# Predicting Monthly Excess Returns of Market Index

Anchal Shaileshkumar Thavrani

Demilade Popoola Samuel

Venus Artus

## Executive Summary

This group project aims to investigate the components influencing the monthly excess returns of the CRSP market index and assess the machine learning models that are the most relevant for this study. The core objective is to develop robust predictive models, compare their performance and derive actionable financial insights to support investment decision-making. The project first includes a preprocessing section, which includes an analysis of the variable's relevance and visualization. The project also dives into the transformation and creation of variables to capture the most intricate relationships. The dataset was split into training and tests sets and thus enabled us to

evaluate model performance on unseen data. A range of machine learning models is then applied, including linear regression (OLS, Lasso, Ridge), decision trees and more advanced methods such as random forests. To assess the efficacy of these models, we compared their predictive accuracy using performance metrics such as Mean Squared Error (MSE) and R-squared values. Our analysis revealed that tree-based models, particularly Random Forest, outperformed linear models like OLS and Ridge regression in capturing complex relationships. However, Lasso regression demonstrated better regularization, leading to stable predictions with minimal overfitting.
These findings could provide investors with additional tools for market forecasting and decision-making.

## Introduction

The ability to predict stock market returns, particularly major indices like the S&P 500, the FT-SE 100 or the CRSP, has long been a subject of interest in both academic research and financial markets. Predicting monthly excess returns is essential for investors and financial institutions to make informed decisions about asset allocation, risk management and portfolio optimization. Despite the volume of economic and financial data available, accurately forecasting market returns remains a challenging task due to the complexity of financial markets and the interplay of various macroeconomic factors.

This project focuses on predicting monthly excess returns for the CRSP value-weighted index using machine learning techniques. More specifically, we leverage the Welch and Goyal (2008, 2022) dataset, which includes a variety of macroeconomic variables and financial indicators believed to influence market returns. By applying different machine learning models, we aim to enhance prediction accuracy and derive insights that can be applied to real-world financial decision-making.

The problem addressed in this project is the difficult in accurately predicting the monthly excess returns of the CRSP, a critical financial indicator that impacts investment decisions. While traditional econometric models have made progress in forecasting returns, they often struggle to capture complex, non-linear relationships between macroeconomic variables and market performance. With the rise of machine learning, there is the potential to improve prediction accuracy by leveraging algorithms that can capture complex relationships and provide more reliable forecasts.

The objectives of this project are as follows:

**Model development:** To develop and train multiple machine learning models including linear regression, decisions trees and random forests to predict monthly excess returns of the CRSP index.

**Model comparison**: To evaluate and compare the performance of the different machine learning models based on metrics such as Mean Squared Error (MSE) and R squared, assessing their predictive power and robustness.

**Deriving financial insights**: To identify the key factors that influence the excess returns of the CRSP and provide actionable insights that can guide investment strategies and decision making.

## Data Presentation & Visualisation

### Data Presentation

The dataset comprises fifteen financial indicators and a target variable, which is the monthly excess returns of the CRSP market index. The CRSP value-weighted market index represents a collection of U.S. stocks listed on major exchanges (NYSE, AMEX, NASDAQ) and reflects the overall performance of the U.S. stock market. As a value-weighted index, it is based on the market capitalization of the included stocks, meaning that companies with higher market values exert a proportionally greater influence on the index than smaller companies. This approach mirrors real-world investment strategies, where larger companies typically dominate investment portfolios. In such portfolios, the performance of large-cap stocks typically drives overall returns, reflecting how institutional investors, fund managers, and other market participants allocate capital based on company size. This aligns with the Diversified Portfolio Theory concept, where larger companies are often considered a stabilising force in portfolios, contributing to their relatively low risk and higher weight in portfolio construction.

This project focuses on forecasting excess returns rather than total returns because excess returns isolate the portion of returns that compensates investors for taking on additional risk beyond the risk-free rate. By focusing on excess returns, we can distinguish between the returns driven by risk-taking and those influenced by broader market or economic conditions. This approach allows for a more accurate performance evaluation, as it enables us to assess whether the returns justify the extra risk taken. Additionally, it supports the use of active management strategies by determining whether riskier positions can outperform the risk-free benchmark. Understanding whether an asset or strategy is likely to outperform a risk-free investment is crucial for assessing the potential value of higher-risk investment approaches.

The fifteen financial regressors are crucial for predicting excess returns, as they encompass a range of macroeconomic, financial, and industry factors that impact investor behaviour, market conditions, and, ultimately, asset pricing. These factors can be broadly grouped into four distinct categories.

| | | |
|---|---|---|
| Macroeconomic factors | Inflation, treasury bill, long-term government bond, term spread | Provide insights into economic conditions and monetary policy |
| Credit risk measures | Default yield spread, default return spread | Highlight market stress and risk appetite |
| Valuation metrics | Dividend to price ratio, dividend yield, earnings to price ratio, book to market ratio | Indicate market valuations and expected returns |
| Market sentiment | Lagged market return, stock variance, net equity expansion | Tracks momentum, volatility and market confidence |

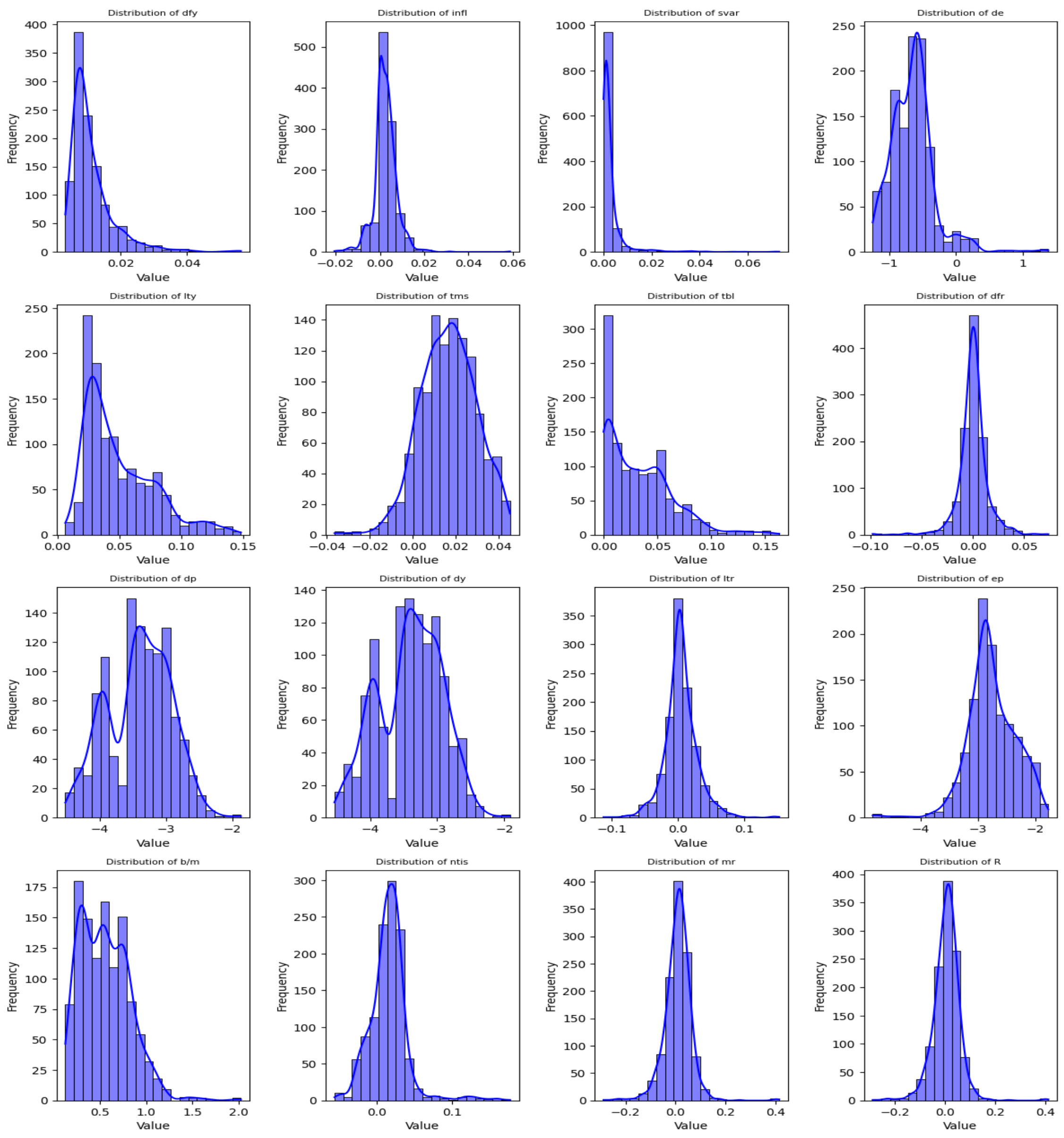
It is important to construct a model that incorporates regressors from various categories, each supported by established economic and financial theories. These theories allow us to predict the potential impact of each regressor on the target variable. For example, stock variance is typically negatively correlated with excess returns. High volatility often signals a market downturn or, at a minimum, heightened uncertainty, which can consequently lead to a reduction in excess returns.

## Distribution shape, skewness and kurtosis

Before constructing our models, it is essential to examine the characteristics of our variables, such as skewness and kurtosis, autocorrelation, and correlation, as well as their distribution to ensure appropriate model assumptions are met.

