FUNDAÇÃO GETULIO VARGAS
ESCOLA DE ECONOMIA DE SÃO PAULO

DANIEL CUNHA OLIVEIRA

EVALUATING GOOGLE TRENDS DATA TO THE TASK OF PREDICTING STOCK RETURNS

SÃO PAULO
2021

DANIEL CUNHA OLIVEIRA

EVALUATING GOOGLE TRENDS DATA TO THE TASK OF PREDICTING STOCK RETURNS

Dissertação apresentada ao Programa de Mestrado da Escola de Economia de São Paulo da Fundação Getulio Vargas, como requisito para a obtenção do título de Mestre em Economia.

Área de concentração:
Engenharia Financeira.

Orientador:
Prof. Dr. Pedro Luiz Valls Pereira
Coorientador:
Prof. Dr. André Fujita

SÃO PAULO

2021

Oliveira, Daniel Cunha.
Evaluating Google Trends data to the task of predicting stock returns / Daniel Cunha Oliveira - 2021.
73 f.

Orientador: Prof. Dr. Pedro Luiz Valls Pereira.
Co-orientador: Prof. Dr. André Fujita.
Dissertação (mestrado profissional MPFE)– Fundação Getulio Vargas, Escola de Economia de São Paulo.

1. Ações (Finanças) - Preços - Previsão. 2. Análise de séries temporais. 3. Aprendizado do computador. I. Pereira, Pedro L. Valls. II. Fujita, André. III. Dissertação (mestrado profissional MPFE) – Escola de Economia de São Paulo. IV. Fundação Getulio Vargas. V. Evaluating google trends data to the task of predicting stock returns.
CDU 336.763.2

Ficha Catalográfica elaborada por: Isabele Oliveira dos Santos Garcia CRB SP-010191/O

Biblioteca Karl A. Boedecker da Fundação Getulio Vargas - SP

DANIEL CUNHA OLIVEIRA

EVALUATING GOOGLE TRENDS DATA TO THE TASK OF PREDICTING STOCK RETURNS

Dissertação apresentada ao Programa de Mestrado da Escola de Economia de São Paulo da Fundação Getulio Vargas, como requisito para a obtenção do título de Mestre em Economia.

Área de concentração:
Engenharia Financeira.

Data da Aprovação: 06/08/2021

Banca Examinadora:

Prof. Dr. Pedro Luiz Valls Pereira
(Orientador)
EESP-FGV

Prof. Dr. André Fujita
(Co-orientador)
IME-USP

Prof. Dr. Diogo de Prince Mendonça
UNIFESP

ACKNOWLEDGMENTS

I thank professor Pedro Valls and professor Andre Fujita for their patience and guidance through the research project. I thank Felipe Salvatore for all the key insights and code review he has provided me. I thank my family and wife for the emotional support during the process. Finally, I thank my colleagues at the Getulio Vargas Foundation and at Fujita's research group.

ABSTRACT

The problem of predicting financial assets returns is one of the main problems of the empirical finance literature. In particular one of it's main challenges is to evaluate the usefulness of the so called alternative data to this task. One of the most common alternative datasets is Google Trends data which have gained popularity in recent years. In this work we want to evaluate the usefulness of this data to the task of predicting U.S. stock indices returns. To achieve this goal we break up the problem in two steps: first we employ feature selection methods, and second we employ forecasting models. We use 15 feature selection methods and 10 forecasting models to achieve this goal. In contrast to what the literature have found, we do not found evidence that the Google Trends data contributes to predict the returns of the stock indices in question. The conclusions seems to be robust across feature selection methods, forecasting models, accuracy and risk and return metrics.

Keywords: Asset pricing. Forecasting. Google trends. Machine learning. Time series.

RESUMO

A previsão dos retornos de ativos financeiros é um dos principais problemas da literatura de finanças empíricas. Em particular, um dos desafios atuais da literatura é avaliar a utilidade dos chamados dados alternativos para esta tarefa. Um dos dados mais comuns caracterizados como tal são os dados do Google Trends, e este tem alcançado popularidade elevada na literatura. Neste trabalho pretendemos avaliar a utilidade dos dados do Google Trends para a tarefa de previsão de índices de ações americanos. Para atingir este objetivo, separaremos o problema de previsão em duas etapas: primeiro a etapa de seleção de covariaveis, e segundo a etapa de previsão. Utilizamos 15 métodos de seleção de features e 10 métodos de previsão. Ao contrário do que a literatura anterior relatou, nós não encontramos evidencia de que os dados do Google Trends contribui para prever os retornos dos índices de ações estudados. As conclusões parecem ser consistentes entre modelos de seleção de covariaveis, modelos de previsão, e em relação a medidas de acurácia e de risco e retorno.

Palavras-chave: Precificacao de ativos. Previsao. Google Trends. Aprendizado de maquina. Series Temporais.

LIST OF FIGURES

Figure 1 – Cumulative returns for $1 invested at the beginning of the sample for all U.S. market indices used in our study. 38
Figure 2 – Rolling mean (60 days) of Google Trends search volume for the word "financial" for different samples. The word was sampled across a week on different times along the day. 40
Figure 3 – For a given level of AUC, we have simulated different strategies for the S&P 500 and computed it's corresponding Sharpe ratio. The variance is induced by the fact that the order that the corresponding hit rate of a given level AUC have been realized is important for the final Sharpe ratio. 46
Figure 4 – Boxplot for the out-of-sample AUC aggregating across all U.S. stock indices and forecasting models. 48
Figure 5 – Boxplot for the out-of-sample AUC aggregating across all U.S. stock indices and feature selection methods. 49
Figure 6 – Boxplot for the out-of-sample AUC for the S&P 500 index aggregating across all feature selection methods. 49
Figure 7 – Boxplot for the out-of-sample AUC for the S&P 500 index aggregating across all forecasting models. 50
Figure 8 – Cumulative returns of $1 unit of money invested in a strategy that times the S&P 500 based on the predictions of each of the forecasting models and given the method ALL as the feature selection method. 51
Figure 9 – Out-of-sample cummulative returns for each forecasting model given the feature selection method ALL. 61
Figure 10 – Out-of-sample cummulative returns for each forecasting model given the feature selection method SFI. 62
Figure 11 – Out-of-sample cummulative returns for each forecasting model given the feature selection method MDI. 62
Figure 12 – Out-of-sample cummulative returns for each forecasting model given the feature selection method MDA. 63
Figure 13 – Out-of-sample cummulative returns for each forecasting model given the feature selection method GRANGER. 63
Figure 14 – Out-of-sample cummulative returns for each forecasting model given the feature selection method HUANG. 64
Figure 15 – Out-of-sample cummulative returns for each forecasting model given the feature selection method IAMB. 64
Figure 16 – Out-of-sample cummulative returns for each forecasting model given the feature selection method MMMB. 65

Figure 17 – Out-of-sample cummulative returns for each feature selection method given the forecasting model LOGIT. 65
Figure 18 – Out-of-sample cummulative returns for each feature selection method given the forecasting model RIDGE. 66
Figure 19 – Out-of-sample cummulative returns for each feature selection method given the forecasting model LASSO. 66
Figure 20 – Out-of-sample cummulative returns for each feature selection method given the forecasting model ENET. 67
Figure 21 – Out-of-sample cummulative returns for each feature selection method given the forecasting model RF. 67
Figure 22 – Out-of-sample cummulative returns for each feature selection method given the forecasting model GB. 68
Figure 23 – Out-of-sample cummulative returns for each feature selection method given the forecasting model NN3. 68

LIST OF TABLES

Table 1 – Example of features within each of the groups defined by Curme et al. (2014) 39
Table 2 – Hyperparameter space for penalized regression models 44
Table 3 – Hyperparameter space for the Random Forest regression 44
Table 4 – Hyperparameter space for the Gradient Boosting 45
Table 5 – Hyperparameter space for the three hidden layer Neural Network 45
Table 6 – Median in-sample AUC over all stock indices for all combinations of Feature Selection method with the Forecasting methods. 48
Table 7 – Median out-of-sample AUC over all stock indices for all combinations of Feature Selection method with the Forecasting methods. 48
Table 8 – Median out-of-sample Sharpe ratio over all stock indices for all combinations of Feature Selection methods with Forecasting methods. To put the results into context, the median Sharpe ratio for each strategy based on the combination of models that times the portfolio of $\frac{1}{12}$ units of each stock index is **0.881**. . . 52
Table 9 – Median out-of-sample Sharpe ratio for all combinations of Feature Selection methods and Forecasting methods for a given stock index. To put the results into context, the median Sharpe ratio for the buy-and-hold of the S&P 500 is **0.914**. 52
Table 10 – Median out-of-sample Sharpe ratio for all combinations of Feature Selection methods and Forecasting methods for a given stock index. To put the results in context, the Sharpe for the buy-and-hold for the S&P Communications index is 1.057. The Sharpe values in bold are better than the buy-and-hold 69
Table 11 – Median out-of-sample Sharpe ratio for all combinations of Feature Selection methods and Forecasting methods for a given stock index. To put the results in context, the Sharpe for the buy-and-hold fot he S&P Technologies index is 1.25. The Sharpe values in bold are better than the buy-and-hold. 70
Table 12 – Median out-of-sample Sharpe ratio for all combinations of Feature Selection methods and Forecasting methods for a given stock index. To put the results in context, the Sharpe for the buy-and-hold fot he S&P Consumer index is 1.193. The Sharpe values in bold are better than the buy-and-hold. 70
Table 13 – Median out-of-sample Sharpe ratio for all combinations of Feature Selection methods and Forecasting methods for a given stock index. To put the results in context, the Sharpe for the buy-and-hold fot he S&P Utilities index is 0.673. The Sharpe values in bold are better than the buy-and-hold. 70

Table 14 – Median out-of-sample Sharpe ratio for all combinations of Feature Selection methods and Forecasting methods for a given stock index. To put the results in context, the Sharpe for the buy-and-hold fot he Nasdaq index is 1.083. The Sharpe values in bold are better than the buy-and-hold. 71
Table 15 – Median out-of-sample Sharpe ratio for all combinations of Feature Selection methods and Forecasting methods for a given stock index. To put the results in context, the Sharpe for the buy-and-hold fot he S&P Industrials index is 0.916. The Sharpe values in bold are better than the buy-and-hold. 71
Table 16 – Median out-of-sample Sharpe ratio for all combinations of Feature Selection methods and Forecasting methods for a given stock index. To put the results in context, the Sharpe for the buy-and-hold fot he S&P 500 index is 0.914. The Sharpe values in bold are better than the buy-and-hold. 71
Table 17 – Median out-of-sample Sharpe ratio for all combinations of Feature Selection methods and Forecasting methods for a given stock index. To put the results in context, the Sharpe for the buy-and-hold fot he S&P Financials index is 0.715. The Sharpe values in bold are better than the buy-and-hold. 72
Table 18 – Median out-of-sample Sharpe ratio for all combinations of Feature Selection methods and Forecasting methods for a given stock index. To put the results in context, the Sharpe for the buy-and-hold fot he S&P Consumer Cyclical index is 0.794. The Sharpe values in bold are better than the buy-and-hold. . 72
Table 19 – Median out-of-sample Sharpe ratio for all combinations of Feature Selection methods and Forecasting methods for a given stock index. To put the results in context, the Sharpe for the buy-and-hold fot he Russel 2000 index is 0.673. The Sharpe values in bold are better than the buy-and-hold. 72
Table 20 – Median out-of-sample Sharpe ratio for all combinations of Feature Selection methods and Forecasting methods for a given stock index. To put the results in context, the Sharpe for the buy-and-hold fot he S&P Energy index is 0.058. The Sharpe values in bold are better than the buy-and-hold. 73
Table 21 – Median out-of-sample Sharpe ratio for all combinations of Feature Selection methods and Forecasting methods for a given stock index. To put the results in context, the Sharpe for the buy-and-hold fot he S&P Basic Materials index is 0.663. The Sharpe values in bold are better than the buy-and-hold. 73

CONTENTS

1 INTRODUCTION . . . 13
1.1 MAIN GOALS . . . 14
1.2 CONTRIBUTIONS . . . 14
1.3 ORGANIZATION . . . 15

2 LITERATURE REVIEW . . . 16
2.1 EXPECTED RETURNS OF FINANCIAL ASSETS . . . 16
2.1.1 Asset Pricing Theory . . . 16
2.1.2 Modern Asset Pricing Theory . . . 17
2.1.3 Machine Learning in Asset Pricing . . . 19
2.2 ALTERNATIVE DATA IN FINANCE AND ECONOMICS . . . 20
2.3 MACHINE LEARNING METHODS . . . 23
2.3.1 Logistic Regression . . . 25
2.3.2 Penalized Regression . . . 25
2.3.3 Regression Trees and Ensembles . . . 27
2.3.4 Neural Networks . . . 28
2.4 FEATURE SELECTION METHODS . . . 28
2.4.1 Mean Decrease Accuracy . . . 30
2.4.2 Mean Decrease Impurity . . . 30
2.4.3 Single Feature Importance . . . 31
2.4.4 Granger Causality . . . 31
2.4.5 The Markov Boundary . . . 31

3 DATA AND PREPROCESSING . . . 37
3.1 DATA DESCRIPTION . . . 37
3.2 DATA PROCESSING . . . 40

4 EXPERIMENT SETUP AND RESULTS . . . 42
4.1 FEATURE SELECTION AND FORECASTING SETUP . . . 43
4.2 MODEL PERFORMANCE . . . 44
4.2.1 Forecasting Perspective . . . 46
4.2.2 Portfolio Perspective . . . 47
4.3 DISCUSSION . . . 52

5 FINAL CONSIDERATIONS . . . 54

REFERENCES . . . 56

APPENDICES 60

APPENDIX A CUMMULATIVE RETURNS FOR THE S&P 500 GROUPING BY FORECAST MODEL AND FEATURE SELECTION METHODS 61

APPENDIX B MEDIAN SHARPE RATIO FOR ALL COMBINATIONS OF FEATURE SELECTION AND FORECASTING MODELS, GIVEN THE UNDERLYING STOCK INDEX. 69

1 INTRODUCTION

Predicting financial asset returns is a central problem in the field of asset pricing. The most well-established approach to this problem is to perform linear regression models to test whether a set of risk factors, typically financial or economical characteristics, have predictive power over the expected returns of the considered cross-section of assets.

One of the best examples of how the financial literature has approached the problem of predicting the expected return of financial assets can be found in the work of Fama and French (1993). The dataset used to search for predictive power in financial and economic variables consists of a cross-section of stocks along with proxies for intrinsic value, growth, size, default risk, among others. The employed statistical model is simply a linear regression model. The goal is to test whether the values of the financial and economical features have significant predictive power over future expected returns. Fama and French (1993) launched the search for factors that explain the expected returns of all asset classes and across all types of datasets.

Recently, the financial literature has approached the problem of predicting expected returns across assets in two mains ways: by employing flexible time series models and by using alternative data.

The first approach the literature uses is time-series regression models to forecast an asset's expected returns using mainly financial and macroeconomic variables Welch and Goyal (2007). These regression models are possibly non-linear. One interesting recent example of this is in the work of Gu et al. (2020). The authors comprehensively evaluate several financial variables related to firm characteristics building upon many machine learning predictive models.

The second approach builds upon the idea of using new collections of data, sometimes called alternative datasets, to build higher frequency proxies for economic and financial phenomena. One of the most famous of such datasets is the Google Trends search index data. The Google Trends data was popularized in the work of Choi and Varian (2012). Their work applied Google Trends to build a nowcast model for unemployment in the United States. The first work in finance to apply Google Trends was Da et al. (2011). The authors build upon Google Trends to construct a proxy for investor attention. They derive a trading rule based on their indicator. More recently, Huang et al. (2020) has coupled both the first approach described in the previous paragraph with the Google Trends data. They build an argument in favor of Google Trends using Granger causality and survival analysis. Also, they implement machine learning techniques using Google Trends as regressors applied to predict the expected return of the S&P 500 index.

Our work follows on the works of Gu et al. (2020) and Huang et al. (2020). Continuing on the problem of predicting expected returns of equity indices, we extend the work of Huang et al.

(2020). We systematically applied feature selection techniques and machine learning models and used Google Trends data as regressors. More specifically, our work differentiates from the work of Huang et al. (2020) by four points: First, we extend the set of equity indices from 1, as is done in Huang et al. (2020), to 15 indices. Secondly, we employ 15 feature selection techniques and ten machine learning methods to test the robustness of the Google Trends data as regressors. Thirdly, we use the average of 5 samples of the Google Trends data as suggested by Medeiros and Pires (2021) to avoid any sample bias that may be present. Lastly, we employ a rolling window time-series cross-validation procedure and use ten years of data as the out-of-sample data.

1.1 MAIN GOALS

The main goal of this work is to test whether Google Trends data has predictive power over the expected returns of US equity indices. More specifically, we can divide our goals as follows:

- We want to test whether any of a large number of feature selection techniques help to select features that consistently improve predictions over the benchmarks in terms of the prediction error and/or Sharpe ratio;
- We want to test whether any of a large number of prediction models help to consistently improve predictions over the benchmarks in terms of the prediction error and/or Sharpe ratio;
- Answer if a portfolio based on the models predictions help to build more desirable (more return and/or less risk) portfolio alternatives compared to the buy-and-hold strategy?;

1.2 CONTRIBUTIONS

This work seeks to extend the work of Huang et al. (2020) which answers whether Google Trends data have predictive power over the time series of expected returns of the S&P 500 index.

The most straightforward modification, in line with what was proposed as an extension by Huang et al. (2020), is that we consider daily returns of 12 United States (US) equity indices instead of weekly.

Secondly, although we included the feature selection techniques and forecasting models presented by Huang et al. (2020), we do not restrict ourselves to any particular model. Instead, we make the feature selection and predictive models as broad as possible to evaluate the robustness

of the data's predictive power. Thirdly, we use a broad range of US market indices to evaluate if the hypothetical predictive power of the data is pervasive across indices. Last, we employ a robustness procedure by sampling Google Trends data many times as proposed by Medeiros and Pires (2021) to avoid sample bias.

1.3 ORGANIZATION

We divide this study into the following sections: Section 2 is the literature review regarding previous works related to the purpose of this paper and the methodologies used; Section 3 presents the data used with a brief description of sources and characteristics; Section 4 presents the experimental setup, and results; and Section 5 contains the final considerations.

2 LITERATURE REVIEW

2.1 EXPECTED RETURNS OF FINANCIAL ASSETS

2.1.1 Asset Pricing Theory

Assets earn risk premiums because of their exposure to risk factors. According to Ang (2014), the first theory to explain risk premiums as a function of the underlying risk factors was developed in the 1960s by Jack Treynor (1961), William Sharpe (1964), John Lintner (1965), and Jan Mossin (1966), building upon the principle of diversification and mean-variance utility introduced by Markowitz (1952). This theory is better known by its name Capital Asset Pricing Model (CAPM), given by William Sharpe.

The CAPM predicts that asset risk premiums depend only on the assets beta, that is, its sensitivity to the market portfolio. Therefore it is a model with one underlying risk factor.

The fundamental pricing relationship derived from the CAPM is the traditional beta pricing relationship, which is formally called the security market line (SML). The SML states that any stock risk premium is proportional to the market risk premium. Formally, let us define R_i as the stock i return, R_f as the risk-free return, and R_m as the market return, then the Sharpe-Lintner version of the CAPM Campbell et al. (1997) is given by

$$\mathbb{E}[R_i] - R_f = \frac{Cov(R_i, R_f)}{Var(R_i)} \left(\mathbb{E}[R_m] - R_f\right) = \beta_i \left(\mathbb{E}[R_m] - R_f\right) \tag{2.1}$$

The CAPM model of equation (2.1) is a single-period model. Therefore, for econometric analysis of that model, we need to make assumptions regarding the time-series behavior of the returns. Typically, the assumption is that the returns are independently and identically distributed over time (IID) and jointly multivariate normal.

