TU Dortmund

Case Studies

Project II: Forecasting The Equity Premium Using Machine Learning

Lecturers:
Prof. Dr. Matei Demetrescu
Dr. Paul Navas

Author: Bushra Tariq Kiyani

Group number: 4

Group members: Sarmistha Bhattacharyya, Oliver Fischer

December 14, 2023

Contents

1 Introduction

In previous analysis, the focus was on leveraging AR and linear regression models to forecast stock market returns. Now, we endeavor to elevate the predictive accuracy and refine forecasting techniques by employing advanced machine learning methodologies in financial analysis.

The dynamism and complexity of financial markets continually challenge investors, analysts, and portfolio managers to harness more sophisticated predictive models. Accurately forecasting stock returns remains a critical pursuit, holding the potential to redefine investment strategies and fortify risk management practices. The capacity to generate precise predictions in financial markets is crucial for optimizing asset allocation decisions and enhancing overall portfolio performance. (Julio et al., 2022).

This project diverges from traditional statistical approaches and instead embraces the power of machine learning algorithms to predict excess stock returns. The focus is to utilize versatility and robustness of regression trees and random forests, to forecast stock returns based on a diverse range of lagged predictors. The project aims to construct predictive models that effectively capture the intricate relationships between various economic indicators, stock market indices, interest rates, corporate bond yields, and macroeconomic variables. Using a dataset comprising monthly data points only this time, these machine learning models will undergo rigorous evaluation through the Mean Squared Forecast Error (MSFE). The objective is to harness the potential of these models to provide more accurate and reliable forecasts of excess stock returns.

The subsequent sections will delve into the intricacies of employing regression trees and random forests, showcasing their application, interpreting results, and drawing meaningful insights to enhance our understanding of stock market predictions. The second section describes the structure and quality of the dataset in more detail. Additionally, the goals of the project are stated in the second section. The third section explains the statistical methods. The fourth section focuses on the application of these methods and the interpretation of the plots and results. Finally, the fifth section summarizes the most important findings.

2 Problem statement

2.1 Dataset and Data Quality

This project relies on the data collected by the authors of the paper "A Comprehensive Look at the Empirical Performance of Equity Premium Prediction" (Welch and Goyal, 2007). Updated data until 2022 sourced from Amit Goyal's webpage is utilized (Goyal, 2022). Collection details are outlined in the paper. The dataset, ranging from 1871 to 2022, centers on the equity premium as the dependent variable, calculated across monthly, quarterly, and yearly intervals. Only the monthly data is utilized in this project. Comprising 18 independent variables and 1824 observations, a comprehensive dataset overview is available in the initial report, excluding any missing values from the analysis.

2.2 Project Objectives

The project objectives have shifted to an extensive comparison between regression trees, and random forests for forecasting monthly series. Firstly, one-step-ahead forecasts will be made using each method based only on the excess return lags, followed by forecasts using both covariates and excess return lags. The Root Mean Squared Forecast Error (RMSFE) will be computed for the obtained forecasts using suggested R packages such as rpart, rpart.plot (for regression trees), and ranger, or randomForest (for random forests).

A significant consideration is the interpretability of machine learning methods, often perceived as "black boxes" due to their opaque performance rationale, contrasting with the transparency of previous linear models. To address this, variable importance measures will be applied to rank the importance of variables. The Mean Squared Forecast Error (MSFE) will be used to compute variable importance for forecasts based on covariates and excess return lags. Subsequently, a discussion on the most crucial variables and any discrepancies between the two selected prediction methods will be undertaken, exploring differences in variable importance.

3 Statistical methods

3.1 Regression Tree

Regression trees are a type of decision tree used in machine learning for predictive modeling, specifically designed for regression tasks. They are used to predict continuous numeric outcomes rather than discrete categorical labels. These trees excel in applications like price predictions, quantity estimations, and forecasting continuous variables due to their specialized focus on the continuous outcome forecasting.

They represent a flowchart-like structure where each internal node denotes a test on an attribute or feature, each branch represents an outcome of the test, and each leaf node represents a predicted value. The tree-building process involves recursive binary splitting based on predictor variables. At each node of the tree, the algorithm chooses a predictor, and a split point to partition the data into subsets that minimizes suitable criteria related to the target variable.