During the analysis of the variables, their distribution is examined, and ideally, the variables should approximate a normal distribution. This means that most data points should cluster around the mean, with the tails of the distribution symmetrically decreasing as they move away from the mean. Achieving normality is crucial for the effective functioning of statistical models, as many financial theories assume that returns and errors follow a normal distribution. This assumption allows risk and return to adequately describe portfolio performance. Normality is also essential for the validity of statistical tests and confidence intervals. If the distribution does not exhibit normality, it may be necessary to transform the relevant variables or employ models that accommodate more complex relationships between the regressors and the target variable.

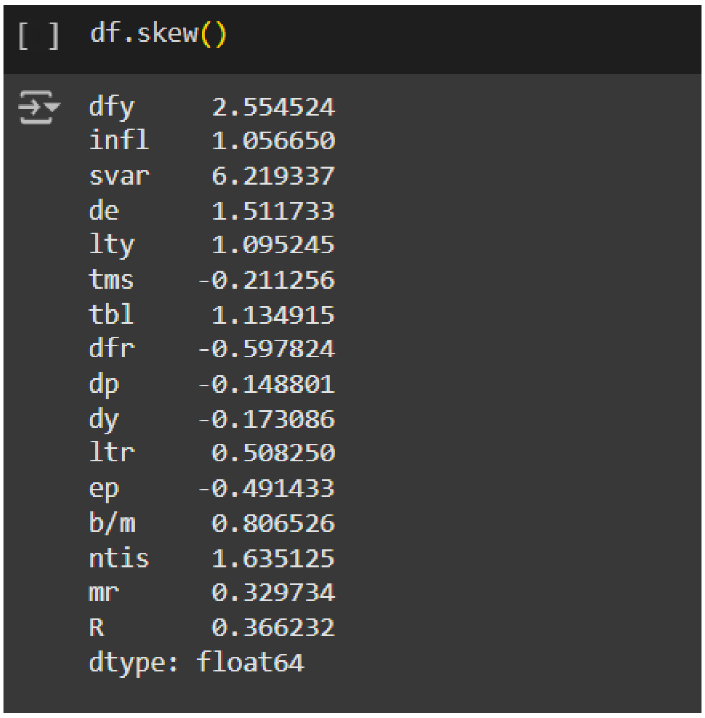
Initially, we visually inspect the distribution of our regressors to assess whether they are approximately normally distributed.



Based on the visual inspection of the distribution shapes of the regressors, we can infer that only a few variables appear to approximate a normal distribution, specifically “tms”, “dfr”, “ltr”, “ep”, “ntis”, “mr”, and “R”. While these variables visually seem to follow a normal distribution, it is important to conduct a more thorough analysis of their normality. To achieve this, we calculate both the skewness and kurtosis of the variables.

**Skewness:**

We calculate the skewness of our regressors to evaluate the symmetry of their distribution. A variable with skewness will have a distribution that leans toward either the right or left tail. We then compute the skewness and present the following results.

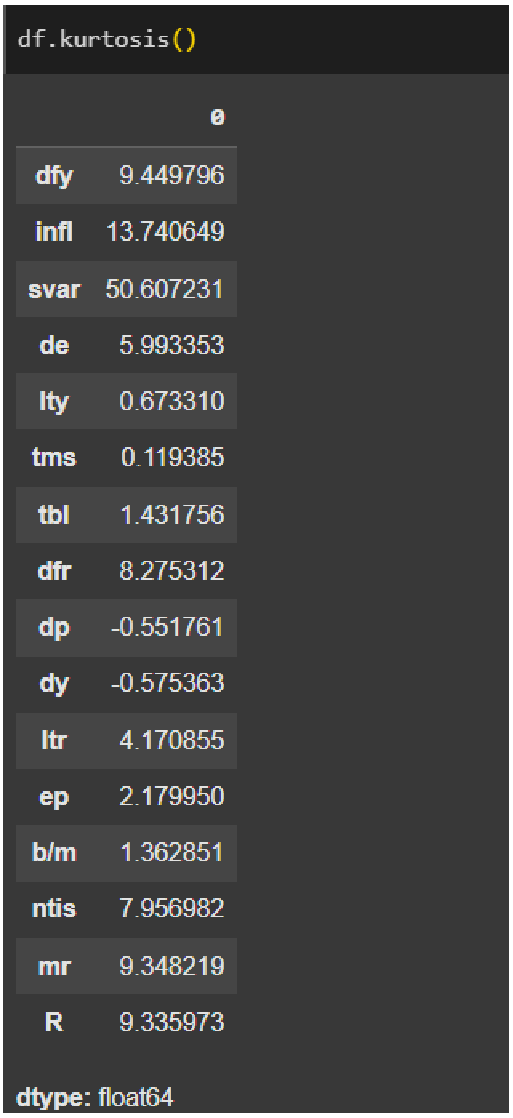


```
[ ]  df.skew()

     dfy      2.554524
     infl     1.056650
     svar     6.219337
     de       1.511733
     lty      1.095245
     tms     -0.211256
     tbl      1.134915
     dfr     -0.597824
     dp      -0.148801
     dy      -0.173086
     ltr      0.508250
     ep      -0.491433
     b/m      0.806526
     ntis     1.635125
     mr       0.329734
     R        0.366232
     dtype: float64
```

To interpret these results, it is important to establish a benchmark for comparison. We consider a skewness value between -1 and +1 to be ideal, while a value between -2 and +2 is still acceptable. Any value outside these ranges indicates excessive skewness, which requires correction. As shown in the results, most of our variables exhibit an acceptable level of skewness, with many being close to zero. However, we observe higher skewness in the default yield spread (dfy) and stock variance (svar), with values of 2.55 and 6.22, respectively, both exceeding the established bounds. The significant skewness in stock variance may suggest occasional but large price movements, leading to extreme volatility peaks. Skewness must be addressed, as it can result in biased parameter estimates, inaccurate risk assessments, and unreliable predictions.

**Kurtosis:**

Kurtosis measures whether a distribution is more peaked or flatter than a normal distribution. A normal distribution has a kurtosis of 3, and excess kurtosis occurs when the value deviates from this benchmark. Excess kurtosis indicates that the distribution contains more extreme values or outliers than a normal distribution. We expect kurtosis values to range from 0 to 6, with 3 being the ideal level. However, in financial data, higher kurtosis values (ranging from 5 to 10) are not uncommon and may be expected.



```
df.kurtosis()
```

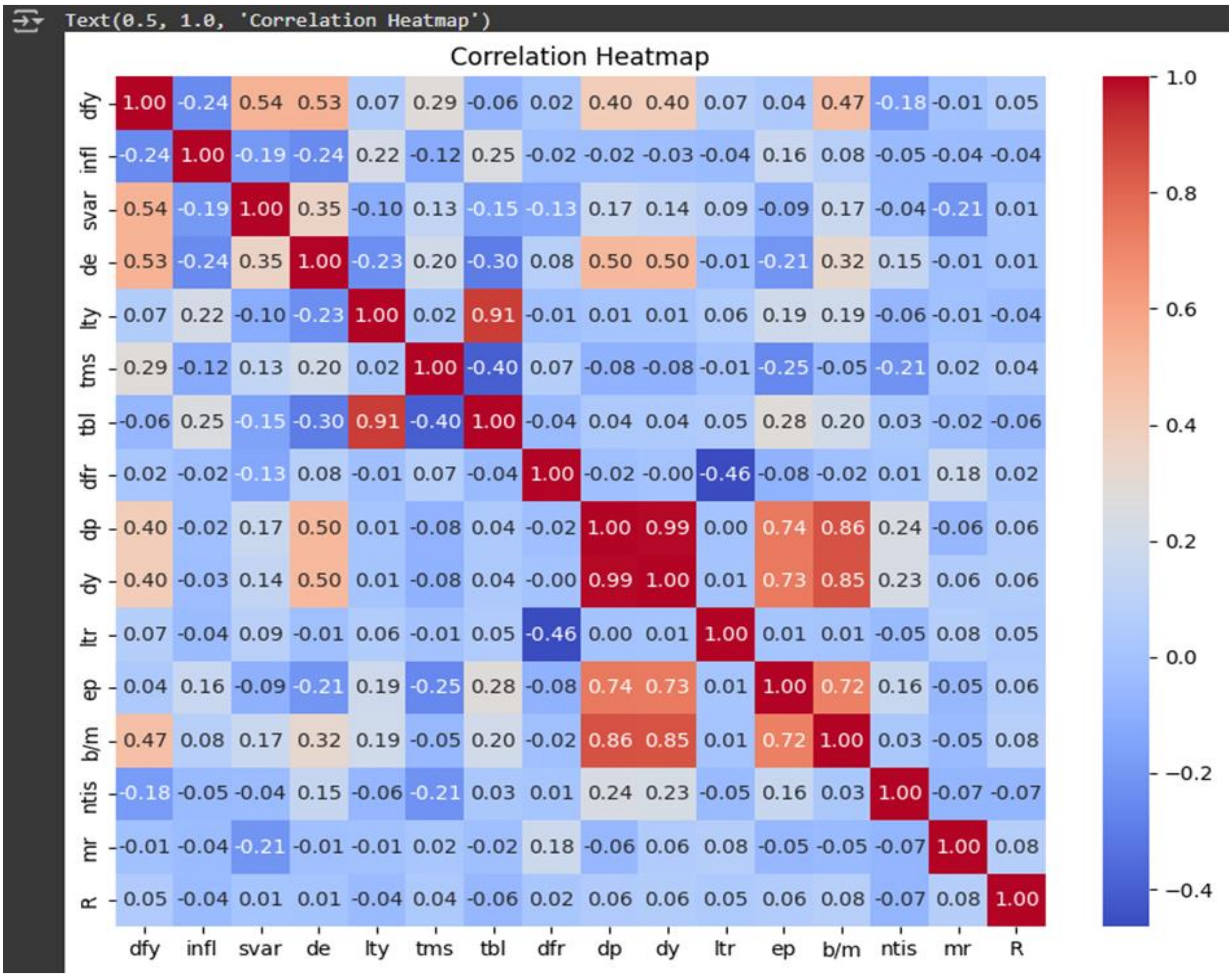
```
              0
dfy    9.449796
infl  13.740649
svar  50.607231
de     5.993353
lty    0.673310
tms    0.119385
tbl    1.431756
dfr    8.275312
dp    -0.551761
dy    -0.575363
ltr    4.170855
ep     2.179950
b/m    1.362851
ntis   7.956982
mr     9.348219
R      9.335973

dtype: float64
```