Fama and MacBeth (1973) provides the seminal empirical test of the CAPM model. The authors take a different approach to the CAPM. They note that the CAPM implies a linear relationship between expected returns and market betas which thoroughly explains the cross-section of expected returns. Following the explanation for the Fama-Macbeth regression provided in Campbell et al. (1997), the basic idea behind this procedure is that, for each cross-section of stock excess returns, one must linearly project the returns into the market betas and then aggregate the estimates in the time dimension.

2.1.2 Modern Asset Pricing Theory

The first model to generalize the SML was the Arbitrage Pricing Theory (APT) developed by Ross (1976). The APT differentiates from the CAPM in two main ways:

1. In a sense, it is more general than the CAPM because it allows for multiple risk factors, and;
2. It does not require the identification of the market portfolio.

According to Campbell et al. (1997), in its most general form, the APT approximates for expected asset returns as a linear function of many risk factors. The APT assumes markets are competitive and that there are no frictions. In this context, the data generating process for an asset's return is given by

$$R_i = \alpha_i + \boldsymbol{\beta}_i^{'}\boldsymbol{f} + \epsilon_i \tag{2.2}$$

$$\mathbb{E}[\epsilon_i|\boldsymbol{f}] = 0 \tag{2.3}$$

$$\mathbb{V}(\epsilon_i) = \sigma_i^2 = \sigma^2 < \infty \tag{2.4}$$

According to Campbell et al. (1997), there are two approaches to identifying the risk factors in (2.2): the statistical approach and the theoretical approach. The statistical approach seeks to explain the covariance matrix of the factors with a parsimonious number of representative components. In contrast, the theoretical approach builds mainly on economic theory to construct portfolios of tradeable factors and to argue in favor of specific ones.

Campbell, Lo, and Mackinlay Campbell et al. (1997) offer us two examples for each of the two approaches to test empirically the APT. In the statistical approach, the authors highlight the work of Connor and Korajczyk (1988) and Lehmann and Modest (1988). On the theoretical approach, they highlight the work of Fama and French (1993) and Kryzanowski and Zhang (1992).

The work of Connor and Korajczyk (1988) employed a modified version of the Principal Component Analysis (PCA) to a set of 5 and 10 risk factors related to each firm characteristic and macroeconomic variables. They found little explanatory power of models beyond five factors. Moreover, the work of Lehmann and Modest (1988) employed Factor Analysis with the same goal as Connor and Korajczyk (1988) and arrived at the same results.

With the theoretical approach, Fama and French (1993) uses firm characteristics to form factor portfolios to project stock excess returns. On the other hand, Kryzanowski and Zhang (1992) does the same analysis but uses macroeconomic characteristics to form factors.

In particular, Fama and French (1993) uses a different identification strategy and increases the number of factor portfolios tested compared to the work of Fama and French (1992). First, Fama and French (1993) choose to construct factor portfolios that employ a log-short strategy on the following risk factors related to firm characteristics

- Market portfolio factor (R_m)
- Mimicking portfolio for the size effect (SMB)
- Mimicking portfolio for the value effect (HML)

where the market portfolio is proxied by a US equity market index, the size portfolio is constructed by sorting stocks based on their market capitalization and going long the last decile (large size) firms and shorting the first decile (small size) of firms, and last the value portfolio is constructed by sorting portfolios based on their book-to-price (value proxy) ratio and going long on the first decile (high value) firms while shorting the last decile (low value) firms. The authors provide economic intuition for their choice of proxies to include in the model. Furthermore, instead of using the Fama-Macbeth regression model Fama and MacBeth (1973), they opt to directly regress the time-series of excess stock returns into the portfolio of risk factors.

We can classify the models surveyed in both sections 2.1.1 and 2.1.2 into two groups. The first group of models builds on utility theory to derive economic implications for asset pricing. These models are most notably the CAPM and the APT. The second group tries to empirically test the theoretical claims of the first group. For instance, Fama and French (1992) and Fama and French (1993) use different statistical strategies to answer the question: What factors are the excess returns of stocks are exposed to? Therefore, it's one way of empirically testing the APT model.

There is yet one last group of the literature that focuses specifically on stock return predictability. Koijen and Van Nieuwerburgh (2011) and Rapach and Zhou (2013) give great surveys of this literature. An asset returns predictability is typically examined via the predictive regression model below Rapach and Zhou (2013)

$$R_{i,t+1} = \alpha + \boldsymbol{\beta}' \boldsymbol{X}_{i,t} + \epsilon_{i,t+1} \tag{2.5}$$

where $R_{i,t+1}$ is typically the excess returns of a particular asset i (most often a stock or stock index) at time $t+1$, $\boldsymbol{X}_{i,t}$ is a set of variables available at time t, earlier than the equity premium, whose predictive power we want to test, and $\epsilon_{i,t+1}$ is a white noise process. Furthermore, Rapach and Zhou (2013) state that there is plenty of evidence on out-of-sample equity premium predictability.

2.1.3 Machine Learning in Asset Pricing

Machine learning applications in finance are indeed a hot topic. Despite the sporadic appearances in the literature, the trend has been stronger in the last 5-10 years.

Most of the work on applying machine learning models in asset pricing is focused on the last group of the literature stated in section 2.1.2: predicting the equity risk premium.

Arguably, the most important recent work on applying machine learning models to predicting the equity risk premium is the work of Gu et al. (2020). The authors state the problem as follows

$$R_{i,t+1} = \mathbb{E}[R_{i,t+1}|\boldsymbol{X}_{i,t} = \boldsymbol{x}_{i,t}] + \epsilon_{i,t+1} \tag{2.6}$$

where

$$\mathbb{E}[R_{i,t+1}|\boldsymbol{X}_{i,t} = \boldsymbol{x}_{i,t}] = f(\boldsymbol{X}_{i,t}) \tag{2.7}$$

for each stock $i = 1, ..., N$ and months $t = 1, ..., T$. Again $R_{i,t+1}$ is the excess returns of each stock and $\boldsymbol{X}_{i,t}$ is a set of covariates. The key idea of the work is to estimate the function $f(\boldsymbol{X}_{i,t})$ which maximizes the out-of-sample predictive power of $R_{i,t+1}$. The authors note that equation (2.6) is essentially a prediction problem, which is exactly the problem in which machine learning methods excel.

The implementation setup for Gu et al. (2020) is as follows:

- Monthly stock returns from 1957 to 2016
- 94 covariates that represent firm characteristics and macroeconomic variables along with 74 industry dummies corresponding to the first two digits of Standard Industrial Classification (SIC)
- They employ a time series cross-validation procedure having the R^2 as the loss function to be maximized.

Furthermore, the machine learning methods used are

- Linear regression estimated via Ordinary Least Squares (OLS) with all covariates
- Linear regression estimated via OLS with only three covariates, the portfolios sorted by size, book-to-market, and momentum
- Linear regression estimated via Partial Least Squares (PLS)

- Principal component regression
- Elastic Net
- Generalized linear model with group lasso
- Random forest
- Gradient boosted regression trees
- Neural network architectures with one to five layers

The authors report significant improvements in out-of-sample predictive performance in terms of R^2. Among the best performers, The Neural Network with three hidden layers is consistently the best model, accompanied by the other Neural Network architectures, the Random Forest model, and the Gradient boosted regression trees.

For more details on the hyperparameter space, cross validation setup and out-of-sample results, please refer to Gu et al. (2020).

2.2 ALTERNATIVE DATA IN FINANCE AND ECONOMICS

The first and the most well-known application of alternative data in economics is the application provided by Choi and Varian (2012), which was first published as a working paper in 2009. The central intuition behind using alternative data in economics and finance is to have an additional proxy for phenomena for which there are, at best, low-frequency proxies. We will focus on using alternative data for sentiment analysis and its derivations, mainly using Google Trends data to proxy for investor attention.

Arguably, the field of finance in which alternative data has been more widely applied is financial markets sentiment analysis. For example, Chan (2003) examines the monthly returns of stocks from The Center for Research in Security Prices (CRSP) following headlines from samples of The Dow Jones Interactive Publications Library. The work of Albuquerque and Vega (2008) analyzes the co-movement of selected stocks between a large (United States) and a small (Portugal) open economy. They create measures for news surprises using data from Bloomberg, International Money Market Services (IMMS), Reuters and Institutional Broker's Estimate System (IBES). In 2009, Fang and Peress (2009) published a work relating media coverage and US stocks' expected returns. They found that a portfolio of stocks with no media coverage outperforms a portfolio of stocks with high media coverage by 3% per year after controlling for market, size, momentum, book-to-market, and liquidity. The work of Engelberg and Parsons (2011) opts for a natural experimental approach. The authors used brokerage accounts data for 19 non-overlapping cities in the US along with local information sources (city newspaper) and

tested whether announcements directly related to US stock indices had a causal effect on the local trade volume. More recently, Ackert et al. (2016) tracked what they define as "influential investors" on finance forums and tested the ability of these investors to predict the direction of the stocks accurately.

Another more specific usage of alternative data in finance is that of building proxies for investor attention. Indeed, this literature is not new. Theoretical justifications have been given for the impact of investor attention, or inattention, on financial prices Mertons (1987); Sims (2003); Peng and Xiong (2006). Following Huang et al. (2020), we divide the applied literature in two ways: the first strand uses news and headlines as proxies for investor attention, and the second uses Google Trends data.

The work of Barber and Odean (2007); Yuan (2015) well represents the first strand. Both papers have the common goal of testing if individual investors are more likely to buy rather than sell the stocks that catch their attention. The work of Barber and Odean (2007) does this by using two financial proxies for investor attention and one alternative proxy. On the alternative side, they use the daily news feed from the Dow Jones News Service and count whether the firm appeared in that day's news. Their main results are that investors follow what they call attention-driven buying behavior. That is, they are net buyers on high-volume days, following both highly negative and highly positive one-day returns and when stocks are in the news. This behavior implies a temporary positive pressure on asset prices.

Moreover, Yuan (2015) analyze the impact of front-page articles as an attention-grabbing investor mechanism to predict trading patterns and asset returns. They find that high market attention generates the aggregate of individual investors to reduce their positions sensibly when the index is high and moderately when the index is low. Furthermore, this selling pattern lowers market price levels.

The second strand of the literature will be characterized by the work of Da et al. (2011), which was the first to apply Google Trends data as a measure of investor attention, the works of Preis et al. (2013); Curme et al. (2014), which presented a word-selection strategy and trading rule for the same data, and finally the work of Huang et al. (2020), which is our primary reference. We will devote more time to the second strand because it is used as a base for the current work.

The first work on proxying investors' attention using Google Trends dates back to Da et al. (2011). In this study, the authors propose Google Trends as a new direct proxy for investor attention and attempt to study its relation to asset prices. The authors gave two advantages of Google Trends data over other investor attention proxies. Firstly, they argue that by the time they published the paper, Google accounted for 72.1% of all search queries performed in the U.S. [1]. Secondly, they argue that Google Trends is a direct and unambiguous measure of attention.

To capture the attention paid by investors, Da et al. (2011) focus on approximately 3000

[1] In January 2021 Google accounted for 61.7% of U.S. desktop queries and 93.22% of U.S. mobile queries (https://www.statista.com/statistics/265796/us-search-engines-ranked-by-number-of-core-searches/).

stocks traded at Russel 3000 from January 2004 to January 2008. They query for weekly Google Trends data for the specific stock tickers. The authors raise the sampling issue recently pointed out in more detail by Medeiros and Pires (2021). However, they argue that the sample variations are minor and the series of the same searched stock ticker had a correlation of more than 90%. The authors discussed no other issues.

We summarize their key findings below:

1. The correlation between their search proxy is positive on average but low regarding alternative measures of attention. They find evidence that their proxy outperforms the others.
2. Using retail order execution data from the SEC Rule 11Ac1-5 (Dash-5) report, they find a strong relationship between their proxy and retail investors' trading volume
3. They find evidence in favor of the hypothesis raised by Barber and Odean (2007)

The works of both Preis et al. (2013) and Curme et al. (2014) rely on a semi-automatic manner of choosing the words to be tested. In particular, Preis et al. (2013) began with a financially biased set of words (i.e. "buy", "sell", "finance", "buy-and-hold", etc) and used the Google Sets service to complete the dataset with similar words. The critical hypothesis that they want to test is whether Google Trends data can anticipate future trends in the stock market. To do so, they build upon a dataset of 98 words and its respective Google Trends search volume from 2004 to 2011. Each word was searched conditioning to the U.S. Furthermore, the asset used to evaluate the prediction power of this dataset was the weekly Dow Jones Industrial Average index (DJIA) returns. To test this hypothesis, the authors perform a backtest of the following trading rule:

1. Define the changes in the search volume as:

$$\Delta S_t = S_t - \frac{1}{T}\sum_{i=1}^{T} S_{t-i} \tag{2.8}$$

   where we measure t and T in weeks.

2. If P_t is the closing price of the DJIA at week t, then we sell the DJIA at P_{t+1} if $\Delta S_t > 0$, otherwise we buy it.

The above trading rule is in accordance with the evidence found in Barber and Odean (2007) and Da et al. (2011), in which a temporary price increase follows an increase in investor attention at week t but both find subsequent reversals on $t + 1$.The authors found that most of the trading strategies built upon the 93 words individually outperformed a passive buy-and-hold portfolio on the DJIA in terms of cumulative returns.

The work of Curme et al. (2014) is essentially an extension of Preis et al. (2013) in the following ways:

1. They implement a more complex analysis. They apply a latent Dirichlet model Blei et al. (2003) to identify topics in the text corpus within Wikipedia;
2. They use the Amazon Mechanical Turk service to give the automatically identified topics a name;
3. They use the most representative words within each identified topic to search for its correspondent Google Trends search volume;
4. They apply the same trading rule as in Preis et al. (2013) using the search indices in the previous step and the prices for the S&P 500 index.

The last work of the second strand is indeed our primary reference. Huang et al. (2020) extends the work of Preis et al. (2013) and Curme et al. (2014) by the following:

1. They argue that using Granger Causality Granger (1969) and the Kaplan-Meyer estimator for the survival function Kaplan and Meier (1958) of a subset of the Google Trends dataset from both Preis et al. (2013) and Curme et al. (2014) indeed has predictive power over S&P 500 returns direction
2. Instead of relying on any *ex-ante* trading rule, the authors build prediction models based on a set of machine learning models (Lasso, Ridge, Support Vector Machine, and Elastic Net) plus feature selection models and use the out-of-sample returns classification as the trading rule.

They find that the Ridge regression model with features selected using Granger causality and a Logistic regression had the lowest prediction error. Furthermore, a trading strategy based on the out-of-sample predictions of this model outperforme the buy-and-hold strategy.

2.3 MACHINE LEARNING METHODS

Throughout this thesis, we will use the following definitions for the models. Let $\{(R_{i,t}, \boldsymbol{Z}_{i,t})\}_{i=1}^{T}$ be a stochastic process such that $R_{i,t}$ is the return for the i-th asset and $\boldsymbol{Z}_{i,t}$ is a d-dimensional vector of covariates for the same asset. We make the following assumption for the process at hand

Assumption 1 *We assume that* $\{(R_{i,t}, \boldsymbol{Z}_{i,t})\}_{i=1}^{T}$ *is a covariance-stationary process taking values in* $\mathbb{R}^{d+1}$

We are often interested in the autoregressive terms related to the covariates and the target variable. We define $\boldsymbol{X}_{i,t} = (\boldsymbol{Z}_{i,t}, ..., \boldsymbol{Z}_{i,t-r})'$ as the n-dimensional vector of covariates for the

current work, including lagged terms, where $n = (d+1)(r+1)$. In the context of predicting the time series of asset returns, we follow Gu et al. (2020) to define

$$R_{i,t+1} = \mathbb{E}[R_{i,t+1}|\boldsymbol{X}_{i,t} = \boldsymbol{x}_{i,t}] + \epsilon_{i,t+1}, \tag{2.9}$$

where our essential problem is to estimate the expected value in (2.6) with any functional form available

$$\mathbb{E}[R_{i,t+1}|\boldsymbol{X}_{i,t} = \boldsymbol{x}_{i,t}] = f(\boldsymbol{X}_{i,t}) \tag{2.10}$$

for $i = 1, ..., N$ and $t = 1, ..., T$. Our goal is to find $\hat{f}(\boldsymbol{X}_{i,t})$ that maximizes the out-of-sample predictive power of $R_{i,t+1}$, given a particular loss function.

One key difference between our work and Gu et al. (2020) is that we follow Huang et al. (2020) to define equation (2.6) as a classification problem. To do so, we define

$$C_{i,t+1} = \mathbb{I}_{(R_{i,t+1}>0)}, \tag{2.11}$$

which is a discrete random variable that takes values on the set $\mathbb{S} = \{1, 0\}$, which is a discrete random variable that thakes the value one if $(R_{i,t+1} > 0)$ and the value zero otherwise. Furthermore, we need to define a transformation to map values from (2.7) to probabilities. We define this function in a general manner as $\Phi(\boldsymbol{X}_{i,t})$ and the only restriction we make is that $\Phi : \mathbb{R}^n \to [0, 1]$. This transformation is called a link function in the Generalized Linear Models literature. For example, within the Logit model Φ is defined as

$$\Phi(\boldsymbol{x}_{i,t}) = \frac{\exp\{f(\boldsymbol{x}_{i,t})\}}{1 + \exp\{f(\boldsymbol{x}_{i,t})\}}. \tag{2.12}$$

One important fact to notice Elliott and Timmermann (2016) is that since $C_{i,t+1} \sim Ber(p_i)$, then $\mathbb{E}[C_{i,t+1}] = p_i$. Furthermore, it's easy to see that the conditional version of the previous expectation is also given by $\mathbb{E}[C_{i,t+1}|\boldsymbol{X}_{i,t} = \boldsymbol{x}_{i,t}] = p_i^*$, where p_i^* is

$$p_i^* = \mathbb{P}(C_{i,t+1} = 1|\boldsymbol{X}_{i,t} = \boldsymbol{x}_{i,t}). \tag{2.13}$$

This means that, unlike the case of a continuous target variable, the discrete target variable case implies that both density and point forecasts are the same.

2.3.1 Logistic Regression

The ordinary Logistic regression model with binary response is given by the probability of the response success as

$$\pi_i = \mathbb{P}(C_{i,t+1} = 1 | \boldsymbol{X}_{i,t} = \boldsymbol{x}_{i,t}) = \frac{\exp\{\boldsymbol{x}_{i,t}^{'}\boldsymbol{\beta}\}}{1 + \exp\{\boldsymbol{x}_{i,t}^{'}\boldsymbol{\beta}\}}, \tag{2.14}$$

where $\boldsymbol{\beta}$ is a n-dimensional vector of parameters. In the context of the model defined in the previous section, the Logistic regression model has a linear target function given by

$$f(\boldsymbol{X}_{i,t}) = \boldsymbol{X}_{i,t}^{'}\boldsymbol{\beta}, \tag{2.15}$$

and the link function is the logistic function given by

$$\Phi(\boldsymbol{X}_{i,t}) = \frac{\exp\{f(\boldsymbol{X}_{i,t})\}}{1 + \exp\{f(\boldsymbol{X}_{i,t})\}}. \tag{2.16}$$

One way of estimating the parameters of (2.11) is by maximizing the log-likelihood function given below

$$\begin{aligned} l(\boldsymbol{\beta}) &= \sum_{t=0}^{T-1} [C_{i,t+1}\log(\pi_i) + (1 - C_{i,t+1})\log(1 - \pi_i)] \\ &= \sum_{t=0}^{T-1} \left[C_{i,t+1}\log\left(\frac{\pi_i}{1-\pi_i}\right) + \log(1 - \pi_i)\right] \\ &= \sum_{t=0}^{T-1} \left[C_{i,t+1}\boldsymbol{x}_{i,t}^{'}\boldsymbol{\beta} - \log(1 + e^{\boldsymbol{x}_{i,t}^{'}\boldsymbol{\beta}})\right] \end{aligned}$$

We note that the vector of coefficients $\boldsymbol{\beta}$ can include an intercept if needed without loss of generality. Moreover, we emphasize that the Logistic regression model does not have any hyperparameters to set.

