Learning from Extremes: Tree-Based Portfolio Construction - A Tail-Focused Machine Learning Approach

Peiran Wang · Kunmin(Rose) Yu

Department of Statistics & Actuarial Science, University of Waterloo

ABSTRACT In this paper, we investigate the effectiveness of tail filtering in training decision tree models for stock return prediction and portfolio construction. Building on the framework proposed by Coqueret and Guida (2019), we assess whether filtering the training data to include only extreme return observations improves out-of-sample prediction accuracy and Sharpe ratios when applied to a different dataset. We further investigate the optimal choice of tail quantile q in a given dataset and examine whether the performance gains from tail filtering are consistent across firm size categories. Our results demonstrate that tail filtering significantly enhances portfolio returns and Sharpe ratios when a sufficiently long historical training window is used. Additionally, we develop an R code framework to identify the optimal quantile for any dataset and find that a filtering threshold of q=0.15 achieves the best performance in our dataset. We also observe that small-cap stocks continue to outperform large-cap stocks under the tail filtering method. These findings suggest that tail filtering is an efficient technique for decision tree modeling in asset pricing and factor investing applications.

Keywords Tail filtering · Decision trees · Portfolio construction · Sharpe ratio · Optimal quantile · Size anomaly · Long-Short

1 RESEARCH QUESTIONS

This paper investigates the following three research questions:

First, does training decision trees using tail values continue to improve prediction accuracy and Sharpe ratios when applied to a different dataset? Although the original authors reported performance improvements, we aim to verify the robustness of their claim using data_ml.RData, a dataset provided in this course that differs slightly from the one originally used.

Second, what is the optimal quantile q for tail filtering in a given dataset? The original study selected q=0.2 somewhat arbitrarily; however, this threshold may not be universally optimal. We aim to identify the most appropriate quantile for our dataset and to develop code that can help practitioners efficiently determine the optimal q when applying the tail value method to other datasets.

Third, does tail filtering enhance prediction accuracy and portfolio returns uniformly across different stock characteristics, such as firm size (small-cap versus large-cap stocks)? Based on assessments conducted during this course, we observe that small-cap stocks have historically tended to outperform large-cap stocks over long periods. Using the data_ml.RData dataset, which contains 244 monthly observations from November 1998 to March 2019, we aim to verify whether the size anomaly (the tendency for small firms to outperform large firms over very long periods) still holds when applying the tail-value method.

2 VARIABLES AND MEASURES

2.1 Dataset Description

This paper uses the data_ml.RData dataset, which contains information on 1,207 U.S.-listed stocks (i.e., N=1,207) over a period of 244 months, spanning from November 1998 to March 2019. For each stock at each point in time, there are 93 predictor variables capturing various firm characteristics, including accounting ratios, market activity measures, and profitability indicators. The dataset also includes four response variables: R1M_Usd, R3M_Usd, R6M_Usd, and R12M_Usd, which correspond to the 1-month, 3-month, 6-month, and 12-month forward realized returns of the stocks, respectively.

2.2 Predictor Variables (features)

All 93 available firm characteristics in the dataset are used as predictors to train the decision tree models. This choice is motivated by the recommendation in *"Training Trees on Tails with Applications to Portfolio Choice"* (Coqueret & Guida, 2019), which states that if there are few variables, competition among predictors will be limited and the tail-value filter will have minimal impact, particularly when one variable strongly dominates others in terms of initial gains. Furthermore, the original study utilized a dataset comprising 108 characteristics for 1,182 U.S.-listed firms, which is comparable to the 93 characteristics for 1,207 firms in our dataset. Adopting

a similar comprehensive approach ensures methodological consistency and enables a fair and robust comparison with the original results.

