12-2021

Forecasting Stock Market Prices: A Machine Learning Approach

Abraham Alhomadi
Utah State University

Forecasting Stock Market Prices: A Machine Learning Approach

By

Abraham Alhomadi

Abstract:

There has been extensive literature written about the efficiency of the stock market. Practitioners and academicians have debated whether investors can exploit publicly available information to generate excess returns. Clearly predicting the stock market's return with high accuracy has been enormously difficult, but we are interested in contributing to the continuous exploration of the efficiency of the stock market using machine learning techniques. We also want to examine the relationship between our dataset's macroeconomic indicators and foreign nations' stock markets with our target feature—the S&P 500. In this paper, we will be using supervised machine learning models, like Linear Regression, Penalized Regression-Elastic Net, Support Vector Regression, Random Forest, and XGBoost models, to predict monthly stock market returns using historical data from 1992 to 2021. Our results show that it is difficult to forecast stock market returns with high accuracy using the monthly SP&500 monthly returns, and that if investors even rely on the high computational power of machine learning techniques to attempt to forecast stock market returns, they will likely end up making a high-risk bet and lose out substantially on their investment. We also report our dataset features' importance in relation to the U.S. SP&500 generated by our machine learning model.

Keywords: Stock Market Prediction; Stock Prices; XGBoost, machine learning, Linear Regression, Support Vector Regression, Linear Regression, time-series, ADF-test, Random Forest, Elastic Net, feature importance, efficient,

Acknowledgments:

I want to express my deepest gratitude and thanks to my committee advisors and academic professors Todd Griffith, Pedram Jahangiry, Tyler Brough, and Paul Fjeldsted for their guidance, support, and mentorship throughout my time at Utah State University and while working and completing this master's thesis research project. I want to further thank Professor Pedram Jahangiry for his selfless help and guidance with machine learning models and the Python code, as well as Professor Todd Griffith for his support and guidance during the data collection and data processing phase. Lastly, I want to thank Professors Tyler Brough and Professor Paul Fjeldsted for their academic support and mentorship throughout the semester. With their help, I am a better and a more capable financial economics student. To all my professors, thank you. I hope to carry your teachings to help make this world a better place in the future.

A special thank you goes out to my father and mother who selflessly, patiently, and admirably pushed me throughout my educational journey. Further, a special thank you goes to my remarkable, loving, and unwaveringly supportive wife, Mounia Alhumedi, who kept me strong and motivated throughout my journey. There is nothing I can do or give to reciprocate her support. I also want to send my love and gratitude to my daughters, Maryam and Hatune, whose presence in my life inspire me every single day. Without my family, I would not have been a better student and human being today.

Thank you to all my colleagues and friends for their support during my time at Utah State University.

Introduction

The stock market efficiency has been studied extensively in the past four decades. Academicians and investors have debated whether stock market returns can be predicted using historical data and various statistical methods. Prior to the introduction of machine learning techniques, academicians and investors relied on traditional statistical methods to attempt to forecast the future return of the stock market. Some of these predictive methods used were intrinsic value analysis, technical analysis, linear regression, and Autoregressive integrated moving average models (ARIMA) (Saurav Agrawal, 2019). These methods are still used today, but they have their own limitations and biases. Recently, academicians and investors started using machine learning techniques which have proven to have remarkable computational capabilities to make forward multiple periods predictions. These capabilities have introduced a new dimension in predictive modeling, whereby historical and high-dimensional data can be used to predict the stock market returns but with low accuracy.

This paper contributes to the existing literature that assess the predictability of stock market returns. We want to add more evidence to the literature evaluating stock market returns predictability and show that forecasting stock market returns with high accuracy—even with machine learning techniques, is extremely difficult due to the efficiency of the stock market. We also want to explore how each of our machine learning algorithms are going to rank the importance of each feature in our dataset in relation to the U.S. SP&500 stock market to gain valuable information about the stock market movement. We will be looking at and comparing the feature importance output of our linear and non-linear machine learning models.

It is clear from the theory of capital markets hypothesis developed by Eugene Fama that predicting the stock market is not possible when markets are efficient. An efficient market is one

where the stock market prices include all publicly and non-publicly available information. Writing in his notorious paper, "Efficient Capital Markets: A Review of Theory and Empirical Work", Fama states that there are three forms of market efficiency forms: the weak form, the semi-strong, and the strong form (Fama, 1970). All three forms describe different gradation of an efficient market where past stock market prices movements are independent and cannot be used to predict their future movement. According to Fama, in a weak-form efficient market, stock market prices cannot be predicted using publicly available information, such as historical prices. In a semi-strong-form efficient market, the theory states that this form incorporates the weak-form and adds that stock market prices immediately and fully reflect new public information, such as earning announcements or surprising events. It also states that fundamental/technical analysis will not be useful in predicting future price movement. Thus, if one tries to use this information to predict future stock market prices, the information would not add to the future price of the stock market, because the market has already priced in the information immediately after its release. In a strong-form efficient market, Fama states that the stock market reflects both public and private information. As a result, investors cannot exploit any information that would enable them to generate excess returns (Fama, 1970).

We write this paper under the belief that markets are efficient and that the form of efficiency markets reflect is the semi-strong form. We believe that the stock market reflects all publicly available information and consequently historical data cannot be used to predict future stock market returns. Nonetheless machine learning techniques have made it possible with their computational capabilities to use historical and high dimensional data to make multiple periods predictions and enable us to test the market's efficiency with their advanced computational capabilities. There is some literature that have utilized both supervised and unsupervised

machine learning techniques for stock market predictive modeling and found that the models can make predictions with some accuracy: (Usmani et al, 2016), (Deepak et al, 2017),(Vijh et al, 2019). We share the belief that even machine learning techniques have not been able to predict monthly stock market returns with high accuracy and we back up this belief in this paper.

This research topic will always be important for investors who seek to generate excess returns, make buy-sell decisions, and determine portfolio allocation, particularly considering development in machine learning techniques and algorithmic improvement. In an inefficient market, investors will have a difficult time achieving those ends because there will be volatility in the market. Volatility creates opportunities for gains and losses, but it makes investors' investment decision-making harder. This paper is contributing to this ongoing research of assessing stock market returns predictability and market efficiency. Even though we believe predicting stock market returns with high accuracy using monthly returns is difficult, investors can still use the paper's findings to help them guide their asset allocation, make buy-sell decisions knowing the U.S. stock market is efficient, and formulate optimal portfolios that best meet their clients' required return (Rossi, 2018). Plus, machine learning techniques are still helpful in providing insights into which predictor features, such as the ones used in this paper, are important in influencing the monthly stock market returns.

In this paper, we use five supervised machine learning models to predict stock market returns: Linear Regression, Elastic Net Regularization, Support Vector Regression, Random Forest, and XGBoost. These supervised machine learning models are described later in this paper. In linear regression, the algorithm seeks to estimate the parameters by fitting the model to the training data and using the parameters to make predictions about the target feature. In penalized regression-elastic net, the model seeks to build a less complex model by shrinking and

or eliminating features' coefficients that make the model needlessly complex. In essence, it seeks to regularize the model's coefficients, whereby the algorithm introduces more bias into the model while reducing the model's variance substantially (Jahangiry, 2021). In the support vector regression (SVR), the algorithm fits a line to the data similar, conceptually, to linear regression, except that in SVR the line is called the hyperplane. It identifies the hyperplane that has the maximum observations within the hyperplane boundary. In linear regression, the algorithm is seeking to minimize the variance between the real and predicted values. In SVR, the model "tries to fit the best line within a threshold value. The threshold value is the distance between the hyperplane and boundary line." (Raj, 2020).

In addition, the Random Forest (RF) algorithm is an ensemble learning method, where it combines the predictions from a collection of decision trees to produce more accurate and stable predictions. Put differently, rather than relying on one decision tree to make predictions, the RF algorithm takes the predictions of all subset of trees and based on the average predictions of those trees, it produces a final, optimal output (Jahangiry, 2021). In XGBoost, the algorithm tries to minimize the model's loss function by including weak learners using gradient descent. Gradient descent is an "iterative optimization algorithm for finding a local minimum of differentiable function. The contribution of each weak learner to the final prediction is based on a gradient optimization process to minimize the overall error of the stronger learner." (Gupta, 2021). To evaluate the performance of our models' prediction, we will use the root-mean-square error (RMSE) as our primary evaluation metric and include the R-squared as a secondary metric.

The paper will be presented in the following manner: In section I, we will describe our dataset and focus on data processing. In section II, we will revisit and briefly describe the machine learning models we use in this paper. In section III, we define the features in our dataset

and provide their summary statistics and correlation matrix. In section VI, we will report the empirical findings of our models. Lastly, in section V, we will provide concluding remarks and cite the references in this paper.

II. Data Description and Data Processing

The data used for this research paper was generated entirely from the Bloomberg Terminal. The dataset includes historical price data of 30 macroeconomic, stock markets' indices, U.S. treasury securities, and other key global financial indicators, like the U.S. dollar spot price. These are all supposed to serve as indicators to evaluate whether they predict the SP&500 monthly return. The features are all defined at a later section in this paper. In a world where world economies are integrated, we thought it would be useful to include global indicators like advanced and emerging nations' stock markets' monthly returns to glean insights into whether their movement have impact on the U.S.'s SP&500's monthly return. Using Bloomberg, we generated the monthly last price of each feature in the dataset, then computed the stock market return using the following arithmetic return formula:

Monthly Return = (Current Monthly Closing Price/Last Month Closing Price)-1

The monthly returns calculation enables us to compare features across the board using the same metric. This is even more helpful when we are dealing with a high dimensional time-series dataset with different features' values. Also, we could have calculated the logarithmic return of our features rather than using the arithmetic return to account for continuous compounding. However, when we did, the difference between the logarithmic returns and arithmetic returns were very small giving that we are calculating monthly returns and therefore we decided to stick to arithmetic returns. If we were calculating annual returns, we would have used logarithmic returns.