2.3.2 Penalized Regression

According to Masini et al. (2021), penalized regression models arise when the regression parameters are not uniquely defined. This is when model identification problems are implied by highly correlated covariates and/or a large number of variables d, possibly much bigger than the number of observations T. The idea of Penalized regression models is to restrict the Ordinary Least Squares solution by shrinking the estimated values of the model parameters towards zero.

Let $p(\boldsymbol{\beta}) = p(\boldsymbol{\beta}; \lambda, \alpha, \boldsymbol{X}_{i,t}) \geq 0$ be a penalty function that depends on the hyperparameter $\lambda \geq 0$, possibly on another hyperparameter $\alpha \in [0, 1]$, and the data Masini et al. (2021).

Then, we define the likelihood function for penalized models under the logistic transformation and in Lagrangian form as follows

$$l(\boldsymbol{\beta}) = \sum_{t=0}^{T-1} \left[C_{i,t+1} \boldsymbol{x}_{i,t}^{'} \boldsymbol{\beta} - \log(1 + e^{\boldsymbol{x}_{i,t}^{'} \boldsymbol{\beta}}) \right] + p(\boldsymbol{\beta}) \tag{2.17}$$

The parameter λ, and possibly α, controls the imposed penalization in the model. This is the central piece of the model that controls the trade-off between goodness of fit and the regularization term.

We formulate the penalized regression models presented below in terms of the penalty function defined in the previous paragraph. Moreover, we will maintain the first term of the likelihood function and the logistic link function as in the Logistic regression model.

The first model presented here is the Ridge regression. Hoerl and Kennard (1970) first proposed the Ridge regression model to introduce a bias towards the minimum residual sum squares solution of the Linear regression model in the presence of non-orthogonal covariates. It corresponds to penalizing the likelihood function with the ℓ_2 norm of the parameter vector. The penalty function under the Ridge model is given by

$$p(\boldsymbol{\beta}) = \lambda \sum_{i=1}^{n} \beta_i^2 \tag{2.18}$$

Tibshirani (1996) and Chan (2003) first proposed the Least Absolute Shrinkage and Selection Operator (LASSO) as a method for regularization and variable selection. It corresponds to estimating the model vector of parameters subject to the ℓ_1 norm. The penalty term to restrict the vector of parameter estimates is given by

$$p(\boldsymbol{\beta}) = \lambda \sum_{i=1}^{n} |\beta_i| \tag{2.19}$$

The last model we review is a combination of the Ridge and the LASSO models. Zou and Hastie (2005) proposed the Elastic-net model as a way of performing regularization and variable selection at the same time. They do it by restricting the parameter estimates concerning a convex combination of the ℓ_2 and ℓ_1 terms. Thus, the penalty term is given by

$$p(\boldsymbol{\beta}) = \lambda \left[\alpha \sum_{i=1}^{n} \beta_i^2 + (1 - \alpha) \sum_{i=1}^{n} |\beta_i| \right] \tag{2.20}$$

where $\alpha \in [0, 1]$. The elastic-net model has both the Ridge and the LASSO as a special case. For the theoretical properties of the models presented in this section, we refer the reader to Masini et al. (2021).

2.3.3 Regression Trees and Ensembles

A Regression tree is a nonparametric technique that approximates the unknown function $f(\boldsymbol{X}_{i,t})$ with local predictors. The covariates for each model are obtained by recursively partitioning the covariate space Masini et al. (2021).

Let $L \in \mathbb{N}$ be the depth of the terminal nodes (leaves) and $\mathcal{R}_j(L)$ be the region associated with the j-th leaf of the tree, $j = 1, ..., K$. The regression tree is given by

$$f(\boldsymbol{X}_{i,t}) = \sum_{j=1}^{K} \beta_j \boldsymbol{I}_{(\boldsymbol{X}_{i,t} \in \mathcal{R}_j(L))} \tag{2.21}$$

where the coefficients β_j in (2.21) are the sample averages of outcomes within the partition, and $\boldsymbol{I}_{(\boldsymbol{X}_{i,t} \in \mathcal{R}_j(L))}$ is a product of up to L indicator functions of the covariates, which effectively defines the partition. Growing a tree is then to find the optimal number and values of splits within the covariate space subject to the leaf's depth L, which maximizes the predictive power of the target variable.

It is well known that the flexibility of Regression tress is also what makes them prone to overfitting Hastie et al. (2001). Many techniques have been proposed to avoid this pitfall. We are going to survey the class of ensemble of trees. More specifically, we will focus on Boosting methods and Random Forests.

Random Forests (RF) Breiman (2001) are made of collections of Regression trees, each of which is fitted on samples from the original data via the bootstrap procedure. One important thing to notice is that, we need to apply a block bootstrap procedure instead of a simple bootstrap since we are dealing with time-series data.

We follow Masini et al. (2021) and Gu et al. (2020) to describe the Random Forest model. For each sample $b = 1, .., B$ a tree with N_b regions is estimated for a randomly selected subset of the original covariates. The number of regions N_b is specified to leave the minimum number of observations in each region. The model is then given by

$$f(\boldsymbol{X}_{i,t}) = \frac{1}{B} \sum_{b=1}^{B} \left[\sum_{j=1}^{K_b} \beta_j \boldsymbol{I}^b_{(\boldsymbol{X}_{i,t} \in \mathcal{R}_j(L))} \right] \tag{2.22}$$

The last ensemble method we survey was proposed by Friedman (2001) and is called Gradient Boosting (GB). This method combines the forecast of different oversimplified trees to build a more robust and stable forecast. The theory behind GB suggests that many weak learners fitted on "incomplete" trees combined result in a strong learner. The GB algorithm can be found

in the box below.

Algorithm 1: GB algorithm

1 **Input:** The maximum depth of the tree L, and the number J of trees in the forest
2 **Result:** Fitted value $\hat{f}$ for equation (2.22)
3 Initialize a shallow tree with depth $L = 1$
4 **for** *j in 1:J* **do**
5 Compute the residual for the initial shallow tree
6 Grow a small tree to fit the residual in step 5
7 Add to the forecast of the model from step 3 the forecast of the step 6 multiplied by a shrinking parameter $v \in (0, 1)$

2.3.4 Neural Networks

The most common Neural Network is the Feed-forward Neural Network (FFNN). The FFNN representation with one hidden layer can be represented by

$$f(\boldsymbol{X}_{i,t}) = \alpha + \sum_{j=1}^{J} \beta_j S(\gamma_{0,j} + \boldsymbol{\gamma}_j^{'} \boldsymbol{X}_{i,t}) \tag{2.23}$$

where $S(.)$ is called the basis or the activation function, and J is the number of neurons in the hidden layer of the FFNN architecture. Historically, the most popular activation function is the Logistic and the Hyperbolic tangent function. However, more recently, the Rectified Linear Unit (ReLu) has received attention Yarotsky (2017).

More layers may be added to the representation in (2.23) to build a deep neural network. We refer to Masini et al. (2021) for a review of these methods.

2.4 FEATURE SELECTION METHODS

The main goal of Feature Selection (FS) methods is to find a subset of the covariates for building predictive models. Aside from this goal, FS methods also have other motivations Guyon et al. (2006), for example:

1. Data reduction to avoid storage limitations
2. Improve predictive performance of predictive models
3. Gain knowledge regarding the underlying characteristics of the data generating process

Feature Selection methods have been classified into three categories: the search algorithm, the objective function they attempt to optimize, and the stopping criterion.

The first group of FS methods Li et al. (2018) is the Wrapper method. Wrapper methods rely on the predictive performance of a predefined learning algorithm to evaluate the quality of the selected features. Generally, a wrapper method repeats the following steps until a stopping criterion is met:

1. Search for a subset of the covariates
2. Evaluate the subset in a machine learning algorithm in regards to an objective function

The first step typically covers all feature combinations, which is computationally costly. With this difficulty in mind, many heuristic algorithms have been developed to improve the search over feature combinations.

The second group is referred to as Filter methods. Unlike Wrappers, Filter methods are independent of any Machine Learning algorithm. These methods rely primarily on the data alone and therefore are more computationally efficient than Wrapper methods. Due to the lack of any Machine Learning algorithm to guide the evaluation of subsets of the features, they may not be optimal for prediction tasks. A typical Filtering method has the following two basic steps:

1. Feature importance is computed and ranked according to some evaluation criterion
2. Low ranked features are filtered out

The last group is called Embedded methods. The methods within this group seek to balance the trade-off between Filters and Wrappers. They include a Machine Learning algorithm into the feature evaluation process while not evaluating sets of features iteratively. The most widely used Embedded methods are the regularization models (i.e., Ridge, LASSO).

To further define FS methods, we first introduce some notation and definitions. Recall that $\boldsymbol{X}_{i,t} = (\boldsymbol{Z}_{i,t}, ..., \boldsymbol{Z}_{i,t-r})' = (X_{i,1}, ..., X_{i,n})'$ was our data matrix for the i-th asset and $R_{i,t+1}$ was our target random variable for the same asset. For simplicity, we drop the i notation and define $\boldsymbol{X}_t$ and R_{t+1} for a generic asset. We follow guy (2007) to define independence and conditional independence as follows: two random variables A and B are said to be conditionally independent given a set of random variables $\boldsymbol{C}$, denoted $A \perp\!\!\!\perp B|\boldsymbol{C}$, if and only if $\mathbb{P}(A, B|\boldsymbol{C}) = \mathbb{P}(A|\boldsymbol{C})\mathbb{P}(B|\boldsymbol{C})$, where $\boldsymbol{C}$ can be the empty set, and we denote $A \perp\!\!\!\perp B$. Furthermore, we denote $\boldsymbol{V}_t$ as a subset of features of $\boldsymbol{X}_t$, and $\boldsymbol{V}_t^{-j}$ and $\boldsymbol{X}_t^{-j}$ as the same sets of random variables but excluding the j-th covariate.

With the above definitions at hand, we define feature relevance/irrelevance as follows guy (2007):

Definition 1 *A feature X_j is **irrelevant** to the target variable R_{t+1} if and only if (that will be denoted by iff) for all subsets of $\boldsymbol{V}_t^{-j}$ we have that $\mathbb{P}(X_j, R_{t+1}|\boldsymbol{V}_t^{-j}) = \mathbb{P}(X_j|\boldsymbol{V}_t^{-j})\mathbb{P}(R_{t+1}|\boldsymbol{V}_t^{-j})$.*

The authors in guy (2007) also define relevance by breaking the concept into strong and weak relevance.

Definition 2 *A feature X_j is **strongly relevant** to the target variable R_{t+1} iff there exists the values x_j, r_t, and $\boldsymbol{v}_t$ with $\mathbb{P}(X_j = x_j, \boldsymbol{X}_{j,t}^{-j} = \boldsymbol{v}_t) > 0$ such that $\mathbb{P}(R_{t+1} = r_{t+1}\boldsymbol{V}_t^{-j} = \boldsymbol{v}_t) \neq \mathbb{P}(R_{t+1} = r_{t+1}|\boldsymbol{V}_t^{-j} = \boldsymbol{v}_t)$.*

Definition 3 *AX_j is **weakly relevant** to the target variable R_{t+1} iff it is not strongly relevant and if there exists a subset of features $\boldsymbol{V}_t^{-j}$ for which there exists values x_j, r_{t+1}, and $\boldsymbol{v}_t$ with $\mathbb{P}(X_j = x_j, \boldsymbol{X}_{j,t}^{-j} = \boldsymbol{v}_t) > 0$ such that $\mathbb{P}(R_{t+1} = r_{t+1}|X_j = x_j, \boldsymbol{V}_t^{-j} = \boldsymbol{v}_t) \neq \mathbb{P}(R_{t+1} = r_{t+1}|\boldsymbol{V}_t^{-j} = \boldsymbol{v}_t)$.*

In what follows, we provide a brief description of some FS methods and classify them into the categories described above.

2.4.1 Mean Decrease Accuracy

The Mean Decrease Accuracy (MDA) method, as explained by de Prado (2018), can be grouped into Wrapper methods because it assigns a score to each feature based on its out-of-sample predictive power.

Given a predictive model of any kind, the MDA computes the out-of-sample prediction error for a model having the covariates given by $\boldsymbol{X}_t^{-j}$, for each $j = 1, ..., n$. The features are ranked based on the out-of-sample predictive error the models had when they were excluded. Features with high out-of-sample predictive error in their absence are preferable.

An essential characteristic of this method is that it does not rely on any particular prediction model nor on any particular performance metric.

2.4.2 Mean Decrease Impurity

The Mean Decrease Impurity (MDI) method surveyed by de Prado (2018) can be classified as a Wrapper method because it depends on tree-based machine learning methods.

Given a tree impurity measure, the MDI method attributes to each feature a measure of how much of the overall tree impurity has decreased because of this particular feature. When growing a forest of trees, the MDI can be assessed by averaging the decrease in each of the trees' impurity attributed to a particular variable. It is important to note that this is an in-sample measure and thus does not account for generalization errors.

2.4.3 Single Feature Importance

In contrast to the MDA, the Single Feature Importance (SFI) method iterates between the covariates in $\boldsymbol{X}_t$ and computes the out-of-sample predictive error in a given model. Since the SFI relies on a predictive model, we can group it with the Wrapper methods.

We can apply the SFI to any prediction model and any performance metric in line with the MDA. Nevertheless, unlike the MDA, it cannot identify the dependence of features as defined in definitions 2 and 3 when the conditioning set $\boldsymbol{V}_t^{-j}$ is not the empty set.

2.4.4 Granger Causality

Granger (1969) proposed the univariate Granger causality method to derive the direction of information between two related variables. Although not directly related to the Feature Selection problem, many authors have argued that it can be used to assess the predictive power of one process over another.

For instance, Huang et al. (2020) have proposed a feature selection method with the Granger causality at its core. For a given significance threshold, the method proposed by Huang et al. (2020) performs a univariate Granger causality test between R_t and each of the random variables in $\boldsymbol{X}_t$, individually. In a second step, the authors propose to build a Logistic regression model having as covariates only the features that pass the univariate Granger causality for the given threshold and select all the features which appear to be significant in the Logistic regression.

2.4.5 The Markov Boundary

Bayesian Networks (BN) provide a convenient framework to represent dependent and independent relationships between random variables. In particular, the Markov Boundary (MB) is a technique that builds upon BN to find the set of random variables that possess enough information to forecast a target variable.

Before defining the key MB algorithms, we define some of the key concepts of BN theory. The essential ingredients of a BN are a set of nodes, which are represented as random variables, a set of paths between nodes, and the graph itself. Following Koller and Friedman (2009), we provide a more formal definition for these critical concepts

Definition 4 *A **graph** is a tuple $\langle \mathcal{X}, \mathcal{E} \rangle$ where $\mathcal{X} = \{C_{t+1}, X_1, ..., X_n\}$ is a set nodes or random variables and $\mathcal{E}$ is a set of edges between random variables.*

Definition 5 *Let $\mathcal{K} = \langle \mathcal{X}, \mathcal{E} \rangle$ be a graph. We say that $X_1, ..., X_n$ forms a **path** in $\mathcal{K}$ of, for any $j = 1, ..., n-1$, or $X_j \rightarrow X_{j+1}$, or $X_j - X_{j+1}$.We say that a path is **undirected** if $X_j - X_{j+1}$. Furthermore, we say that a path is **directed** if, for at least one j, we have that $X_j \rightarrow X_{j+1}$.*

Definition 6 *Let $\mathcal{X}$ be a set of random variables, and let $\mathbb{P}(\mathcal{X})$ be the joint distribution over these variables. We say that PA_j is the set of **Markovian parents** of X_j if PA_j is the smallest set of predecessors of the variable X_j such that it renders X_j independent of all its non-descendants given PA_j.*

It is interesting to note that Definition 6 enables us with semantics to decompose the joint distribution as a product of conditional distributions where PA_j is the conditioning set. Once we state the above definitions, we can now enunciate one fundamental assumption of BN Pearl (1988):

Assumption 2 *Let $\mathcal{G} = \langle \mathcal{X}, \mathcal{E}, \mathbb{P}(\mathcal{X}) \rangle$. We say that $\mathcal{G}$ satisfies the **Markov condition** if every node in $\mathcal{G}$ is independent of any subset of its non-descendants given PA_j.*

We can now state the definition of BN as follows:

Definition 7 *We call $\mathcal{G}$ a **Bayesian Network** (BN) iff it satisfies the Markov condition.*

To further define what a MB is we will need two more assumptions which are given below Pearl (1988):

Assumption 3 *Let $\mathcal{G}$ be a BN. We say that $\mathcal{G}$ is **Faithful** to the joint distribution $\mathbb{P}(\mathcal{X})$ iff every conditional independence present in $\mathbb{P}$ is entailed by $\mathcal{G}$ and the Markov condition. On the other hand, we say that $\mathbb{P}(\mathcal{X})$ is Faithful to $\mathcal{G}$ iff there exists a BN $\mathcal{G}$ such that it is faithful to $\mathbb{P}$.*

Assumption 4 *We state that a BN $\mathcal{G}$ is **Causal sufficient** if any common cause of two or more variables in $\mathcal{X}$ are also in $\mathcal{X}$.*

Before stating the definition of a MB we define the concept of a Markov Blanket Pearl (1988):

Definition 8 *Let C_{t+1} be a class variable in $\mathcal{X}$. We say that $Mb(C_{t+1})$ is a **Markov Blanket** of C_{t+1} if for every $X_k \in \mathcal{X} \setminus (Mb(C_{t+1}) \bigcup C_{t+1})$ we have $C_{t+1} \perp\!\!\!\perp X_k | Mb(C_{t+1})$.*

It is not hard to see that there can be different sets of many sizes that satisfy the definition of a Markov Blanket. In other words, they are not unique. In practice, we are interested in the smallest set that satisfies the Markov Blanket, and that is the definition of an MB:

Definition 9 *We say that the set $Mb(C_{t+1})$ is the **Markov Boundary** (MB) set if none of its proper subsets satisfies the Markov condition.*

Moreover, MBs have two crucial properties stated in Pearl (1988). They are:

Theorem 1 *Under Assumption 3 (the Faithfulness assumption) the MB of a node in a BN is unique and it consists of the nodes:*

- *Parents (direct causes)*
- *Children (direct effects)*
- *Spouses (parents of node children)*

Proposition 1 *Let $MB(C_{t+1})$ be the MB of a given variable in $\mathcal{X}$, and let $\mathcal{G}$ be a BN. We claim that for every $X_k \in \mathcal{X} \setminus (Mb(C_{t+1}) \bigcup C_{t+1})$, it is true that $C_{t+1} \perp\!\!\!\perp X_k | Mb(C_{t+1})$.*

We can find the proof for the above results in Pearl (1988).

Many MB algorithms have been proposed in the literature, and we can find a comprehensive survey in Yu et al. (2019). Here we opt to describe two of these algorithms:

- Incremental Association Markov Boundary (IAMB) proposed by Tsamardinos and Aliferis (2003)
- Max-Min Markov Boundary (MMMB) proposed by Tsamardinos et al. (003a)

According to Yu et al. (2019), MB feature selection techniques are based on the standard forward-backward feature selection algorithm. The forward phase starts with a set of features, that is usually empty, and adds features to each by continually testing independence or conditional independence between the target and the covariate in question. This procedure runs until a stopping criterion is met. The backward phase eliminates features from the final set of the previous step until a stopping criterion is met.