Splitting Criteria: The primary goal of splitting in regression trees is to create homogeneous subsets with small variance in the target variable within each subset. Common splitting criteria include Residual Sum of Squares (RSS)

$$\text{RSS} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

and Mean Squared Forcast Error (MSFE)

$$\text{MSFE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

where n is total number of observations, y_i is observed value of the dependent variable and $\hat{y}_i$ is predicted value of the dependent variable for the ith observation.

Leaf Nodes and Prediction: The terminal nodes (leaf nodes) of the tree contain predicted values. When a new data point traverses the tree, it reaches a leaf node, and the predicted value for that data point is the average (or another aggregation) of the target variable values in that leaf node. (Hastie et al., 2009, p. 295-336)

3.1.1 Pruning

Pruning is process of reducing the size or complexity of a tree after it has been fully grown or constructed. The main goal of pruning is to prevent overfitting and improve the tree's generalization ability on unseen data.

Pre-pruning: Pre-pruning takes place within the tree-building phase, preceding the tree's full expansion to its maximum complexity or depth. It involves imposing conditions or constraints to regulate the tree's growth. Which includes imposing an upper limit on the tree's depth or number of levels, establishing the minimum count of samples necessary to proceed with splitting a node further or setting a benchmark for the minimum enhancement in the error metric (e.g., RSS or variance reduction) required to validate a split. Pre-pruning is computationally less expensive since it restricts tree growth during construction. It helps control tree size and complexity early in the process, which can speed up training and reduce memory usage. However, pre-pruning might oversimplify the tree, leading to underfitting if the constraints are too stringent.

Post-pruning: Post-pruning serves as a technique applied subsequent to constructing a decision tree, with the purpose of diminishing its scale, intricacy, and the risk of overfitting. Its objective is to simplify the tree structure by eliminating branches or nodes that hold minimal significance in enhancing predictive accuracy. It can be done using two techniques Cost-Complexity pruning technique involves growing the complete tree and then systematically eliminating nodes to identify the subtree that strikes a balance between reduced complexity and optimal accuracy. Subtree Replacement technique involves substituting entire subtrees with a solitary node. Post-Pruning techniques allow the tree to grow to its full size before pruning, potentially capturing more complex relationships. They aim to strike a balance between simplicity and accuracy. Cost-complexity pruning, particularly, is known for its effectiveness in producing simpler trees while maintaining predictive performance. (Hastie et al., 2009, p. 295-336)

3.1.2 Algorithm

- **Input:** Training dataset with predictor variables X and target variable Y.
- **Initialization:** Start with the entire dataset represented as the root node and choose a stopping criteria (e.g., minimum node size or maximum depth of the tree) to control the tree's growth.

- **Recursive Splitting:**
  - **Step 1:** Select Predictor and Split Point.
    - Evaluate all predictors and their possible split points.
    - Choose the predictor and split point that minimize the Residual sum of square (RSS) or maximize variance reduction in the node.
  - **Step 2:** Create Child Nodes.
    - Split the data into two child nodes based on the selected predictor and split point from Step 1.
  - **Step 3:** Recursive Loop.
    - Recursively apply Steps 1 and 2 to each child nodes until the stopping criteria are met.
- **Terminal Nodes (Leaves):** These leaves represent the final partitions or segments of the dataset where predictions are made.
- **Output:** Predictions that can be made on new or unseen data based on decision rules learned from the tree structure. (Hastie et al., 2009, p. 295-336)

3.1.3 Considerations

Advantages: Intuitive and easy to interpret, making them suitable for explaining relationships between predictors and continuous outcomes. Nonlinear relationships between predictors and the target variable can be captured effectively.

Limitations: Prone to overfitting, especially when the tree grows too large or lacks proper pruning. Can be sensitive to small variations in the data, leading to different trees for slightly different datasets. (Hastie et al., 2009, p. 295-336)

3.2 Random Forest

Random Forests are an ensemble learning technique used for both classification and regression tasks. They combine the predictive power of multiple decision trees to make more accurate predictions. Unlike a single decision tree, Random Forests employ a collection of trees and then aggregate their outputs to arrive at a final prediction. This ensemble approach tends to yield more robust and accurate results.

