the data and implement the models?

To my understanding, it seems that most of the machine learning literature deals with classification rather than time series regression. Classification problems and cross-sectional analysis allow for splitting methods where the data is randomly shuffled before train and test indices are split. However, for time series regressions, it wouldn't make a lot of sense to train the model on sets without respect to sequentiality. For example, suppose we have a small training set containing only observations on February 1994, December 2001, and March 2016. Now, let our test set contain observations on March 2005 and October 2007. If we fit the model to the training set, we would train it using observations many years ahead of the observations in the test set. This is not realistic, nor feasible in real life, so we should therefore use a different approach. The TimeSeriesSplit function provides train/test indices to split time series data samples that are observed at fixed time intervals. In each split, test indices must be higher (i.e. more recent) than before. After experimenting with different numbers of splits and strategies, the best results were obtained using a grid search for cross-validation and hyperparameter tuning and then test on a different subset. This might be somewhat confusing, so let's explain this

a little further: First, we set TimeSeriesSplit to make five splits on the entire data set. The data will then be split into five pairs of train and test sets. Suppose one of these train sets contains observations 50 to 150, and that it is complemented with a test set containing observations 150 to 180. We then use the GridSearchCV function within this train set, and feed it with an array of hyperparameters. Grid search cross-validation will then use all combinations between hyperparameters across five new subsets *within* this train set and store the best combination - the one that achieves the highest score. This combination of hyperparameters will then be used to make predictions on the unseen test set. That is, a forecast of observations 150 to 180. This is especially important for the Random Forest regression as it takes many hyperparameters, which would have been very difficult to determine manually.

6 Results

Due to the large number of regressions, interpreting the results one by one will not be done here. Instead, summaries of the results is shown in Tables 1.

Starting with the univariate GLS (Table 1): The coefficients of each regression tells us something about the directional relationship between each feature and the portfolios, on average. The asterisks denote the level of significance each estimated coefficient has. A positive coefficient indicates that the feature varies *with* the portfolio returns, while a negative one indicates that the feature varies *against* the portfolio returns. For example, the negative coefficients for $cefd$ indicates that an *increase* in the closed-end fund discounts, the portfolio returns would *decrease* on average. This is consistent with Lee et al. (1991), and strengthens the hypothesis that closed-end fund discounts widen when the stock market is performing poorly and vice versa. However, note that the coefficients for $cefd$ seem to be slightly more significant for large stocks than for small stocks, which is not what the paper shows. One possible explanation for that has to do with the definition of $cefd$. Here, it is an equal weight average of all closed-end fund discounts in the database, as opposed to using market sizes as weights like they do in the paper. Thus, the discount as defined here may not capture the proper size-corresponding relationships. Another take-away from this figure is the obvious power of $vixret$ - the change in 'the fear guage' seems to be the best in explaining the variation of relatively risky equity-only portfolios. It suggests that when the VIX Index is increasing, risk aversion turns investors away

from the stock market and position themselves in other markets. As for the COT features, the commercial and non-commercial traders' aggregate positioning both seem to be significant in explaining the variation in the portfolio returns. As expected, the commercials or 'hedgers', increase their net long position when stock market is falling, whereas the non-commercials, i.e. leveraged money funds, increase their net long position when the markets rising. This makes intuitive sense. Moreover, the results show that trader positioning better explains the change in stocks with high book-to-market ratios than low book-to-market ratios.

The results are not as clear to interpret when it comes to the joint regressions. As mentioned, the features are correlated and that comes with some covenants. First, the $X^\top X$ matrix may not be invertible, which prevents one from estimating the coefficients all together. Now, in this case, the algorithm is able to obtain the inverse, but its values could be inaccurate and we have no means to assess this. Assuming it is accurate, another problem with multicollinearity is that the standard errors of the estimated coefficients will be overestimated, which exposes the model to make Type 2 errors (Brooks, 2019, p. 170-175). Thus, the coefficients may come out as insignificant when in fact they are not. This can perhaps be what is seen when comparing the significance levels in Tables 1 and 2. Another related consequence of this, is that the R^2_{adj} of 34.6% measure is also likely to be overestimated. Using the features' principal components instead fully removes the collinearity issue and nearly doubles the fit of the model, to 68.1%, despite having used only three variables.