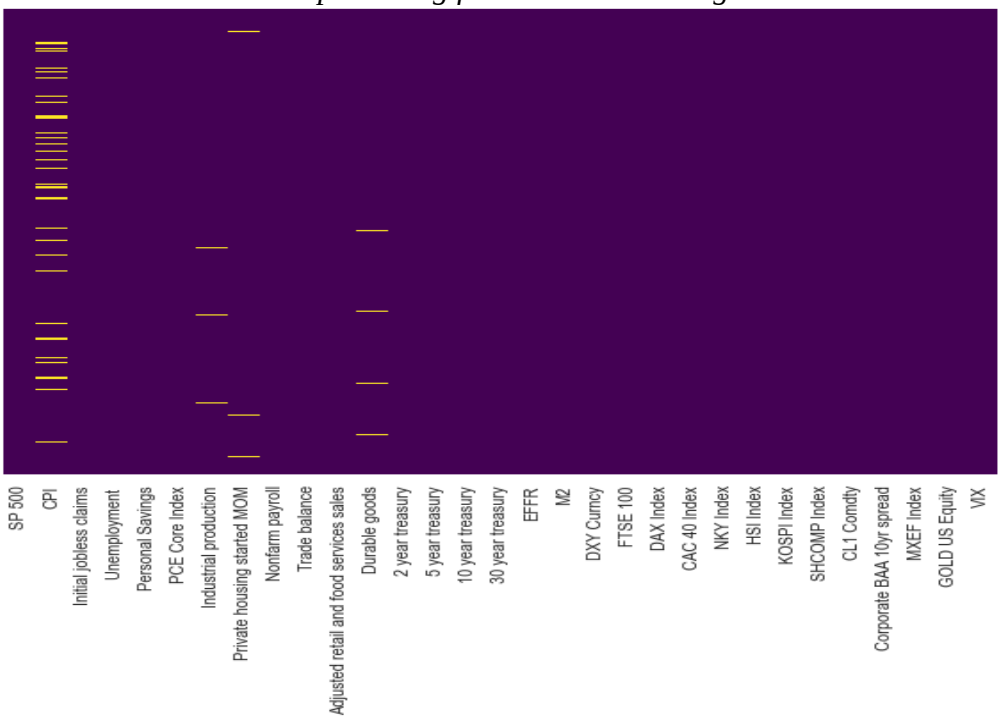
Moreover, when we generated the data, we at first wanted our sample period to extend from January 1980 through September 2021. One challenge we came across was that not all the features had data extending back to 1980. Across our 30 features, the availability of data varied for some features where some had data extending back to the 1990s or 2000s, while others had data for our entire sample period. We realized it would be an enormous task to continue with our original sample size, knowing that we do not have all the data. In fact, if we had filled the missing data with zero values, then this would have affected the integrity of our data and skewed our models' predictive findings. For example, inflation ("CPI") is one of our main macroeconomic features in the dataset. The variable contains many missing values in our sample data. If we simply substitute the missing monthly values with zeros, it will be inappropriate as the model will assume that inflation percent change for those particular months were constant. However, this is not the case and if we proceed with the substitution strategy, we would create a more complex problem than just finding the most appropriate way to handle missing values without diluting the predictability of our data or impacting the integrity of our models (Huey Fern Tay, 2021). To address this issue, we identified March 1992 to September 2021 as the appropriate period in which we have observations for the entire sample period.

Handling Missing values

After we reduced our sample size period, there were four features in our dataset that contained some missing values. Three of the features—durable goods, industrial production, and private housing started MOM—had less than five periods missing values, except CPI which had above 25 periods of monthly missing values. We did not want to drop our missing values, because it would

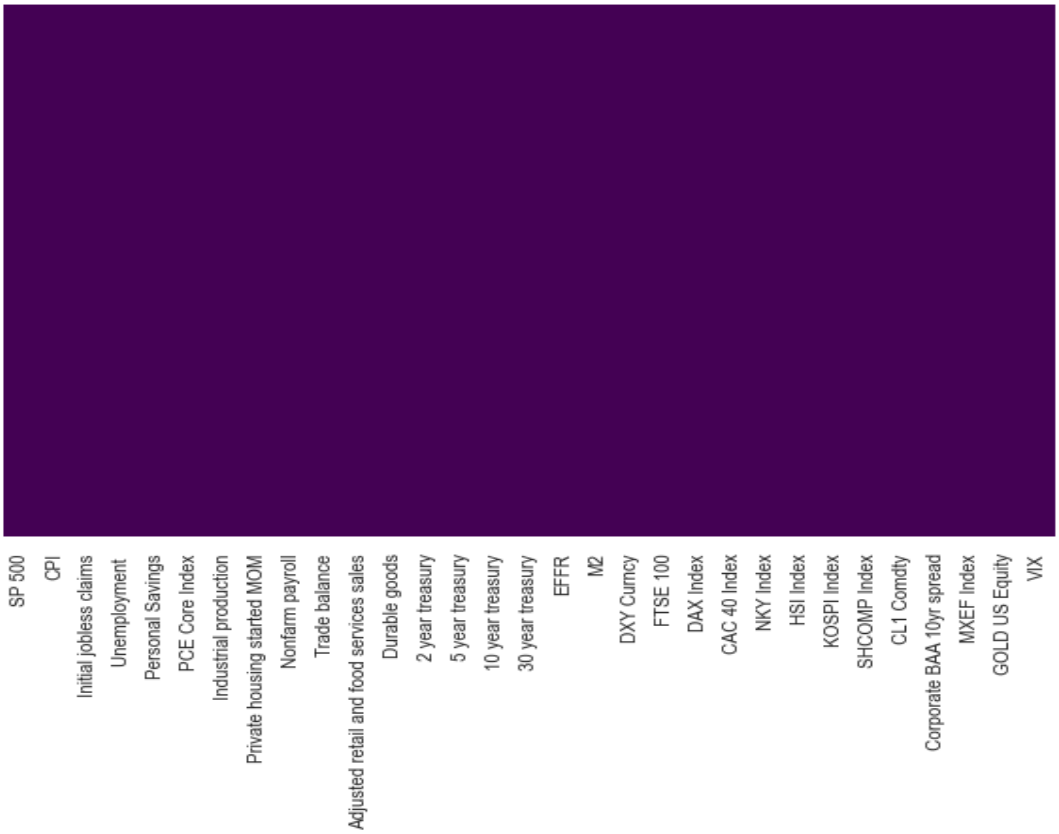
have further reduced our sample size and our models will lose important insights from other features' data.

A heatmap showing features with missing values



To tackle the missing values problem in our dataset, we decided to use the linear regression method to impute the missing values. We thought the regression method would provide more reliable values based on the relationship between our features with missing values and other features in our dataset. We also wanted to refrain from using the mean, media, and mode imputation method, because the method, first, may render values that can introduce bias into our dataset. Second, the method only looks at the variable itself and thus may come up with values that are not truly representative of trends in the dataset. For example, for our CPI variable with the most missing values, if we take the CPI data and find that the median percent monthly change is .08% but the actual missing value was supposed to be .02%, this will introduce bias into our dataset.

The heatmap showing our features after imputing missing values using regression



Additionally, in our dataset, we include key macroeconomic indicators that we think influence monthly stock returns. These are inflation, unemployment, labor force participation, and industrial production data that are tracked by asset managers and Wall Street analysts to attempt to forecast the future path of the SP&500. We also include key monetary policy indicators we think may be helpful in evaluating the stock market return, including the Effective Federal Funds Rate (EFFR), money velocity (M2), and the U.S. short- and long-term tenor treasuries. We understand that prediction has its uncertainty, but we think these indicators have helped financial economists in the past understand the future movement of the SP&500. Prior research on the relationship between macroeconomic indicators and stock market returns have been explored and the research findings indicate that there are key macroeconomic indicators, like the ones we include in our research, that

demonstrated the existence of a correlation between those features and stock markets' returns (Sirucek, Martin, 2012). Also see other research that have explored the relationship between macroeconomic features and stock market return in other countries (Saseela Balagobei, 2017); (Issahaku et al., 2013); and (Wisam Rohilina, 2009).

Mutual Information test

To examine how much information each of our features contributes in relation to our target feature, we decided to use the Mutual Information (MI) method from the Ski-Learn package to achieve that. The benefits of this method relative to other methods like Pearson Correlation is that MI captures both linear and non-linear relationships between our variables. Defining MI, Jason Brownlee states that, "Information Gain, or IG for short, measures the reduction in entropy or surprise by splitting a dataset according to a given value of a random variable. A larger information gain suggests a lower entropy or groups of samples, and hence less surprise." (Brownlee, 2019). Additionally, Halil Ertan puts it differently by stating, "[The Mutual Info] method calculates mutual information value for each of independent features with respect to dependent variable, and selects the ones which has most information gain" (Ertan, 2020). Essentially, the method looks at our data's parameters and assesses how much they contribute to explaining the target feature. The features with a high MI score will be ranked higher than those with a low MI score.