At the core of Random Forests lie decision trees, which are inherently prone to overfitting when grown too deep or with high variance in the dataset. To address this, Random Forests introduce two key concepts: bagging and random feature selection. **Bagging (Bootstrap Aggregating):** Random Forests use bootstrapping, a resampling technique, to create multiple datasets by sampling with replacement. Decision trees are built on each of these bootstrapped datasets and then aggregate the output. **Random Feature Selection:** At each node of the decision tree, instead of considering all features for splitting, Random Forests randomly select a subset of features. This reduces the chance of a single dominant feature influencing the tree and enhances diversity among the trees. (Hastie et al., 2009, p. 587-604)

3.2.1 Algorithm

- **Bootstrapping:** Randomly sample from the dataset to create multiple bootstrap samples.
- **Regression Tree Building:**
  - For each bootstrap sample, construct individual regression trees.
  - Grow trees by recursively splitting nodes based on random feature subsets.
  - Stop splitting based on defined criteria (e.g., maximum depth, minimum samples per leaf).
- **Ensemble Creation:** Aggregate predictions from all trees by averaging the outputs. (Hastie et al., 2009, p. 587-604)

3.2.2 Considerations

Advantages: The ensemble approach reduces overfitting, enhancing generalization to unseen data. Random Forests perform well on large datasets without heavy preprocessing. Provides insights into the importance of features in the prediction process.

Limitations: Training Random Forests can be computationally expensive for large datasets and numerous trees. Due to their ensemble nature, Random Forests are considered "black box" models, making interpretation challenging compared to simpler models like linear regression. (Hastie et al., 2009, p. 587-604)

3.3 Permutation Feature Importance

Permutation Feature Importance is a technique used in machine learning to assess the significance of different features in a predictive model. It evaluates the importance of each feature by observing how the model's performance deteriorates when the values of that feature are randomly shuffled or permuted, disrupting their original relationship with the target variable. This disruption helps in understanding the impact of individual features on the model's predictive capability.

This technique is valuable as it allows us to discern the influence of individual features on the model's predictive accuracy. By measuring the change in model performance, typically evaluated by metrics like accuracy, mean squared forcating error, or others, when a feature's values are randomly shuffled, we can determine the relative importance of that feature. If the model's performance notably declines after the permutation, it suggests that the feature holds substantial predictive power. Conversely, if the model remains relatively unaffected, it implies that the feature might have lesser importance or correlation with the target variable. (Molnar, 2023, Chapter 8.5)

3.3.1 Algorithm

Inputs: A trained model $\hat{f}$, predictor matrix X, target variable vector y, and an error measure/loss $L(y, \hat{f})$.

1. Start by calculating the original model error: $e_{\text{o}} = L(y, \hat{f}(X))$ using a specified error measure such as mean squared forecast error.
2. For each predictor j in the set of predictors $1, ..., p$:
    - Create a covariate matrix X_{p} by permuting feature j within the dataset X.
    - Calculate the error $e_{\text{p}} = L(Y, \hat{f}(X_{\text{p}}))$ by making predictions using the permuted data.
3. Compute the permutation feature importance:
    - Determine the permutation feature importance either as the quotient $FI_j = e_{\text{p}}/e_{\text{o}}$ or as the difference $FI_j = e_{\text{p}} - e_{\text{o}}$.
4. Arrange features in descending order based on their permutation feature importance (FI).

(Molnar, 2023, Chapter 8.5)

3.4 K-Fold Cross Validation

K-fold cross-validation is used in machine learning to assess model performance and generalization ability. It involves splitting the dataset into K equally sized subsets (or folds). The model is trained K times, each time using K-1 folds for training and the remaining fold for validation. This process yields K performance scores, which are averaged to compute a more robust estimation of the model's performance. K-fold cross-validation helps in understanding a model's bias-variance tradeoff, identifying potential issues like overfitting or underfitting, and aids in hyperparameter tuning and model selection. Ultimately, it provides a reliable evaluation of a model's predictive capability by reducing the reliance on a single train-test split. (Hastie et al., 2009, Chapter 7)

3.5 Bar Charts

Bar charts are graphical representations that display categorical data using rectangular bars of different heights or lengths. Each bar typically represents a specific category or group, and the length or height of the bar corresponds to the numerical value or frequency associated with that category. These charts offer a visual comparison of data across different categories, making it easy to identify patterns, trends, or variations among the groups. They are effective in facilitating comparisons, and highlighting differences or relationships between distinct categories. (Christopher, 2019, Chapter 7)