Interpreting the coefficients in a VAR model is difficult, since all variables are endogenous (Johansen, 2005). To avoid making spurious claims about these relationships, we refrain from concluding anything other than whether or not a causal relationship is found. The VAR model results in Table 5 show that the causal relationships between the selected investor sentiment features and portfolio returns may exist, but the evidence is underwhelming. Of the 36 causality tests in the feature $\rightarrow$ response direction, only 4 tests reveal evidence of a causal relationship, where 2 of those are significant on a 5% level. However, it is worth to note that all 4 are COT features, and that they seem to only affect stocks with a high book-to-market ratio. As for causality in the response $\rightarrow$ feature direction, Table 6 shows evidence of 5 causal relationships: None of which suggest a feedback system. That is, there does not seem to exist a bi-directional causal relationship between any of these features and portfolios.

For the machine learning approach, we will not have any 'yes' or 'no' answers on whether the results show evidence of causality. Even though F-tests are performed on each lagged variable, the interpretation of these remains difficult. It is the joint regression that would be of interest, but even then, the violations of some important CLRM assumptions makes any strong claims questionable. Instead, we have to settle with comparing AIC of the forecasts with and without the sentiment features. Figures 2 to 4 display the final splits' AIC values for the OLS, Ridge and Lasso regressions. In Figures 2 and 3, a total of 36 regressions are made, one for each sentiment feature on each portfolio. Note that the regressions that are made without including sentiment features are repeated 6 times due to the pipeline design. This makes it easier to compare each features' significance - each 'without feature'-forecast result is adjacent to its corresponding 'with feature'-forecast result. Note that a lower AIC value indicates a better forecast. Comparing these AIC values with the AIC values from the traditional VAR model done earlier, these results are generally higher. There might be several reasons for this, but the first obvious reason is that the AIC computed here is calculated on unseen data, whereas the AIC values from the VAR are computed on the 'training' data. Another reason for this has to do with the robustness of the regressions. The errors are, as mentioned, not always normally distributed with mean zero and constant variance. Nevertheless, there is an emerging pattern in Figures 2 and 3. The third feature, $CPNL$, seems to improve the AIC considerably on all forecasts of portfolio returns. In addition, the fifth feature, OI, improves the forecast on small stocks with a low book-to-market ratio. This is somewhat consistent with the traditional VAR model, where $NONPNL$, OI and $CPNL$ were the only features whose test results rejected the null-hypothesis of Granger non-causality. The features' principal components, however, do not seem to improve the forecasts very much unless they are bundled together as in the random forest regression. Figure 5 shows that for all portfolios, except the large stocks with medium book-to-market ratio, adding the features decreases the mean squared error of the forecasts, though with unconvincing magnitude. The feature importances in Figure 6 show that the features get an average of 32.5% of the weights.

6.1 Conclusion

This paper investigated both linear and non-linear relationships between a selected set of investor sentiment features and a range of stock portfolio returns. While some of the implemented

models are fairly straightforward to implement, the models using machine learning algorithms had to be partly constructed. This discrepancy in design may have suppressed the goal of achieving one-to-one comparisons between the traditional models and the machine learning models. Regardless, the results from the machine learning approach do stay relatively consistent with their traditional counterparts. Specifically, the results show that when it comes to explaining the variation of stock portfolio returns, four features are particularly significant: $vixret$, $NCPNL$, $CPNL$ and $cefd$. For a 95% confidence interval or higher, all of their coefficients are significant in explaining changes in small stocks with high book-to-market ratios. The first three are also significant on a 99% confidence interval for large stocks with high book-to-market ratio. When considered jointly, the collinearity issue seems to be too strong to draw any good conclusion. Instead, what can be deduced from the multivariate regressions is that the first three principal components of the features are highly significant in explaining the variation of all the portfolios. In some cases, the explained variation reaches levels above 85%. This should provide sufficient evidence to answer the question on whether there exists a contemporaneous linear relationship between some of these investor sentiment features and stock portfolios, and should indicate that further research on this topic is needed.