Our results show that most regressors exhibit a kurtosis value below 10. However, inflation and stock variance display particularly high kurtosis levels of 13.74 and 50.61, respectively. Excess kurtosis indicates that the distribution has more extreme values or outliers than a normal distribution. While excess kurtosis can lead to larger errors in parameter estimation, it is important to approach corrections with caution. Outliers and extreme values may be the result of significant market events, such as crashes, and can contain valuable information. Removing kurtosis could result in underestimating the risk associated with such events and lead to misleading conclusions.

## Correlation analysis

### *Correlation between regressors*

We also chose to compute a correlation heatmap to analyze the relationships between the variables. Correlation is a statistical measure that indicates the strength and direction of the relationship between two variables. Examining the correlations between the regressors is crucial, as it helps assess the potential presence of multicollinearity. When two independent variables are highly correlated, it can be challenging to isolate the individual effects of each variable on the dependent variable.



```
Text(0.5, 1.0, 'Correlation Heatmap')
```

| | dfy | infl | svar | de | lty | tms | tbl | dfr | dp | dy | ltr | ep | b/m | ntis | mr | R |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| dfy | 1.00 | -0.24 | 0.54 | 0.53 | 0.07 | 0.29 | -0.06 | 0.02 | 0.40 | 0.40 | 0.07 | 0.04 | 0.47 | -0.18 | -0.01 | 0.05 |
| infl | -0.24 | 1.00 | -0.19 | -0.24 | 0.22 | -0.12 | 0.25 | -0.02 | -0.02 | -0.03 | -0.04 | 0.16 | 0.08 | -0.05 | -0.04 | -0.04 |
| svar | 0.54 | -0.19 | 1.00 | 0.35 | -0.10 | 0.13 | -0.15 | -0.13 | 0.17 | 0.14 | 0.09 | -0.09 | 0.17 | -0.04 | -0.21 | 0.01 |
| de | 0.53 | -0.24 | 0.35 | 1.00 | -0.23 | 0.20 | -0.30 | 0.08 | 0.50 | 0.50 | -0.01 | -0.21 | 0.32 | 0.15 | -0.01 | 0.01 |
| lty | 0.07 | 0.22 | -0.10 | -0.23 | 1.00 | 0.02 | 0.91 | -0.01 | 0.01 | 0.01 | 0.06 | 0.19 | 0.19 | -0.06 | -0.01 | -0.04 |
| tms | 0.29 | -0.12 | 0.13 | 0.20 | 0.02 | 1.00 | -0.40 | 0.07 | -0.08 | -0.08 | -0.01 | -0.25 | -0.05 | -0.21 | 0.02 | 0.04 |
| tbl | -0.06 | 0.25 | -0.15 | -0.30 | 0.91 | -0.40 | 1.00 | -0.04 | 0.04 | 0.04 | 0.05 | 0.28 | 0.20 | 0.03 | -0.02 | -0.06 |
| dfr | 0.02 | -0.02 | -0.13 | 0.08 | -0.01 | 0.07 | -0.04 | 1.00 | -0.02 | -0.00 | -0.46 | -0.08 | -0.02 | 0.01 | 0.18 | 0.02 |
| dp | 0.40 | -0.02 | 0.17 | 0.50 | 0.01 | -0.08 | 0.04 | -0.02 | 1.00 | 0.99 | 0.00 | 0.74 | 0.86 | 0.24 | -0.06 | 0.06 |
| dy | 0.40 | -0.03 | 0.14 | 0.50 | 0.01 | -0.08 | 0.04 | -0.00 | 0.99 | 1.00 | 0.01 | 0.73 | 0.85 | 0.23 | 0.06 | 0.06 |
| ltr | 0.07 | -0.04 | 0.09 | -0.01 | 0.06 | -0.01 | 0.05 | -0.46 | 0.00 | 0.01 | 1.00 | 0.01 | 0.01 | -0.05 | 0.08 | 0.05 |
| ep | 0.04 | 0.16 | -0.09 | -0.21 | 0.19 | -0.25 | 0.28 | -0.08 | 0.74 | 0.73 | 0.01 | 1.00 | 0.72 | 0.16 | -0.05 | 0.06 |
| b/m | 0.47 | 0.08 | 0.17 | 0.32 | 0.19 | -0.05 | 0.20 | -0.02 | 0.86 | 0.85 | 0.01 | 0.72 | 1.00 | 0.03 | -0.05 | 0.08 |
| ntis | -0.18 | -0.05 | -0.04 | 0.15 | -0.06 | -0.21 | 0.03 | 0.01 | 0.24 | 0.23 | -0.05 | 0.16 | 0.03 | 1.00 | -0.07 | -0.07 |
| mr | -0.01 | -0.04 | -0.21 | -0.01 | -0.01 | 0.02 | -0.02 | 0.18 | -0.06 | 0.06 | 0.08 | -0.05 | -0.05 | -0.07 | 1.00 | 0.08 |
| R | 0.05 | -0.04 | 0.01 | 0.01 | -0.04 | 0.04 | -0.06 | 0.02 | 0.06 | 0.06 | 0.05 | 0.06 | 0.08 | -0.07 | 0.08 | 1.00 |

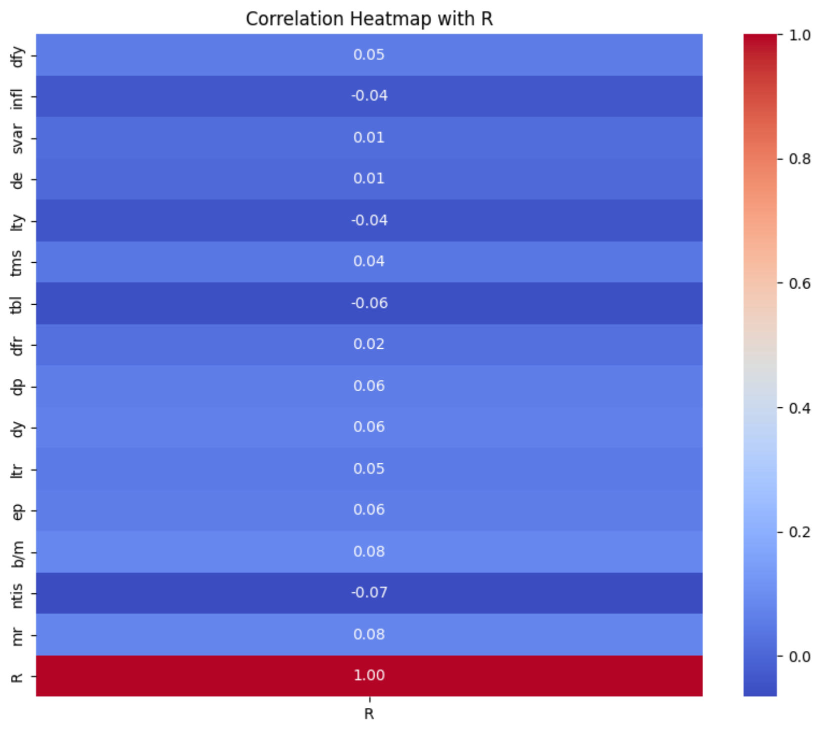
To determine whether the correlation between variables is reasonable, a benchmark is typically set. Correlation values between -0.7 and +0.7 are generally considered acceptable, as they indicate a moderate relationship without suggesting excessive overlap in the information conveyed by the variables. Correlations exceeding ±0.7, however, may indicate a strong relationship, raising concerns about multicollinearity.

Among our variables, a few exhibit very high correlation. For example, the dividend-to-price ratio (dp) and the dividend yield (dy) show an almost perfect correlation of 0.99. This high correlation arises because both variables measure essentially the same concept. Specifically, the dividend-to-price ratio calculates the difference between the logarithm of the sum of 12 months of dividends and the logarithm of prices. Similarly, the dividend yield represents the same measure but uses lagged prices, which explains their near-perfect correlation.

Since these variables convey nearly identical information, retaining both in the model could lead to redundancy and multicollinearity issues. To address this, it would be more effective to drop one of the variables, ideally retaining the one that provides the most relevant and distinct information for the analysis.