The data features with their respective MI score

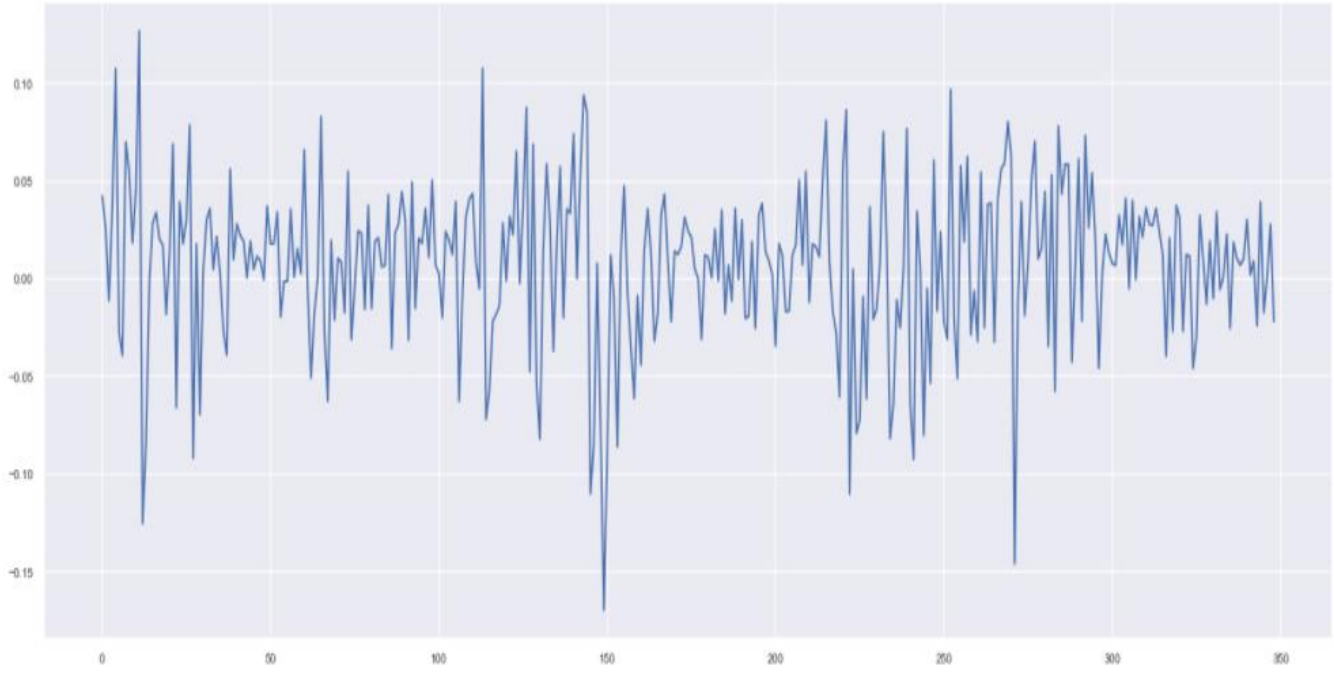
```
CAC 40 Index                          0.453441
DAX Index                             0.446547
FTSE 100                              0.433945
MXEF Index                            0.310398
VIX                                   0.276519
HSI Index                             0.230916
KOSPI Index                           0.226385
NKY Index                             0.204746
Corporate BAA 10yr spread             0.137713
Trade balance                         0.112323
EFFR                                  0.098226
CL1 Comdty                            0.075478
Nonfarm payroll                       0.070494
10 year treasury                      0.035407
PCE Core Index                        0.030813
5 year treasury                       0.027183
Initial jobless claims                0.023312
Industrial production                 0.019406
30 year treasury                      0.007811
Adjusted retail and food services sales    0.006944
GOLD US Equity                        0.003849
CPI                                   0.002214
Private housing started MOM           0.000000
Durable goods                         0.000000
SHCOMP Index                          0.000000
DXY Curncy                            0.000000
Personal Savings                      0.000000
Unemployment                          0.000000
M2                                    0.000000
2 year treasury                       0.000000
Name: MI Scores, dtype: float64
```

When we ran the MI test, some of our features' MI scores were zero, as shown in the above figure. We could have used the score for feature selection purposes and thus eliminated those with a zero MI score. However, we decided to keep them in our model because they will be useful for our Random Forest algorithm when it decides the optimal split of the features and selects the nodes that make up the trees. They may also provide insights collectively than they do individually (Aznar, 2021).

Time-Series Data Stationarity Test

In order for our predictive modeling analysis to work, our time-series data must be tested for stationarity. A stationarity data is one where the statistical properties of the data, like the average and variance, do not change with time. This test is important to understand the underlying trends behavior in our data and to produce effective predictive analysis (Kumar, 2021). We tested our dataset for stationarity using the Augmented Dickey-Fuller Test (ADF Test). The results are reported in the below figures and both show that our data does meet the stationarity test:

A visual of our stationarity test



Results of the ADF Test

```
ADF Statistic: -17.986949
p-value: 0.000000
Critical Values:
        1%: -3.449
        5%: -2.870
        10%: -2.571
```

It is evident from the plot above that no trend or seasonality can be detected. Plus, from the ADF computation figure above, we see that the p-value giving a 95% confidence level is below .05 and the ADF statistic is -17.99. Our critical value (t-stat) is -2.870, which is large giving a 95% confidence level. Therefore, we can safely say that our data is stationarity.

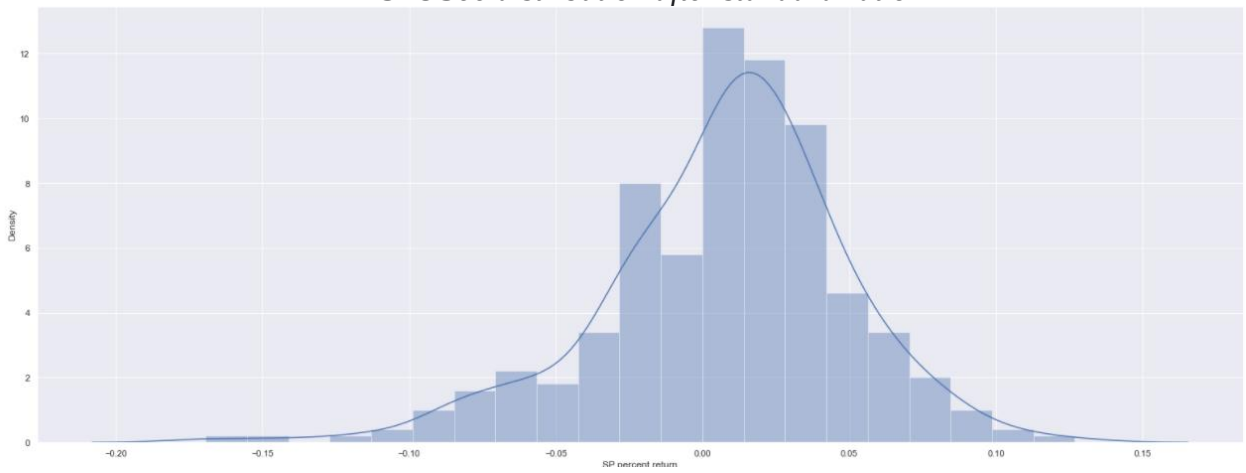
Standardization of the dataset:

There are 30 variables in our dataset with different unit measurements. If we proceed with our dataset without standardization, this will make comparability of results across our features and our models' results difficult. Also, standardizing the data helps speed up the computation of our machine learning algorithms. To standardize our datasets, we use the following formula:

$$\textbf{Standardized value} = X - \mu / \sigma$$

Where **X** represents each feature's observation, **μ** is the feature's average, and **σ** is the feature's standard deviation. This will "standardize the features around the center and 0 with a standard deviation of 1. Standardization assumes that [our] data has a Gaussian (bell curve) distribution. This does not strictly have to be true, but the technique is more effective if [our] attribute is Gaussian" (Lakshmanan, 2019). Below is the chart distribution of our target feature after standardization, which looks normally distributed with some skewness to the right:

SP&500 distribution after standardization



III. Machine Learning Methodology

The machine learning field is constantly evolving. There are many machine learning algorithms that have been used to analyze data and make predictions. For purposes of this paper, we will be employing supervised machine learning algorithms that we learned in our Machine Learning course. In exploring machine learning algorithms and the various research that have utilized ML models to analyze data, I realized that Neural Networks and Deep Learning algorithms are more preferred to some of the supervised models we use in this paper (Raut Sushrut Deepak et al., 2017) and (Adil, 2016). However, we believe that even our used supervised models can still be effective at providing information about the causal relationship between our features and the SP&500 monthly return.

With that, below we provide a brief summary of each of the machine learning algorithms employed in analyzing our time-series dataset:

Linear Regression:

Linear Regression is a supervised machine learning algorithm and the easiest to implement out of all supervised machine learning models. The algorithm takes historical data of one or more features called parameters and attempts to explain or predict one variable called the target variable. In linear regression, we use the following mathematical equation to explain relationship between the target and the parameters:

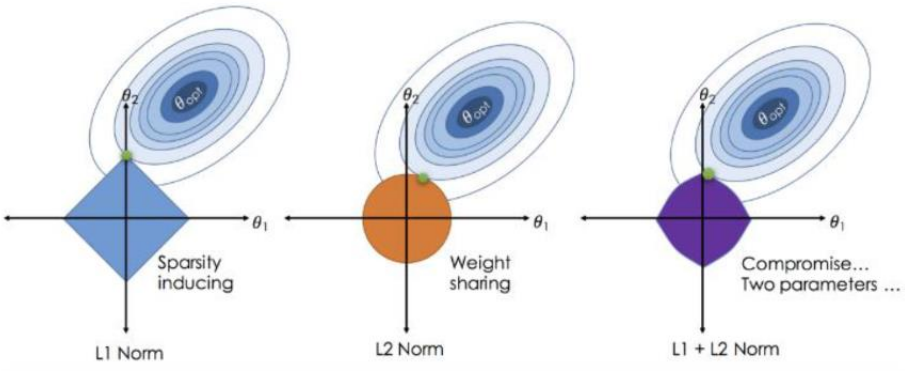
$$f_{w,b}(X) = WX + b$$

Where *fw,b(X)* is a "linear combination of features and parameterized by **W** and **b**" and **W** is a "D-dimensional vector of parameters" **b** is "a real number" **(**Jahangiry, 2021).

Simply put, the linear regression model estimates the parameters of our equation above by fitting the model to a training dataset. Once we have fitted the line and compared the trained and test datasets and evaluated our model's prediction performance using the **Root Mean Square Error (RMSE)** and the R squared metrics, we then estimate the RMSE and the R squared in the test-set using time series cross validation. Then we look at the RMSE to identify how much our model in the test set has improved. A lower RMSE means that cross validation helped improve the model prediction accuracy.