As for causal relationships, the results show that some COT features indeed affect the following portfolio returns. From the causality test conclusions in the traditional VAR(p) model, $NONPNL$, $CPNL$ and OI show significant F-statistics. This suggests that these features do indeed Granger cause some returns, particularly on stocks with high book-to-market equity. In the machine learning approach, the same features show some or even substantial improvements in forecasting portfolio returns, relative to omitting them. In particular, commercial traders' positioning on the futures market (read: hedgers' positioning) seems to affect stock portfolio returns the most. The open interest, seems to As for the principal components, they once again do not seem to contribute much to the model. Only in the case of small stocks with high book-to-market do they seem to reduce the mean squared error of the forecast, though not by much.

Appendices

A

Contemporeneous Relationships

Table 1: Univariate GLS:

	cefd	NONPNL	CPNL	NCPNL	OI	vixret
SMALLLoBM	-0.1999	0.0148	-0.0393*	0.0119*	-0.0367*	-0.1897***
ME1BM2	-0.1772*	0.0124	-0.0347**	0.0124**	-0.0260	-0.1560***
SMALLHiBM	-0.2300**	0.0112	-0.0337**	0.0129***	-0.0246	-0.1466***
BIGLoBM	-0.1357*	0.0106	-0.0234*	0.0074*	-0.0001	-0.1414***
ME2BM2	-0.2189***	0.0137*	-0.0310***	0.0106***	-0.0078	-0.1358***
BIGHiBM	-0.1798*	0.0162*	-0.0393***	0.0137***	-0.0128	-0.1381***
* $\Rightarrow \alpha = 10\%$	** $\Rightarrow \alpha = 5\%$	*** $\Rightarrow \alpha = 1\%$				

Table 2: Multivariate GLS: Features

	cefd	NONPNL	CPNL	NCPNL	OI	vixret
SMALLLoBM	-0.1018	-0.0011	-0.0203	0.0032	-0.0122	-0.1858***
ME1BM2	-0.0981	0.0009	-0.0146	0.0061	-0.0035	-0.1526***
SMALLHiBM	-0.1546	0.0018	-0.0125	0.0073	-0.0032	-0.1429***
BIGLoBM	-0.0518	0.008	-0.0057	0.0043	0.0224**	-0.1406***
ME2BM2	-0.1422**	0.0099	-0.0059	0.0071	0.0156	-0.1332***
BIGHiBM	-0.0951	0.0108	-0.0105	0.0094*	0.015	-0.1356***
Mean R^2_{adj}	0.3458					

Table 3: Feature Correlation Matrix

	cefd	NONPNL	CPNL	NCPNL	OI	vixret
cefd	1.000	-0.030	0.040	0.022	-0.039	0.084
NONPNL	-0.030	1.000	-0.748	0.097	-0.336	-0.051
CPNL	0.040	-0.748	1.000	-0.465	0.374	0.069
NCPNL	0.022	0.097	-0.465	1.000	-0.231	-0.072
OI	-0.039	-0.336	0.374	-0.231	1.000	0.084
vixret	0.084	-0.051	0.069	-0.072	0.084	1.000

Table 4: Multivariate GLS: PCs

	PC_1	PC_2	PC_3
SMALLLoBM	0.8571***	-0.6255***	-0.0836***
ME1BM2	0.6329***	-0.3807***	0.1049***
SMALLHiBM	0.6174***	-0.3406***	0.1964***
BIGLoBM	0.4862***	-0.3696***	0.1652***
ME2BM2	0.4400***	-0.2179***	0.3652***
BIGHiBM	0.4922***	-0.1874***	0.4837***
Mean R^2_{adj}	0.6806		

B

Causal Relationships: Traditional

Table 5: Causality test p-values. Rows,columns represent causing and caused variables, respectively.