We can divide forward-backward feature selection methods into two subgroups. The first group is called the Standard forward-backward feature selection (SFBS) strategy. The second group is called the Intervealing forward-backward feature selection (IFBS) strategy. The critical difference between them is that while the SFBS performs both the forward and the backward phase separately, the IFBS performs both phases simultaneously. Moreover, we note that the IAMB algorithm is an SFBS method, while the MMMB algorithm is an IFBS method. Following

Yu et al. (2019), we provide a basic algorithm for the SFBS and IFBS below. We emphasize that

these algorithms will find a candidate MB, which we will denote $CMB(C_{t+1})$, of $MB(C_{t+1})$.

Algorithm 2: SFBS for MB learning

Input: Feature set $\boldsymbol{X}_t$, a target C_t and $CMB(C_{t+1}) = \emptyset$
Result: $CMB(C_{t+1})$
// Forward phase
while *New features are added to* $CMB(C_{t+1})$ **do**
  Randomly select a feature $X_j \in \boldsymbol{X}_t$;
  if $C_{t+1} \not\perp\!\!\!\perp X_j | CMB(C_{t+1})$ **then**
    $CMB(C_{t+1}) = CMB(C_{t+1}) \bigcup X_j$;
    $\boldsymbol{X}_t = \boldsymbol{X}_t \backslash X_j$;
  end
end
// Backward phase
while *Features of* $CMB(C_{t+1})$ *are removed* **do**
  Randomly select a feature $X_j \in CMB(C_{t+1})$;
  if $X_j \perp\!\!\!\perp C_{t+1} | CMB(C_{t+1}) \backslash X_j$ **then**
    $CMB(C_{t+1}) = CMB(C_{t+1}) \backslash X_j$
  end
end

Algorithm 3: IFBS for MB learning

Input: Feature set $\boldsymbol{X}_t$, a target C_t and $CMB(C_{t+1}) = \emptyset$
Result: $CMB(C_{t+1})$
// Forward phase
while *New features are added to* $CMB(C_{t+1})$ **do**
  Select a feature $X_j \in \boldsymbol{X}_t$ with the highest association with C_{t+1};
  // Backward phase
  if $C_{t+1} \not\perp\!\!\!\perp X_j | CMB(C_{t+1})$ **then**
    $CMB(C_{t+1}) = CMB(C_{t+1}) \bigcup X_j$;
    $\boldsymbol{X}_t = \boldsymbol{X}_t \backslash X_j$;
    while *Features of* $CMB(C_{t+1})$ *are removed* **do**
      Randomly select a feature $X_j \in CMB(C_{t+1})$;
      if $X_j \perp\!\!\!\perp C_{t+1} | CMB(C_{t+1}) \backslash X_j$ **then**
        $CMB(C_{t+1}) = CMB(C_{t+1}) \backslash X_j$
      end
    end
  end
end

Tsamardinos and Aliferis (2003) proposed the IAMB algorithm as a modification of the Growing-Shrinking MB (GSMB) learning algorithm, which implements Algorithm 1 exactly. The main difficulty of the GSMB that Tsamardinos and Aliferis (2003) try to solve with IAMB are steps 4 to 9 in the Forward phase. The authors in Yu et al. (2019) note that these steps imply many more false positives than we would expect, which slows down the algorithm speed. To solve this issue, the IAMB introduces a dynamic heuristic in step 5, which adds to $CMB(C_{t+1})$ features X_j with the highest association with C_{t+1} conditioning on $CMB(C_{t+1})$. This heuristic makes features that belong to $MB(C_{t+1})$ enter the $CMB(C_{t+1})$ as early as possible, which increases efficiency.

Algorithm 4: SFBS for PC learning

Input: Feature set $\boldsymbol{X}_t$, a target C_t and $CPC(C_{t+1}) = \emptyset$
Result: $CPC(C_{t+1})$
// Filtering out irrelevant features
$R = \boldsymbol{X}_t \backslash S'$ (for every $X_j \in S, X_j \perp\!\!\!\perp C_{t+1} | \emptyset$)
// Forward phase
while *New features are added to* $CPC(C_{t+1})$ **do**
    Select the best feature $X_j \in R$ with a greedy strategy
    $CPC(C_{t+1}) = CPC(C_{t+1}) \bigcup X_j$
    $R = R \backslash X_j$
end
// Backward phase
while *Features of* $CPC(C_{t+1})$ *are removed* **do**
    Consider every feature $X_j \in CPC(C_{t+1})$
    if $\exists S \in CPC(C_{t+1})$ *s.t.* $X_j \perp\!\!\!\perp C_{t+1} | S$ **then**
        $CPC(C_{t+1}) = CPC(C_{t+1}) \backslash X_j$
    end
end

The latest algorithm considered here is employing the SFBS strategy along with what is called Divide-and-conquer MB learning Yu et al. (2019). The main goal is to reduce the computational issues of the other approaches. These algorithms break the problem of learning the $MB(C_{t+1})$ into two problems: the first is to find the direct causes (parents) and direct effects (children) of C_{t+1} ($PC(C_{t+1})$; the second step is to find the causes of C_{t+1} effects (spouses, $SP(C_{t+1})$). The MMMB algorithm is an example of an SFBS divide-and-conquer algorithm. The MMMB algorithm uses it as its base Algorithm 3 in the box.

3 DATA AND PREPROCESSING

This section intends to describe all processes related to data: its characteristics, acquisition, and processing decisions.

3.1 DATA DESCRIPTION

We divide our dataset into two main categories:

1. Financial data
2. Google Trends data

We use total daily returns (returns including the payment of dividends) data for three indices of the U.S. stock market regarding financial data. More specifically, we chose the following indices:

- Standard and Poors 500 index (S&P 500)
- NASDAQ
- Dow Jones

To further increase the set of returns to be used, we decided to break the S&P 500 into nine industry groups as Bloomberg's stock classification suggested. The groups are as follows:

- Basic Materials
- Communications
- Consumer Cyclical
- Consumer Non-cyclical
- Energy
- Financial
- Industrial
- Technology

- Utilities

We end up with 12 total return time series to build our forecast models. All financial data were acquired using the Bloomberg terminal.

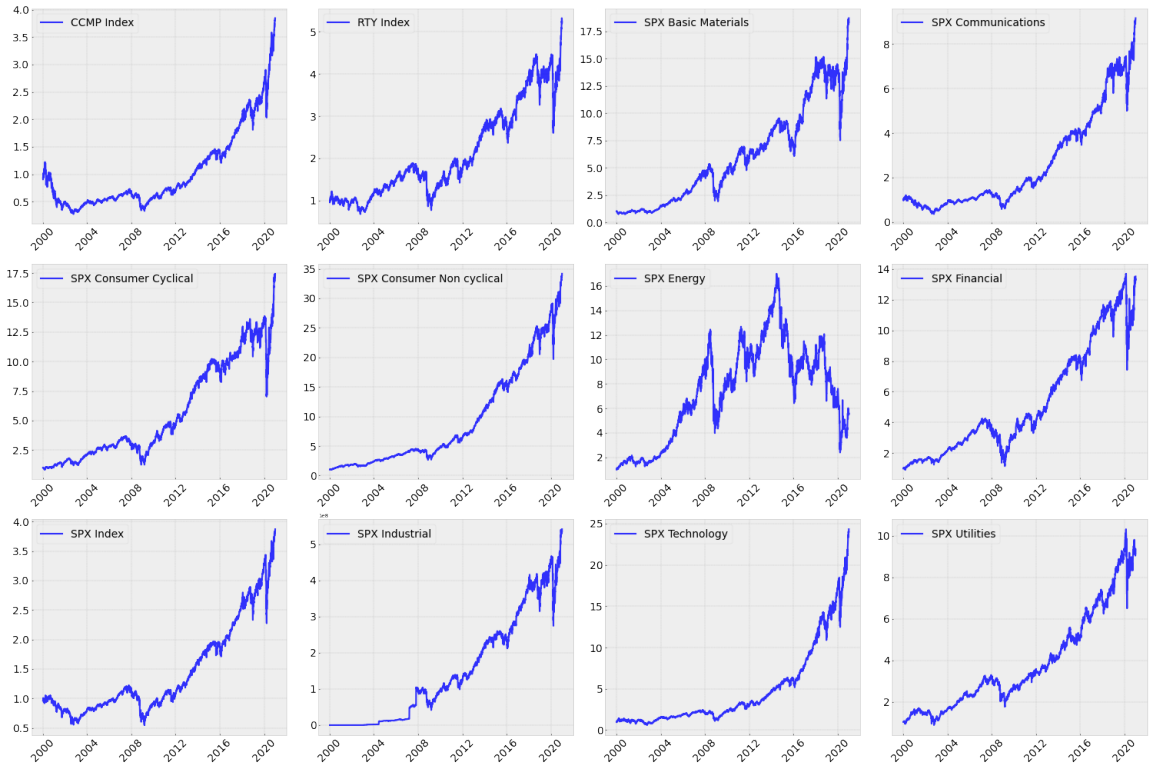

Figure 1 – Cumulative returns for $1 invested at the beginning of the sample for all U.S. market indices used in our study.

Regarding the Google Trends data, we follow Huang et al. (2020) in most of its steps. We select the words presented in Curme et al. (2014) under the major categories: *Politics 1*, *Politics 2*, and *Business*. We note that we have discarded the words "notability" and "rare earth" because the value for these words is less than one for all time windows selected. We have also added the words presented in Preis et al. (2013). Hence, there are **181** words in total.

Using the public data from the Google Trends API, we collect daily information about these words from January 1st of 2004 to December 31st, 2020.

We follow Medeiros and Pires (2021) by sampling from the Google Trends data five times across a week to control sample variability. The final value for each daily entry is the median of all samples.

Another issue that we must address is the comparability between each of the time series of words. One way to do that is to search for all the target words together. For API limitation reasons, we cannot do this. To resolve this, we follow Curme et al. (2014) by downloading search volume data for each word together with a controlling term "google" alongside search volume data for each of our terms.

Politics 1	Politics 2	Business
republican	notability	blacklist
wisconsin	party	business
york	law	management
county	government	services
served	court	companies
senate	minister	development
president	political	bank
district	act	founded
ohio	president	million
virginia	votes	financial
carolina	police	industry
massachusetts	council	products
pennsylvania	election	global
democratic	rights	market
illinois	elections	marketing
representatives	committee	ceo
washington	democratic	technology
texas	security	investment
governor	union	inc
election	case	social
elected	members	media
law	justice	project
vermont	voters	firm
missouri	legal	service
kentucky	seats	energy
william	movement	economic
judge	secretary	community
tennessee	federal	corporation
college	stats	government
georgia	ministry	ltd

Table 1 – Example of features within each of the groups defined by Curme et al. (2014)

The Google Trends API does not allow downloading extensive periods of data. In order to circumvent this issue, we have downloaded data in a six-month window. We have combined the different windows by simply averaging the repeated information across windows.[1]

[1] Although some practitioners create different methods of combining these windows <https://towardsdatascience.com/reconstruct-google-trends-daily-data-for-extended-period-75b6ca1d3420>, by experimentation, we have observed that just by averaging the information, we were able to usefully reconstruct the daily information with small error.

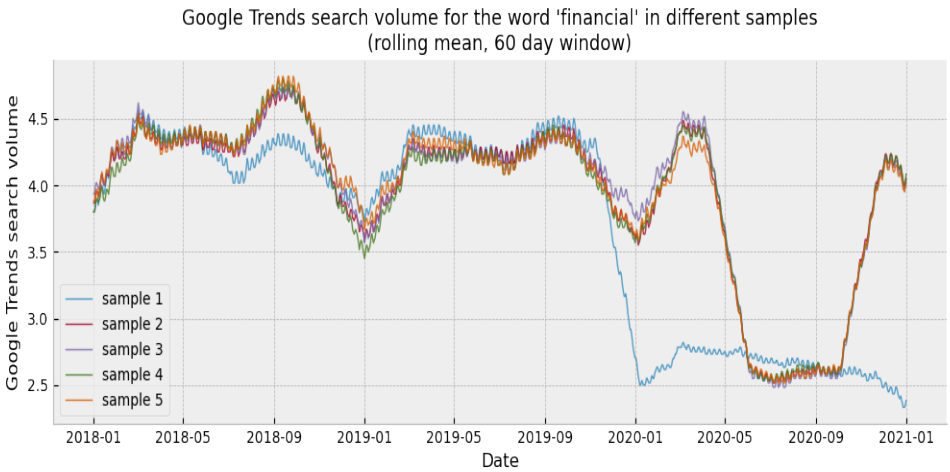

Figure 2 – Rolling mean (60 days) of Google Trends search volume for the word "financial" for different samples. The word was sampled across a week on different times along the day.

3.2 DATA PROCESSING

We proceed within the data groups defined in section 3.1 and provide more details about their processing. We start with the financial data because its data processing is much simpler than the Google Trends data.

Since our goal is to classify the discrete version of financial returns with the ultimate goal of backtesting a trading strategy that builds on these predictions to time the underlying asset, we have two main concerns to deal with:

1. Financial returns must reflect the true out-of-sample return of the financial asset in practice;

2. Returns must be stationary

Typically the logarithmic of the first difference of financial returns can guarantee the second issue above. More specifically, we choose to model the first difference of the U.S. market indices price level, which results in daily returns for the assets. The stationary of the returns was checked by running the Augmented Dickey Fuller test (ADF).

Regarding the first issue, since we are dealing with indices that reflect the return of stocks and stocks have dividends as one source of income reflected in their prices, we need to compute the returns of the U.S. market indices considering the payment of dividends on the underlying stocks. In practice, we can solve both the first and second issues relatively easily by using Bloomberg's terminal. More specifically, we use Bloomberg's function "Bloomberg data history" (BDH) with the "DAY TO DAY TOT RETURN GROSS DVDS" option, which yields precisely what we expected to easily solve the issues raised above.

With the financial data at hand, we can describe the processing of the Google Trends data. Recall that we have defined the n-dimensional vector of covariates for the i-th asset at time t as $\boldsymbol{X_{i,t}} = (\boldsymbol{Z}_{i,t}, ..., \boldsymbol{Z}_{i,t-r})'$, where $\boldsymbol{Z}_{i,t}$ is the d-dimensional vector of covariates. We note that the covariates for the models we implement are essentially the Google Trends search volume for the different words and their respective lags. Thus we can concentrate on the description of $\boldsymbol{Z}_{i,t}$ and its lagged version here.

Given a d-dimensional vector $\boldsymbol{Z}_{i,t}$, we first need to make sure that the Google Trends data is stationary. To do so, we employ the following procedure. For a given asset i and word $Z_{j,t} \in \boldsymbol{Z}_{i,t}$, we take the first d-th difference, for $d = 1, ..., 5$, of the series. It rejects the null hypothesis of non-stationarity of the ADF test at the 0.001 significance level. In practice, for all time series the first difference was enough to reject the null hypothesis. Moreover, with the stationary series of words at hand, we expand these dataset of 181 words to include the lags of each word up to lag 20, which yields a dataset of $181 \times 20 = 36420$ features. We point out that, since our data frequency is daily, $l = 20$ means that we have lags up to a month in the past for each word at time t.

4 EXPERIMENT SETUP AND RESULTS

In this section, we give a description of the experimental setup and our main results. The experiment consists of three steps: The feature selection step, the hyperparameter search, and the forecasting step. We give a detailed description of each step in the sections below. Moreover, we organize the results to address the questions raised in the first chapter of this thesis.

Roughly speaking, we can describe the experimental setup as follows: We split the out-of-sample data into two groups consisting of 50% of the days. The first sample is used to select words (features) that will be used as covariates in the models of the second step. All features selected by the feature selection methods will be stored to be used in the second step. The last step consists of a recursive foresting procedure. For each year in the last 50% of the sample and given a set of covariates previously selected, we split the current year into train and test samples and perform a hyperparameter search. Once the hyperparameters are selected, we forecast the class of the return for the following year (test set) and store the results. The last step is performed recursively for each of the proposed forecasting methods.

The acronyms of the feature selection and forecasting methods that will be used are summarized below:

1. Feature selection methods
    - Control group with all words (“ALL”)
    - Mean decrease accuracy (“MDA”)
    - Single feature importance (“SFI”)
    - Mean decrease impurity (“MDI”)
    - Univariate Granger causality (“GRANGER”)
    - Univariate Granger causality and Logistic regression proposed by Huang et al. (2020) (“HUANG”)
    - Incremental association Markov boundary (“IAMB”)
    - Max-Min Markov boundary (“MMMB”)
2. Forecasting methods
    - Logistic regression (“LOGIT“)
    - Ridge regression (“RIDGE”)
    - LASSO regression (“LASSO”)
    - Elastic net regression (“ENET”)

- Random forest (“RF”)
- Gradient boosting (“GB”)
- Feed-forward three-hidden-layer Neural Network (“NN3”)

4.1 FEATURE SELECTION AND FORECASTING SETUP

We can divide the forecasting framework used in our work into two major steps:

1. Feature selection and;
2. Forecasting step

Recall that our sample consists of daily data from U.S. stock indices and Google Trends search volume from January 1st, 2004 to December 31st, 2020. We start by splitting our data into two groups: January 1st of 2004 to June 26th, 2012, and the other from June 27th, 2012 to December 31st, 2020.

The first sample will be used for the feature selection that will be set in the second step. The FS methods previously listed can be roughly divided into two groups: the first group performs in-sample selection. The other performs selection based on some loss function of the out-of-sample predictions.

Feature Selection methods such as “GRANGER”, “HUANG”, “IAMB”, “MMMB”, and “MDI” can be classified into the first group. Furthermore, exceptions for “MDI”, “MDA” and “SFI” all perform some hypothesis testing related to the relevance of the features, which can be univariate or multivariate. We have set the significance level to 0.05 as a threshold for all the hypothesis testing performed by these methods. On the other hand, methods such as “MDA” and “SFI” perform feature selection based on out-of-sample prediction errors. This means that we split the feature selection sample into two parts and the second part, the test sample, is used by the “MDA” and “SFI” methods to select the features based on some forecasting error metric.

The FS process results in a set of features for each of the FS methods. It results in seven sets of features and the control set, which is all the words and their respective lags.

With the eight sets of features at hand, we can begin to describe the forecasting step. Methods such as the K-fold cross-validation are not adequate for dependent data, as is well known in the machine learning literature (Bergmeir et al. (2018); Cerqueira et al. (202)). A simple and effective way to search for hyperparameters while accounting for dependencies among observations is using recursive cross-validation procedures. Generally speaking, these methods employ a rolling or expanding window scheme to perform parameter optimization by splitting the current window into train and validation sets. Once the optimal parameters have been found,

the forecast error is obtained using this model to make predictions on a test set. Finally, after the prediction has been made, the test set becomes part of the train plus validation window. The beginning of this sample can stay set or roll one year forward.

In our case, we chose a recursive expanding window cross-validation scheme. Starting from year 1, we first set the current year as the training sample and the next year as the test sample. For each draw of the ten total draws of a hyperparameter in the parameter space detailed in Tables 2, 3, 4, and 5, we split the training sample into $S = \{s_{t_1}, s_{t_2}, s_{t_3}, s_{t_4}, s_{t_5}\}$. For each of the validation samples in S, we fit the model for the associated drawn hyperparameter on the training sample (i.e., $T = \{s_{t_1}, s_{t_2}, s_{t_3}, s_{t_4}\}$) and validate the remaining data (i.e., $V = \{s_{t_5}\}$) while preserving the time order. This procedure is repeated recursively until the last year of the sample.

The hyperparameters optimized using the cross-validation procedure described in the last paragraph are shown in table 2, 3, 4 and 5. Each table shows the hyperparameters (rows of the table) used in the cross-validation exercise for a given model (columns of the table). For a quick review on the key hyperparameters for each model please refer to Chapter 3. Each of the them are referenced according to the name used in the python package Scikit-Learn. For further explanations on the hyperprameters we refer to Pedregosa et al. (2011) which contains a description on each of them. Furthermore, for the ranges of the hyperparameters and how to set them we refer to Hastie et al. (2001) and Pedregosa et al. (2011).