2.3 Response Variables

Two response variables, R12M_Usd and R1M_Usd, are used during the training and testing stages, respectively in this paper. During the training stage, R12M_Usd is used as the response variable, with the objective of identifying stocks that fall into the highest and lowest predicted 12-month return categories, which are subsequently used to construct a portfolio. In the testing stage, R1M_Usd serves as the response variable to assess the model's short-term predictive performance. The selection of R12M_Usd for training is motivated by the desire to utilize the longest available historical return horizon, providing a more comprehensive representation of stock performance over time. Nevertheless, recognizing that investors often prioritize short-term returns, R1M_Usd is chosen for evaluation during the testing stage. This approach to response variable selection is consistent with the methodology outlined in *"Training Trees on Tails with Applications to Portfolio Choice"* (Coqueret & Guida, 2019), thereby enabling a fair comparison with the original results.

2.4 How the Response Variables Are Measured

First, we apply a quantile filter to the dataset based on the chosen quantile threshold, retaining only observations within the extreme tails. A decision tree is then trained on the filtered data, where the training label is R12M_Usd (the 12-month historical return) and the split criteria are based on the predictor variables. After training, the decision tree is used to predict the R12M_Usd values for each stock. The stocks are divided into 6 subgroups based on their predicted R12M_Usd ranking. We take long positions in stocks predicted to fall into the highest return categories of R12M_Usd, and short positions in those predicted to fall into the lowest return categories. A long-short portfolio is then constructed based on the selected stocks. Finally, the portfolio's performance is evaluated by calculating the realized R1M_Usd (1-month forward return) using an average of the returns of the selected stocks.

3 THE APPLICATION OF THE ML APPROACH TO FI

In this project, we explore how decision tree-based machine learning models can be applied to portfolio construction and cross-sectional stock return prediction. Instead of directly forecasting the next period's return for each stock, we train a tree-based model to learn the characteristics of historical winners and losers. This is exactly what factor investing aims to achieve.

The key idea is that stocks with similar financial traits (such as size, volatility, and momentum) often exhibit similar return patterns. The decision tree algorithm captures this relationship by recursively partitioning the feature space into subgroups of stocks with homogeneous historical returns. Each path through the tree is a sequence of if-else rules, and each terminal node (or leaf) represents a group of stocks with a common return profile. Once the model is trained, each stock is passed down the tree according to its features and is assigned the average return of the leaf node it falls into. This average becomes the stock's predicted return, not as a point forecast for the next month, but as a score used for ranking. Then, we construct portfolios consisting of long positions in the top-performing stocks and short positions in the losing stocks.

Our primary objective is to evaluate whether training the tree using only the tails of the historical return distribution — that is, the most extreme winners and losers — leads to better out-of-sample portfolio construction and performance.

To do so, we use an XGboost decision tree model, which is trained to predict historical 12-month return (R12M_Usd) based on 93 firm-level features such as size, value, and momentum indicators. However, the key innovation is that we first filter the dataset to only retain the top and bottom q quantile of forward return stocks. This is inspired by the idea presented in *"Training Trees on Tails with Applications to Portfolio Choice"* (Coqueret & Guida, 2019), in which the authors argue that mid-range values are less informative and introduce more noise, so removing them before training can lead to better model prediction.

This approach is relatively easy to reproduce, as it only requires filtering the training dataset and does not alter the tree-training logic.

4 EXPERIMENTAL METHODOLOGIES

4.1 Tools and Environment:

We conducted a series of systematic experiments using open-source tools:

- Programming language: R
- Machine learning library: xgboost, a gradient-boosted decision tree package
- Data Manipulation libraries: dplyr, tidyverse
- Data visualization library: ggplot2
- Timing Execution: R's proc.time() function was used to measure total runtime

4.2 Experimental Setup:

- We use data_ml.RData, which contains 1,207 U.S.-listed stocks over a period of 244 months, starting from 1998 to 2019.
- The training window is roughly 8~9 years, from November 1998 to January 2007. Out-of-sample test date begins from January 2008. Note that there is a one-year gap between the end of the training data and the start of the test data to avoid forward-looking bias, ensuring that no information from the test period is inadvertently used in model training.