Elastic Net Regularization:

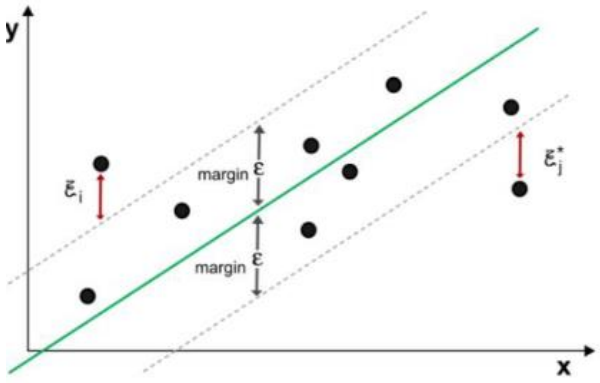
Elastic net is one variation of regularization in supervised machine learning. The other two variations are Ridge and Lasso. All of these methods are designed to shrink the coefficient estimates toward zero or make them zero to address the overfitting problem. In Ridge, the algorithm shrinks coefficients toward zero to make the model less complex, but coefficients are not eliminated. In LASSO, the algorithm actually eliminates unimportant coefficients to make the model less complex. In Elastic Net—which is the model we will be using, it is a combination of Ridge and LASSO, whereby the model seeks to minimize and or shrink coefficient estimates to zero that are not important in the model to achieve a balanced bias-variance tradeoff. These three techniques help reduce the complexity of the model by introducing some bias into the model to achieve large reduction in the model's variance (Pedram Jahangiry, 2021). The figure below provides a geometric illustration of how the algorithm in the background optimizes the function in LASSO on the left, Ridge in the middle, and Elastic Net Regression on the right:



Source: Pedram Jahaniry, Machine Learning Course, Fall 2021

Support Vector Regressors (SVR)

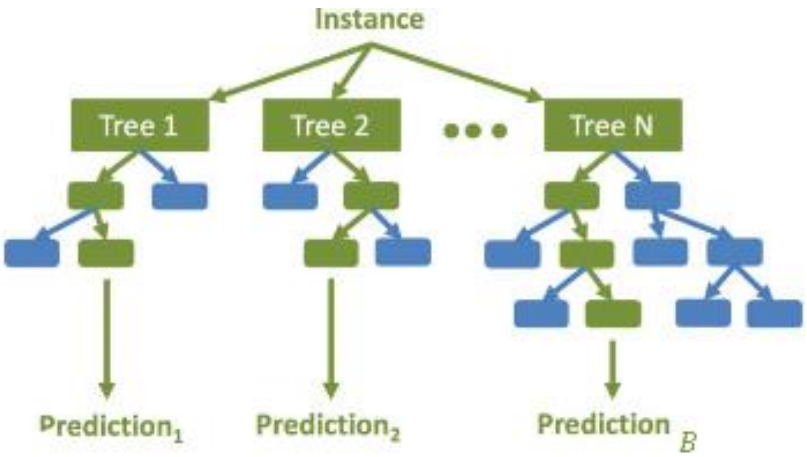
The SVR is another supervised machine learning algorithm that falls within the Support Vector Machine Algorithms family. The SVR is also used to analyze times series data and find the best fitted line into the data. The line that the SVR algorithm fits is considered the hyperplane that "has the maximum number of points" (Ashwin Raj, 2020). What makes SVR different from the linear regression algorithms is that SVR does not necessarily attempt to minimize the variances between the best fit line and the real data. Rather, in SVR, the algorithm tries to "fit the best line within a threshold value. The threshold value is the distance between the hyperplane and boundary line. (Ashwin Raj, 2020). Put differently, the objective is to come up with a hyperplane that has the maximum training observations within the margin ε, which represents the tolerance level (Jahangiry, 2021). See the below figure for illustration:



Source: Pedram Jahaniry, Machine Learning Course, Fall 2021

Random Forest Regression

The Random Forest Regression (RFR) is another supervised machine learning algorithm. Random Forest is an ensemble machine learning algorithm that falls within the tree-based algorithms family. The algorithm relies on multiple decision trees for learning the data and making decisions, such that it combines the output decisions of all the trees and produces one optimized decision output (Gurucharan M K, 2020). See the figure below as provided for illustration:

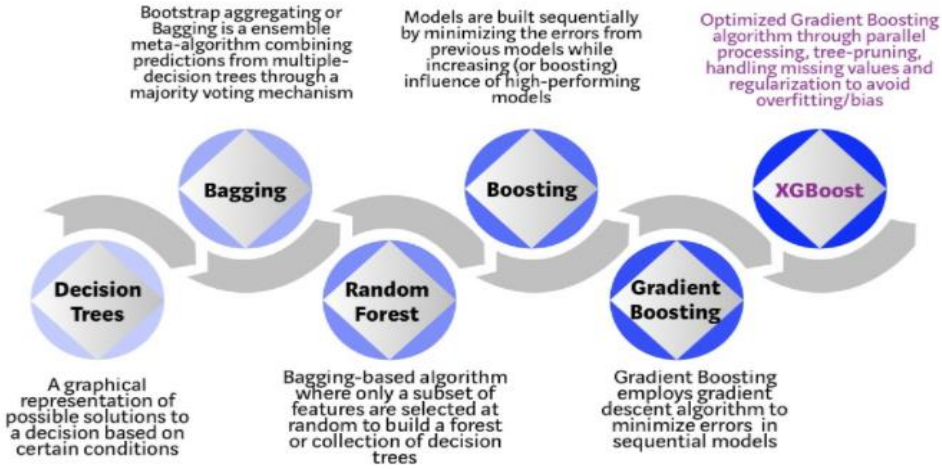


Source: Pedram Jahangiry, Machine Learning Course, Spring 2021

Extreme Gradient Boosting (XGBoost)

According to Vishal Morde, "XGBoost is a decision-tree based ensemble Machine Learning algorithm that uses a gradient boosting framework" (Vishal Morde, 2019). It is one variation algorithm of decision tree-based algorithms and considered one of the best gradient boosting algorithms to address the variance/bias tradeoff problem. Therefore, it renders better performance and is more efficient relative to other gradient boosting algorithms (Jahangiry, 2021). To understand the gradient boosting-based algorithms, we can look at the below figure to understand XGBosst advantages relative other bagging models:

Source: Pedram Jahangiry, Machine Learning Course, Fall 2021



Source: Vishal Morde, "XGBoost Algorithm: Long May She Reign!. 2019

There are two distinct features that make XGBoost a better alternative than its sister algorithms. It improves upon its sister gradient boosting algorithms through system optimization and algorithmic enhancement. In system optimization, the algorithm achieves the improvement through **parallelization, tree pruning, and hardware optimization.** On the other hand, in algorithmic enhancements, the process is achieved through **regularization** (penalizing complex models), **sparsity awareness, weighted quantile sketch, and cross validation.** With this, we could expect XGBoost to render the best prediction performance than its sister gradient boosting algorithms (Vishal Morde, 2019). To simplify, Jason Brownlee states that XGBoost "is an ensemble of decision trees algorithm where new trees fix errors of those trees that are already part of the model. Trees are added until no further improvements can be made to the model (Brownlee, 2020).

IV. Variable Definitions and Summary Statistics

In this section, we provide brief definitions of our dataset's features and include a summary statistics table. The definitions are derived from the Bloomberg Terminal and they are as follows:

SP&500: This is the U.S. Standard and Poor's 500 stock market index that tracks the performance of the largest publicly listed 500 companies.

CPI: The CPI is an inflation measure that tracks the prices consumers paid for a market basket of goods and services.

Initial Unemployment Claims: This is an unemployment metric that track the number of people who have filed jobless claims for the first time during the specified period with the appropriate government labor office.

Unemployment Rate: This unemployment indicator tracks the number of unemployed persons as a percentage of the total labor force that includes both employed and unemployed persons.

Personal Savings: This is defined as household disposable income less household consumption.

U.S. Personal Consumption Expenditures Ex Food & Energy (PCE): An index that measures prices that people in the United States pay for goods and services, excluding food and energy.

U.S. Industrial Production: An indicator that measures U.S. manufacturing, mining, electrical, and gas output facilities.

Private Housing Units Started by Structure: Housing (or building) starts track the number of new housing units (or buildings) that have been started during the reference period.

U.S. Employees on Nonfarm Payrolls Total: A macroeconomic indicator that measures the number of U.S. workers in the economy, excluding proprietors, private household employees, unpaid volunteers, farm employees, and unincorporated self-employed. The indicator accounts for a total of approximately 80% of the workers who contribute to Gross Domestic Product (GDP).

U.S. Trade Balance of Goods and Services: A macroeconomic indicator that measures the difference between the movement of merchandise trade and/or services leaving a country (exports) and entering a country (imports).

Adjusted Retail & Food Services Sales Total: This indicator tracks the U.S. retail and food services sales estimates, adjusted for seasonal variation and holiday and trading-day differences, but not for price changes.

U.S. Durable Goods New Orders Industrials: An index performance that tracks U.S. durable goods new orders and help explain ongoing industrial activity.

U.S. 2-year Treasury: This is a two-year U.S. government debt note that has a maturity of 2 years.

U.S. 5-year Treasury: This is a two-year U.S. government debt note that has a maturity of 5 years.

U.S. 10-year Treasury: This is a two-year U.S. government debt note that has a maturity of 10 years.

U.S. 30-year Treasury: This is a two-year U.S. government debt note that has a maturity of 30 years.

U.S. Effective Federal Funds Rate (EFFR): The EFFR is the interest rate that depository institutions charge each other for overnight loans of funds.

Bloomberg Velocity of Money M2 Money Supply: The average number of times a unit of money (as measured by monetary aggregate) turns over during a specified period of time.

U.S. Dollar Index (DXY Curncy): The index indicates the general international value of the U.S. dollar. The index averages the exchange rates between the USD and major world currencies.

FTSE 100 Index: The FTSE 100 Index is a capitalization-weighted index of the 100 most highly capitalized companies traded on the London Stock Exchange.