***Correlation of parameters with R***

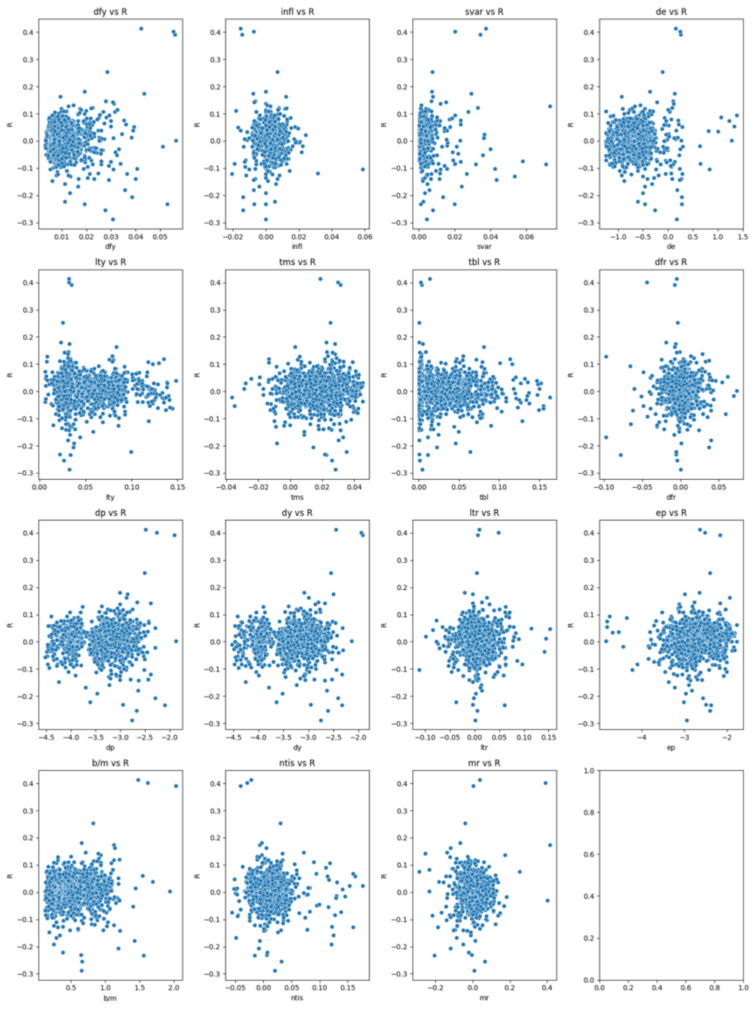
It is also important to examine the correlation between the regressors and the target variable, as this provides insight into which variables are most effective for forecasting excess returns. Identifying the regressors that have the greatest impact on the target variable will enable us to construct robust and accurate models for predicting excess returns.



We observe that all the variables show only a slight correlation with the target variable. However, this correlation analysis captures only the linear relationship between variables and does not account for causality. It is prudent to investigate potential non-linear relationships before proceeding with transformations or the creation of new variables.

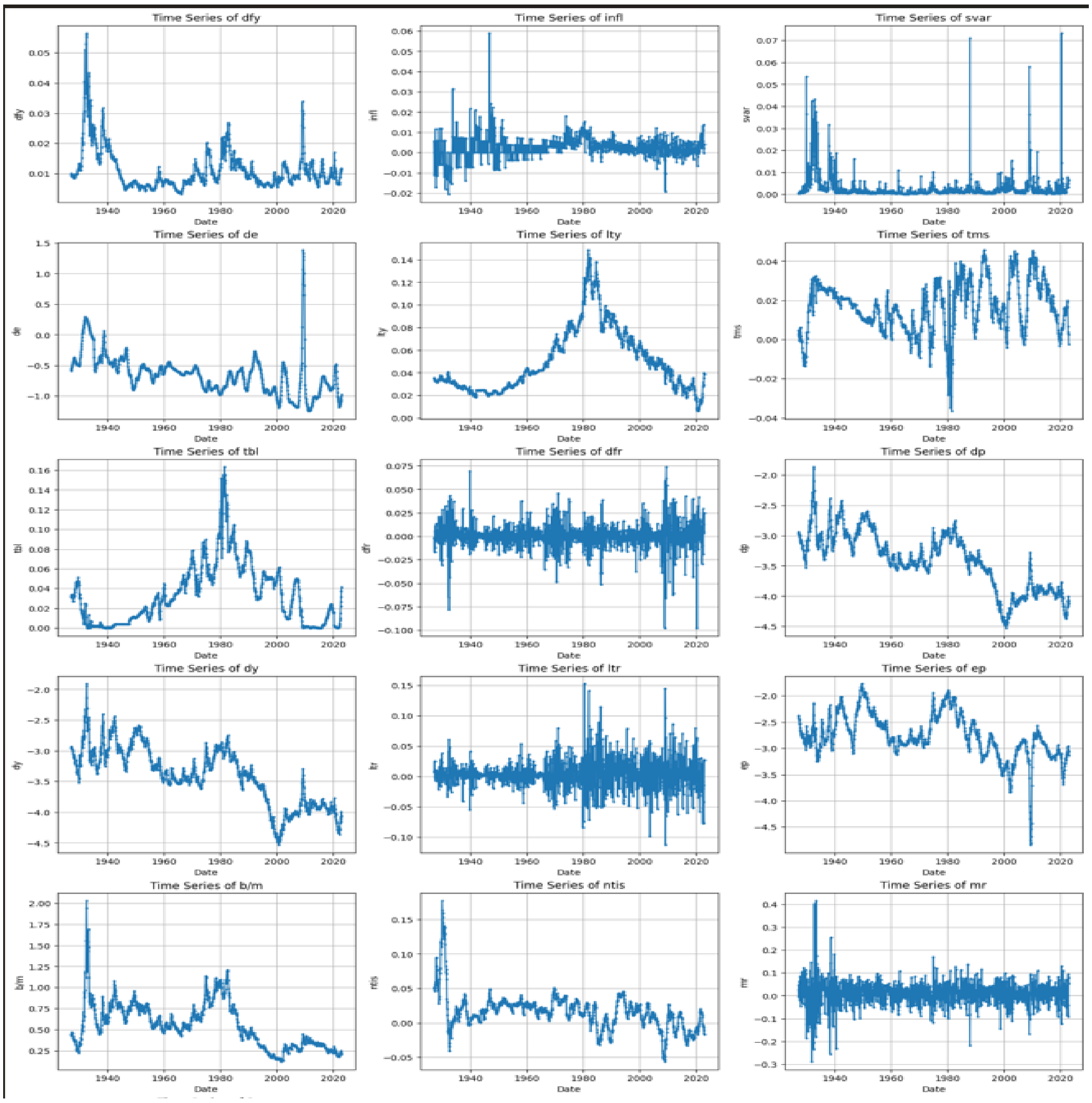
## Scatterplot against R

We also computed the scatterplot of the regressors against the target variable ‘R’. We mostly observed clusters of points with the presence of outliers. Overall, we cannot discern linear relationships between the variables, which suggested we should investigate feature engineering.



## Outliers

Addressing outliers requires first understanding their nature. It is crucial to determine whether an outlier results from a computational error or represents an extreme economic event, as the latter can provide valuable insights for forecasting excess returns. To investigate this, we plot the time series of the variables to identify the specific periods in which the outliers occur.



As we can observe from the different graphs all the variables exhibit similar trends around key time periods. Indeed, we can observe severe drops from most of the variables around the following years: 1930s, 1987, 2008 and 2020. Thos specific years represent periods of financial crisis and market downturns that have severally impacted the overall economy. Indeed, the 1930s were hit by the biggest economic downturn of the century and was caused by the stock market crash of 1929. Similarly, 1987 and

2008 were also characterized by financial market crashes and economic slowdown. More recently, the 2020 sanitary crisis cause a global economic crisis marked by a sharp decline in global GDP, disruptions in trade and stock and market volatility. We can also differentiate a special trend for inflation which was particularly high around the 1950s and which could be attributed to the post-war economic boom and the period of prosperity enjoyed in the US at that time. Overall, those historical events have caused our variables to sharply increase and decrease and thus they may alter our regression models. It could be relevant to create different machine learning models to assess whether financial crises have a significant impact on the excess return of our index.

## Methodology

### Introduction

The methodology section of this report is an account of the steps taken to apply machine learning techniques to predict monthly excess returns of the CRSP index using the Welch and Goyal (2008, 2022) dataset. This methodology consists of presenting the workflow, data preprocessing, model selection, implementation, and evaluation of the models explored. The objective is to develop and compare different predictive models, compare their performance and derive financial insight to suggest investment strategies based on the models developed.

### Data Collection and Preprocessing

#### *Data Sources*

For this project, the primary data source is the Welch and Goyal (2008, 2022) dataset, which is renowned for its comprehensive collection of financial predictors. While this dataset provides a wide range of variables, a subset of these predictors was **selected for** forecasting the excess returns of the CRSP index. This selected subset serves as the foundation for building and evaluating our predictive models

**Data Cleaning**

Data cleaning is a critical step to ensure the dataset is free from inconsistencies and suitable for robust machine learning model training. The following detailed steps were taken:

Data Handling: Although the dataset did not contain traditional missing values (e.g., blank cells or "NA" entries), certain feature engineering steps, such as the creation of lagged variables, naturally introduced NaN values at the beginning of the time series. To address this and prevent data leakage, a backward-fill (bfill) approach was employed.

Justification of bfill in this context:
Preventing Data Leakage: Using forward-fill (ffill) in this scenario would have meant using future values to predict past values, leading to data leakage. Backward-fill, on the other hand, fills the NaN values introduced by lagging with values from the past, maintaining the temporal integrity of the data.