	LASSO	Ridge	Elastic Net
C	$[0.01, 50]$	$[0.01, 50]$	$[0.01, 50]$
intercept	True	True	True
l1_ratio			$[0.001, 0.999]$

Table 2 – Hyperparameter space for penalized regression models

	Random Forest
max_features	['auto', 'sqrt', 'log2']
min_samples_split	$x \sim Unif\{2, 31\}$
n_estimators	$x \sim Unif\{2, 301\}$
max_depth	$x \sim Unif\{2, 20\}$

Table 3 – Hyperparameter space for the Random Forest regression

4.2 MODEL PERFORMANCE

In this section, we present the main results obtained in the experiment described in the previous section. The performance of the models will be evaluated from two different perspectives:

	Gradient Boosting
num_leaves	$x \sim Unif\{6, 50\}$
min_child_samples	$x \sim Unif\{100, 500\}$
min_child_weight	$[1e^{-5}, 1e^{4}]$
subsample	$x \sim Unif(0.2, 0.8)$
n_estimators	$x \sim Unif\{500, 1000\}$
max_depth	$x \sim Unif\{2, 100\}$
learning_rate	$[0.001, 0.99]$
colsample_bytree	$x \sim Unif(0.4, 0.6)$
reg_alpha	$[0, 1e^{-1}, 1, 2, 5, 7, 10, 50, 100]$
reg_lambda	$[0, 1e^{-1}, 1, 5, 10, 20, 50, 100]$
objective	huber

Table 4 – Hyperparameter space for the Gradient Boosting

	Neural Network
early_stopping	True
learning_rate	invscaling
learning_rate_init	$[0.001, 0.999, 100]$
alpha	$[0.001, 0.999]$
activation	relu
hidden_layer_sizes	$\{32, 16, 8\}$

Table 5 – Hyperparameter space for the three hidden layer Neural Network

1. Forecasting perspective
2. Portfolio perspective

The forecasting perspective is more common in the statistical and machine learning literature. It is centered on the evaluation of the models based on some measure of accuracy. In our case, we wish to compare the classes of the returns predicted by our models against the benchmarks. The chosen metric of accuracy was the Area Under the Receiver Operating Characteristic curve (ROC-AUC, but AUC for short).

The ROC curve is a technique used to evaluate the performance of a classifier Fawcett (2006). The ROC graph is a two-dimensional graph which presents the true positive rate (rate of positives correctly classified) on the y-axis, whereas the false positive rate (rate of negatives incorrectly classified) is presented on the x-axis. Thus, the ROC graph depicts the trade-off between benefits (true positives) and costs (false positives). Given a classifier that outputs a value between zero and one, there can be different thresholds to discriminate which values will be attributed to zero and one classes. The ROC curve is essentially a step function where each point in the function corresponds to a true positive rate and false positive rate. Furthermore, it is often useful to reduce the ROC curve to a single number with the goal of comparing this value

between classifiers. One popular metric for this is the Area Under the RUC curve (AUC).

Despite the importance of evaluating the forecasting error of a particular model, we highlight that it is not enough to choose between models to build a profitable trading strategy. In particular, since our predictions are binary (1 for a positive return and 0 for a negative return), we do not explicitly account for the volatility of the returns. Furthermore, since we are dealing with time-series and compounded returns, the order in which the predictions match or not the target also matters. To illustrate this fact, for a given level of AUC and corresponding hit rate, we have simulated different trading strategies that times the S&P 500 and computed the corresponding Sharpe ratios. The results of the simulations are plotted on Figure 3. As we can see, there is a considerable variance between the trials for a given AUC. The variance is induced by the fact that the order in which the correct predictions are matters. To take these issue into account, we also compare Feature Selection methods and Forecasting models in terms of cumulative returns and the Sharpe ratio (SR).

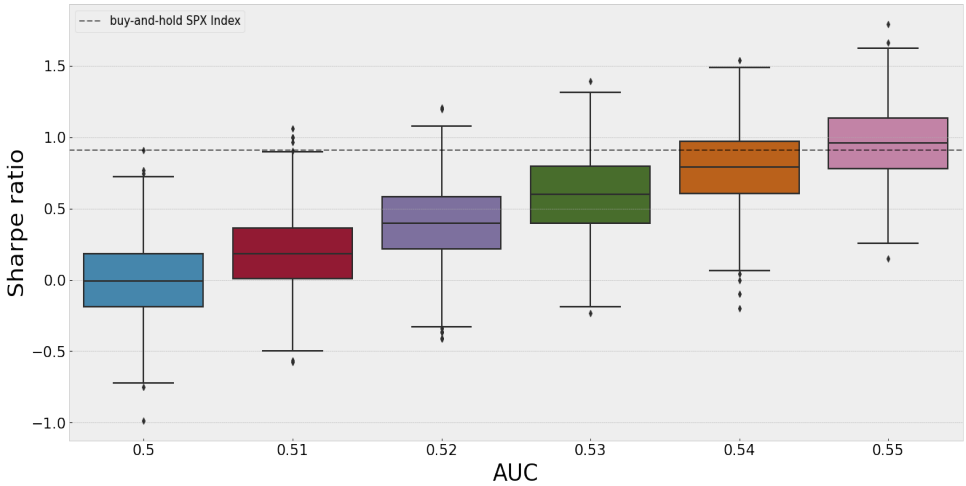

Figure 3 – For a given level of AUC, we have simulated different strategies for the S&P 500 and computed it's corresponding Sharpe ratio. The variance is induced by the fact that the order that the corresponding hit rate of a given level AUC have been realized is important for the final Sharpe ratio.

4.2.1 Forecasting Perspective

We summarize the results for each combination of feature selection method and forecasting model in terms of AUC in Table 6 and Table 7. Each element of the tables corresponds to the median AUC across all U.S. stock indices tested for a given combination of feature selection methods (rows) and forecasting models (columns). Furthermore, Table 6 provides the in-sample AUC for each combination, whereas Table 7 provides the out-of-sample AUC.

The first thing that we can notice comparing both tables is that the maximum AUC drops significantly between both tables. The maximum in-sample AUC is 61.8%, whereas it is 50.8% in Table 7. The IAMB stands out among the feature selection methods within the in-sample data with an average AUC of 59%. It is followed by the HUANG method, which achieves an AUC of 55%. Moreover, regardless of the feature selection method, there is no apparent winner when it comes to the best forecasting model. The ENET model achieves the highest average AUC of 55% across the feature selection methods and U.S. stock indices, followed by the RIDGE, which achieves an AUC of 54%.

When it comes to analyzing the results for the out-of-sample data, they become even less clear. Figure 4 reports the boxplot for the forecasting models. We can see that the LASSO, ENET, RIDGE, and the LOGIT models have a median AUC above the random classifier (a.k.a. random guess). Despite this, the lower quartiles of the three methods are below the random guess, and none of the forecasting models appear to have the lower quartile above the random guess. Figure 5 reports the same result but for the feature selection methods. Once again, there is no clear gain when comparing the median AUC for the feature selection methods compared to the random guess. Methods such as the IAMB, GRANGER and MDA have a median AUC above the random guess. They have lower quartiles far below the same benchmark.

The results shown in the previous paragraphs raise the question: When it comes to the S&P 500 index studied by Huang et al. (2020), can we find a model that consistently improves the AUC of the random guess? Figures 6 and 7 try to answer precisely this question. Figure 6 reports the boxplot for the forecasting models and the S&P 500 index. Although none of the models seem to have a median AUC above 51%, the LASSO and the ENET models have a median AUC reasonably above the random guess (50.57%, 50.37% respectively). Furthermore, both of the methods have lower bound standard deviations above the random guess. Models such as the LOGIT and the RIDGE also have a median AUC above the random guess. In line with the results found by Huang et al. (2020), the penalized regression models seem to outperform the random guess in terms of AUC for the S&P 500. Figure 7 reports the same result for the feature selection models. The methods that stand out are the HUANG and IAMB, which perform a median AUC of 51.77% and 50.8%, respectively. We also found that the lower bound standard deviations for both methods are above the random guess highlighting the consistency of the methods. Furthermore, methods such as the MDI and MDA also have a median AUC above the random guess. Again, we found results in line with the work of Huang et al. (2020), that is, the feature selection method HUANG is the best performer across all other methods.

4.2.2 Portfolio Perspective

We have previously highlighted that when it comes to using a classifier's predictions to build a profitable quantitative strategy, it is not enough to evaluate its performance based on

	LOGIT	RIDGE	LASSO	ENET	RF	GB	NN3
ALL	48.8	49.5	49.1	49.6	49.5	48.8	50.1
SFI	56.4	56.5	55.0	59.4	52.0	51.7	53.1
MDI	50.8	50.3	51.0	51.1	50.5	49.6	50.2
MDA	50.4	50.5	49.5	50.6	50.5	50.3	50.5
GRANGER	54.8	54.8	55.3	55.0	51.8	51.4	52.5
HUANG	56.7	56.7	56.7	56.8	53.1	53.0	53.9
IAMB	61.8	61.8	61.8	61.8	55.6	54.7	57.9
MMMB	56.4	56.4	56.4	56.4	53.3	53.2	51.6

Table 6 – Median in-sample AUC over all stock indices for all combinations of Feature Selection method with the Forecasting methods.

	LOGIT	RIDGE	LASSO	ENET	RF	GB	NN3
ALL	50.0	49.9	50.1	50.4	49.5	50.1	49.5
SFI	50.8	50.0	50.3	50.4	48.5	49.6	50.0
MDI	49.7	50.1	50.3	49.6	49.2	49.1	49.8
MDA	50.4	50.2	50.3	49.8	50.1	49.6	49.8
GRANGER	50.5	50.5	50.6	50.6	49.5	49.7	49.8
HUANG	50.1	50.1	50.1	50.1	49.6	50.1	49.6
IAMB	50.5	50.5	50.4	50.5	50.1	49.6	50.4
MMMB	49.2	49.2	49.2	49.2	49.1	49.6	49.6

Table 7 – Median out-of-sample AUC over all stock indices for all combinations of Feature Selection method with the Forecasting methods.

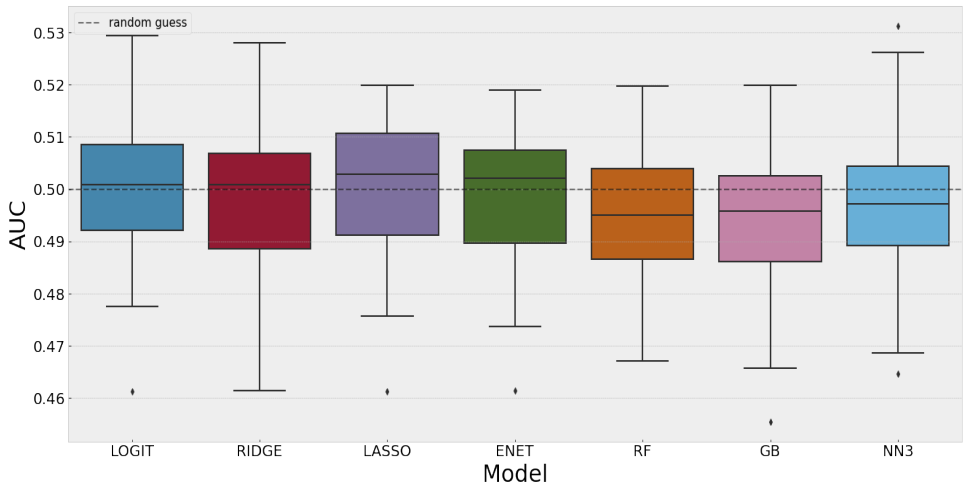

Figure 4 – Boxplot for the out-of-sample AUC aggregating across all U.S. stock indices and forecasting models.

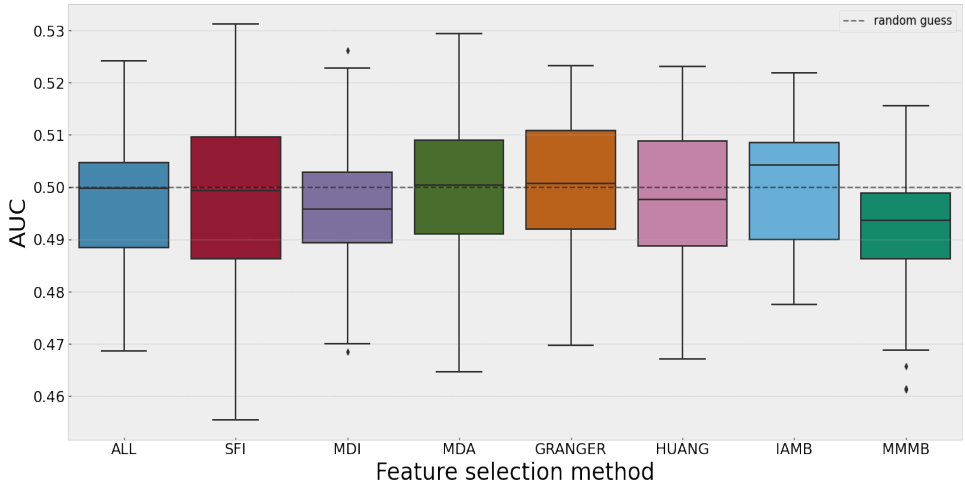

Figure 5 – Boxplot for the out-of-sample AUC aggregating across all U.S. stock indices and feature selection methods.

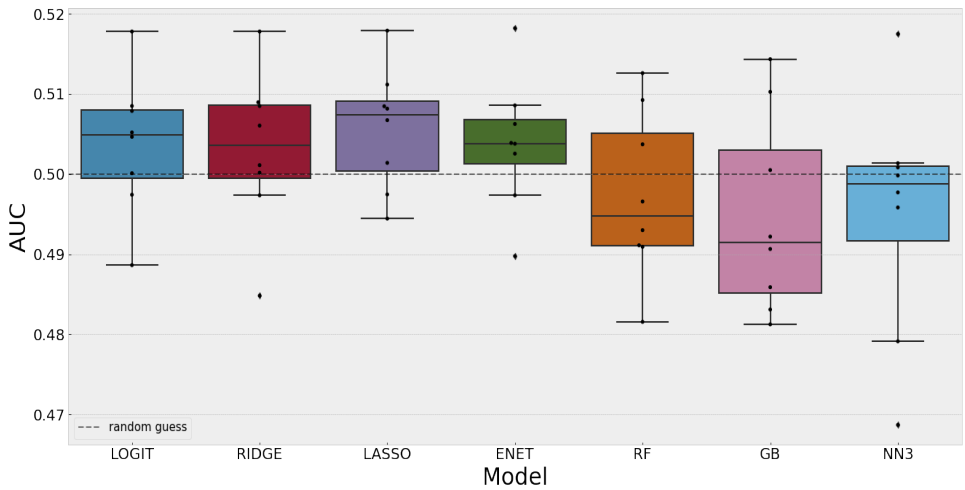

Figure 6 – Boxplot for the out-of-sample AUC for the S&P 500 index aggregating across all feature selection methods.

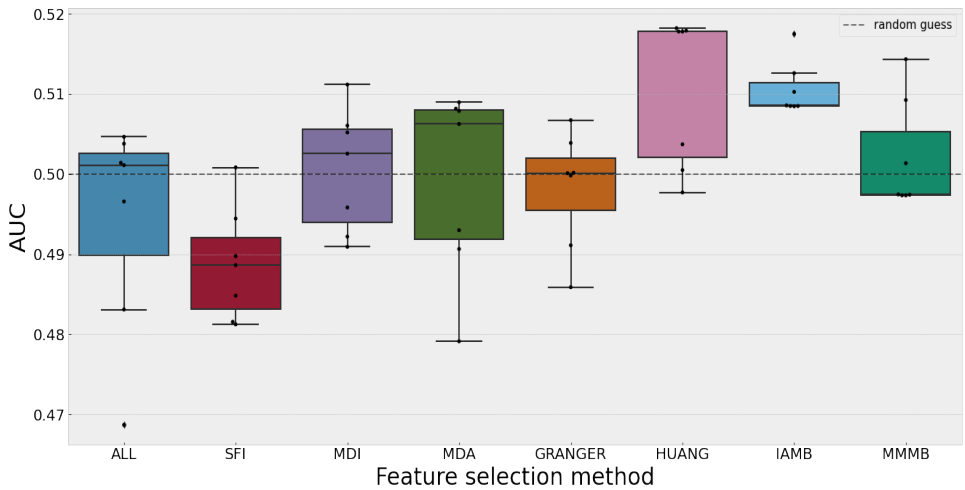

Figure 7 – Boxplot for the out-of-sample AUC for the S&P 500 index aggregating across all forecasting models.

prediction error metrics alone. To fill this gap, we evaluate in this section the feature selection and forecasting models based on a risk-return metric known as the Sharpe ratio. To compute the Sharpe ratio, we build a portfolio that uses the binary results (zeros and ones) predicted by each combination of models to simulate a trading strategy that buys or sells the underlying stock index. With the returns of each strategy based on the combinations of models at hand, we compute the Sharpe ratio as

$$SR_k = \frac{\mu_k}{\sigma_k} \times \sqrt{252} \tag{4.1}$$

for each combination of feature selection and forecasting models $k = 1, ..., K$, where μ_k is the sample average, and σ_k is the sample standard deviation. The Sharpe ratio tells us how many return units an investor is earning for each unit of risk.

We present Figure 8 to illustrate the returns of a portfolio based on the predictions of our models. In particular, we set the feature selection model to ALL and plot the cumulative returns for a strategy that times the S&P 500 index based on the predictions of each forecasting model. The black line is the cumulative return of the buy-and-hold strategy on the S&P 500. We computed the cumulative returns assuming the reinvestment of gains. We plot the compounded returns of $1 unit of money invested at the beginning of the out-of-sample data. The cumulative returns for all breaks of feature selection and forecasting models can be found in Appendix A.

In Table 8 we present the Sharpe ratio for a trading strategy that uses the predictions of each pair of feature selection and forecasting model to anticipate the direction of the 12 stock indices. The portfolio buys or sells the indices with equally weighted position sizes. That is, it buys or sells $\frac{1}{12}$ units of the indices at hand based on the predictions of each pair of feature

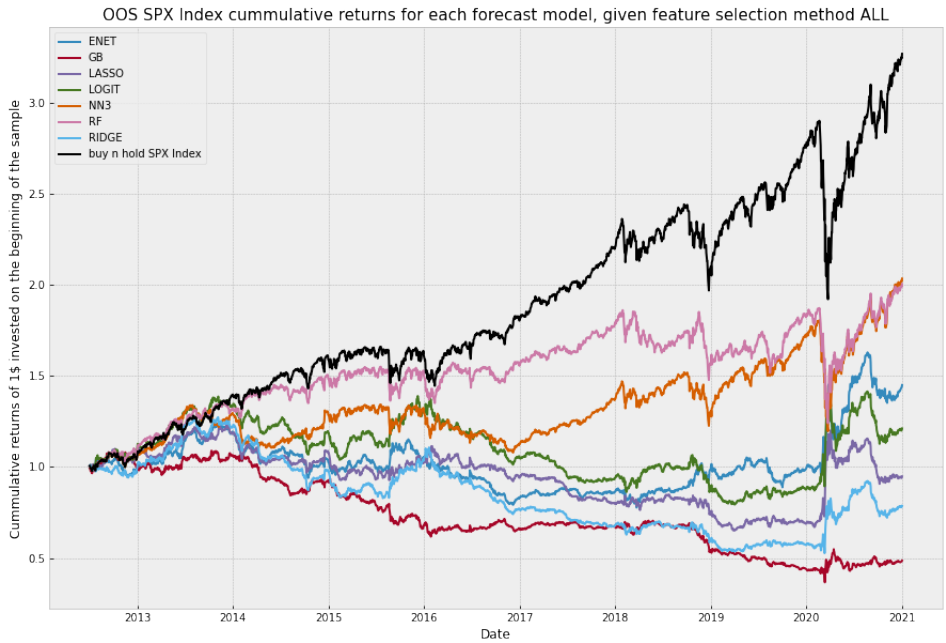

Figure 8 – Cumulative returns of $1 unit of money invested in a strategy that times the S&P 500 based on the predictions of each of the forecasting models and given the method ALL as the feature selection method.

selection and forecasting model in the table.