	SMALLLoBM	ME1BM2	SMALLHiBM	BIGLoBM	ME2BM2	BIGHiBM
cefd	0.3514	0.2841	0.218	0.1911	0.101	0.1961
NONPNL	0.4709	0.3731	0.0138**	0.1506	0.4355	0.1705
CPNL	0.3873	0.1705	0.0812*	0.682	0.2853	0.0953*
NCPNL	0.5308	0.5911	0.2585	0.1852	0.3298	0.2232
OI	0.7423	0.5092	0.0326**	0.266	0.1238	0.7043
vixret	0.4707	0.244	0.2753	0.5519	0.638	0.9487

Table 6: Causality test p-values. Rows,columns represent causing and caused variables, respectively.

	cefd	NONPNL	CPNL	NCPNL	OI	vixret
SMALLLoBM	0.4071	0.2635	0.202	0.4381	0.0258**	0.1719
ME1BM2	0.353	0.32	0.0849*	0.4831	0.6176	0.7566
SMALLHiBM	0.6641	0.2209	0.1881	0.1764	0.8114	0.8686
BIGLoBM	0.4508	0.6123	0.1592	0.0151**	0.7866	0.9696
ME2BM2	0.6381	0.0985*	0.07*	0.2011	0.839	0.16
BIGHiBM	0.8997	0.8359	0.62	0.653	0.3822	0.7426