4 Statistical analysis

In this section, a comprehensive analysis of the dataset is presented using machine learning models discussed previously. For model training and testing, and graphical representations R software (R Core Team, 2022) Version, 4.2.1 is used with additional packages **dplyr** (Wickham et al., 2022), **ggplot2** (Wickham, 2016), **readr** (Wickham et al., 2023), **rpart** (Therneau and Atkinson, 2022), **rpart.plot** (Milborrow, 2022), **randomForest** (Liaw and Wiener, 2022), and **caret** (Kuhn and Others, 2022).

The dataset is initially imported from the file "PredictorData2022.xlsx - Monthly.csv." The column names of the dataset are then updated for better readability and understanding. The excess returns series is generated based on the stock returns and the risk-free rate. The excess returns are computed as the difference between the growth rates of the "Index" series and the corresponding lagged "Risk-free Rate" values. To conduct forecasts, the dataset is processed to include lagged excess returns and lagged predictors. The columns related to the stock returns, date, index, and risk-free rate are excluded as they are used to derive the dependent variable.

4.1 Generation of One-Step-Ahead Forecasts Using Lagged Excess Returns

In this task, the predictive models are developed exclusively based on lagged predictors of excess returns spanning from lag 1 to lag 10. The Regression Trees and Random Forests models are constructed using the rpart and randomForest functions in R, respectively. The models are trained and evaluated for their forecasting performance using a dataset split into training and testing sets with an 80:20 ratio. The regression tree models, relying on the lagged excess returns, are tuned through the rpart package, while Random Forest models are trained using the randomForest function. Cross-validation, employing a five-fold partitioning strategy, is applied to the dataset to train and assess the models' performances. Subsequently, predictions are generated for each model based on the test data.

The accuracy of the one-step-ahead predictions is quantified using the Mean Squared Forecast Error (MSFE). As depicted in Figure 1, the MSFE for the Random Forest model (0.0414) outperforms that of the Regression Tree model (0.0472). This comparison highlights the better predictive accuracy achieved by the Random Forest model over the Regression Tree model based on the lagged excess returns.

4.2 Enhanced Forecasting Using Additional Covariates in One-Step-Ahead Predictions

The goal of this task is to boost the forecasting capability of each method by integrating additional covariates alongside lagged excess returns. To accomplish this, we once more use cross-validation on the training data to fit models using both regression trees and random forests. This iterative process entails training models on five subsets of the data and evaluating their performance using the Root Mean Squared Forecast Error (MSFE) metric on the test data.

For each iteration, both regression trees and random forest models are trained utilizing the entire array of available predictors, incorporating lagged excess returns and other covariates. These models generate forecasts for a one-step-ahead prediction on the test dataset.

The MSFE, serving as a metric for the models' predictive accuracy, is computed for both regression tree and random forest models considering all covariates and lagged excess returns. The resulting MSFE values are then compared to assess each model's performance in capturing the intricate relationships between the covariates and the predicted excess returns. Figure 1 shows the MSFEs of both models with full covariates. Once again, random forests exhibit a lower MSFE (0.0405) in contrast to regression trees (0.0467).