The comparisons of Table 8 will be based on the benchmark models represented by the acronym ALL and LOGIT and against the median Sharpe ratio of portfolios that buy-and-hold $\frac{1}{12}$ units for each of the 12 stock indices (this median Sharpe ratio is 0.881). We notice that the only negative Sharpe ratio is produced by the ALL and RIDGE model, our control group. Regarding the forecasting method, we see that the RF and NN3 stand out as the best performers in the Sharpe ratio across all feature selection methods. In particular, the combination RF and HUANG achieve the highest Sharpe ratio, followed by the LOGISTIC and HUANG (0.716), LASSO and HUANG (0.714), and RIDGE and HUANG (0.714). Moreover, the HUANG feature selection method appears to be consistent across forecasting models and stock indices. One interesting result is that the LOGIT model, along with the HUANG as the feature selection model achieves a Sharpe ratio that approximates those of the best performers. When comparing the best combination of models (HUANG+RF) with the Sharpe ratio of the aggregated portfolio, we found that the timed indices perform significantly worse than the buy-and-hold strategy. Thus, none of the combinations of models tested was able to beat the buy-and-hold strategy.

We present the Sharpe ratios for each of the U.S. stock indices in Appendix B to break down our results further. In particular, Table 9 presents the Sharpe ratio for each combination of feature selection method and forecasting model given the S&P 500 index. Note that the Sharpe

Portfolio of Indices	LOGIT	RIDGE	LASSO	ENET	RF	GB	NN3
ALL	0.143	-0.126	0.030	0.334	0.605	0.129	0.669
SFI	0.064	0.210	0.006	0.155	0.661	0.011	0.638
MDI	0.025	0.070	0.100	0.218	0.713	-0.018	0.657
MDA	0.097	0.081	0.143	0.228	0.714	0.054	0.509
GRANGER	0.396	0.396	0.664	0.480	0.635	0.140	0.688
HUANG	0.716	0.716	0.714	0.684	**0.717**	0.367	0.708
IAMB	0.172	0.172	0.177	0.160	0.620	0.256	0.503
MMMB	0.616	0.616	0.640	0.625	0.690	0.575	0.641

Table 8 – Median out-of-sample Sharpe ratio over all stock indices for all combinations of Feature Selection methods with Forecasting methods. To put the results into context, the median Sharpe ratio for each strategy based on the combination of models that times the portfolio of $\frac{1}{12}$ units of each stock index is **0.881**.

S&P 500	LOGIT	RIDGE	LASSO	ENET	RF	GB	NN3
ALL	0.218	-0.084	0.047	0.344	0.571	-0.421	0.581
SFI	-0.163	-0.324	-0.041	-0.037	0.666	-0.348	0.444
MDI	-0.088	-0.023	0.644	0.687	0.741	-0.196	0.643
MDA	0.590	0.637	0.621	0.867	0.678	-0.306	0.386
GRANGER	0.211	0.207	0.433	0.374	0.449	0.148	0.703
HUANG	0.727	0.727	0.724	0.641	**0.946**	0.360	0.772
IAMB	0.245	0.245	0.245	0.278	0.850	0.332	0.755
MMMB	0.555	0.555	0.616	0.555	0.676	0.621	0.753

Table 9 – Median out-of-sample Sharpe ratio for all combinations of Feature Selection methods and Forecasting methods for a given stock index. To put the results into context, the median Sharpe ratio for the buy-and-hold of the S&P 500 is **0.914**.

ratio achieved by the buy-and-hold strategy in our out-of-sample period is 0.914. We can see that, in line with the results found for the aggregated portfolio of stock indices, the RF and NN3 models appear to have the highest Sharpe ratios, on average about 0.7. In particular, the combination of RF and HUANG achieves the highest Sharpe ratio (0.94) among all the combinations tried, which slightly outperforms the buy-and-hold benchmark. Furthermore, the feature selection method proposed by Huang et al. (2020) achieves the highest mean Sharpe ratio. Despite this fact, both the combinations tested in their work (HUANG + RIDGE/LASSO/ENET) have Sharpe ratios significantly smaller than the buy-and-hold portfolio.

4.3 DISCUSSION

In the previous section, we analyzed the performance of several combinations of feature selection and forecasting models. We broke down our analysis into two perspectives: the prediction perspective and the portfolio perspective. Despite this breakdown, both perspectives point

to similar results.

Comparing all out-of-sample predictions in terms of the AUC, we could not find evidence that any of the combinations tested can significantly outperform the random guess. Even when comparing the feature selection methods against the method with all features (ALL), we could not find any significant difference in terms of median and comparing the variance of the AUC in the boxplot. Some methods seem to reduce the overall variance (MMMB and MDI) but cannot outperform the random guess in terms of median. The same conclusions can be drawn from the forecasting models. We find that none of the methods could outperform consistently in terms of median AUC or the random guess.

We can further compare the AUC for the S&P 500 index, as is done by Huang et al. (2020). We find that when conditioning on the S&P 500, methods such as the IAMB and HUANG were able to achieve a median significantly higher than the random guess and the control method ALL. Furthermore, when comparing forecasting models, we find that RIDGE, LASSO, and ENET were able to beat the random guess but are close to the LOGIT model, which is similar to what Huang et al. (2020) found.

We also compare the combinations of models in terms of the Sharpe ratio. When comparing the Sharpe ratio that buy-and-hold an equally weighted portfolio of the 12 stock indices, we found that none of the combinations achieved a higher Sharpe ratio than the passive strategy, which achieves a Sharpe ratio of 0.881. In terms of feature selection methods, HUANG, MMMB, and GRANGER were significantly better than the control method, but could not outperform the buy-and-hold. In terms of forecasting models, the RF and the NN3 were able to outperform the LOGIT model. Despite this, the best combination was RF+HUANG, which achieved a Sharpe ratio of 0.717 against 0.881 for the buy-and-hold.

Finally, when breaking down the equally weighted combination portfolio to condition only on the S&P 500, we find that the combination HUANG+RF slightly outperform the buy-and-hold strategy of the S&P 500, achieving a Sharpe ratio of 0.946 against 0.914, respectively. Despite this, none of the other combinations have Sharpe ratios above 0.77, in line with the results reported before. In particular, we found that the methods tested in Huang et al. (2020), which are HUANG + RIDGE/LASSO/ENET, achieve a maximum Sharpe ratio of 0.727 in our out-of-sample data, which is significantly worse than the passive strategy.

5 FINAL CONSIDERATIONS

To begin our final considerations, we recall the main goals of this project, which were:

- To test whether either individually of any combination of feature selection and forecasting models was able to achieve better performance than the benchmarks and, in a more general way;
- Answer if the Google Trends dataset would help us to build profitable trading strategies across U.S. stock indices.

When comparing the predictive performance of the feature selection methods and forecasting models for the S&P 500 stock index, we found evidence of outperformance in AUC and Sharpe ratio. In particular, we found similar results to the ones found in Huang et al. (2020). In other words,forecasting models such as the Logistic regression and penalized regression models (Ridge, LASSO, and Elastic Net) significantly improved the AUC performance compared to the random guess and compared to their peer models. Furthermore, feature selection methods such as the method proposed by Huang et al. (2020), the IAMB, and the MMMB algorithm were also able to outperform the random guess.

We expanded the analysis proposed by Huang et al. (2020) and computed the Sharpe ratio for a simulated trading strategy which buy/sells the S&P 500 index based on the predictions of each combination of models. Furthermore, the benchmark for each trading strategy was the Sharpe ratio of the buy-and-hold strategy. In contrast to the results in terms of AUC, we found that none of the feature selection methods nor the forecasting models were able to outperform the buy-and-hold strategy in terms of the Sharpe ratio. One exception to this was the combination of feature selection methods proposed by Huang et al. (2020) and the Random Forest model, which achieved a Sharpe ratio of 0.94 against 0.91 of the benchmark.

We further extended the results of Huang et al. (2020) by having conducted the same comparison as previously but for a large set of U.S. stock indices. The main goal of our analysis was to provide a robustness check of the results presented before. We highlighted that if Google Trends data is indeed a good proxy for investors attention and the hypothesis that attention does affect stock trading provided by Barber and Odean (2007) is true. The effects found in the context of the S&P 500 must hold for the other indices.

We found evidence that, when comparing aggregated AUC results across all twelve U.S. stock indices, none of the feature selection models or forecasting models could significantly improve the random guess. Moreover, we found the same results in terms of the Sharpe ratio. We achieved the highest Sharpe ratio by combining feature selection methods proposed by Huang

et al. (2020) and the Random Forest model, which had a Sharpe of 0.717 against 0.88 of the buy-and-hold of the portfolio of indices.

REFERENCES

(2007). Chapter 4 - causal feature selection. In H. Liu and H. Motoda (Eds.), *Computational Methods of Feature Selections*, Volume 1 of *Computational Methods of Feature Selections*, pp. 63–83. Routledge.

aaaa.

Ackert, L. F., L. Jiang, H. S. Lee, and J. Liu (2016). Influential investors in online stock forums. *International Review of Financial Analysis 45*, 39–46.

Albuquerque, R. and C. Vega (2008, 08). Economic News and International Stock Market Co-movement*. *Review of Finance 13*(3), 401–465.

Ang, A. (2014). *Asset Management: A Systematic Approach to Factor Investing*. Oxford University Press.

Barber, B. M. and T. Odean (2007, 12). All That Glitters: The Effect of Attention and News on the Buying Behavior of Individual and Institutional Investors. *The Review of Financial Studies 21*(2), 785–818.

Bergmeir, C., R. J. Hyndman, and B. Koo (2018). A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics Data Analysis 120*, 70–83.

Blei, D. M., A. Y. Ng, and M. I. Jordan (2003, March). Latent dirichlet allocation. *J. Mach. Learn. Res. 3*(null), 993–1022.

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

Campbell, J. Y., A. W. Lo, and A. MacKinlay (1997). *The Econometrics of Financial Markets*. Princeton University Press.

Cerqueira, V., L. Torgo, and I. Mozetič (202). Evaluating time series forecasting models: an empirical study on performance estimation methods. *Machine Learning 109*, 70–83.

Chan, W. S. (2003). Stock price reaction to news and no-news: drift and reversal after headlines. *Journal of Financial Economics 70*(2), 223–260.

Choi, H. and H. Varian (2012). Predicting the present with google trends. *Economic Record 88*(s1), 2–9.

Connor, G. and R. Korajczyk (1988). Risk and return in an equilibrium apt: Application of a new test methodology. *Journal of Financial Economics 21*(2), 255–289.

Curme, C., T. Preis, H. E. Stanley, and H. S. Moat (2014). Quantifying the semantics of search behavior before stock market moves. *Proceedings of the National Academy of Sciences 111*(32), 11600–11605.

Da, Z., J. Engelberb, and P. Gao (2011). In search of attention. *The Journal of Finance 66*(5), 1461–1499.

de Prado, M. L. (2018). *Advances in Financial Machine Learning* (1st ed.). Wiley Publishing.

Elliott, G. and A. Timmermann (2016). *Economic Forecasting* (1 ed.). Princeton University Press.

Engelberg, J. E. and C. A. Parsons (2011). The causal impact of media in financial markets. *The Journal of Finance 66*(1), 67–97.

Fama, E. F. and K. R. French (1992). The cross-section of expected stock returns. *The Journal of Finance 47*(2), 427–465.

Fama, E. F. and K. R. French (1993). Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics 33*(1), 3–56.

Fama, E. F. and J. D. MacBeth (1973). Risk, return, and equilibrium: Empirical tests. *Journal of Political Economy 81*(3), 607–636.

Fang, L. and J. Peress (2009). Media coverage and the cross-section of stock returns. *The Journal of Finance 64*(5), 2023–2052.

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

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics 29*(5), 1189–1232.

Granger, C. W. J. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica 37*(3), 424–438.

Gu, S., B. Kelly, and D. Xiu (2020, 02). Empirical Asset Pricing via Machine Learning. *The Review of Financial Studies 33*(5), 2223–2273.

Guyon, I., S. Gunn, M. Nikravesh, and L. A. Zadeh (2006). *Feature Extraction: Foundations and Applications (Studies in Fuzziness and Soft Computing)*. Berlin, Heidelberg: Springer-Verlag.

Hastie, T., R. Tibshirani, and J. Friedman (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. New York, NY, USA: Springer New York Inc.

Hoerl, A. E. and R. W. Kennard (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics 42*(1), 80–86.

Huang, M. Y., R. R. Rojas, and P. D. Convery (2020). Forecasting stock market movements using google trend searches. *Empirical Economics 59*(6), 1435–8921.

Kaplan, E. L. and P. Meier (1958). Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association 53*(282), 457–481.

Koijen, R. S. and S. Van Nieuwerburgh (2011). Predictability of returns and cash flows. *Annual Review of Financial Economics 3*(1), 467–491.

Koller, D. and N. Friedman (2009). *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press.

Kryzanowski, L. and H. Zhang (1992). Economic forces and seasonality in secirity returns. *Review of Quantitative Finance and Accounting 2*(3), 227–244.

Lehmann, B. N. and D. M. Modest (1988). The empirical foundations of the arbitrage pricing theory. *Journal of Financial Economics 21*(2), 213–254.

Li, J., K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu (2018, Jan). Feature selection. *ACM Computing Surveys 50*(6), 1–45.

Masini, R. P., M. C. Medeiros, and E. F. Mendes (2021). Machine learning advances for time series forecasting.

Medeiros, M. C. and H. F. Pires (2021). The proper use of google trends in forecasting models.

Mertons, R. C. (1987). A simple model of capital market equilibrium with incomplete information. *The Journal of Finance 42*(3), 483–510.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12*, 2825–2830.

Peng, L. and W. Xiong (2006). Investor attention, overconfidence and category learning. *Journal of Financial Economics 80*(3), 563–602.

Preis, T., H. S. Moat, and H. E. Stanley (2013). Quantifying trading behavior in financial markets using google trends. *Scientific Reports 3*(1), 2045–2322.

Rapach, D. and G. Zhou (2013). Chapter 6 - forecasting stock returns. In G. Elliott and A. Timmermann (Eds.), *Handbook of Economic Forecasting*, Volume 2 of *Handbook of Economic Forecasting*, pp. 328–383. Elsevier.

Ross, S. A. (1976). The arbitrage theory of capital asset pricing. *Journal of Economic Theory 13*(3), 341–360.

Sims, C. A. (2003). Implications of rational inattention. *Journal of Monetary Economics 50*(3), 665–690. Swiss National Bank/Study Center Gerzensee Conference on Monetary Policy under Incomplete Information.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological) 58*(1), 267–288.

Tsamardinos, I. and C. F. Aliferis (2003, 03–06 Jan). Towards principled feature selection: Relevancy, filters and wrappers. In C. M. Bishop and B. J. Frey (Eds.), *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, Volume R4 of *Proceedings of Machine Learning Research*, pp. 300–307. PMLR. Reissued by PMLR on 01 April 2021.

Tsamardinos, I., C. F. Aliferis, and A. Statnikov (2003a). Time and sample efficient discovery of markov blankets and direct causal relations. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, New York, NY, USA, pp. 673–678. Association for Computing Machinery.

Welch, I. and A. Goyal (2007, 03). A Comprehensive Look at The Empirical Performance of Equity Premium Prediction. *The Review of Financial Studies 21*(4), 1455–1508.

Yarotsky, D. (2017). Error bounds for approximations with deep relu networks. *Neural Networks 94*, 103–114.

Yu, K., X. Guo, L. Liu, J. Li, H. Wang, Z. Ling, and X. Wu (2019). Causality-based feature selection: Methods and evaluations.

Yuan, Y. (2015). Market-wide attention, trading, and stock returns. *Journal of Financial Economics 116*(3), 548–564.

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

APPENDICES

APPENDIX A – CUMMULATIVE RETURNS FOR THE S&P 500 GROUPING BY FORECAST MODEL AND FEATURE SELECTION METHODS

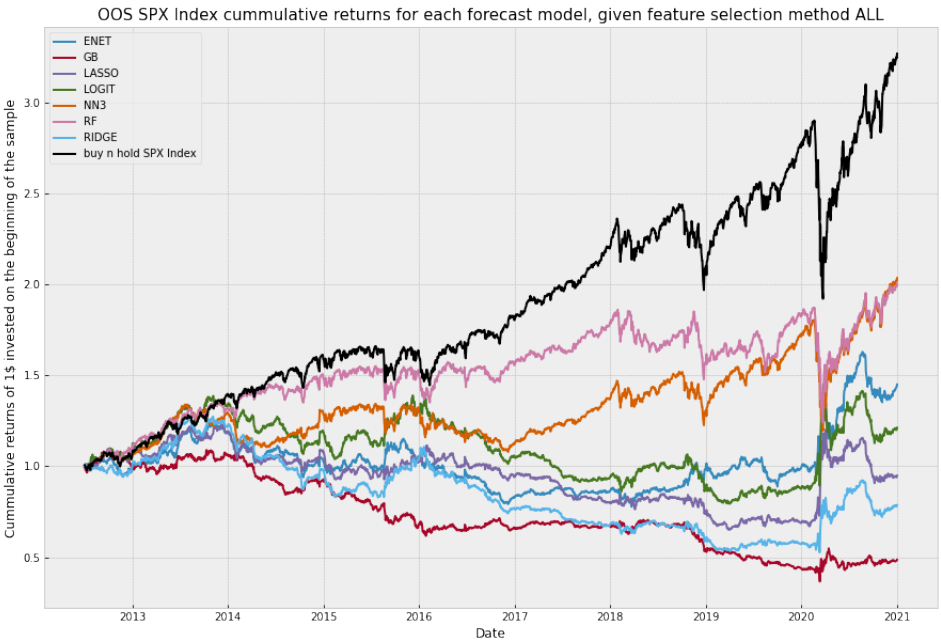

Figure 9 – Out-of-sample cummulative returns for each forecasting model given the feature selection method ALL.

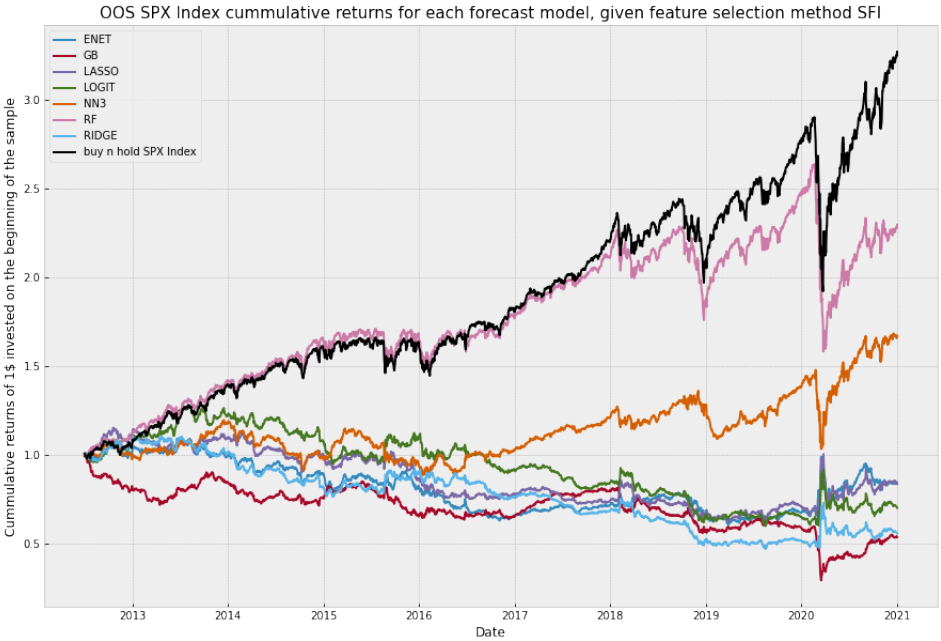

Figure 10 – Out-of-sample cummulative returns for each forecasting model given the feature selection method SFI.