Handling NaNs from Lagging: Lagging a time-series variable shifts the data forward, leaving NaN values at the beginning. bfill effectively fills these gaps with the next valid observation, preventing data loss that would otherwise occur if these rows were simply dropped.

It's important to note that this use of bfill is a strategic choice for data preparation in the context of time-series analysis and feature engineering, rather than an imputation technique for handling pre-existing missing data.

Outlier Detection and Treatment:
Visual Analysis: Boxplots, scatter plots and time series plots were employed to identify anomalies.

Historical Events: Key historical market downturns, such as the Great Depression (1930s), the 2008 financial crisis, and the 2020 COVID-19 market crash, caused extreme values. Instead of removing these outliers, they were retained to provide valuable training data for real-world financial forecasting.

Data Type Consistency: All columns were converted to appropriate data types to avoid computation errors during feature engineering and model training.
Duplicate Removal: Checked for duplicate observations to ensure that no repeated rows distorted the data distribution.
Index and Date Alignment: Verified that the date index was continuous, ensuring no gaps in monthly observations.

## Feature Engineering

Feature engineering was an iterative process, informed by both domain knowledge and insights gained from initial model training. The primary goal was to create new variables that could enhance the predictive power of the models while addressing issues like multicollinearity that arose during preliminary analyses.

### Moving Averages and Rolling Volatility

Moving averages and rolling standard deviations were calculated to capture trends and volatility in the data. For instance, a 3-month rolling mean and standard deviation were computed for the market return (mr), stock variance (svar), and Treasury bill rate (tbl). These features help in capturing the short-term momentum and risk associated with these variables. Additionally, a 3-month moving average was calculated for the default yield spread (dfy).

### Lagged Variables

Lagged variables were created to capture the delayed impact of certain predictors on the market return. Lags of 1 and 2 months were introduced for variables such as market return (mr), default return spread (dfr), inflation (infl), stock variance (svar), and net equity expansion (ntis). A 1-month lag was also created for the book-to-market ratio (b/m). These lagged variables allow the models to learn from past values of these predictors.

### Interaction Terms

Interaction terms were created to capture the combined effect of two predictors on the market return. The following interaction pairs were included: term spread (tms) and inflation (infl), dividend-price ratio (dp) and dividend yield (dy), and default yield spread (dfy) and Treasury bill rate (tbl). These interaction terms aim to capture potential non-linear relationships between these variables and the market return.

**Data Standardization**

Following the calculation of moving averages, rolling standard deviations, lagged variables, and interaction terms, the features were standardized. Data standardization was applied to ensure that all features had a mean of zero and a standard deviation of one. This process is critical for improving the performance and convergence of many machine learning algorithms, particularly those sensitive to feature scales. Standardization helps in giving equal weight to all features during model training.

**Iterative Feature Refinement and Addressing Multicollinearity**

Initial model training, especially with linear regression, revealed issues related to multicollinearity. Multicollinearity, where two or more predictor variables are highly correlated, can destabilize the model's coefficient estimates, making them sensitive to small changes in the data or model specification. This can lead to unreliable and difficult-to-interpret results, especially in a linear regression model.

To address this:

Correlation Analysis: A correlation matrix was computed and visualized as a heatmap to identify highly correlated features.

Variance Inflation Factor (VIF): While not shown in the provided code, computing the VIF for each predictor is a common next step after correlation analysis. VIF quantifies the severity of multicollinearity.

Feature Selection/Dropping: Based on the correlation analysis and insights from the initial models (e.g., examining coefficient magnitudes and p-values), certain features were dropped. Features like long-term yield (lty), default yield spread (dfy), book-to-market ratio (b/m), dividend yield (dy) and dividend-price ratio (dp) were removed due to their high correlation with other predictors, or low individual predictive power in initial model runs.

Model-Driven Insights:

The initial linear regression models (OLS, Ridge, and Lasso) served as valuable tools for refining the features. For example:

Coefficient Analysis: The magnitude and sign of the coefficients in the initial OLS model, along with their statistical significance (p-values), provided insights into the relative importance and direction of influence of each feature.

Regularization Effects: The shrinkage of coefficients in Ridge and Lasso regression highlighted features that were less important for prediction, guiding the decision to drop or modify them.

This iterative process, where model insights informed feature engineering, and vice versa, is a common and valuable practice in machine learning. It reflects a dynamic approach to model building, where the initial model serves as a diagnostic tool to guide further refinement. By addressing multicollinearity and iteratively refining our feature set, we aimed to create a more robust, stable, and interpretable model.

**Data Splitting**

Before any feature engineering or data transformation, the dataset was split into training and testing sets. This is a crucial step to prevent data leakage, ensuring that the model is evaluated on genuinely unseen data and that the feature engineering process doesn't inadvertently introduce information from the test set into the training process. The split was performed chronologically to maintain the temporal order inherent in time-series data. Specifically:

Training Set: Data prior to '2019-01-01' was used to create the training set. This set would be used to train the models and to derive the parameters and transformations during feature engineering.

Testing Set: Data from '2019-01-01' onwards constituted the testing set. This set was held out and used only for the final evaluation of the trained models.

Why Split Before Feature Engineering?

Splitting the data before feature engineering is a best practice, particularly important when dealing with time-series data, for the following reasons:

Preventing Data Leakage: Feature engineering techniques like calculating moving averages, creating lagged variables, or scaling based on the mean and standard deviation use information from the entire dataset. If these techniques are applied before splitting, information from the future (test set) will leak into the training set, leading to overly optimistic performance estimates that won't generalize to truly new data.

Realistic Model Evaluation: By splitting first, we ensure that the feature engineering process is based only on information available up to the training set's cutoff point. This mimics a real-world scenario where we would only have access to past data when making predictions about the future.

Maintaining Time-Series Integrity: In time-series analysis, the order of observations is critical. Chronological splitting ensures that we respect this order and avoid introducing future information into our training process.

Therefore, by splitting the data into training and testing sets at the outset, we preserved the integrity of the time-series data, prevented data leakage, and ensured a more realistic and reliable evaluation of our predictive models. After splitting, feature engineering steps (detailed in Section 3) were applied separately to the training and testing sets, using only parameters derived from the training set to transform the testing set.

## Model Selection and Justification

### Models Considered

The project explored two main types of machine learning models:

Linear Models:

Ordinary Least Squares (OLS): A baseline model used to provide a benchmark for performance.

Ridge Regression: A regularized linear model that mitigates overfitting by penalizing large coefficients.

Lasso Regression: Another regularized linear model that performs feature selection by shrinking irrelevant coefficients to zero.

Tree-Based Models:

Decision Trees: A simple yet powerful model that can capture non-linear patterns but is prone to overfitting.

Random Forest: An ensemble method that reduces overfitting by averaging multiple decision trees.

Gradient Boosting: A sequential ensemble technique that builds models iteratively, correcting previous errors.

### Model Choice

Ridge Regression: Chosen for its ability to handle multicollinearity and produce stable coefficient estimates, particularly important in datasets with highly correlated predictors. We used all the linear models to compare and select which one is the best predictor.

Lasso Preprocessing & Random Forest: Selected for its robustness to outliers and ability to model complex, non-linear relationships.

Hyperparameter Tuning

Hyperparameter Optimization To enhance the predictive performance of the models, hyperparameter optimization was conducted using a combination of grid search and cross-validation. This process systematically explores a predefined range of hyperparameter values to identify the combination that yields the best model performance on unseen data.

Ridge Regression:

For the Ridge regression model, the key hyperparameter tuned was the regularization strength, denoted by alpha (α). A range of alpha values was explored using a logarithmic scale to cover a wide spectrum of regularization strengths, from weak to strong. The specific values tested were [0.001, 0.01, 0.1, 1, 10, 100] The optimal alpha was selected based on the lowest mean squared error (MSE) achieved during cross-validation, as described below.

Random Forest:

The Random Forest model has several hyperparameters that influence its performance. The following were tuned:

n_estimators: This parameter defines the number of decision trees in the forest. A larger number of trees generally improves performance but increases computational cost. The values tested were [50, 100, 200, 300]. max_depth: This parameter controls the maximum depth of each decision tree. Deeper trees can capture more complex patterns but are prone to overfitting. The values tested were [None, 10, 20, 30]. min_samples_split: This parameter sets the minimum number of samples required to split an internal node. Higher values can prevent overfitting by ensuring that nodes are split only if they contain a sufficient number of samples. The values tested were [2, 5, 10]. 4.X.3 Cross-Validation for Robustness:

To ensure that the chosen hyperparameters generalize well to unseen data, 5-fold time-series cross-validation was employed. The training data was divided into five consecutive folds. In each iteration, four folds were used for training, and the remaining fold served as the validation set. The model was trained with a given set of hyperparameters, and its performance (measured by MSE) was evaluated on the validation fold. This process was repeated for each hyperparameter combination, and the average MSE across all folds was used to select the optimal set. This approach helps mitigate the risk of overfitting to a specific training set and provides a more robust estimate of the model's true performance. For the Random Forest model the RandomizedSearchCV was employed to reduce computation time.

## 4.2 Workflow Overview

Data Preprocessing: Handled missing values, standardized features, and ensured data integrity.

Feature Engineering: Created new features, including lagged variables, interaction terms, and moving averages.

Model Training: Implemented Ridge Regression and Random Forest with optimized hyperparameters.