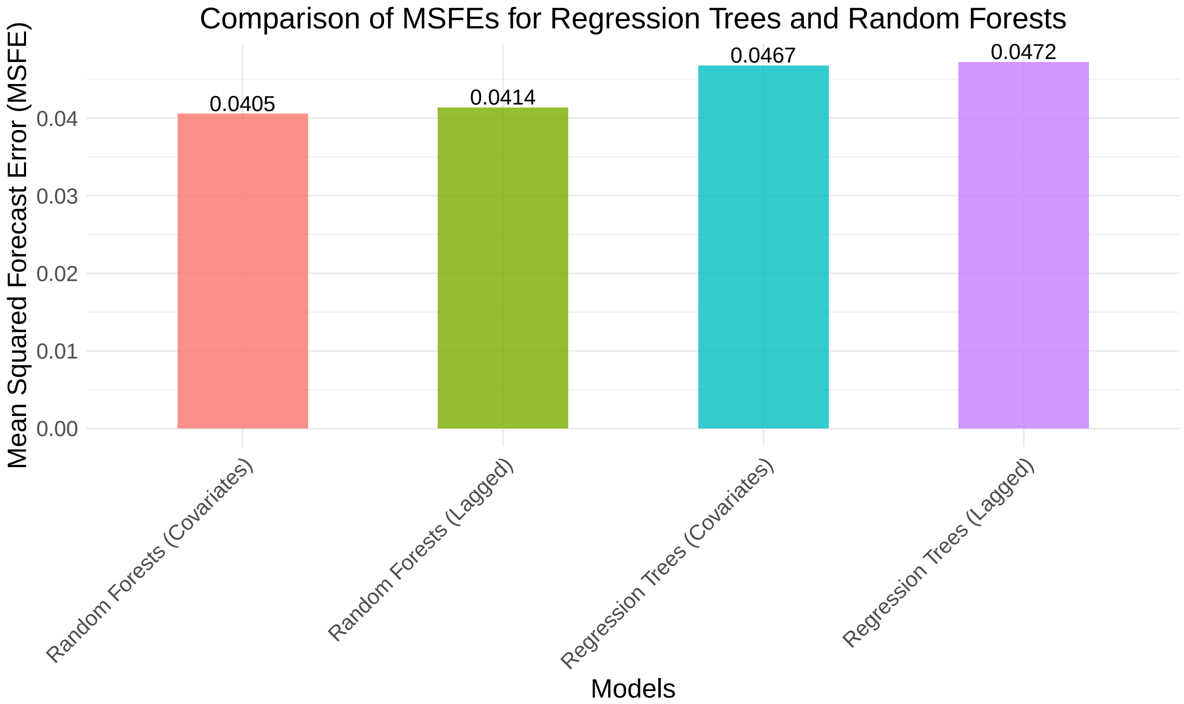

Figure 1: Comparison of Mean Squared Forecast Errors (MSFEs) for Regression Trees and Random Forests using lagged excess returns and models with full covariates

4.3 Computing Importance Measure for Forecast Models Incorporating Covariates and Lagged Excess Returns

This task involves assessing the importance of individual predictors in the context of forecasting models that include both lagged excess returns and covariates. The goal is to compute one preferred importance measure for the forecasts developed in section 4.2.

We sample different predictors individually, generating forecast models for both decision trees and random forests by altering one predictor at a time while holding the rest of the dataset unchanged. Subsequently, the Mean Squared Forecast Error (MSFE) difference between the altered model and the complete covariates and lagged excess returns model from section 4.2 is computed. This difference reflects the change in the MSFE caused by altering a single predictor, indicating its relative importance.

The output of this task includes computed importance measures for individual predictors, displayed through visualizations bar charts. These visualizations allow for a comparative understanding of the influence of each predictor in the context of decision trees and random forests. The graphs illustrate the relative impact of predictors on forecast accuracy, aiding in the identification of influential variables within the models.

Figure 2 shows the analysis of variable importance within the regression tree model and provides insights into the predictors affecting excess returns. The variable excess returns lag 9 (Importance: 0.00299) exhibits the highest importance among the considered factors, suggesting a substantial influence on forecasting excess returns. Lagged Net Equity Expansion (Importance: 0.00233) and excess returns lag 2 (Importance: 0.00185) also demonstrate noteworthy importance, influencing the predictions positively. excess returns lag 3, excess returns lag 8, and lagged Earnings, lagged Corporate Bond Yields on AAA rated Bonds, lagged Corporate Bond Yields on BAA rated Bonds, lagged Long Term Yield, lagged Long Term Rate of Returns, lagged Long term Corporate Bond Returns, and lagged Stock Variance possess similar importance values (0.00180), suggesting a relatively smaller impact on excess returns. The excess returns lag 5 (Importance: 0.00177), excess returns lag 10 (Importance: 0.00176), and lagged Inflation (Importance: 0.00172) also demonstrate moderate importance in predicting excess returns.

Other variables like lagged Cross Sectional beta Premium (Importance: 0.00170), lagged Book to Market Ratio (Importance: 0.00168), and excess returns lag 6 (Importance: 0.00160) contribute positively but with slightly lesser influence. Variables like excess returns lag 1 (Importance: 0.00143), lagged Dividends (Importance: 0.00132), excess returns lag 7 (Importance: 0.00102), and excess returns lag 4 (Importance: 0.00064) display lower importance values compared to others in predicting excess returns. Notably, lagged Treasury Bills (Importance: -0.00181) exhibits a negative importance value.