German Stock Index (DAX): This is the total return index of 40 selected German blue-chip stocks traded on the Frankfurt Stock Exchange.

Paris Stock Market (CAC 40 Index): a free float market capitalization weighted index that reflects the performance of the 40 largest and most actively traded shares listed on Euronext Paris.

Japanese Stock Market (NKY Index): The index is a price-weighted average of 225 top-rated Japanese companies listed in the First Section of the Tokyo Stock Exchange.

Hang Seng Index (HSI Index): The index is a free-float capitalization-weighted index of a selection of companies from the Stock Exchange of Hong Kong.

KOSPI Index: A South Korean capitalization-weighted index of all common shares on the KRX main board.

Shanghai Stock Exchange Composite Index: The index is a capitalization-weighted index. The index tracks the daily price performance of all A-shares and B-Shares listed on the Shanghai Stock Exchange.

CL1 Comdty: This is a Bloomberg crude oil futures contracts index.

Corporate BAA 10yr Spread: This is the spread between Moody's corporate yields for bonds rated BAA and the US government 10-year yield.

MSCI Emerging Markets Index: This is a free-float weighted equity index that captures large and mid-cap representation across Emerging Markets (EM) countries. The index covers approximately 85% of the free float-adjusted market capitalization in each country.

GOLD US Equity: This is the Barrick Gold Corporation's stock. The Barrick Gold Corporation is an international gold company with operating mines and development projects in the United States, Canada, South America, Australia, and Africa.

The VIX Index: The VIX Index is a financial benchmark designed to be an up-to-the-minute market estimate of the expected volatility of the S&P 500 index, and is calculated by using the midpoint of real-time S&P 500 Index (SPX) option bid/ask quotes.

Summary Statistics

This section will comment on some of the observations we encountered from our summary statistics table. It is important to note that we ran the summary statistics of our dataset prior to standardization. Thus, the interpretation of the statistics will be consistent with each feature's normal unit measurement. Our target feature S&P 500 has a monthly average return over the sample period of .70% and standard deviation of 4.2%. The VIX index has an average monthly volatility of 2% with a standard deviation of 22%. The U.S. 10-year treasury has an average

monthly return of -.09% and a standard deviation of 8.7%. Lastly, the famous inflation indicator CPI has an average monthly percent change of -14% with a standard deviation of 139%. These averages are calculated over our entire sample period.

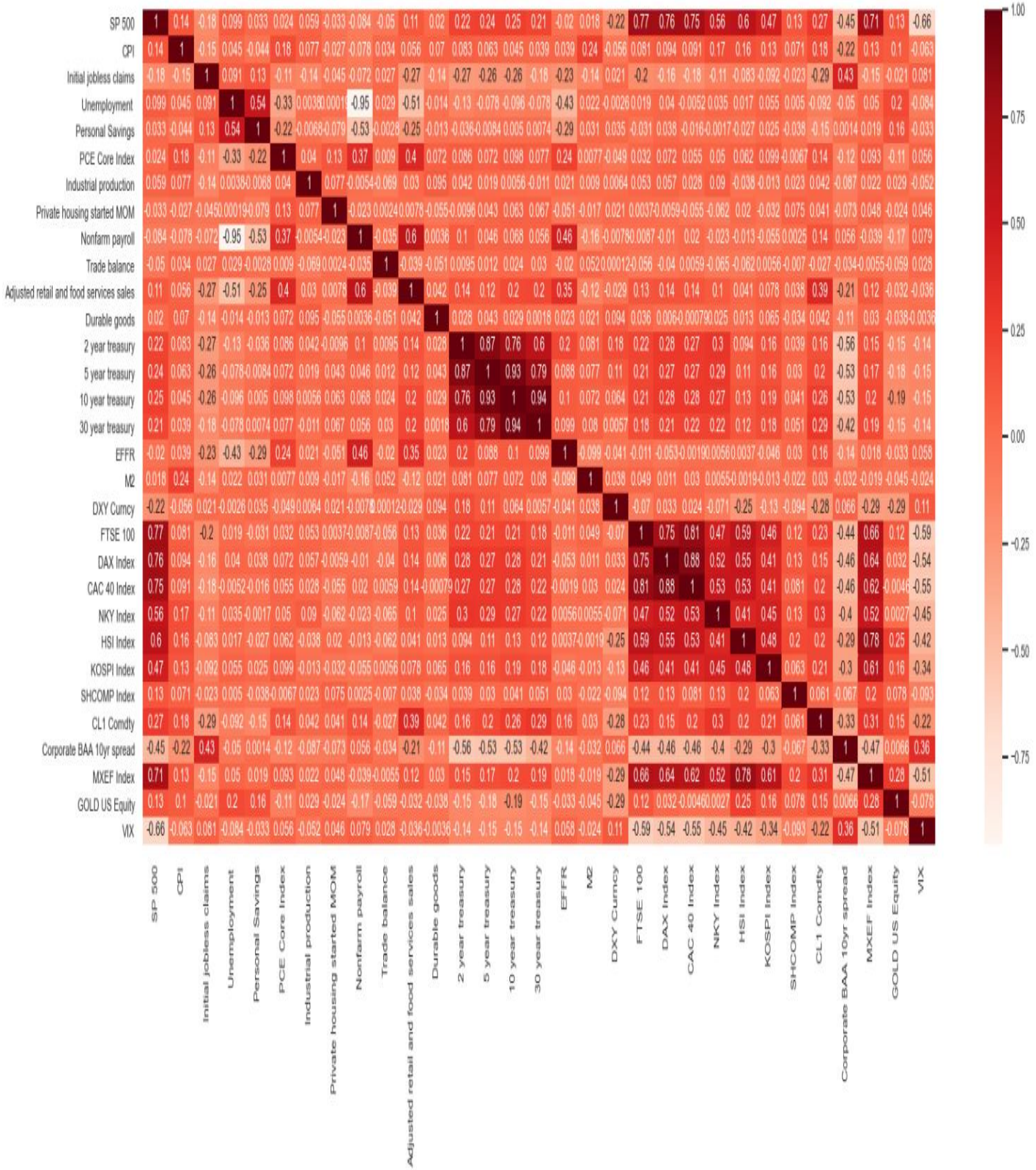
Summary statistics of all features in the sample

	count	mean	std	min	25%	50%	75%	max
SP 500	354.0	0.007663	0.041655	-0.169425	-0.016676	0.011825	0.033332	0.126844
CPI	320.0	-0.143899	1.391292	-10.000000	-0.606250	-0.183333	0.333333	7.000000
Initial jobless claims	354.0	0.068476	1.421806	-0.534917	-0.041859	-0.004624	0.029140	26.708333
Unemployment	354.0	0.002848	0.130410	-0.176471	-0.021277	0.000000	0.016949	2.363636
Personal Savings	354.0	0.018437	0.202311	-0.604286	-0.041815	0.003033	0.046327	1.976163
PCE Core Index	354.0	0.001531	0.001106	-0.005667	0.000971	0.001493	0.002065	0.007042
Industrial production	351.0	-0.637691	6.284108	-34.000000	-1.634682	-0.675000	0.246667	65.000000
Private housing started MOM	351.0	-1.457356	4.151761	-23.666667	-2.420833	-1.459016	-0.571895	42.500000
Nonfarm payroll	354.0	0.000904	0.007913	-0.137092	0.000515	0.001358	0.002053	0.036438
Trade balance	354.0	0.022404	0.168063	-0.349191	-0.040513	0.009180	0.068791	2.178099
Adjusted retail and food services sales	354.0	0.003928	0.018318	-0.146934	-0.001439	0.003948	0.009029	0.181729
Durable goods	350.0	-1.332157	8.803649	-34.000000	-2.367983	-1.383152	-0.447917	127.000000
2 year treasury	354.0	0.002680	0.150476	-0.731106	-0.059059	-0.002351	0.057615	0.766880
5 year treasury	354.0	0.001794	0.125340	-0.593950	-0.053953	-0.006054	0.053656	0.744097
10 year treasury	354.0	-0.000968	0.087197	-0.417117	-0.044042	0.000458	0.035878	0.334343
30 year treasury	354.0	-0.001972	0.062482	-0.222702	-0.030747	-0.005430	0.030086	0.346513
EFFR	354.0	0.002574	0.138159	-0.923077	-0.019633	0.000000	0.030184	1.000000
M2	354.0	-0.001344	0.012599	-0.199433	0.000000	0.000000	0.000000	0.042352
DXY Curncy	354.0	0.000366	0.022564	-0.062236	-0.014873	-0.000105	0.012638	0.080518
FTSE 100	354.0	0.003693	0.039863	-0.138080	-0.018356	0.007922	0.028516	0.123523
DAX Index	354.0	0.008029	0.059180	-0.254222	-0.022974	0.009890	0.042570	0.213778
CAC 40 Index	354.0	0.004848	0.052978	-0.174903	-0.028776	0.010141	0.037651	0.201189
NKY Index	354.0	0.002437	0.057287	-0.238269	-0.033033	0.006181	0.040298	0.161442
HSI Index	354.0	0.007143	0.070226	-0.294068	-0.030611	0.009522	0.043527	0.302807
KOSPI Index	354.0	0.007498	0.076424	-0.272473	-0.030682	0.006080	0.042090	0.507746
SHCOMP Index	354.0	0.014572	0.152902	-0.311530	-0.046693	0.004375	0.049352	1.772262
CL1 Comdty	354.0	0.009099	0.105274	-0.542449	-0.051297	0.011123	0.069399	0.883758
Corporate BAA 10yr spread	354.0	0.002751	0.078267	-0.174199	-0.044201	-0.001821	0.036568	0.627529
MXEF Index	354.0	0.005889	0.062971	-0.292852	-0.026235	0.007301	0.043941	0.166569
GOLD US Equity	354.0	0.007532	0.114952	-0.381056	-0.064174	0.001457	0.072003	0.538462
VIX	354.0	0.020065	0.216183	-0.458969	-0.111179	-0.009233	0.106102	1.345710

Correlation Matrix

In this section, we provide a features correlation heatmap to show the relationship between our target feature and all other parameters. Given the high dimensionality of our dataset, we will just share the results for some features. The features with the highest positive correlations with the S&P 500 are: FTSE 10 (.77), DAX index (.76), CAC 40 index (.75), MXEF index (.71), and NKY (.56). This is an interesting observation because they are all other countries' stock market indices. Conversely, the following are the features with the highest negative correlations with the S&P 500: VIX (-.66) and Corporate BAA 10yr spread (-.45). These two features' correlation results are expected, because they are inversely related to the SP&500. In addition, some of the features with both little positive and negative correlations with the SP&500 are: EFFR (-.02), private housing started MOM (-.03), nonfarm payroll (-.08), PCE core index (.02), durable goods (.02), personal savings (.03).