4.3 Research Question 1 – Does tail filtering improve prediction accuracy and Sharpe ratios on a different dataset?

Filtered Model (Tail-Filtered Tree Approach)

- Apply a tail filter at the 15th and 85th percentiles of R12M_Usd to retain only extreme-performing stocks, with the threshold q=0.15 selected based on the results of Research Question 2, where the filtered long-short portfolio achieved the highest average 1-month return at q=0.15.
- Train an XGBoost decision tree model on the filtered dataset.
- Divide the stocks into six portfolios based on their predicted 12-month forward returns (R12M_Usd), with cut-off quantiles at 0, 1/6, 2/6, 3/6, 4/6, 5/6, and 1.
- Construct a long-short portfolio by taking long positions in the highest portfolio and short positions in the lowest portfolio based on predicted 12-month forward returns.
- Calculate the average 1-month forward returns (R1M_Usd), and the monthly Sharpe ratio of the long-short portfolio.

Unfiltered Model (Baseline Tree Approach and Market Benchmark)

- Trained a separate XGBoost decision tree on the full unfiltered dataset (no tail filtering), and predicted returns, formed portfolios, and calculated average returns and Sharpe ratios in a similar manner.
- Used an equal-weighted method to compute the market return across all stocks in the original dataset (without filtering), which serves as the benchmark for comparison.

Performance Comparison

- Compare the average monthly long-short return and Sharpe ratio across three methods:
  - Tail-filtered decision tree model.
  - Unfiltered decision tree model.
  - Equal-weighted market return.

Expected Outcome

- Tail filtering is expected to enhance model performance by:
  - Increasing average long-short returns.
  - Achieving a higher Sharpe ratio.
  - Producing a statistically significant model.

4.4 Research Question 2 – What is the optimal quantile q for a given dataset?

- Five candidate quantile thresholds (q) were tested: 0.1, 0.15, 0.2, 0.25, and 0.3, corresponding to retaining 20%, 30%, 40%, 50%, and 60% of the dataset, respectively. A maximum of 60% was considered sufficient for training under the tail filtering method.
- For each q, we applied a tail filter, trained an XGBoost decision tree model, and predicted 12-month forward returns (R12M_Usd).
- Based on the predicted returns, a long-short portfolio was constructed by taking long positions in the highest-ranked stocks and short positions in the lowest-ranked stocks.
- The average 1-month forward return (R1M_Usd) was calculated for each long-short portfolio at every quantile threshold.
- The optimal quantile was determined as the one that achieved the highest average 1-month forward return.

4.5 Research Question 3 – Does tail filtering improve prediction accuracy or returns uniformly across different types of stocks, such as by firm size?

- Stocks were split into Small Cap and Large Cap groups based on a median split of market capitalization, with small-cap stocks defined as those with capitalization less than or equal to the median, and large-cap stocks as those with capitalization greater than the median.
- Separate XGBoost decision tree models were trained for each size group, and long-short portfolios were constructed within each group based on predicted returns.
- For each group, the average 1-month forward return (R1M_Usd) and the monthly Sharpe ratio were calculated.
- Results were compared between Small Cap and Large Cap groups, with the expectation that the size anomaly would be observed—that is, Small Cap stocks would demonstrate higher average returns and higher Sharpe ratios than Large Cap stocks.

4.6 Reproducibility

The code base used for data loading, processing, training, and evaluation is provided in the Appendix, with detailed step-by-step comments.