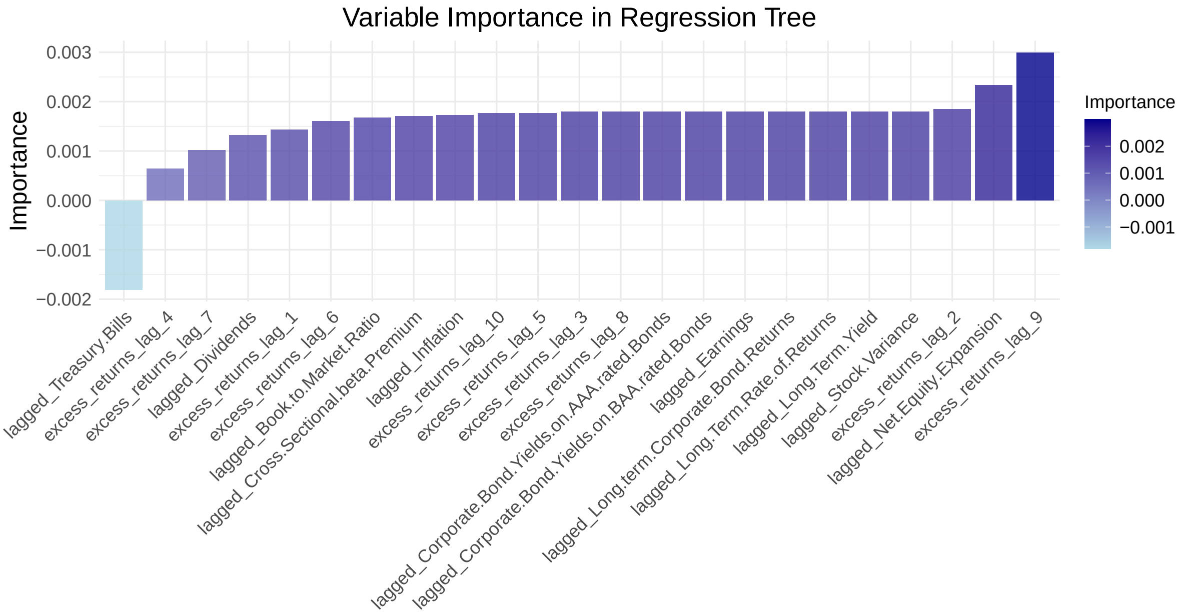

Figure 2: Variable Importance Values for Predictors in the Regression Tree Model: Rankings based on feature importance measures illustrating the relative impact of each predictor on forecasting excess returns

The illustration in Figure 3 depicts a bar graph representing the variable importance observed within the random forest analysis. It shows The variable lagged Cross Sectional beta Premium demonstrates the highest positive importance (Importance: 0.00119), signifying a notable influence on forecasting excess returns. Variables such as lagged Book to Market Ratio (Importance: 0.00063), excess returns lag 3 (Importance: 0.00050), and excess returns lag 5 (Importance: 0.00043) also display moderate positive importance in predicting excess returns. Other variables, including lagged Treasury Bills (Importance: 0.00042), lagged Long Term Rate of Returns (Importance: 0.00039), and lagged Long term Corporate Bond Returns (Importance: 0.00031), contribute with lower positive importance, indicating a comparatively smaller impact on excess returns.

Several variables like lagged Dividends (Importance: 0.00030), lagged Stock Variance (Importance: 0.00022), and lagged Inflation (Importance: 0.00006) have importance values close to zero, suggesting minimal influence on the forecasted excess returns. In contrast, variables such as excess returns lag 1 (Importance: -0.00037), lagged Earnings

(Importance: -0.00028), and excess returns lag 2 (Importance: -0.00007) exhibit negative importance. Variables like excess returns lag 9 (Importance: -0.00001) and lagged Corporate Bond Yields on BAA rated Bonds (Importance: -0.00001) have marginal negative importance, implying a minimal impact on the model predictions.

