UCLouvain

LOUVAIN
School of Management

Louvain School of Management

Prediction of future stocks returns using an ensemble of machine learning models

Author: Victor ANCIAUX
Supervisor: Frédéric VRINS
Academic year 2024-2025
Dissertation for the Master of Business Engineering with specialization in Supply Chain Management and Financial Engineering
Daytime schedule

AI Declaration

During the preparation of this master's thesis, the author utilized `Microsoft Copilot` and `DeepL` for the following purpose:

1. **Code Assitance and reformulations:** `Copilot` assisted in writing parts of the code included in this thesis, helping to better understand various Python and cuML functions and modules, export data for visualization in Matplotlib, and, since English is not the author's native language, using a mix `Copilot` and `DeepL` for rephrase or paraphrase contents to improve clarity and readability.

2. After using those I.A tools I diligently reviewed and edited the content produced by the tool. I take full responsibility for the final content presented in this thesis.

By signing this declaration, I affirm that the content of this master's thesis reflects my original work, augmented by the responsible use of AI.

Louvain-La-Neuve, 15th of July 2025,

Abstract

Over the past few decades, it has proven extremely difficult to consistently outperform financial markets. For instance, about 90% of Large Cap funds fail to beat the major benchmark indices over extended horizons. At the same time, the availability of financial data has exploded and machine learning (ML) methods have advanced significantly. This convergence offers a promising opportunity to revisit traditional approaches to return forecasting and portfolio construction. Drawing on empirical data from large-scale forecasting competitions and recent research on ensemble learning, this thesis examines whether combining multiple machine learning models can improve short-term stock return forecasting and enhance portfolio performance.

Specifically, this thesis develops a set of ML classifiers to predict whether individual stocks will outperform their benchmark index over the following week. The models are trained using a walk-forward approach over a long historical period (January 2007-December 2024), allowing for continuous re-estimation to adapt to changing market conditions. Particular emphasis is placed on threshold selection strategies, which transform the continuous results from the combination into binary signals. These thresholds are optimized according to various criteria and are found to have a significant influence on trading performance.

The predicted signals are integrated into a portfolio construction framework using various weights allocations techniques. Of the 544 portfolios constructed, all those that outperformed the benchmark did so using ensemble machine learning strategies. The best performing strategy combined a mean-variance portfolio allocation with a filtering-based covariance estimator and Ledoit-Wolf shrinkage, as well with a trimmed ensemble of ML models whose threshold was optimized to maximize precision. This configuration results in a Sharpe ratio of **1.11** and an APY of **19.29**%. By comparison, its universe index achieved a Sharpe ratio of **0.74** and an APY of **14.72**% over the same period. Moreover, 11 of the 15 best-performing portfolios used dynamic threshold techniques explicitly designed to maximize classification precision.

Overall, this research demonstrates empirically the practical potential of ensemble learning and adaptive thresholding in financial forecasting and portfolio management.

Acknowledgements

This thesis marks the end of my university studies and, with it, the closing of a rich and intense chapter in my life. Beyond academic rigor, it also represents a unique opportunity to combine two of my greatest interests: financial markets and quantitative programming. Although this journey has been filled with many challenges (some expected, others less so), I quickly realized that I was not alone in facing them.

First of all, I would like to thank my thesis advisor, Professor Frédéric Vrins, for his wise advice, judicious recommendations, and availability throughout the year. His comments, always relevant and constructive, greatly helped me to stay focused and refine my approach.

I would also like to sincerely thank my family and friends. Not only for their unconditional support, but also for their heroic patience every time I tried to explain "why this result didn't make sense," "how the model was supposed to behave," or "why changing a threshold changed everything." Their encouragement kept me grounded, even when my code wasn't.

To all of you: **thank you** for being part of this adventure.

Contents

List of Figures

List of Tables

Introduction

"89.5% of all Large-Caps funds underperformed the S&P 500 over a 15-years period."