C

Causal Relationships: Machine Learning

Figure 2: AIC with OLS

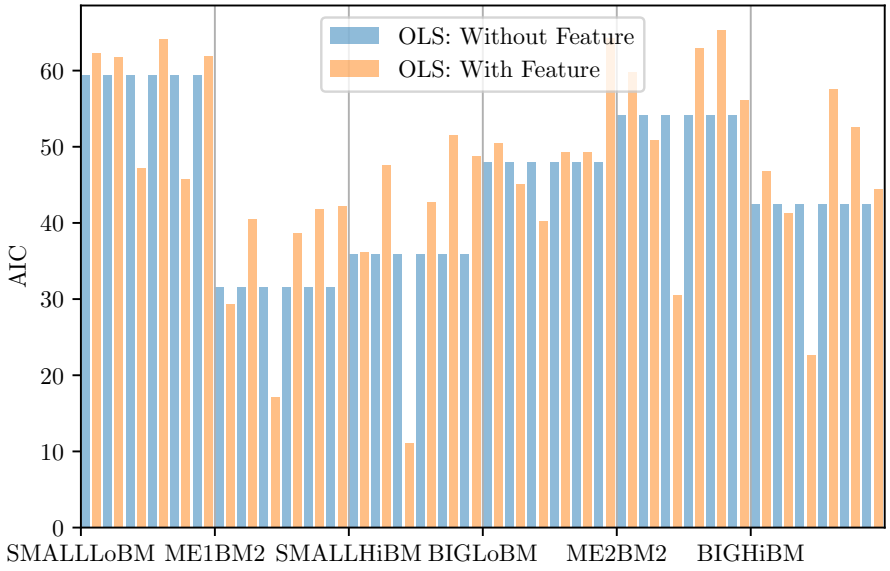


Figure 3: AIC with Ridge and Lasso

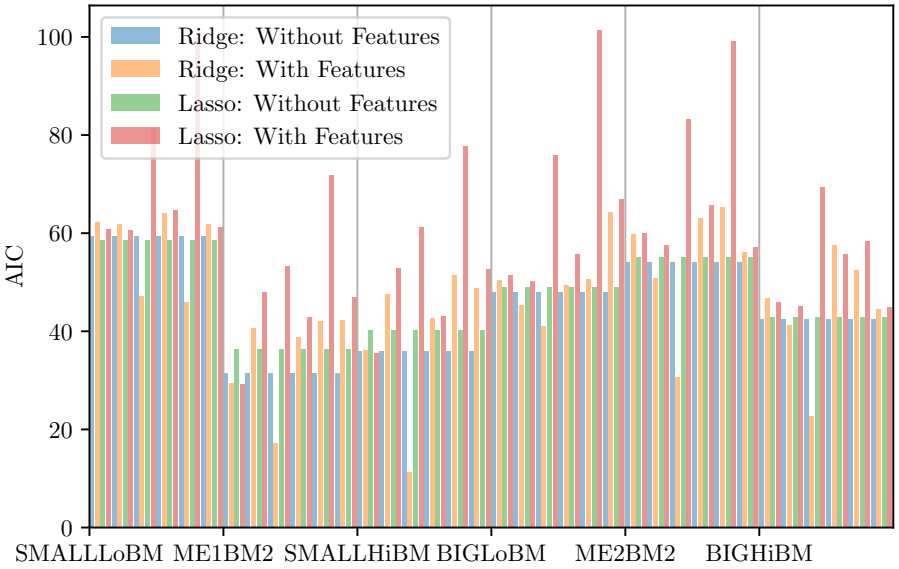


Figure 4: AIC with PCs: OLS, Ridge and Lasso

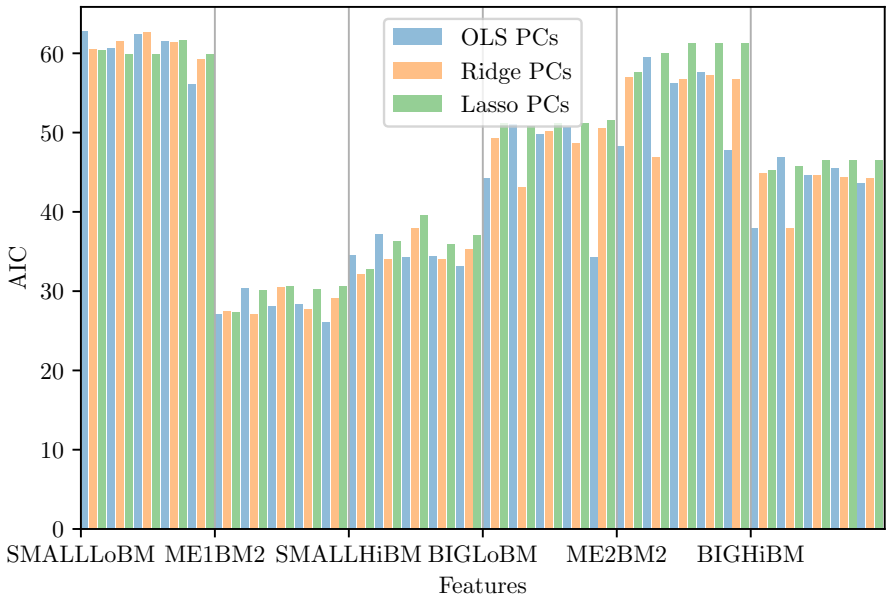


Figure 5: Mean Squared Error: Random Forest

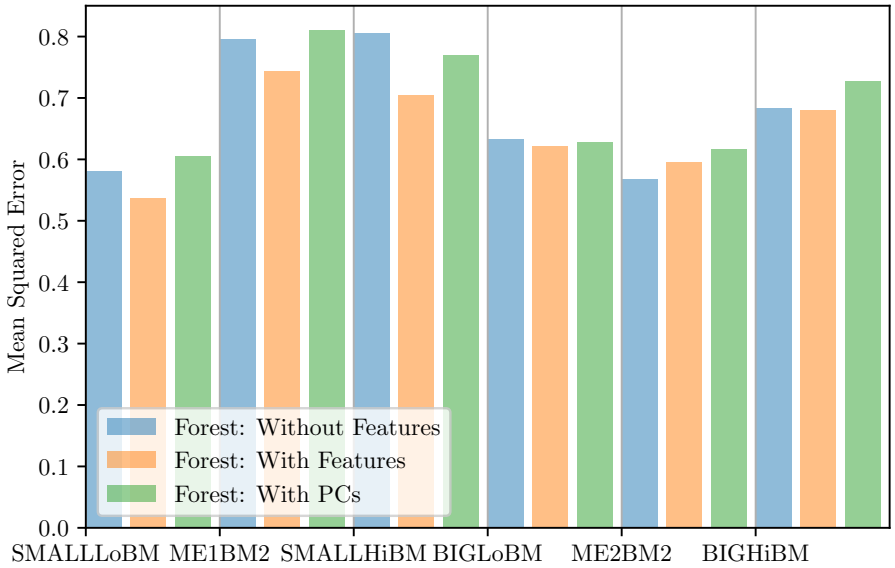
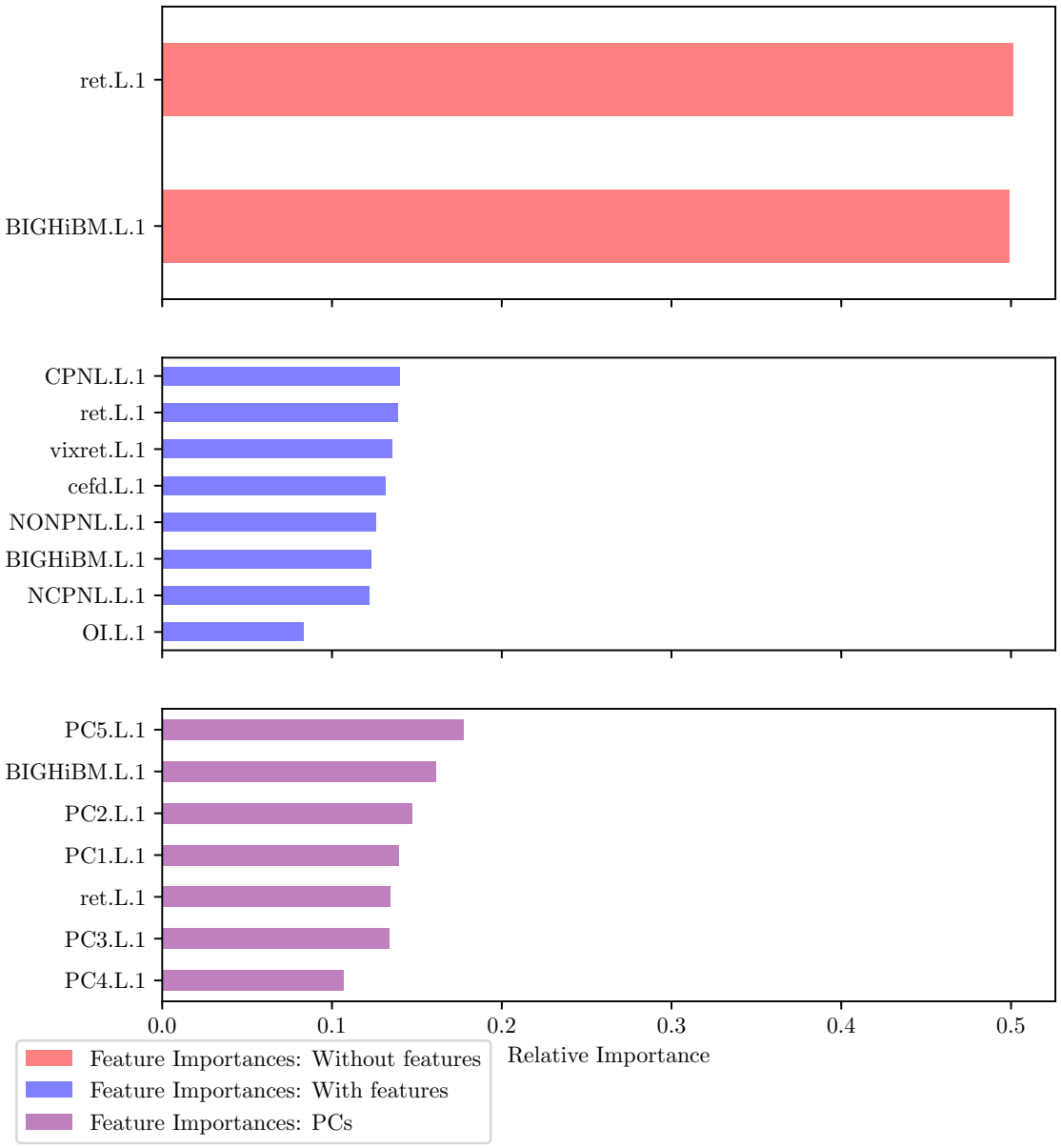


Figure 6: Feature Importances: Random Forest


ret.L.1
BIGHiBM.L.1
CPNL.L.1
ret.L.1
vixret.L.1
cefd.L.1
NONPNL.L.1
BIGHiBM.L.1
NCPNL.L.1
OI.L.1
PC5.L.1
BIGHiBM.L.1
PC2.L.1
PC1.L.1
ret.L.1
PC3.L.1
PC4.L.1
0.0
0.1
0.2
0.3
0.4
0.5
Relative Importance
Feature Importances: Without features
Feature Importances: With features
Feature Importances: PCs

References

Akaike, H. (1974). A new look at the statistical model identification. In *Selected Papers of Hirotugu Akaike*, pages 215–222. Springer.