5 RESULTS AND DISCUSSION

5.1 Results and Discussion – Research Questions 1 & 2:

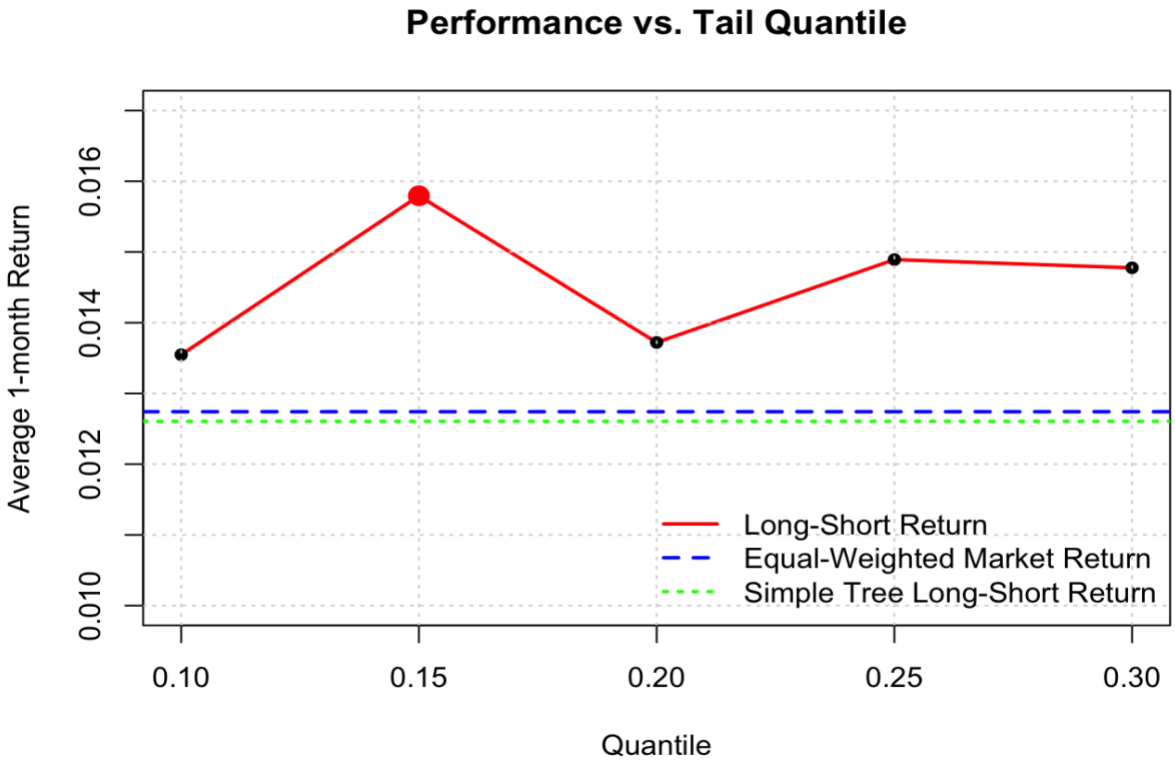

Figure 1: Average 1-month return across different quantiles, with rolling window = 9 years.

Figure 1 illustrates the long-short portfolios' average 1-month return across 5 different decision trees trained on different quantile q's, ranging from (0.1, 0.15, 0.2, 0.25, 0.3). The rolling window is 9 years. We see that the long-short portfolios constructed by these trees consistently outperform the equal-weighted benchmark and the simple tree benchmark (simple tree refers to a tree trained on the entire unfiltered dataset). Also, the best average 1-month return is achieved when q = 0.15, which means only the top and bottom 15% of R12M_Usd stock returns are kept. This is only 30% of the original dataset, thereby significantly reducing the training time.