Heatmap showing correlation matrix of all features



V. Empirical Results

In this paper, we looked at whether we can predict stock market returns using the historical data of 30 macroeconomic indicators and foreign stock markets indices. We began this examination with the belief that stock market returns were not predictable with high accuracy using historical data but wanted to test this belief using machine learning techniques. To make predictions, we split our dataset into a training set (80%) and a testing set (20%). We ensured that our dataset was split and predictions were made without violating the time series prediction rules. We used Grid Search Cross Validation to tune the hyperparameters of our non-parametric models. To make sure that our models did not shuffle our data when implementing cross-validation, we used time-series cross validation to estimate the RMSE and R-squared in the test set. For cross validation, we used the TimSeriesSplit cross validation function from the Scikit learn library for splitting the dataset and estimating the RMSE and the R-squared in the test set. Unlike k-fold cross validation, in times series cross validation the algorithmy_ splits "time series data that are observed at fixed time intervals. In each split, test indices must be higher than before, and thus shuffling in cross validator is inappropriate" (Scikit-learn, 2011). This is consistent with our time-series data analysis and predictive modeling rules. After training our data and making predictions, we examined our predicted values for each model and compared them to each model's test set. Below we report snapshots of our results:

The machine learning results

Model	RMSE (Regression)	R2 (Regression)
Linear Regression	0.721394	0.196675
Elastic Net	0.567129	0.455943
SVR	0.555799	0.497002
Random Forest	0.540615	0.33628
XG Boost	0.535724	0.421522

Linear Regression Model

	y_test	predictions	resid
283	0.059791	1.002238	-0.942448
284	0.193974	0.702013	-0.508040
285	0.887672	0.070879	0.816792
286	-1.013091	-2.141532	1.128441
287	1.093621	1.125397	-0.031776

Elastic Net Regression Model

	y_test	y_hat_net	resid
283	0.059791	0.720748	-0.660957
284	0.193974	0.633233	-0.439260
285	0.887672	0.146951	0.740721
286	-1.013091	-2.106205	1.093114
287	1.093621	1.068987	0.024634

Support Vector Regression Model

	y_test	y_hat_optimized	resid
283	0.059791	0.518961	-0.459170
284	0.193974	0.876105	-0.682131
285	0.887672	0.170133	0.717539
286	-1.013091	-2.234560	1.221470
287	1.093621	1.271418	-0.177797

Random Forest Model

	y_test	y_hat_optimized_RF	resid
283	0.059791	0.511493	-0.451702
284	0.193974	0.559305	-0.365331
285	0.887672	0.221688	0.665984
286	-1.013091	-1.974068	0.960978
287	1.093621	0.699358	0.394263

XGBoost Model

	y_test_XGB	y_hat_XGB	resid
283	0.059791	0.523993	-0.464202
284	0.193974	0.672892	-0.478918
285	0.887672	0.410777	0.476895
286	-1.013091	-1.045972	0.032882
287	1.093621	0.706667	0.386953

When we look at each model's standardized test set values in the figures above and compare them to the predicted values, just as our results show in the table below, the non-parametric models seem to perform better than parametric models at making closer predictions to the test set. However, but even the non-parametric models do not even render highly accurate predictions.

Our results show that our machine learning techniques could not predict the monthly stock market returns with high accuracy. Looking at output of the five machine learning models in the results table above, it is clear the XGBoost model performs the best relative to other models based on its lower root mean square error (RMSE) output. XGboost has a root-mean-squared-error (RMSE) of .5357 and an R-squared of .42. XGBoost has proven to be a remarkable algorithm due to its performance capabilities, but even its advanced performance was not able to make highly accurate predictions. The RMSE for other non-parametric models produce a very close RMSE and in fact some models explain the stock market return better than XGBoost giving their higher R-squared. See table above.

The linear regression model performed the worst relative to other non-parametric models. It can be fairly said that our parameters relationship with the stock market return was not just linear. This explains why the non-parametric models like XGBoost, Random Forest, and SVR were able to outperform the linear regression. We find the Elastic Net model RMSE result of .5671 quite appealing, because they are not far off from the non-parametric models. This is consistent with our expectation that the algorithm is penalizing our model a lot to reduce its complexity. It introduces some bias by regularizing our model's dimensionality to reduce the variance substantially.

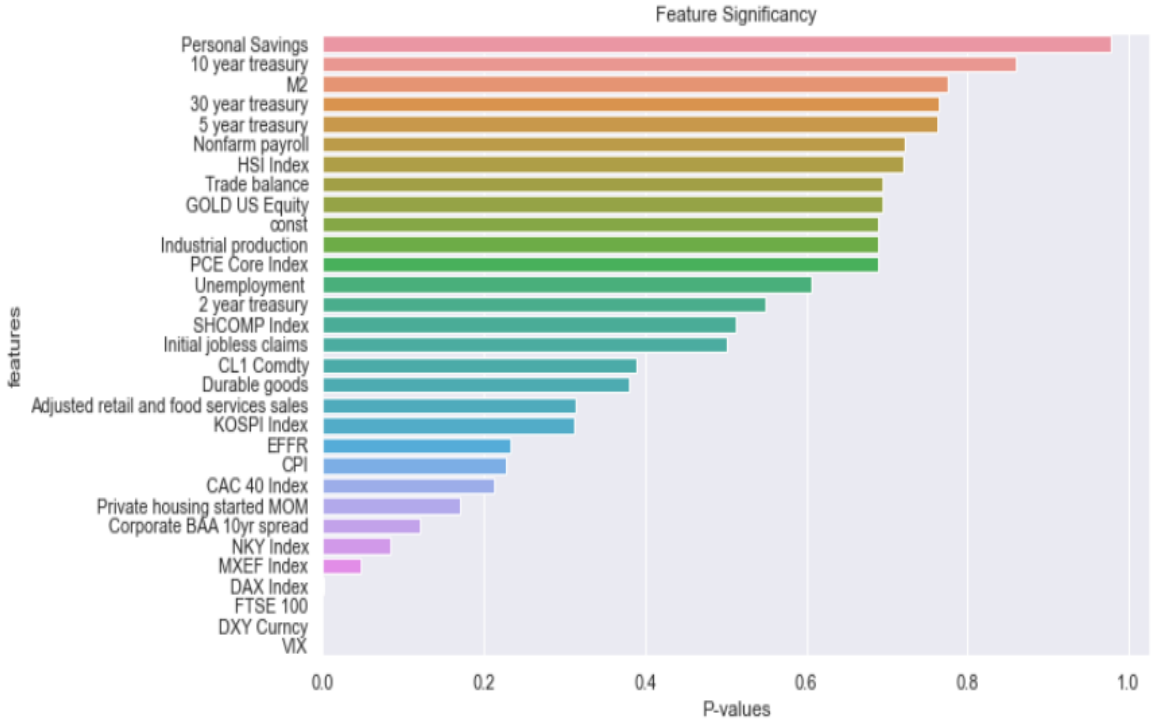
Even though we were able to make some predictions about the stock market returns, they nonetheless have little accuracy. All our machine learning models have an R-squared of below .50,

demonstrating that our data can only explain the variation in the stock market returns by close to or less than 50%. Nonparametric models like Support Vector Regression, Random Forest, and XG Boost seem to do better than linear regression, but even these models cannot make predictions with high accuracy. If an investor tries to rely on the predictions of these models to make investment decisions, they will be making a bid that has a 50-50 or less chance of generating excess returns. These findings support existing literature that have tried to assess the predictability of the stock market. They also support Fama's efficient market hypothesis because our models did not make highly accurate predictions. That said, we can safely state that our stock market is semi-strong efficient and cannot be predicted with high accuracy using monthly historical data, even with machine learning techniques.

Feature Importance

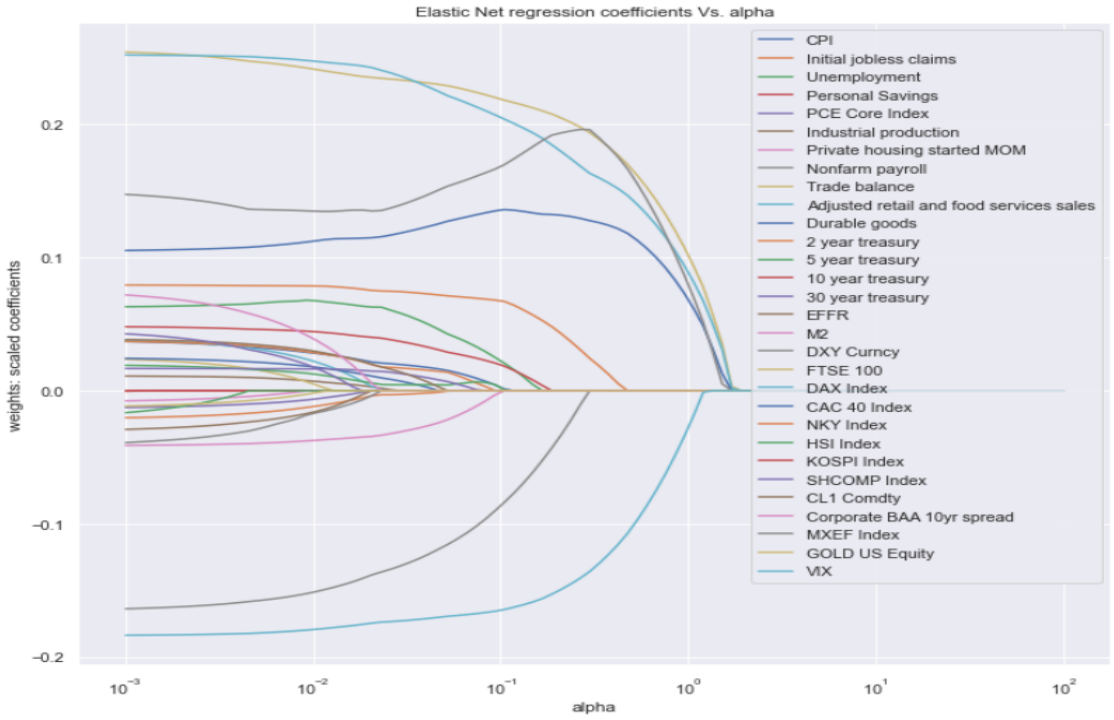
One other objective of this paper was to examine how our models were going to rank the importance of our 30 features in relation to the SP&500. Below we include the finding of our models' feature importance:

Linear Regression Feature Significancy Using P-Value



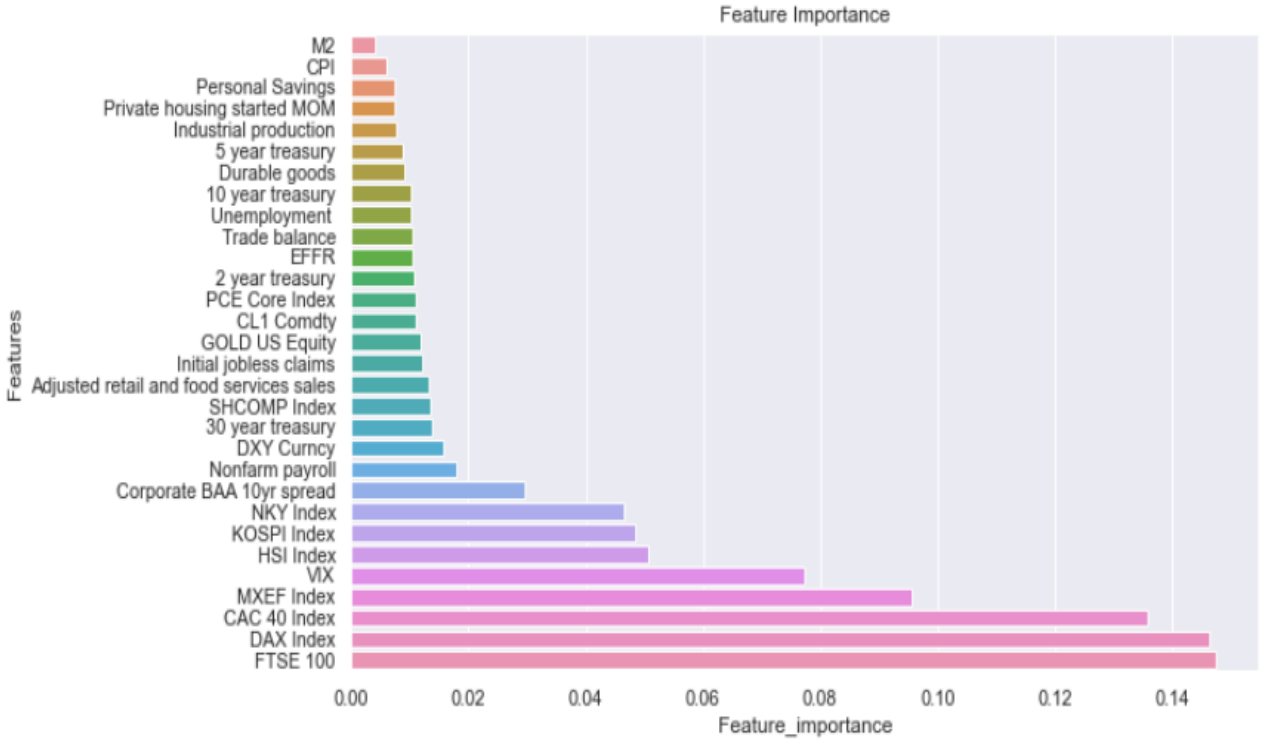
Based on the features statistical significance figure above, we see that the volatility index (VIX), the dollar spot rate index (DXY Curncy), the Financial Times Stock Exchange (FTSE 100) and the German blue-chip stock index (DAX Index) were all statistically significant based on their p-values of .05 and below. For all other features, we kept them in our model because they may still provide important insights into our target feature collectively, even though individually they are statistically insignificant.

Elastic Net Feature Importance with optimized alpha



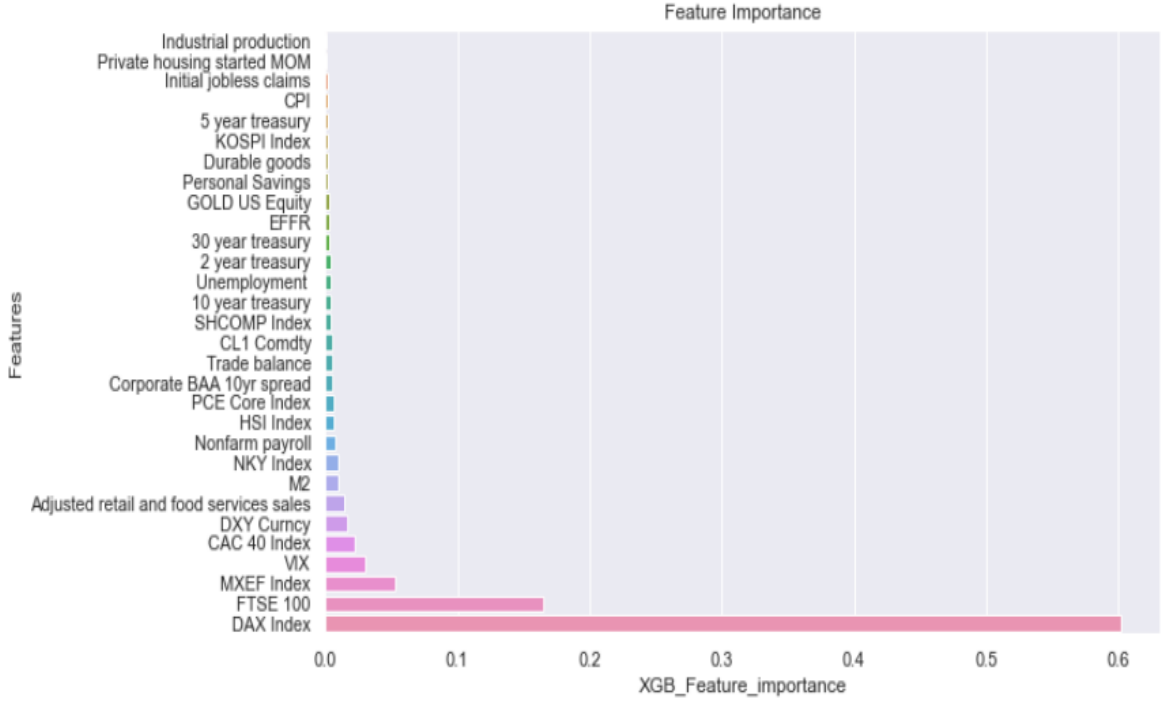
The optimal alpha our Elastic Net model computed was .05. When we apply the optimal alpha to the figure above, we can see that it sits between 10^{-1} and 10^{0}, which means that our model regularized most coefficients to 0, except for the volatility index (VIX), the dollar spot rate index (DXY Curncy), the Financial Times Stock Exchange (FTSE 100), the Tokyo Stcok Index Exchange (Nikkei 225), the French stock market index (CAC 40), and the 40 major German blue-chip companies stock index (DAX Index). Most of these features were also statistically significant as we saw in the p-value and coefficients figure in linear regression. For all other features, it appears that the Elastic Net model shrunk them to zero. The algorithm regularized them to make the model less complex, introducing some bias by getting rid of these features to reduce the model's variance substantially.

Random Forest Feature Importance



The Random Forest feature importance figure tells a little different story than did the Linear Regression and the Elastic Net algorithms. In addition to the FTSE 100, DAX index, VIX index, NKY index, and the CAC 40 index, the Random Forest is adding the Corporate BAA 10yr spread, the emerging markets stock index (MXEF), the Hong Kong stock market index (HSI), the South Korean stock market index (KOSPI) as importance features in the model. These features would be considered the most important ones in making some predictions about the monthly return of the stock market. The way the algorithm determines the importance of the features is that it evaluates how much each feature contributes to the decline in the residual sum of squares (RSS). The RSS is a statistical method used to determine the variance in a dataset. Thus, the importance of the features and their ranking will be determined by how much they each cause the RSS of the model to decline.