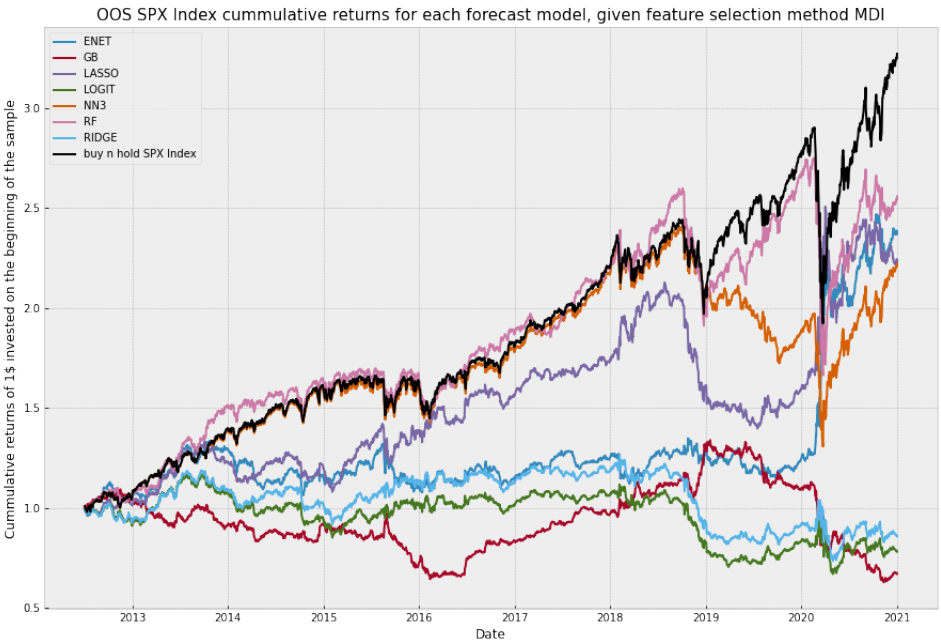

Figure 11 – Out-of-sample cummulative returns for each forecasting model given the feature selection method MDI.

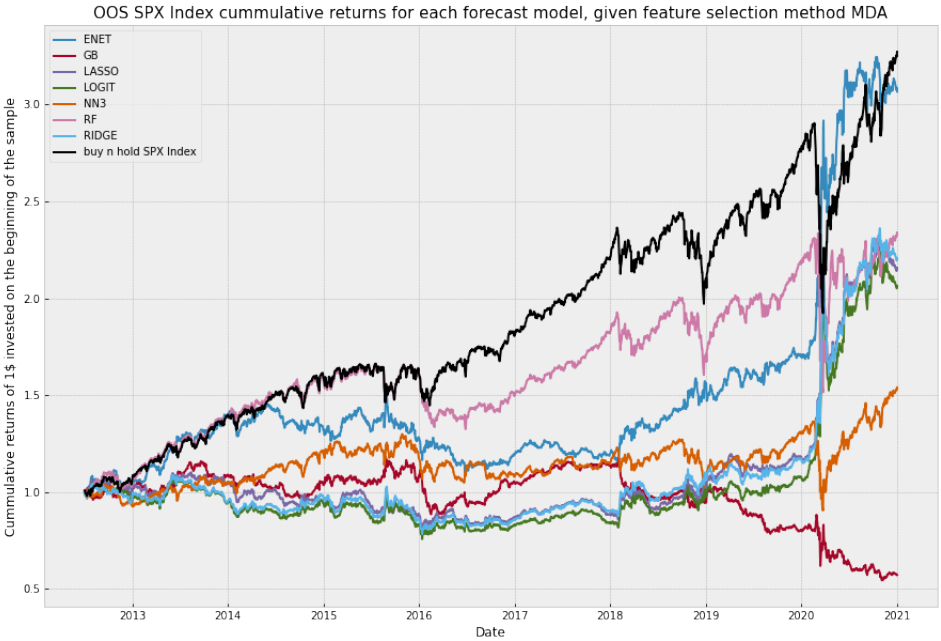

Figure 12 – Out-of-sample cummulative returns for each forecasting model given the feature selection method MDA.

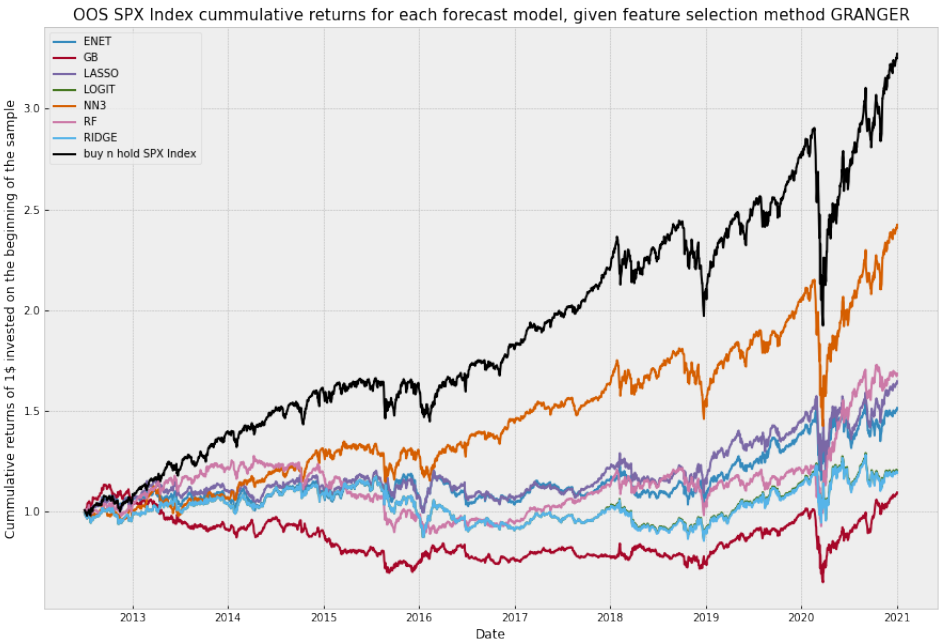

Figure 13 – Out-of-sample cummulative returns for each forecasting model given the feature selection method GRANGER.

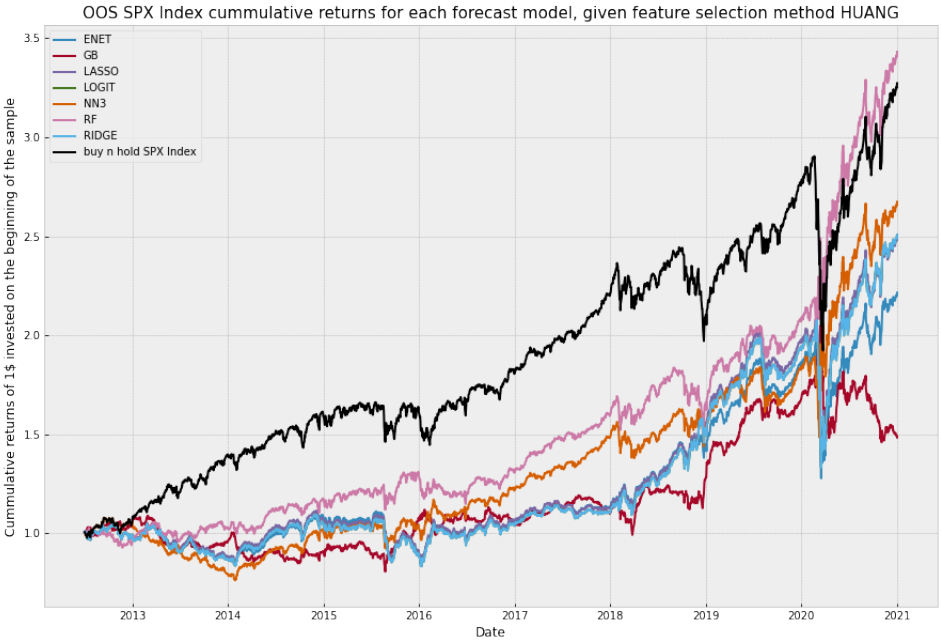

Figure 14 – Out-of-sample cummulative returns for each forecasting model given the feature selection method HUANG.

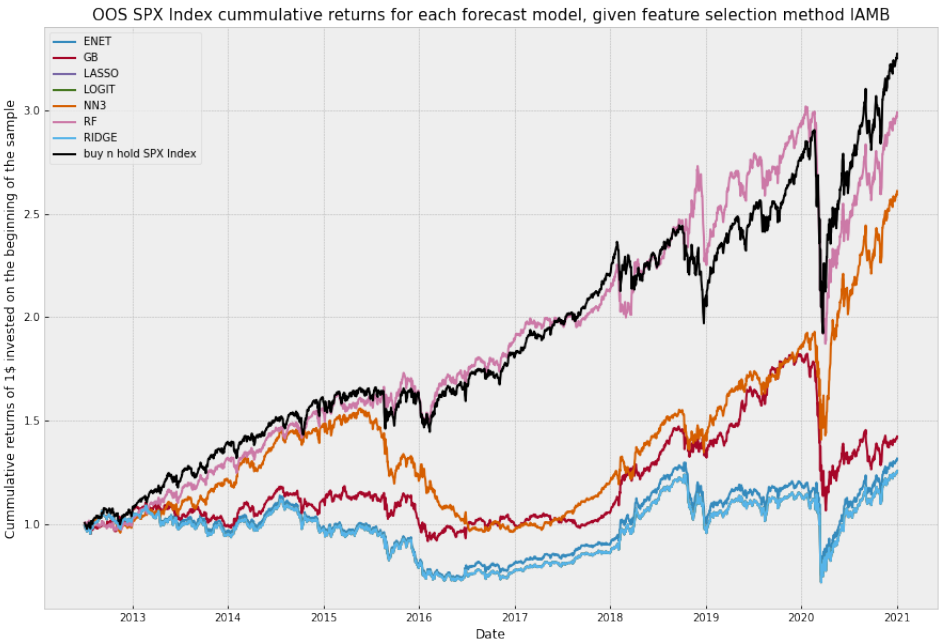

Figure 15 – Out-of-sample cummulative returns for each forecasting model given the feature selection method IAMB.

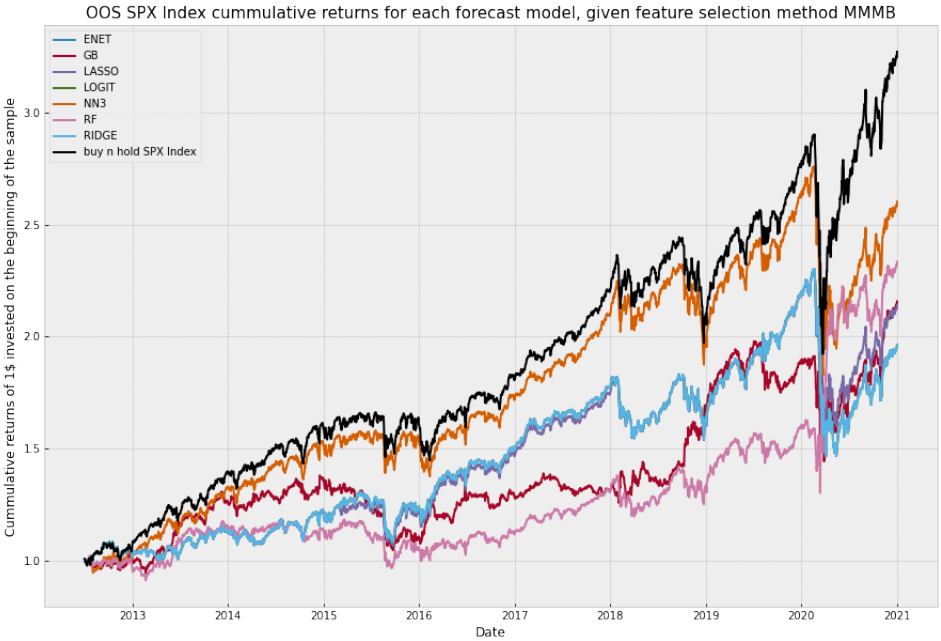

Figure 16 – Out-of-sample cummulative returns for each forecasting model given the feature selection method MMMB.

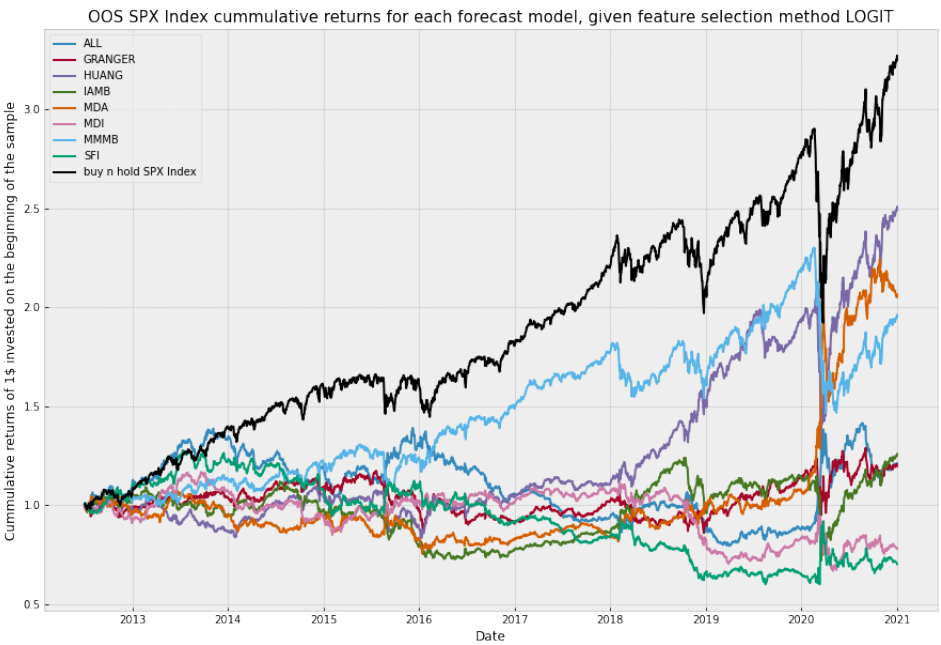

Figure 17 – Out-of-sample cummulative returns for each feature selection method given the forecasting model LOGIT.

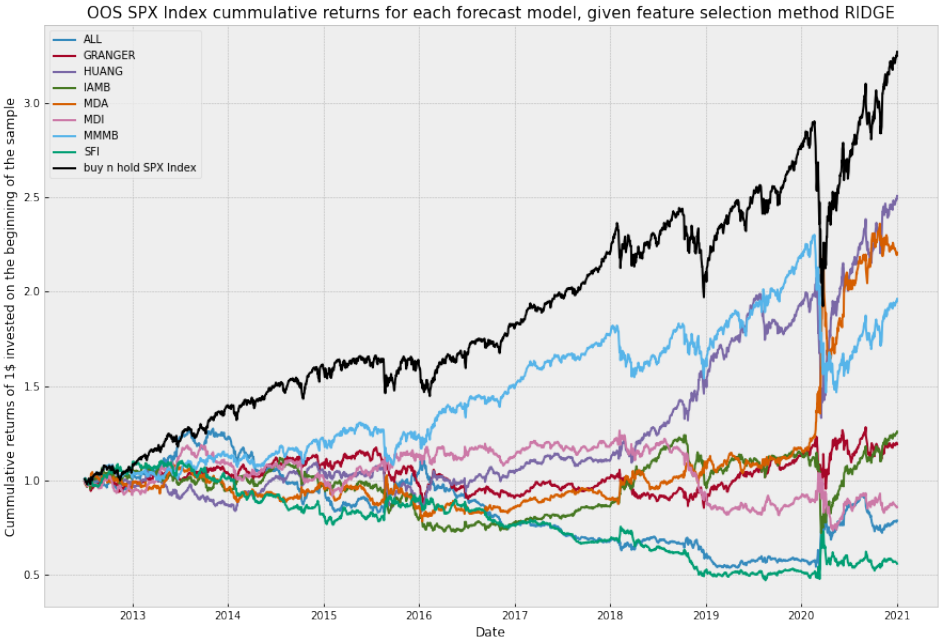

Figure 18 – Out-of-sample cummulative returns for each feature selection method given the forecasting model RIDGE.

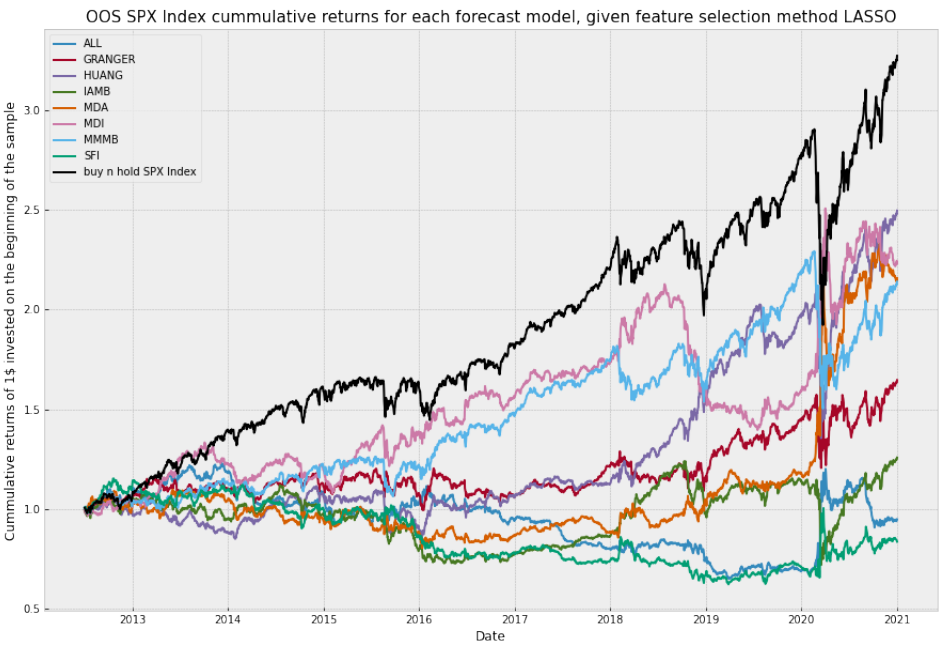

Figure 19 – Out-of-sample cummulative returns for each feature selection method given the forecasting model LASSO.

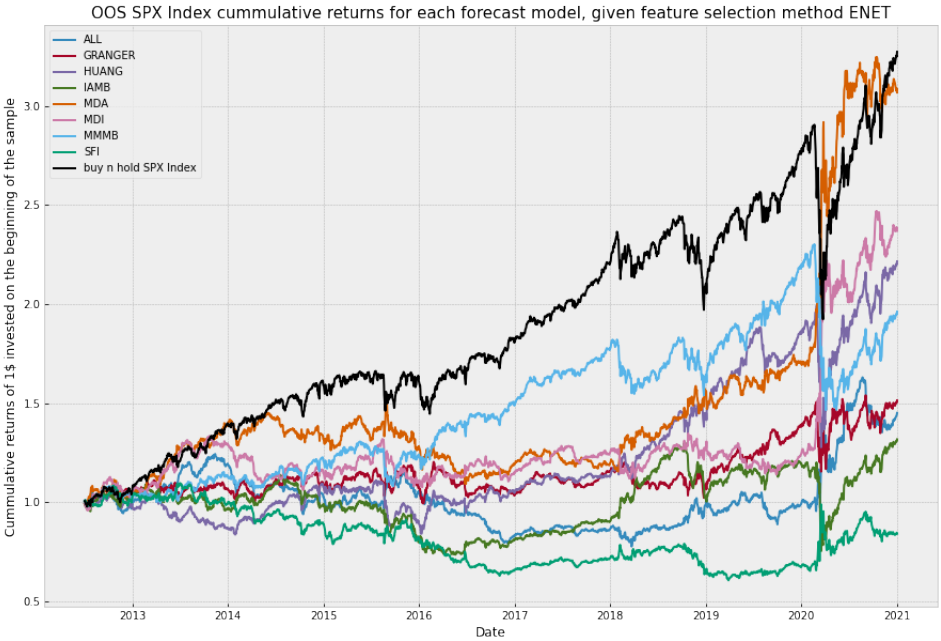

Figure 20 – Out-of-sample cummulative returns for each feature selection method given the forecasting model ENET.