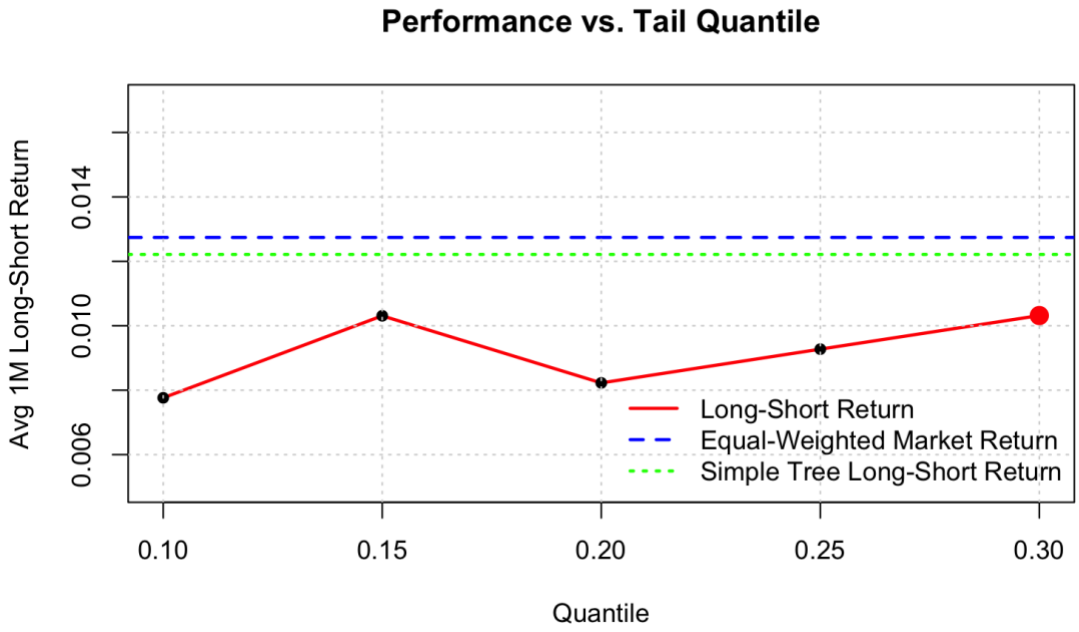

Figure 2: Average 1-month return across different quantiles, with rolling window = 5 years.

Figure 2 shows the performance of the decision tree models that are trained with a 5-year rolling window. This time, however, the models consistently underperform when compared against other benchmarks, for all tested quantiles. One possible explanation is that the filtered subset may be too narrow and too sensitive to temporary noise. With fewer training data, the trees risk learning short-term shocks instead of long-term relationships. This can be further influenced by the 2008 financial crisis.

When training on only 5 years of data, the long-short strategy underperforms compared to training on 9 years of data, especially at lower tail quantiles. This suggests that longer historical context helps tree models learn stronger cross-sectional return signals, particularly when using tail filtering. Therefore, Tail filtering requires sufficient historical data to reliably distinguish winners and losers. 5-year windows may be too narrow, especially in volatile markets.

Interestingly, we observe that Figure 2 exhibits a similar trend as Figure 1, and q = 0.15 is still an optimal split location.

5.2 Results and Discussion – Research Question 3:

group	Avg_LongShort	SD	Sharpe
<chr>	<dbl>	<dbl>	<dbl>
1 LargeCap	0.00651	0.0347	0.188
2 SmallCap	0.0207	0.0673	0.308

Figure 3: Average long-short return, Standard Deviation, and Sharpe Ratio for each size group

Figure 3 shows the result from a tree model trained using q = 0.15. It compares the average long-short return, Standard Deviation, and Sharpe Ratio for small and large firms, respectively. Here, the stocks are divided into small-cap and large-cap, where small-cap contains firms whose predicted R12M_Usd is below the median predicted R12M_Usd, and large-cap contains the remaining firms. Then, each group is further divided into 6 portfolios based on their return ranking within the group, and the same long-short strategy is applied to each group. The table clearly shows that small firms dominate large firms on average 1-month return and Sharpe ratios.