(S&P Global, 2024)

Nowadays, outperforming the market, i.e. benchmark indices such as the S&P 500 in the US, is extremely difficult and rare over the long term. Of all Large-Caps funds, 75.25% underperformed the S&P 500 over 5-years periods and this percentage rises to 84.34% and even 89.5% over 10-years and 15-years periods respectively (S&P Global, 2024). Although these results are direct consequences of the market's complexity and volatility, it underlines the difficulties inherent in financial forecasting and highlights the need for models capable of managing the vast and dynamic data of financial markets. Forecasting financial markets is still a challenging task in data science and finance.

This critical area has major implications for both academics and industry operators. Forecasting stock returns and creating optimal portfolios are crucial for investors, fund managers and other stakeholders seeking to efficiently allocate capital and manage risk in a volatile and uncertain trading environment. For researchers, the complexity of financial markets challenges existing predictive models and offers a clear motivation to explore new advanced forecasting techniques.

Traditional statistical methods often fail to capture the complexity and volatility of financial markets. With the rapid growth in data availability (Graf et al., 2020), Machine learning techniques have become powerful tools to improve the accuracy and efficiency of financial forecasts. Even among machine learning methods, performance varies considerably. Each technique tends to capture certain patterns well while overlooking others, which can limit its predictive power and its ability to fully capture complex market dynamics (Htun et al., 2024). While single models can provide valuable information, their forecasts often exhibit limitations in terms of reliability and robustness when they are used in isolation.

The integration of several forecasting models in an ensemble learning approach appears to be a promising way of overcoming these limitations (Atiya, 2020). By combining different models, it enables us to capture a diversity of signals present in financial data, thereby improving the stability and accuracy of forecasts.

In this context, the objective of this research thesis is to assess the potential of using **an ensemble of machine learning models for predicting future stock returns in order to create an optimal portfolio**, i.e. one that outperforms a benchmark index.

The theoretical foundations of this thesis are based on several key areas both in finance and machine learning.

In finance, the efficient market hypothesis (EMH) suggested by Eugene Fama in 1970 serves as a key basis. This hypothesis suggests that all available information is already accounted for in stock prices (Fama, 1970). In other words, there is never any error in the valuation of stock prices. Assets are never undervalued or overvalued. In such a situation, the price of an asset is always equivalent to its theoretical value. The main implication of this hypothesis is that, if true, it is impossible to make profits on the basis of under- or over-valued assets. However, EMH often fails to account for short-term irregularities and the influence of psychological and behavioral factors. Indeed, the EMH assumes that all individuals act in a rational manner, but actual behavior is influenced by various cognitive biases, emotions and not unpredictable events directly themselves but the unpredictable reactions to those (such as the 2008 crisis or the COVID-19 pandemic). The Adaptive Market Hypothesis (AMH) addresses this limitation by suggesting that market efficiency is not static (Lo, 2004). It proposes that markets behave in a way that may appear efficient during certain periods, but at other times, deviate temporarily in unexpected or irregular ways. These deviations reflect changes in market dynamics, often influenced by changes in investor behavior, market conditions or external factors.

The portfolio optimization of this thesis is also based on a seminal model in the financial theory ; The Modern Portfolio Theory (Markowitz, 1952). Markowitz introduced the concept of the efficient frontier, which identifies the set of portfolios that generate the maximum expected return for a given level of risk. This theory was pioneering in portfolio management, as it introduced the idea of optimal diversification. However, the optimization framework has also been widely criticized due to its unstable solutions (R. Michaud, 1989). Many researchers have studied the issue and proposed potential solutions, such as the shrinkage method (Ledoit & Wolf, 2004), on which this thesis will be based.

To address the research question, this thesis will follow an empirical methodology. Historical stock prices will be collected from sources such as Bloomberg. Several machine

learning models will then be applied to analyze these data and cross-validation techniques will be used to optimize model parameters and ensure robust forecasting results. Finally, forecasts will be optimally combined and four different portfolio construction techniques will be tested and compared to create a dedicated optimal portfolio.