Model Evaluation: Compared model performance using MSE, RMSE, R-squared, and MAE.

Result Interpretation: Derived actionable financial insights from model outputs.

## 4.3 Version Control Workflow

Version Control System: Git.

Branching Strategy: Created feature-specific branches (e.g., feature-engineering, model-training).

Commit Frequency: Made commits after major changes, with descriptive messages for clarity.

Collaboration: Used pull requests to review and merge changes, maintaining high code quality.

## 5. Evaluation Metrics

This project primary focused on Mean Squared Error (MSE) **and** the R-squared as the primary evaluation metrics. Below is a detailed explanation:

Mean Squared Error (MSE):

The MSE is a commonly used metric in regression analysis that measures the average of the squared differences between predicted and actual values.

A lower MSE value indicates better predictive accuracy, with predictions closer to the true values. Since errors are squared, MSE penalizes larger errors more heavily than smaller ones, making it sensitive to outliers. When using financial data, achieving a low MSE is challenging as financial returns are highly influenced by unpredictable factors such as market shocks or news events.

R-squared:

The R-squared is a statistical measure that indicates how effectively the independent variables of a model explain the variation in the dependent variables. It ranges from 0 to 1, where 0 means the model explains none of the variability in the dependent variable, and 1 means it explains all the variability.

This metric is crucial for machine learning and particularly for regression models, because it provides insights into the performance and reliability of our models. A higher

R-squared often suggests that the chosen features contribute significantly to explaining the outcome, thus, this can guide feature selection and engineering. In the context of predicting excess return of a market index, the aim is to approach an R-squared of about 5%.

## 5.2 Justification of Metrics

In predicting financial returns, large prediction errors can signal substantial risk. Using the Mean Squared Error (MSE) ensures that larger deviations are penalized more heavily because the error is squared, aligning the metric with the importance of minimizing high-risk discrepancies. MSE provides a clear and consistent basis for comparing different models. It measures how well each model captures the variability in the data, making it easier to identify the most effective predictive model. Nevertheless, MSE is highly sensitive to outliers because squaring the errors amplifies the impact of extreme values. This can skew results if the dataset contains anomalies or extreme deviations. Additionally, since MSE is measured in the square of the original units, interpreting the results can be less intuitive compared to metrics expressed in the same units as the target variable.

R-squared is also valuable for comparing models. It allows for a straightforward assessment of how well different models explain the variability in the target variable, making it a key tool in model selection. Its interpretability in the context of variability explained by the predictors adds an additional layer of insight when analysing model performance. However, R-squared has limitations. It does not account for model complexity, meaning a higher R-squared value does not always indicate a better model, particularly if overfitting occurs. Furthermore, in the context of financial return prediction, a high R-squared is often unrealistic due to the inherently noisy and unpredictable nature of financial data.

## 6. Assumptions and Limitations

### 6.1 Assumptions

The relationship between predictors and the target variable remains stable over time.

Macroeconomic shocks do not drastically alter the model's predictive capacity during the testing period.

Predictors used are sufficient proxies for broader economic conditions.

### 6.2 Limitations

Financial market data is inherently noisy and non-stationary, making predictions susceptible to fluctuations.

Multicollinearity among predictors may still affect interpretability, even with regularization.

Models may underperform in periods of extreme market uncertainty.

## 7. Reproducibility

### 7.1 Environment Details

Python Version: 3.9.

Development Tools: Jupyter Notebook for exploratory analysis, VS Code for script development.

Operating System: macOS/Linux/Windows.

### 7.2 Configuration Files

# 3 Model Performance and Comparison

## 3.1 Ordinary Least Square Model

We first ran the Ordinary Least Squared model to assess the relationships between the features and dependent variables. For instance, we can observe that the feature “dfy_ma3_lag3” has the largest negative coefficient (-0.0536), implying that this lagged measure of the default yield spread significantly reduces the predicted excess returns. Similarly, “dfy_ma3_lag1” has a positive coefficient (0.0376), indicating that this lag has a positive impact on returns. The magnitude of coefficients highlights which features have the most influence on the dependent variable. Features like “dfy_ma3_lag3”, “dfy_ma3_lag1” and “dp_dy” have relatively larger coefficients comapred to others, suggesting they are more influential predictors. Conversely the lags of the past returns seem to have negligible coefficients, indicating minimal impact on the model’s predictions. Features with larger coefficients may have stronger economic implications. For instance, the default yield spread (dfy) and its lags often reflect credit risk and economic stress, making them critical predictors for financial models. Similarly, the dividend-price ratio (dp_dy) aligns with traditional asset pricing theory.

## 3.2 Lasso Model

From the output of the Lasso model, we observe several coefficients that are non-zero, indicating that the corresponding features are significant predictors in the model. The features with the largest non-zero coefficients include: “dfy_ma3_lag1” (0.0362). This feature has a strong positive effect, meaning that an increase in this variable is associated with an increase in the predicted outcome. This is interesting because the OLS model predicted this feature to have a negative relationship on the target variable.

Furthermore, coefficients such as “ntis_lag2”, “tms_x_infl”, “mr” and “mr_rolling_mean” all appear to contribute to predicting the target variable.

On the contrary, features like “mr_lag1”, “infl_lag1”, “ep”, “infl_lag2” all have coefficients of zero. Lasso has effectively removed these variables from the model by shrinking their coefficients to zero. This means that these features do not contribute to the model’s prediction, either because they are redundant or not informative enough in the context of the other features. Notably, while other models generally improved with increased features and complexity, the Lasso model's performance either stagnated or degraded, suggesting that in this case, a simpler model with fewer features might be more suitable.

## 3.3 Model Comparison

The performance of the OLS, Ridge, and Lasso models was evaluated both on the training (in sample) data and test (out sample) data. In terms of in sample performance, all three models exhibited identical Mean Squared Errors (MSE) of 0.0027, suggesting that they fit the training data similarly. However, the R-squared values revealed key differences in their ability to explain the variance in the target variable. The OLS model had the highest R-squared value of 0.0961, indicating that it explained 9.61% of the variance in the data. In comparison, Ridge and Lasso both showed lower R-squared values, with Ridge at 0.0816 and Lasso at 0.0789, indicating that regularization reduced their ability to capture the underlying patterns in the training data. Despite their similar MSE, Ridge and Lasso were less effective than OLS at explaining the variance in the target variable.

On the test (out sample) data, the performance of all three models deteriorated, showing signs of overfitting. The MSE increased for each model, with OLS showing an MSE of

0.0033, while Ridge and Lasso both exhibited MSEs of 0.0035. This suggests that all models struggled to generalize to new, unseen data. The R-squared values on the test data were negative for all three models, with OLS at -0.0734, Ridge at -0.1366, and Lasso at -0.1265. Negative R-squared values indicate that the models performed worse than a simple baseline model that predicts the mean value of the target variable. This poor test performance highlights the overfitting issue, where the models captured patterns specific to the training data but failed to generalize effectively to new data.

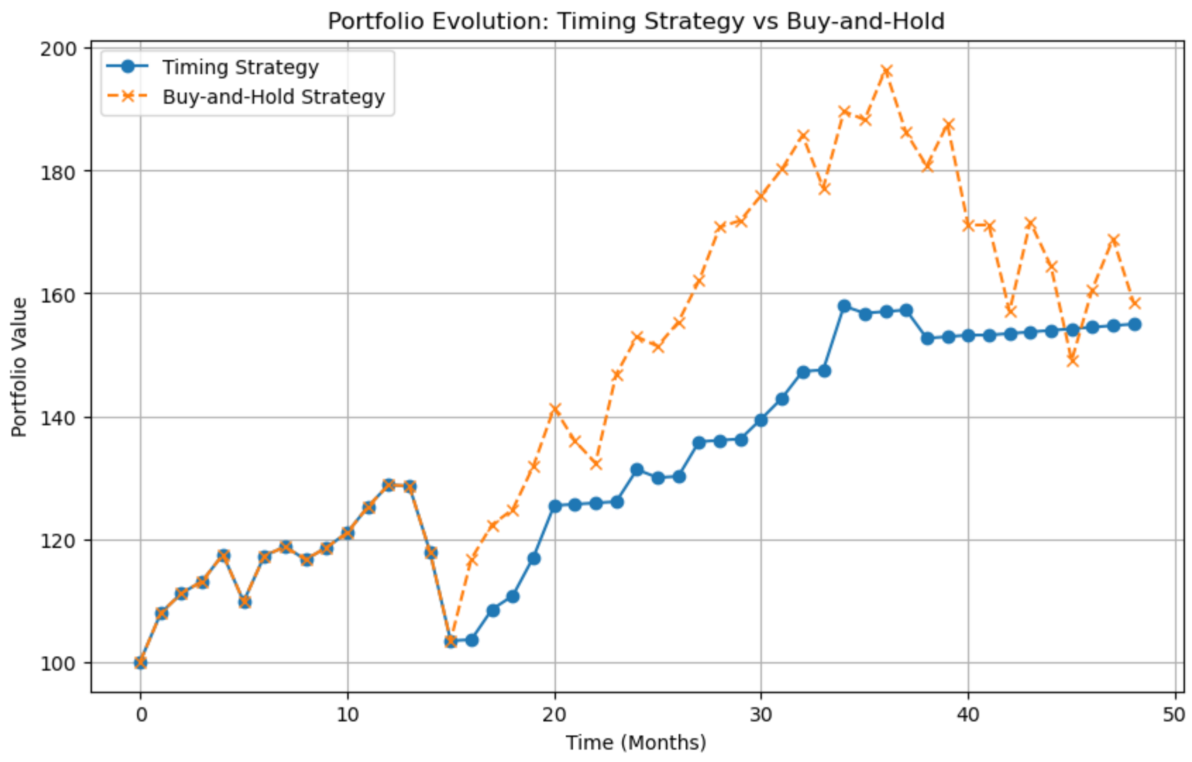
Overall, while Ridge and Lasso added regularization to prevent overfitting, their performance on both the in sample and out sample data was not significantly better than OLS. This suggests that further model refinement, such as additional feature engineering or tuning of regularization parameters, may be necessary to improve the models' ability to generalize and capture the underlying patterns in the data.