XGBoost Feature Importance

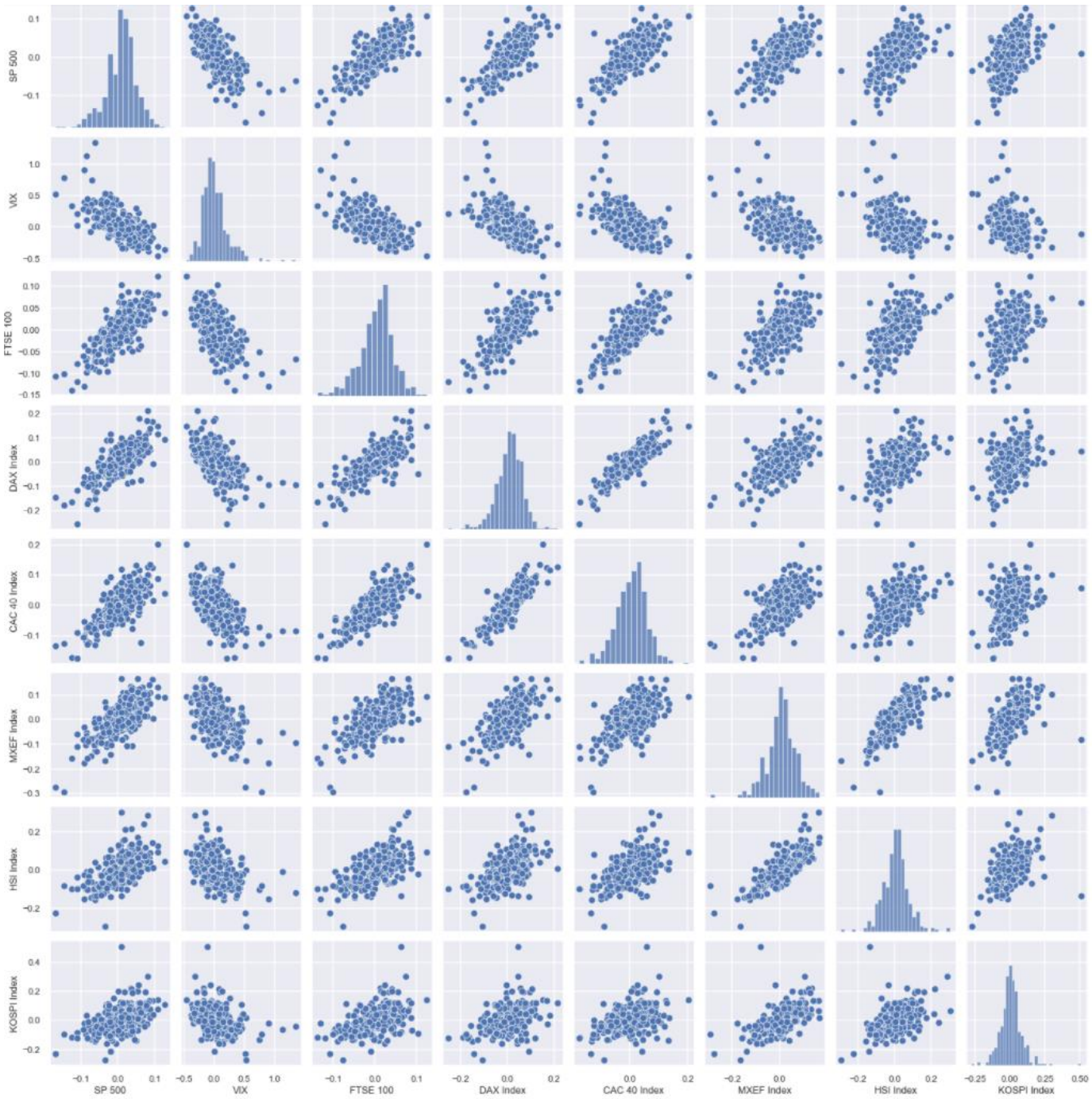


In boosting algorithms, the feature importance is evaluated based on "how useful or valuable each feature was in the construction of the boosted decision trees within the model" (Brownlee, 2016). Put differently, the more a feature is useful for improving decision trees performance, the more important the feature is relative to the others. It appears that the XGBoost algorithm ranks the DAX index, the FTSE 100, the MXEF index, the VIX index, and the CAC 40 index as the top important features, with emphasis on the German blue-chip companies stock index (DAX Index). Again, we see that the algorithm considered most of our attribute features as unimportant, as indicated in the figure above.

Looking at feature importance ranking of each algorithm, it is evident that other nations' stock markets indices tend to rank high relative to other features. Specifically, this is true for the 40 major German blue-chip companies index (DAX index), the Financial Times Stock Exchange 100 Index (FTSE 100), the emerging markets index (MXEF index), the volatility index (VIX), the French stock market index (CAC 40), and the U.S. dollar index (DXY). While we held the belief

that the U.S. stock market is efficient and tested this hypothesis using machine learning algorithms discussed in this paper, one could say that features importance does not necessarily matter because our machine learning models were not able to predict the SP&500 stock return with high accuracy. This is true, but the feature importance technique can still help provide key insights into the performance of our models and how our models' features behave relative to the SP&500.

Viewing the pair-plot of the features that were ranked high in our models



VI. Conclusion

This paper was written to assess the predictability of the U.S. SP&500 stock market return using machine learning techniques. There is extensive literature that have examined the predictability of the stock market using both traditional statistical methods and machine learning techniques, but neither of the two methods have been successful at predicting the stock market return with high accuracy using monthly returns. Thus, investors who attempt to make investment bets using our models' findings have a 50% or less probability of generating excess return. When we ventured to test our hypothesis, we undertook this experiment with the belief that the stock market is semi-strong efficient. However, we wanted to contribute to existing literature and add evidence proving that stock market returns cannot be predicted using historical data, whether one uses traditional statistical methods as existing literature demonstrated or machine learning techniques as in this paper.

It is important to caution that we used supervised machine learning techniques to test the predictability and efficiency of the market. There is some literature that have demonstrated that unsupervised machine learning models tend to do better than the supervised machine learning models we used in this paper. Thus, this paper does not say anything about the predictability of the stock market using unsupervised machine learning techniques. But, what we can say is that our experiment to assess the predictability of the stock market return using supervised machine learning techniques proved that it cannot be predicted with high accuracy. This supports the belief that the U.S. stock market is efficient and backs up the efficient market hypothesis theory introduced by Fama and other economists.

Lastly, even though our results show that the stock market cannot be predicted with high accuracy, we were still able to look at the features' importance. We wanted to examine how each machine

learning technique discussed in this paper was going to rank the importance of the features in relation to the SP&500 returns. Interestingly, all our machine learning techniques feature importance output had features ranked similarly across all models. Some of these features that were ranked high were other nations' stock markets indices, including the major 40 German blue-chip companies index (DAX index), the Financial Times Stock Exchange 100 Index (FTSE 100), the emerging markets index (MXEF index), the volatility index (VIX), the French stock market index (CAC 40), and the U.S. dollar index (DXY). These were among the top ranked features. We had expected that key U.S. macroeconomic indicators like employment, inflation, U.S. treasuries, and monetary policy indicators were going to be among the highly ranked features, but all our models ranked these indicators among the lowest. This was a worthwhile observation we encountered from this paper. The stock market efficiency should always be explored continuously to guide policy-making, investors, and portfolio managers, particularly as machine learning techniques continue to improve their performance and computational capabilities. As of today, and as this paper demonstrates, supervised machine learning techniques are not successful at predicting the U.S.SP&500 returns with high accuracy using monthly returns.

References:

Marin Sirucek. 2012. “Macroeconomic Variables and Stock Market: US Review.” Mendel University, Paper No. 39094.

Saseela Balagobei. 2017. “Macroeconomic Variables and Stock Market Returns in Sri Lanka.” Asian Journal of Finance & Accounting, Vol. 9, No. 2.

Charles Barnor. 2014. “The Effect of Macroeconomic Variables on Stock Market Returns in Ghana.” Walden University ScholarWorks.

Asmy, Mohamed and Rohilina, Wisam and Hassama, Aris and Fouad, Md. 2009. “Effects of Macroeconomic Variables on Stock Prices in Malaysia: An Approach of Error Correction Model.” International Islamic University Malaysia, Paper No. 20970.

Jorge Castanon. 2009. “ 10 Machine Learning Methods that Every Data Scientist Should Know.” Toward Data Science.
https://towardsdatascience.com/10-machine-learning-methods-that-every-data-scientist-should-know-3cc96e0eeee9

Ashwin Raj. 2020. “Unlocking the True Power of Support Vector Regression.” Toward Data Science.
https://towardsdatascience.com/unlocking-the-true-power-of-support-vector-regression-847fd123a4a0

Gurucharan M K. 2020. “Machine Learning Basics: Random Forest Regression.” Towards Data Science.
https://towardsdatascience.com/machine-learning-basics-random-forest-regression-be3e1e3bb91a

Morde, Vishal. 2019. “XGBoost Algorithm: Long May She Reign!”. Toward Data Science.
https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d

Huey Fern Tay. 2021. “When is it Ok to Impute Missing Values with a Zero.” Towards Data Science.
https://towardsdatascience.com/when-is-it-ok-to-impute-missing-values-with-a-zero-6d94b3bf1352

Ertan, Halil. 2020. “ Which Features to use in Your Model?”
https://medium.com/@hertan06/which-features-to-use-in-your-model-350630a1e31c

Gupta, Aman. 2021. “XGBoost Versus Random Forest.” https://medium.com/geekculture/xgboost-versus-random-forest-898e42870f30

Kumar, Vijay. 2021. “Statistical Tests to Check Stationarity in Time Series.” https://www.analyticsvidhya.com/blog/2021/06/statistical-tests-to-check-stationarity-in-time-series-part-1/

Brownlee, Jason. 2020. “Information Gain and Mutual Information for Machine Learning.” Machine Learning Mastery. https://machinelearningmastery.com/information-gain-and-mutual-information/

Aznar, Pablo. 2021. “ What is Mutual Information.” QuantDare. https://quantdare.com/what-is-mutual-information/

Lakshmanan, Swetha. 2019. “How, When, and Why Should You Normalize/ Standardize/ Rescale Your Data?”. Towards AI. https://towardsai.net/p/data-science/how-when-and-why-should-you-normalize-standardize-rescale-your-data-3f083def38ff

Brownlee, Jason. 2020. “Feature Importance and Feature Selectino With XGBoost in Python.” Machine Learning Mastery. https://machinelearningmastery.com/feature-importance-and-feature-selection-with-xgboost-in-python/

Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.

Jahangiry, Pedram. 2021. ML-USU-Fall21[PowerPoint presentation]. GitHub. https://github.com/PJalgotrader/ML-USU-Fall21/tree/main/Classes

Bloomberg (2021), definition of U.S. economic indicators and foreign stock markets. *Bloomberg Professional.* [Accessed November 12 2021].