Bekaert, G. and Hoerova, M. (2014). The vix, the variance premium and stock market volatility. *Journal of Econometrics*, 183(2):1–5.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Brooks, C. (2019). *Introductory econometrics for finance*. Cambridge university press.

Das, S. R. and Chen, M. Y. (2007). Yahoo! for amazon: Sentiment extraction from small talk on the web. *Management science*, 53(9):1375–1388.

De Bondt, W. F. and Thaler, R. (1985). Does the stock market overreact? *The Journal of finance*, 40(3):793–805.

De Long, J. B., Shleifer, A., Summers, L. H., and Waldmann, R. J. (1990). Noise trader risk in financial markets. *Journal of political Economy*, 98(4):703–738.

Dwyer Jr, G. P. and Wallace, M. S. (1992). Cointegration and market efficiency. *Journal of International Money and Finance*, 11(4):318–327.

Engle, R. F. and Granger, C. W. (1987). Co-integration and error correction: representation, estimation, and testing. *Econometrica: journal of the Econometric Society*, pages 251–276.

Granger, C. W. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, pages 5–30, 110–200.

Hastie, T., Tibshirani, R., Friedman, J., and Franklin, J. (2005). The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):61–73.

Hiemstra, C. and Jones, J. D. (1994). Testing for linear and nonlinear granger causality in the stock price-volume relation. *The Journal of Finance*, 49(5):1639–1664.

Jain, P. C. and Joh, G.-H. (1988). The dependence between hourly prices and trading volume. *Journal of Financial and Quantitative Analysis*, 23(3):269–283.

Jegadeesh, N. (1990). Evidence of predictable behavior of security returns. *The Journal of finance*, 45(3):881–898.

Johansen, S. (2005). Interpretation of cointegrating coefficients in the cointegrated vector autoregressive model. *Oxford Bulletin of Economics and Statistics*, 67(1):1–10.

Kara, Y., Boyacioglu, M. A., and Baykan, Ö. K. (2011). 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):5311–5319.

Keele, L. and Kelly, N. J. (2006). Dynamic models for dynamic theories: The ins and outs of lagged dependent variables. *Political analysis*, 14(2):186–205.

Kumar, M. and Thenmozhi, M. (2006). Forecasting stock index movement: A comparison of support vector machines and random forest. In *Indian institute of capital markets 9th capital markets conference paper*.

Lee, C. M., Shleifer, A., and Thaler, R. H. (1991). Investor sentiment and the closed-end fund puzzle. *The journal of finance*, 46(1):4–75.

Lo, A. W. and MacKinlay, A. C. (1990). When are contrarian profits due to stock market overreaction? *The review of financial studies*, 3(2):175–205.

Malkiel, B. G. (1999). *A random walk down Wall Street: including a life-cycle guide to personal investing*. WW Norton & Company.

Malkiel, B. G. and Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The journal of Finance*, 25(2).

Niarchos, N. A. and Alexakis, C. A. (1998). Stock market prices,'causality'and efficiency: evidence from the athens stock exchange. *Applied Financial Economics*, 8(2):167–174.

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

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D.,

Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Samuelson, P. A. (1965). Proof that properly anticipated prices fluctuate randomly. *Management Review*, 6(2).

Sanders, D. R., Boris, K., and Manfredo, M. (2004). Hedgers, funds, and small speculators in the energy futures markets: an analysis of the cftc's commitments of traders reports. *Energy Economics*, 26(3).

Smirlock, M. and Starks, L. (1988). An empirical analysis of the stock price-volume relationship. *Journal of Banking & Finance*, 12(1):31–41.

Tetlock, P. C. (2007). Giving content to investor sentiment: The role of media in the stock market. *The Journal of finance*, 62(3):1139–1168.

Tetlock, P. C., Saar-Tsechansky, M., and Macskassy, S. (2008). More than words: Quantifying language to measure firms' fundamentals. *The Journal of Finance*, 63(3):1437–1467.

Tsai, C.-F., Lin, Y.-C., Yen, D. C., and Chen, Y.-M. (2011). Predicting stock returns by classifier ensembles. *Applied Soft Computing*, 11(2):2452–2459.