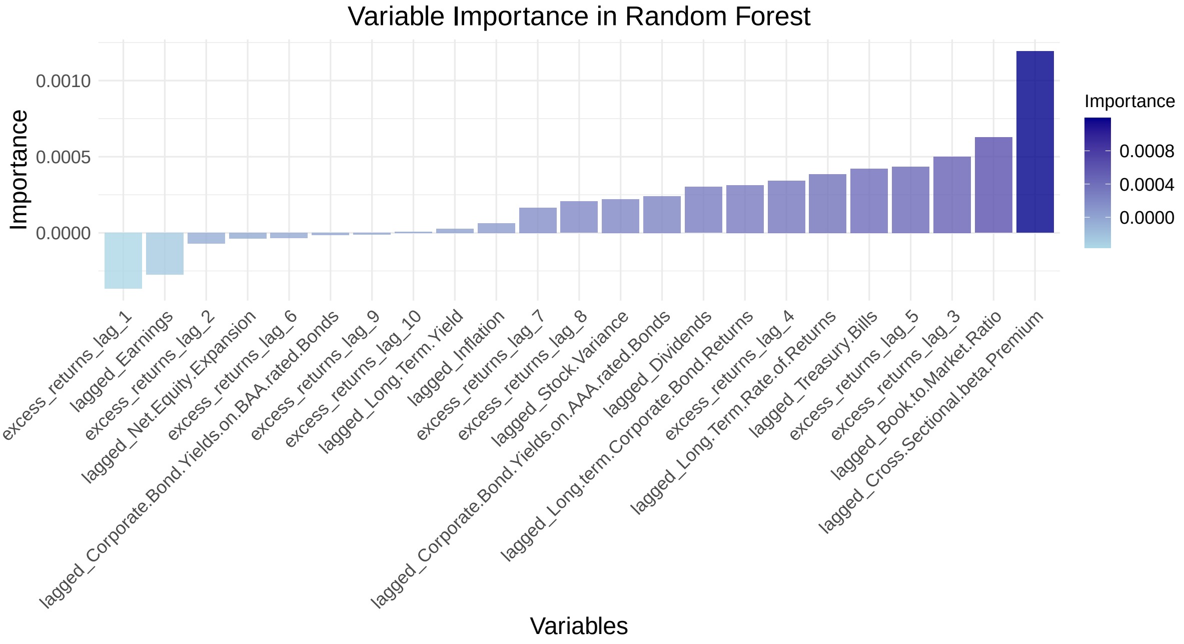

Figure 3: Variable Importance Values for Predictors in the Random Forest Model: Rankings based on feature importance measures, showcasing the relative influence of each predictor on predicting excess returns.

4.4 Assessment of Variable Importance and Comparative Analysis Between Prediction Methods

In the regression tree model, variables like Excess returns lag 9, Lagged Net Equity Expansion, and Excess returns lag 2 hold the highest importance values. These variables demonstrate substantial significance in forecasting excess returns within this method. Notably, Lagged Stock Variance, Lagged long Term Yield, and Excess returns lag 3 and others also contribute significantly to predictive capability, though with slightly lower importance values.

Comparatively, within the random forest analysis, Lagged Cross Sectional beta Premium emerges with the highest positive importance, signifying a notable influence on forecasting excess returns. Additionally, Lagged Book to Market Ratio, Excess returns lag 3, and Excess returns lag 5 also display moderate positive importance in predicting excess returns within the random forest method. The variables Lagged Treasury Bills, Lagged Long Term Rate of Returns, and Lagged Long-term Corporate Bond Returns contribute with lower positive importance within the random forest analysis, indicating a comparatively smaller impact on excess returns.

There are differences in the variables identified as most important between the two prediction methods. While Excess returns lag 9, Lagged Net Equity Expansion, and Excess returns lag 2 exhibit higher importance in the regression tree model, Lagged Cross Sectional beta Premium, Lagged Book to Market Ratio, and Excess returns lag 3 take precedence in the random forest method. In random forest analysis, Lagged Treasury Bills display a positive importance, whereas in regression tree analysis, they notably exhibit negative importance. However, despite these differences, both methods consistently highlight certain variables as significant contributors to predicting excess returns, albeit with some variations in their order of importance and specific values.

4.5 Summary

The primary objective of the project was to employ advanced machine learning techniques, specifically regression trees and random forests, to improve the accuracy of stock return forecasts in financial analysis. Using a dataset covering economic indicators, stock market indices, interest rates, corporate bond yields, and macroeconomic variables from 1871 to 2022, the report aimed to build predictive models capturing complex interrelationships among these variables.