## 3.4 Timing Strategy VS. Buy-and-Hold strategy

### General concept and ideas

In the context of our machine learning project, implementing trading strategies is essential to evaluate whether the models we develop generate meaningful and robust predictive signals. The buy-and-hold strategy represents a passive investment approach, where an investor purchases shares of the CRSP index and holds them over an extended period, regardless of short-term market fluctuations. This strategy is grounded in the belief that markets tend to grow over time, meaning the observed returns are directly linked to the overall performance of the index. In contrast, the timing strategy is an active investment approach that involves dynamically adjusting exposure to the CRSP index based on predictions of future market movements. This strategy leverages predictive signals, indicators, and market forecasts to optimise the timing of entry and exit decisions. By doing so, it aims to achieve returns that exceed those of a passive buy-and-hold approach, albeit with increased reliance on accurate forecasting and timely decision-making

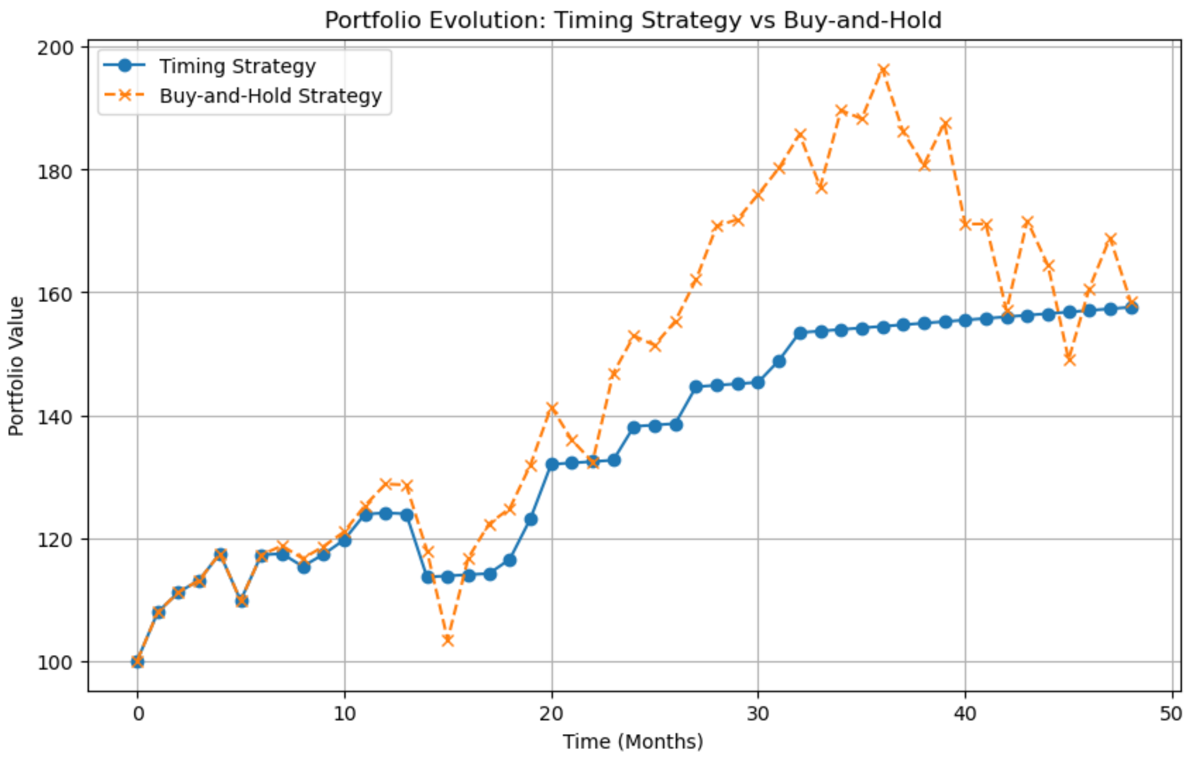
## Ordinary Least Square interpretation



As we can observe, the cumulative return of the OLS timing strategy is noticeably lower than that of the buy-and-hold strategy, indicating underperformance. This suggests that the model's predictions are not effective in identifying profitable timing signals. While both strategies exhibit similar overall patterns, the timing strategy appears flatter and less pronounced, reflecting its limited ability to amplify returns. Additionally, the Sharpe ratio, which measures risk-adjusted returns, is low, highlighting the strategy's inability to effectively balance returns and risk. Furthermore, the timing strategy experienced a significant drawdown of –19.7%, signalling substantial losses during market downturns.

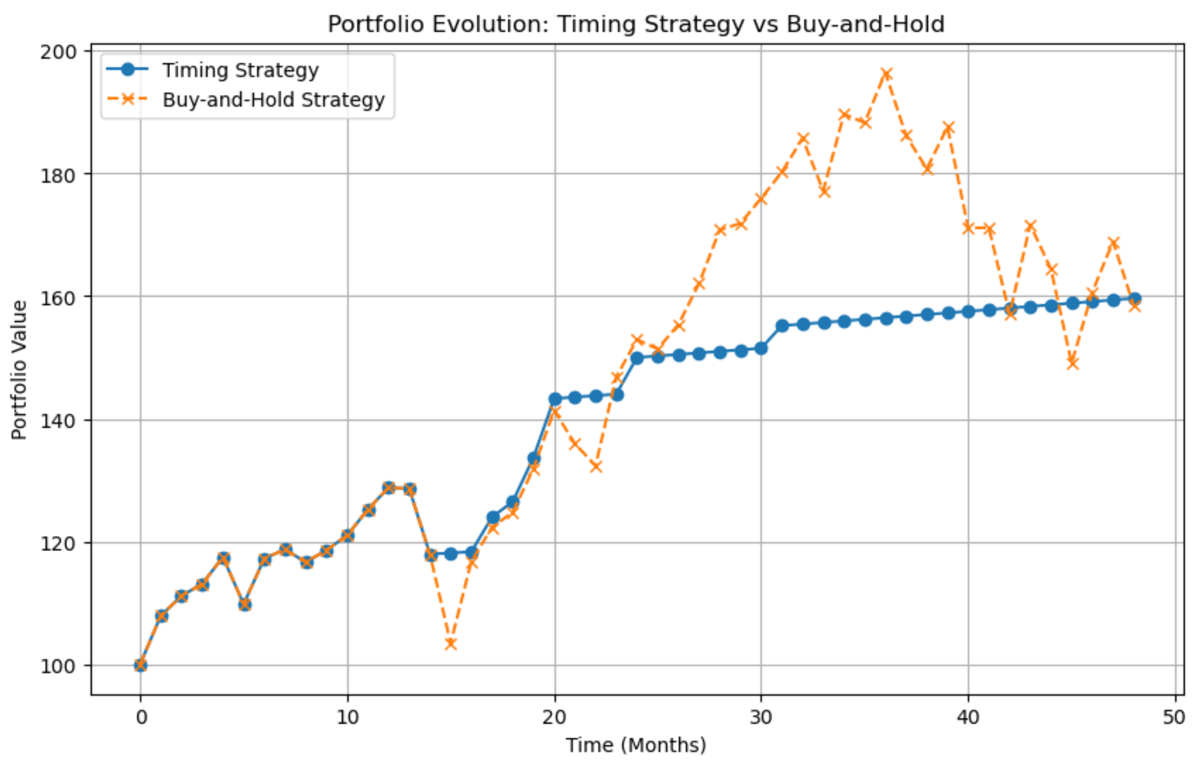
Overall, the OLS model struggled to generate strong predictive signals, resulting in suboptimal returns and inadequate risk management. Its linear nature likely fails to capture the complexity and nuances of the data, limiting its effectiveness in predicting market movements.