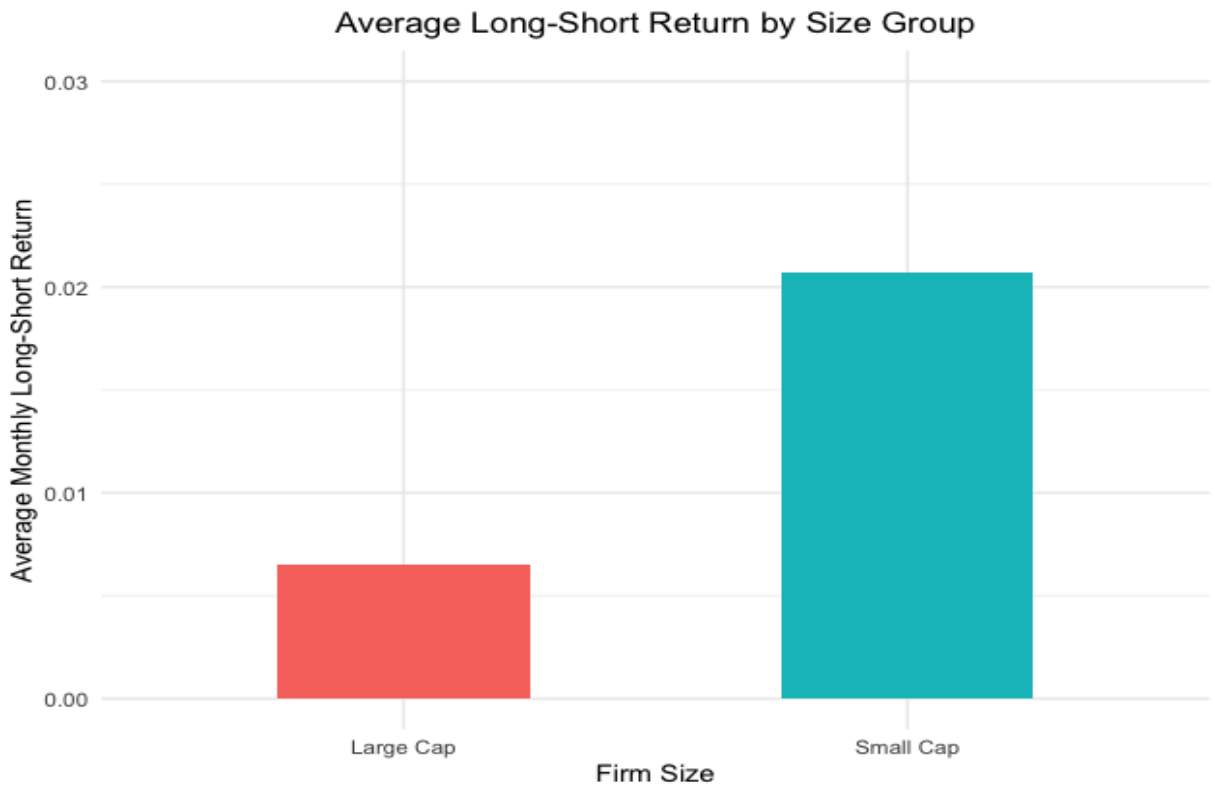

Figure 4: Average 1-month long-short return by firm size

Figures 3 and 4 aim to answer the third research question: whether the well-known size anomaly still holds after filtering. The short answer is yes, the long-short strategy applied to small firms gives an average 1-month return of 2.07%, which is significantly higher than the average long-short return of large firms, 0.65%. A rationale for this is that small firms are more volatile, as the observed standard deviation of 0.0673, so winners win more and losers lose more, creating a large spread that benefits the long-short strategy. Smaller firms also receive less public attention, and therefore, they are more likely to be mispriced. This gives machine learning methods a greater edge if certain models can more efficiently exploit the pricing inefficiencies in small firms. On the other hand, large firms tend to be more stable, priced more efficiently, and much of their return structure may already be explained by traditional factors such as value and momentum. Our XGBoost tree models may be identifying combinations of characteristics that are especially predictive within small firms, but less relevant for large firms.