The analysis involved several key tasks. One-step-ahead forecasts were generated using lagged predictors, resulting in Mean Squared Forecast Errors (MSFEs). These MSFEs indicated that the Random Forest model outperformed the Regression Tree model, achieving an MSFE of 0.0414 compared to 0.0472, respectively. Additionally, models incorporating both lagged excess returns and other covariates exhibited lower MSFEs, with the Random Forest model showing a better performance (MSFE of 0.0405) than the Regression Tree model (MSFE of 0.0467).

Furthermore, permutation feature importance was computed to determine the significance of predictors between the Regression Tree and Random Forest models. In the Regression Tree model, predictors like Excess returns lag 9, Lagged Net Equity Expansion, and Excess returns lag 2 held the highest importance values. Conversely, in the

Random Forest analysis, Lagged Cross-Sectional Beta Premium, Lagged Book to Market Ratio, and Excess returns lag 3 emerged as the most influential predictors.

In summary, the report successfully applied advanced machine learning techniques to forecast stock returns, revealing that Random Forest models generally outperformed Regression Tree models in terms of accuracy. The analysis provided insights into the most influential predictors in financial forecasting, contributing to a comprehensive understanding of predictive modeling in the finance domain.

Additionally, it is recommended to Include rolling window techniques which can enhance the temporal adaptability and robustness of the predictive models, making them more attuned to evolving market dynamics and potentially improving their forecasting capabilities. Experimentation with ensemble methods beyond Random Forests, such as Gradient Boosting Machines (GBM), XGBoost, or stacking models, could be beneficial. These methods often combine multiple models to achieve higher predictive power.

Bibliography

Hay-Jahans Christopher. *Introduction to the theory of statistics.* Taylor Francis Group, 2019. 1st Edition.

Amit Goyal. A Comprehensive Look at the Empirical Performance of Equity Premium Prediction, 2022. URL `https://sites.google.com/view/agoyal145`.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, 2009.

Raky Julio, Andres Monzon, and Yusak O. Susilo. Identifying key elements for user satisfaction of bike-sharing systems: a combination of direct and indirect evaluations. *Journal Name*, 2022. doi: 10.1007/s11116-022-10335-3. URL `https://link.springer.com/article/10.1007/s11116-022-10335-3`.

Max Kuhn and Others. *caret: Classification and Regression Training*, 2022. URL `https://CRAN.R-project.org/package=caret`. R package version 6.0-91.

Andy Liaw and Matthew Wiener. *randomForest: Breiman and Cutler's Random Forests for Classification and Regression*, 2022. URL `https://CRAN.R-project.org/package=randomForest`. R package version 4.6-16.

Stephen Milborrow. *rpart.plot: Plot 'rpart' Models: An Enhanced Version of 'plot.rpart'*, 2022. URL `https://CRAN.R-project.org/package=rpart.plot`. R package version 3.1.5.

Christoph Molnar. Interpretable machine learning, 2023. URL `https://christophm.github.io/interpretable-ml-book/`. Accessed: 2023-12-13.

R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2022. URL `https://www.R-project.org/`.

Terry Therneau and Beth Atkinson. *rpart: Recursive Partitioning and Regression Trees*, 2022. URL `https://CRAN.R-project.org/package=rpart`. R package version 4.1-16.

Ivo Welch and Amit Goyal. A Comprehensive Look at The Empirical Performance of Equity Premium Prediction. *The Review of Financial Studies*, 21 (4):1455–1508, 03 2007. ISSN 0893-9454. doi: 10.1093/rfs/hhm014. URL `https://doi.org/10.1093/rfs/hhm014`.

Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*, 2016. URL `https://ggplot2.tidyverse.org`.

Hadley Wickham, Romain François, Lionel Henry, and Kirill Müller. *dplyr: A Grammar of Data Manipulation*, 2022. https://dplyr.tidyverse.org, https://github.com/tidyverse/dplyr.

Hadley Wickham, Jim Hester, and Jennifer Bryan. *readr: Read Rectangular Text Data*, 2023. URL `https://readr.tidyverse.org`. R package version 2.1.4, https://github.com/tidyverse/readr.