## Ridge interpretation



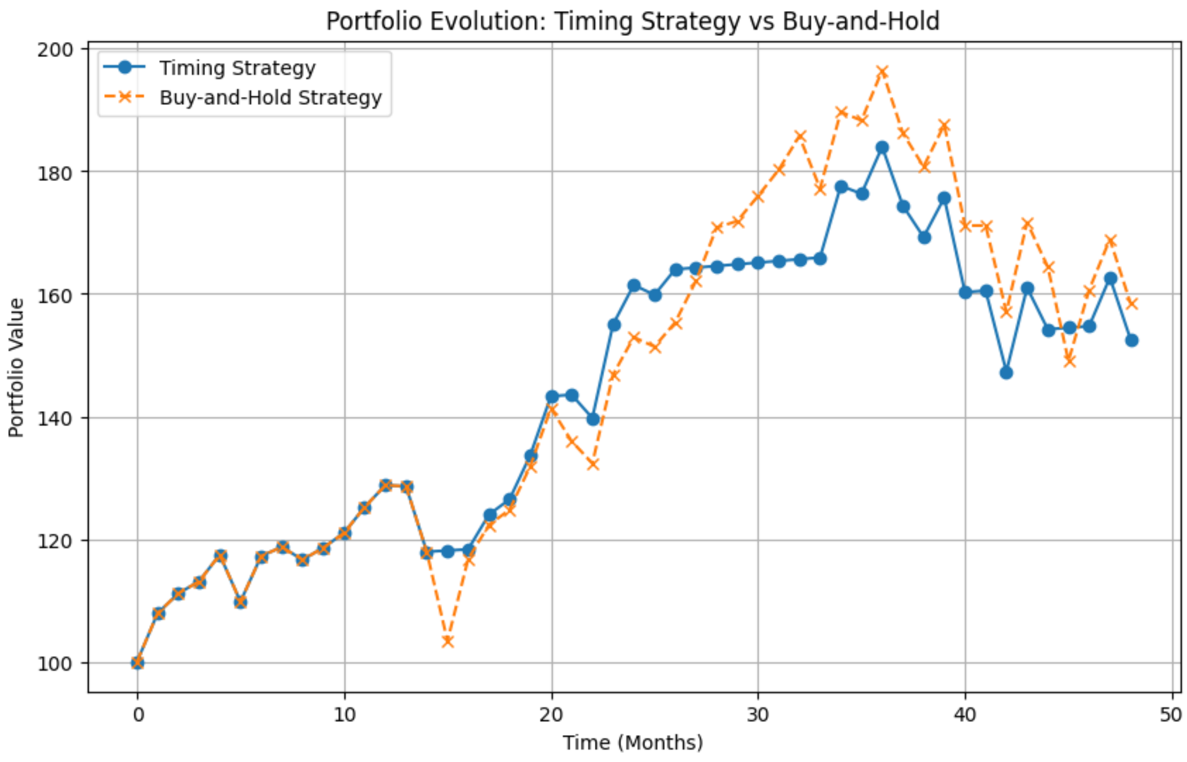
Among the models evaluated, ridge regression demonstrates a moderate cumulative return for the timing strategy, indicating superior predictive performance. Additionally, the Sharpe ratio for ridge regression shows significant improvement at 0.3, reflecting better risk-adjusted returns compared to the OLS model. The drawdown is also the lowest among the models, suggesting effective risk management. These results highlight ridge regression's outperformance, likely due to its ability to regularize coefficients, effectively addressing multicollinearity and reducing noise in the data. Visually, the timing strategy under ridge regression occasionally surpasses the buy-and-hold strategy, particularly during periods of market crises. However, overall, the buy-and-hold strategy continues to deliver better overall performance.

## Lasso Interpretation



The Lasso model demonstrates a cumulative return of 59.65%, indicating a slightly higher return than the Ridge model, suggesting its ability to generate favorable returns during the evaluation period. However, with a Sharpe ratio of 0.31, the model still shows a relatively low risk-adjusted performance, implying that the returns may not be fully compensating for the risk taken. Additionally, the maximum drawdown of -8.42% indicates that the model experienced a peak-to-trough loss of 8.42% at some point, which reflects the potential for significant short-term declines. Overall, while the Lasso model provides a strong cumulative return, its risk-adjusted performance remains modest, and it exhibits similar levels of volatility as the Ridge model. Lastly, while it performs similarly or better than the buy-and-hold model during the first 25 months, it then quickly becomes outperformed by the other strategy.

Random Forest interpretation



The Random Forest model shows a cumulative return of 82.55%, which is significantly higher than both the Ridge and Lasso models, suggesting it can capture more profitable opportunities during the evaluation period. However, with a Sharpe ratio of 0.27, its risk-adjusted performance is relatively low, indicating that the returns achieved might not be sufficient to compensate for the risks undertaken. Additionally, the maximum drawdown of -12.89% highlights a more pronounced downside risk, as the model experienced a peak-to-trough loss of nearly 13%. While Random Forest demonstrates impressive cumulative returns, its relatively low Sharpe ratio and higher drawdown emphasize the need to consider its volatility and potential risk exposure in portfolio management. Graphically, the trend of the Random Forest timing strategy closely mirrors that of the buy-and-hold strategy, indicating that the returns generated are like what could be achieved by simply holding the index. This suggests that the model is not producing distinctive trading signals that outperform the broader market.

## 3.6 Financial implications and recommendations

The results from the different models provide important insights for investors. In terms of in-sample performance, OLS achieves the highest R-squared of 0.0961, suggesting it explains the most variance in returns among the models. Ridge and Lasso, with R-squared values of 0.0816 and 0.0789 respectively, show a slightly lower ability to explain the variability in returns. The key coefficients in Lasso, such as “ntis_lag5” and dfy_ma3_lag3, indicate the relevance of macroeconomic variables, while the Ridge model highlights similar important factors, such as ntis_lag3 and mr_lag48. These suggest that macroeconomic indicators and market returns are crucial for predicting excess returns.

In terms of out-of-sample performance, all models show a decline, with OLS performing the best but still yielding a negative R-squared of -0.0734, indicating that none of the models generalize well to unseen data. This emphasizes the difficulty of accurately predicting market returns outside the training period.

From a financial perspective, the cumulative returns are most notable for the Random Forest model, which achieves 82.55%, compared to 57.59% for Ridge and 59.65% for Lasso. However, Random Forest also has a higher maximum drawdown of -12.89%, indicating higher risk. The Sharpe ratios are relatively modest, with Ridge (0.3) and Lasso (0.31) showing similar risk-adjusted returns, while Random Forest has a slightly lower Sharpe ratio of 0.27.

In conclusion, while Random Forest offers the highest potential return, it carries significant risk. For investors, a balanced approach considering the stability of Ridge and Lasso, along with ongoing model refinement, is recommended. Macro indicators such as ntis_lag5 and dfy_ma3_lag3 should be closely monitored as they influence returns.

## 3.7 Limitations

The results from the various timing strategies can be attributed to the negative R-squared values obtained from our models. This suggests that the models' predictions did not generalize effectively, resulting in poor performance from the timing strategies. Given the unreliability of our models, it is not surprising that the timing strategies underperformed the buy-and-hold approach, as we were unable to generate high-quality signals. However, it is important to note that financial markets are inherently noisy, and predicting excess returns remains a challenging task. Timing strategies often experience diminishing returns because market prices typically incorporate most available information.

## 3.8 Conclusion and future insights

This project aimed to predict the monthly excess returns of the CRSP value-weighted index by leveraging financial predictors and applying machine learning models to identify robust patterns and derive actionable insights for market forecasting and investment decision-making. Throughout the analysis, several critical steps were undertaken, including data preprocessing, feature engineering, model training, and performance evaluation, to ensure a comprehensive approach toward building effective predictive models.

Our exploration of multiple machine learning models, including linear regression (OLS, Ridge, Lasso) and tree-based models (Decision Trees, Random Forest), highlighted key strengths and limitations in capturing the complex relationships between financial predictors and excess returns. Among the linear models, Ridge regression demonstrated strong regularization capabilities, mitigating overfitting and providing stable coefficient estimates. The Lasso model effectively performed feature selection, identifying significant predictors while excluding less informative variables. However, both models faced challenges in generalizing well to unseen data, as evidenced by negative R-squared values and increased mean squared errors on the test set.

On the other hand, tree-based models, particularly Random Forest, excelled at capturing non-linear interactions and performed significantly better in terms of cumulative returns in the timing strategy. However, despite its high cumulative return, the relatively low Sharpe ratio and considerable drawdown highlighted the trade-off between return and risk. This underscores the inherent volatility and unpredictability of financial markets, which limit the capacity of even advanced models to consistently outperform simpler buy-and-hold strategies.

## 3.9 Key Insights and Implications

1. **Predictive Power of Features:**

2. The analysis reaffirmed the importance of key financial indicators such as the default yield spread, stock variance, and dividend-price ratio in explaining variations in market returns. The significance of lagged and interaction terms emphasized the need to account for time-dependent and joint effects when building predictive models for financial data.

3. Model Selection:

While linear models provide interpretability and are valuable for understanding the relationships between variables, they may struggle to capture the non-linear dynamics present in financial time series. In contrast, tree-based models can capture complex interactions but may introduce higher volatility in predictions.

4. **Performance of Timing Strategies:**

The implementation of timing strategies based on predictive signals revealed that, despite improvements in cumulative returns over time, the Sharpe ratios and drawdown levels indicated suboptimal risk-adjusted performance. This suggests that relying solely on machine learning-based timing signals may not consistently outperform passive investment strategies, particularly during volatile market conditions.

5. **Financial Crises and Market Events:**

The presence of extreme values corresponding to historical market downturns such as the Great Depression, the 2008 financial crisis, and the 2020 COVID-19 pandemic demonstrated the importance of contextualizing model performance in the face of major economic events. These events introduced substantial noise and risk, complicating the task of accurate prediction

## 3.10 Future Directions

To improve the robustness and predictive power of the models:

1. **Feature Enrichment:** Incorporating additional economic indicators, such as geopolitical risk indices, commodity prices, or investor sentiment indices, could enhance predictive capabilities.
2. **Time-Series-Specific Models:** Utilizing time-series models such as ARIMA, VAR, or Long Short-Term Memory (LSTM) networks could capture sequential dependencies more effectively.

3. **Dynamic Strategies:** Developing adaptive trading strategies that adjust to market conditions in real-time could improve performance compared to static timing signals.

## Appendices