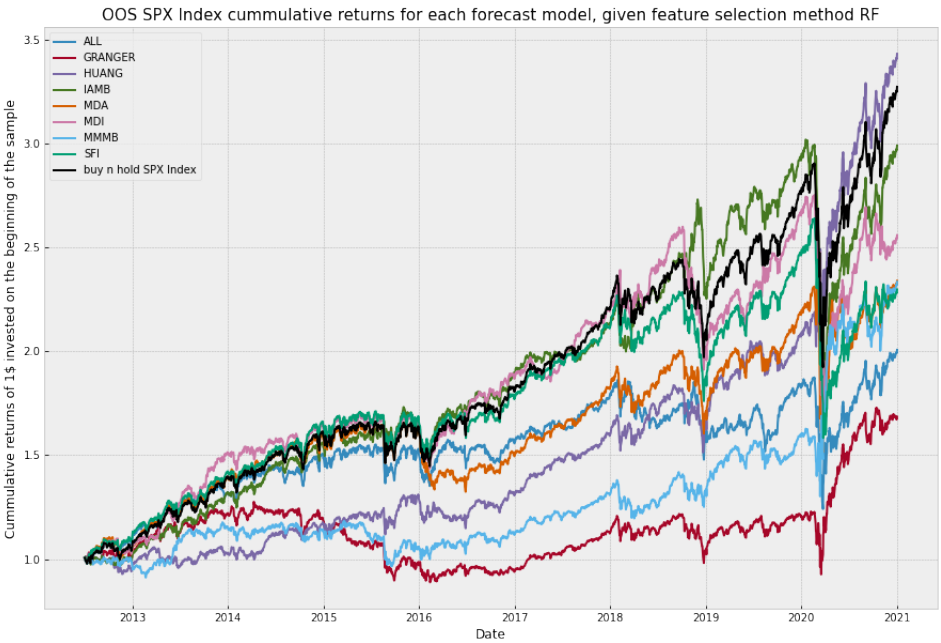

Figure 21 – Out-of-sample cummulative returns for each feature selection method given the forecasting model RF.

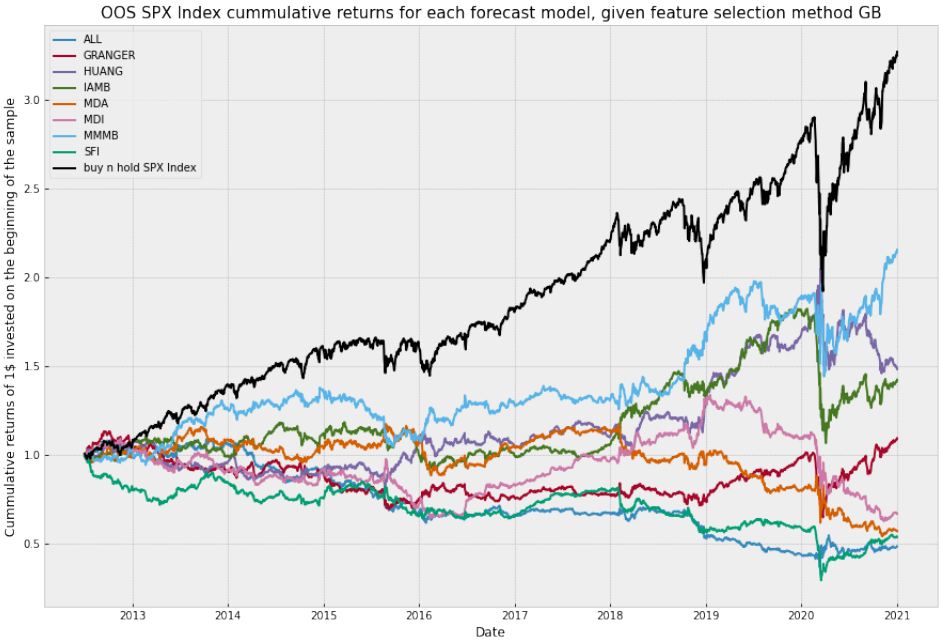

Figure 22 – Out-of-sample cummulative returns for each feature selection method given the forecasting model GB.

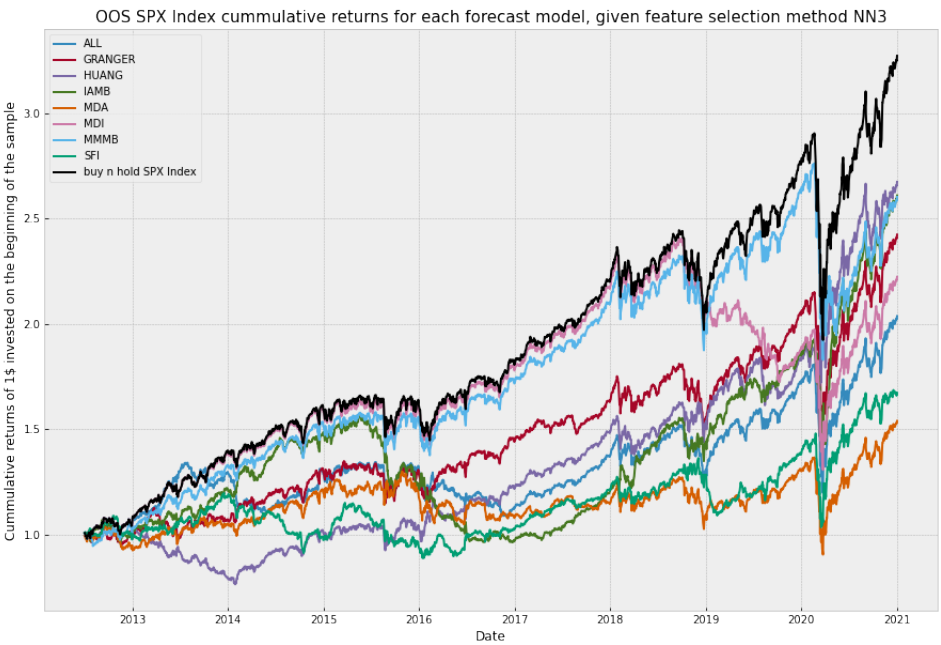

Figure 23 – Out-of-sample cummulative returns for each feature selection method given the forecasting model NN3.

APPENDIX B – MEDIAN SHARPE RATIO FOR ALL COMBINATIONS OF FEATURE SELECTION AND FORECASTING MODELS, GIVEN THE UNDERLYING STOCK INDEX.

S&P Communications	LOGIT	RIDGE	LASSO	ENET	RF	GB	NN3
ALL	-0.250	-0.197	0.013	0.323	0.889	0.517	1.017
SFI	0.141	0.200	0.070	0.108	**1.390**	0.492	0.879
MDI	-0.136	-0.203	-0.667	0.118	0.821	-0.306	1.017
MDA	-0.142	0.040	0.017	0.160	0.989	-0.802	0.865
GRANGER	-0.346	0.125	-0.303	-0.146	**1.084**	-0.202	0.673
HUANG	0.041	0.038	0.045	0.045	0.964	-0.503	0.824
IAMB	0.172	0.172	0.128	0.153	0.852	0.481	0.487
MMMB	0.702	0.702	0.704	0.697	0.723	0.559	0.606

Table 10 – Median out-of-sample Sharpe ratio for all combinations of Feature Selection methods and Forecasting methods for a given stock index. To put the results in context, the Sharpe for the buy-and-hold for the S&P Communications index is 1.057. The Sharpe values in bold are better than the buy-and-hold .

S&P Tech	LOGIT	RIDGE	LASSO	ENET	RF	GB	NN3
ALL	-0.298	-0.246	-0.218	0.244	0.884	0.262	1.106
SFI	-0.013	0.377	-0.296	-0.184	1.118	-0.222	0.759
MDI	0.549	0.313	0.447	0.380	1.040	0.085	1.112
MDA	-0.210	-0.398	-0.303	-0.085	**1.270**	0.611	0.954
GRANGER	0.393	0.392	0.683	0.510	1.181	0.395	1.003
HUANG	1.083	1.102	1.138	1.001	**1.340**	0.132	0.955
IAMB	0.172	0.172	0.172	0.151	0.540	0.085	0.938
MMMB	0.801	0.801	0.818	0.801	0.704	0.812	1.220

Table 11 – Median out-of-sample Sharpe ratio for all combinations of Feature Selection methods and Forecasting methods for a given stock index. To put the results in context, the Sharpe for the buy-and-hold fot he S&P Technologies index is 1.25. The Sharpe values in bold are better than the buy-and-hold.

S&P Consumer Non-cyclical	LOGIT	RIDGE	LASSO	ENET	RF	GB	NN3
ALL	0.085	-0.168	-0.476	0.532	1.155	0.076	0.772
SFI	0.281	0.333	0.445	0.409	1.108	0.222	0.759
MDI	0.267	0.314	0.132	0.474	1.154	0.010	1.146
MDA	-0.160	-0.261	0.044	0.297	1.077	0.172	1.180
GRANGER	0.920	0.920	0.917	0.920	**1.284**	0.243	0.781
HUANG	0.839	0.839	0.841	0.839	1.144	0.037	0.758
IAMB	0.934	0.951	0.932	0.930	0.884	0.938	0.649
MMMB	**1.273**	**1.273**	**1.253**	**1.273**	**1.337**	**1.278**	**1.203**

Table 12 – Median out-of-sample Sharpe ratio for all combinations of Feature Selection methods and Forecasting methods for a given stock index. To put the results in context, the Sharpe for the buy-and-hold fot he S&P Consumer index is 1.193. The Sharpe values in bold are better than the buy-and-hold.

S&P Utilities	LOGIT	RIDGE	LASSO	ENET	RF	GB	NN3
ALL	0.459	0.216	0.373	0.080	0.459	0.349	0.366
SFI	**0.887**	**0.675**	**1.192**	**0.927**	**0.781**	0.452	0.089
MDI	0.375	0.388	0.639	0.242	**0.974**	-0.046	0.520
MDA	0.493	0.186	0.610	**1.032**	0.534	0.069	0.574
GRANGER	0.400	0.400	0.332	0.403	0.625	0.084	0.580
HUANG	**0.761**	**0.761**	**0.758**	**0.761**	**0.735**	0.505	0.626
IAMB	0.141	0.113	0.056	0.139	0.623	0.249	0.331
MMMB	0.581	0.581	0.629	0.595	0.434	0.204	0.525

Table 13 – Median out-of-sample Sharpe ratio for all combinations of Feature Selection methods and Forecasting methods for a given stock index. To put the results in context, the Sharpe for the buy-and-hold fot he S&P Utilities index is 0.673. The Sharpe values in bold are better than the buy-and-hold.

Nasdaq	LOGIT	RIDGE	LASSO	ENET	RF	GB	NN3
ALL	-0.450	-0.316	-0.207	0.098	0.809	0.108	**1.088**
SFI	0.393	0.864	0.236	0.641	0.591	0.529	1.011
MDI	-0.094	-0.143	0.187	0.238	0.685	-0.101	0.486
MDA	-0.492	-0.354	-0.582	0.031	0.405	0.322	**1.112**
GRANGER	0.908	0.946	0.943	0.923	0.915	0.306	0.907
HUANG	0.975	0.955	1.024	0.979	**1.167**	0.526	**1.103**
IAMB	0.444	0.444	0.480	0.444	0.980	0.930	1.027
MMMB	0.651	0.651	0.651	0.654	**1.095**	0.535	0.864

Table 14 – Median out-of-sample Sharpe ratio for all combinations of Feature Selection methods and Forecasting methods for a given stock index. To put the results in context, the Sharpe for the buy-and-hold fot he Nasdaq index is 1.083. The Sharpe values in bold are better than the buy-and-hold.

S&P Industrials	LOGIT	RIDGE	LASSO	ENET	RF	GB	NN3
ALL	0.200	0.207	0.527	0.807	0.451	0.704	0.666
SFI	0.474	0.220	0.049	0.451	0.534	0.163	0.645
MDI	0.086	0.172	0.072	0.056	0.779	0.231	0.330
MDA	0.554	0.656	0.451	**0.918**	0.790	0.437	0.912
GRANGER	0.789	0.742	0.752	0.785	**1.152**	0.411	**0.955**
HUANG	0.794	0.822	0.800	0.793	0.688	0.601	0.853
IAMB	-0.167	-0.167	-0.166	-0.156	0.646	-0.311	0.663
MMMB	**0.942**	**0.942**	**0.942**	**0.942**	0.872	0.839	**0.973**

Table 15 – Median out-of-sample Sharpe ratio for all combinations of Feature Selection methods and Forecasting methods for a given stock index. To put the results in context, the Sharpe for the buy-and-hold fot he S&P Industrials index is 0.916. The Sharpe values in bold are better than the buy-and-hold.

S&P	LOGIT	RIDGE	LASSO	ENET	RF	GB	NN3
ALL	0.218	-0.084	0.047	0.344	0.571	-0.421	0.581
SFI	-0.163	-0.324	-0.041	-0.037	0.666	-0.348	0.444
MDI	-0.088	-0.023	0.644	0.687	0.741	-0.196	0.643
MDA	0.590	0.637	0.621	0.867	0.678	-0.306	0.386
GRANGER	0.211	0.207	0.433	0.374	0.449	0.148	0.703
HUANG	0.727	0.727	0.724	0.641	**0.946**	0.360	0.772
IAMB	0.245	0.245	0.245	0.278	0.850	0.332	0.755
MMMB	0.555	0.555	0.616	0.555	0.676	0.621	0.753

Table 16 – Median out-of-sample Sharpe ratio for all combinations of Feature Selection methods and Forecasting methods for a given stock index. To put the results in context, the Sharpe for the buy-and-hold fot he S&P 500 index is 0.914. The Sharpe values in bold are better than the buy-and-hold.

S&P Financials	LOGIT	RIDGE	LASSO	ENET	RF	GB	NN3
ALL	0.279	0.565	0.470	**0.728**	**0.855**	-0.403	0.700
SFI	-0.204	-0.169	-0.188	0.314	**0.938**	0.193	0.083
MDI	0.218	0.162	0.099	0.199	0.540	-0.337	0.242
MDA	0.643	**0.716**	0.435	0.393	**0.818**	0.357	0.075
GRANGER	**0.927**	**0.930**	**0.915**	**0.919**	0.543	0.133	0.598
HUANG	0.483	0.483	0.483	0.483	0.386	-0.273	0.574
IAMB	0.064	0.062	0.064	0.080	0.617	0.505	0.520
MMMB	0.052	0.041	0.182	0.041	0.264	0.591	0.252

Table 17 – Median out-of-sample Sharpe ratio for all combinations of Feature Selection methods and Forecasting methods for a given stock index. To put the results in context, the Sharpe for the buy-and-hold fot he S&P Financials index is 0.715. The Sharpe values in bold are better than the buy-and-hold.

S&P Consumer Cyclical	LOGIT	RIDGE	LASSO	ENET	RF	GB	NN3
ALL	-0.433	-0.536	-0.493	0.749	0.481	0.134	0.371
SFI	-0.173	-0.218	-0.275	-0.543	0.478	-0.520	0.631
MDI	-0.036	-0.056	-0.069	0.005	0.399	0.379	0.671
MDA	-0.323	-0.172	-0.089	-0.159	0.750	-0.073	0.177
GRANGER	**0.896**	**0.890**	**0.907**	**0.885**	0.646	0.365	0.774
HUANG	0.705	0.705	0.705	0.727	0.699	0.373	0.624
IAMB	0.047	0.047	0.006	0.042	0.412	0.263	0.191
MMMB	0.757	0.757	0.757	0.765	0.793	0.720	0.575

Table 18 – Median out-of-sample Sharpe ratio for all combinations of Feature Selection methods and Forecasting methods for a given stock index. To put the results in context, the Sharpe for the buy-and-hold fot he S&P Consumer Cyclical index is 0.794. The Sharpe values in bold are better than the buy-and-hold.

Russel 2000	LOGIT	RIDGE	LASSO	ENET	RF	GB	NN3
ALL	-0.122	-0.283	-0.188	-0.191	0.639	0.300	0.673
SFI	-0.210	-0.016	-0.037	-0.044	0.655	-0.701	**0.700**
MDI	0.289	0.210	0.100	-0.224	0.532	0.188	**0.862**
MDA	0.145	0.122	0.253	-0.046	0.357	0.038	-0.398
GRANGER	-0.043	0.010	-0.036	0.026	-0.317	0.025	0.400
HUANG	0.287	0.287	0.287	0.302	0.494	**0.766**	0.610
IAMB	0.168	0.168	0.181	0.168	0.542	-0.384	0.195
MMMB	0.423	0.423	0.423	0.423	0.123	0.188	0.367

Table 19 – Median out-of-sample Sharpe ratio for all combinations of Feature Selection methods and Forecasting methods for a given stock index. To put the results in context, the Sharpe for the buy-and-hold fot he Russel 2000 index is 0.673. The Sharpe values in bold are better than the buy-and-hold.

S&P Energy	LOGIT	RIDGE	LASSO	ENET	RF	GB	NN3
ALL	**0.425**	**0.301**	0.049	0.027	0.058	-0.004	-0.030
SFI	-0.175	-0.287	-0.326	-0.348	-0.304	-0.296	-0.319
MDI	-0.176	-0.129	-0.247	**0.277**	-0.165	0.090	**0.221**
MDA	**0.797**	**0.564**	**0.242**	-0.441	**0.243**	-0.494	-0.079
GRANGER	**0.297**	**0.336**	**0.644**	**0.449**	-0.485	0.006	0.094
HUANG	**0.349**	**0.349**	**0.338**	**0.332**	0.064	0.031	**0.128**
IAMB	**0.347**	**0.347**	**0.339**	**0.337**	-0.027	-0.078	**0.418**
MMMB	**0.100**	**0.100**	**0.100**	**0.107**	-0.106	**0.154**	-0.137

Table 20 – Median out-of-sample Sharpe ratio for all combinations of Feature Selection methods and Forecasting methods for a given stock index. To put the results in context, the Sharpe for the buy-and-hold fot he S&P Energy index is 0.058. The Sharpe values in bold are better than the buy-and-hold.

S&P Basic Materials	LOGIT	RIDGE	LASSO	ENET	RF	GB	NN3
ALL	0.265	0.219	0.157	0.365	0.298	0.123	0.657
SFI	0.597	0.631	0.225	0.203	0.180	-0.141	0.072
MDI	-0.467	-0.407	-0.468	-0.547	0.051	-0.463	**0.720**
MDA	0.050	-0.127	-0.325	0.657	0.137	-0.041	0.443
GRANGER	0.349	0.349	0.394	0.346	0.336	-0.123	0.503
HUANG	0.639	0.639	0.639	0.639	0.545	0.596	0.657
IAMB	0.503	0.503	0.482	0.482	0.379	0.153	0.339
MMMB	0.378	0.378	0.378	0.378	0.150	0.355	**0.677**

Table 21 – Median out-of-sample Sharpe ratio for all combinations of Feature Selection methods and Forecasting methods for a given stock index. To put the results in context, the Sharpe for the buy-and-hold fot he S&P Basic Materials index is 0.663. The Sharpe values in bold are better than the buy-and-hold.