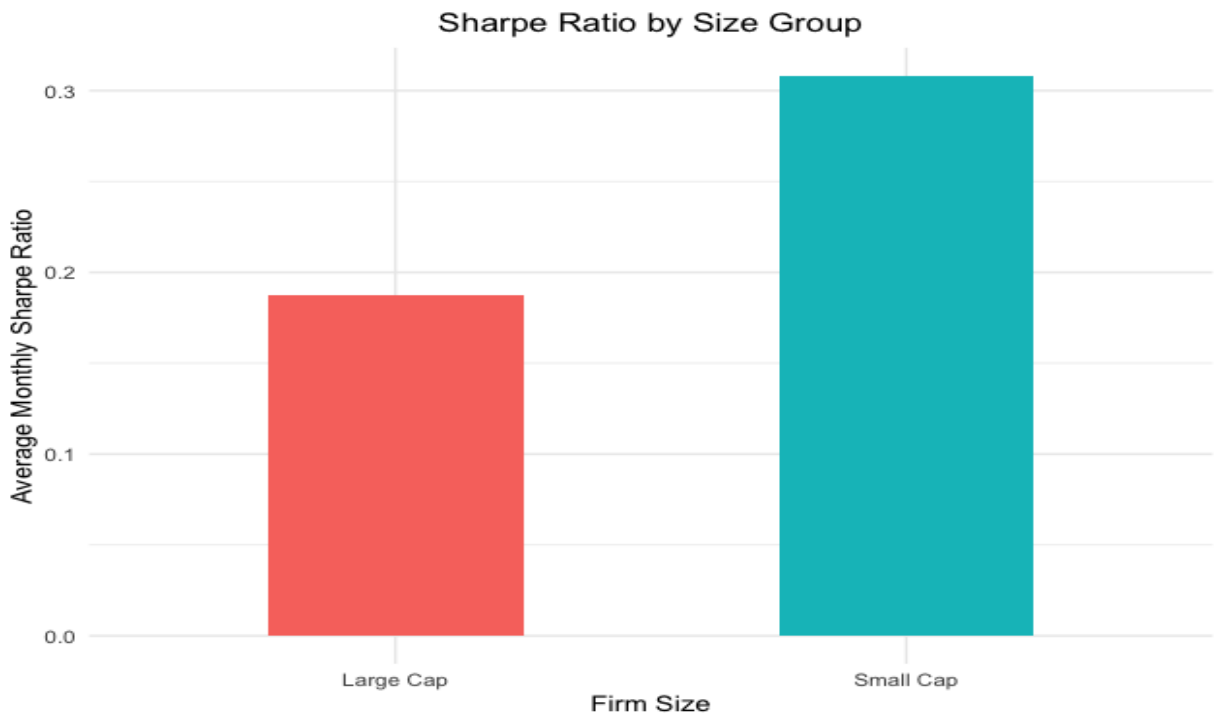

Figure 5: Average monthly Sharpe Ratio by size group

Lastly, we examine average returns on a risk-adjusted basis, which is measured by the Sharpe Ratio. Small-cap long-short strategy delivers a higher Sharpe ratio (~0.31), and the large-cap long-short strategy has a lower Sharpe ratio (~0.18). Even though small-cap portfolios are more volatile, they deliver even better returns relative to that risk. While small firms are riskier, they also exhibit stronger alpha opportunity. These results align well with the size anomaly.

6 RELATED WORK

Projects 1 and 2 are highly related, therefore, the reader is referred to our GPR # 1 Report for related work.

7 CONCLUSIONS

This report validated the new tree-based machine learning approach inspired by Coqueret & Guida (2019), while diving deeper to research three questions aimed to address the limitation of their strategy. We found that q = 0.15 is an optimal quantile for filtering the data_ml.RData dataset, and our model can effectively construct long-short portfolios that consistently outperform benchmarks, if a sufficiently long training window is used. Most importantly, the tree-based model trained on post-filtered data is great at identifying factor combinations that drive excess return specifically for small firms, which aligns well with the long-standing size anomaly.

However, due to time constraints, we could only test 5 quantile values, and we do not have a theoretical derivation to explain the optimal quantile choice. We are only able to find an optimal quantile through simulation, but we cannot provide rigorous mathematical proofs and derivations. Further, even if our dataset is somewhat similar to that of Coqueret & Guida (2019), we obtained seemingly contradictory results at first, when we attempted to fully mimic their process and parameters (i.e., 5-year training window led to great results for them, but not for us. We had to use a 9-year training window in order for our model to beat the benchmarks). The difference in the number of predictors in the two datasets might have contributed to the discrepancy.

Furthermore, we have used all 93 predictors for training, which led to significant computation burdens. Future research can explore training using a smaller predetermined set of predictors and test more quantile values. This could more systematically validate the approach of training using extreme tail values, since researchers can control the predictor set and keep track of their performance across different datasets.

REFERENCE

Coqueret, G., & Guida, T. (2019, June 20). *Training trees on tails with applications to portfolio choice*. SSRN. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3403009

APPENDIX

Wang-Peiran. (2025, April 20). *AFM-423-/Peiran_Rose_GP2_FINAL1.html at main · Wang-Peiran/AFM-423-*. GitHub. https://github.com/Wang-Peiran/AFM-423-/blob/main/Peiran_Rose_GP2_FINAL1.html