This thesis is organized as follows:

- Chapter 1 and 2 provide a review of the literature on machine learning techniques applied to stock market return forecasting. Particular attention is given to methods of combining forecasts, highlighting both the theoretical foundations, motivations and empirical evidence supporting ensemble approaches.
- Chapter 3 highlights different portfolio optimization techniques. It begins with the seminal work of Markowitz on Modern Portfolio Theory (Markowitz, 1952). To address the constraints explained previously, some alternative optimization techniques are introduced, each designed to improve and enhance the capabilities of the Markowitz framework to generate robust and optimal portfolios.
- Chapter 4 details the methodological framework adopted in this thesis. It describes the process of training individual predictive models, combining their results and systematically varying the classification thresholds applied to the combined forecasts.
- Chapter 5 reports empirical results obtained from a dataset on US equities (NASDAQ-100). The predictive metrics of individual models and combinations of sets are evaluated and compared statistically to a random benchmark. The portfolios constructed are then compared to a passive investment strategy that replicates the index of the equity universe under consideration. The analysis includes a comparative assessment of returns, risk-adjusted performance and the sensitivity of the results to the threshold parameter governing binary classification decision.
- Finally, the conclusion synthesizes the main findings, discusses the practical and theoretical limitations of the approach and proposes directions for further research in the area of stock return prediction and the design of ensemble-based portfolio strategies.

1 Machine Learning Models

Machine learning refers to models that automatically learn patterns in data, including complex relationships often missed by traditional statistical methods. These patterns are used to make predictions or decisions on future outcomes based on new input data.

Machine learning is based on three fundamental concepts: **data**, a **model** and **learning**. As an inherently data-driven discipline, its aim is to extract meaningful patterns from data with a minimum of domain-specific input. A model represents a structured way of describing the relationship within the data, for example, mapping inputs to outputs in a regression. Learning is the process by which the model improves its performance on a task as it is exposed to more data, with the ultimate aim of generalizing well to new, unseen examples (Deisenroth et al., 2020).

Several recent studies have shown that machine learning techniques may outperform classical models in stock return forecasting tasks (Gu et al., 2020). Traditional linear models may have limitations to capture the complex and nonlinear dynamics of financial markets. In contrast, machine learning models can adapt to high-dimensional data and discover hidden, complex and implicit interactions between variables/features, making them well-suited for forecasting tasks under uncertainty.

Theoretical Framework of Machine Learning Models Used

In this thesis, machine learning is used as a predictive tool to classify stocks based on expected future returns. Although a full discussion of machine learning models is beyond the scope of this main document, a brief overview is provided here to maintain consistency in the structure of this thesis.

To give structure to our analysis, we follow a simple and well-known way of organizing machine learning methods: by separating them based on how they learn from data and the type of problem they are meant to solve.

The first distinction is between *supervised* and *unsupervised* learning. Supervised learning is used when we have both input features (such as technical indicators, fundamentals, or historical prices) and a known target outcome, here, whether a stock will outperform a benchmark. This is the core of our predictive task. Unsupervised learning, on the other hand, is applied without any predefined target variable. In our case, we use it primarily through dimensionality reduction (PCA), to simplify and clean the feature space before training models.

Another key difference lies in the prediction task itself. While regression aims to predict a continuous outcome (such as expected return), we formulate our problem as a *binary classification*: each stock is labeled 1 if it is expected to outperform over a certain horizon, and 0 otherwise. This binary framework is closer to the decision-making process of portfolio managers, who are more interested in rankings than in exact estimates of return.

Based on this framework, we apply a wide range of supervised machine learning models to generate predictions. These include linear models, tree-based algorithms, support vector machines, k-nearest neighbors, and neural networks.

While the goal here is to keep this brief overview concise, all the general definitions, theoretical foundations, and mathematical formulations of each model used are fully detailed in Appendix A. This includes not only how each algorithm works in practice, but also the underlying assumptions, the learning mechanisms, and their relevance in a financial forecasting context. Moreover, if the reader is not familiar with classification metrics used to analyze the performance of machine learning models, they may refer to Appendix C for detailed information.

This theoretical base sets the foundation for the ensemble learning strategies developed in the next chapter, where individual model forecasts are aggregated into a single decision signal.

2 Ensemble of Machine Learning Models

In this chapter, the main motivations for using ensembles of machine learning models are presented. Their typology is reviewed and the different categories considered throughout the thesis are described, ranging from simple averaging techniques to trimming, screening and more advanced approaches based on constrained optimization and penalization. Finally, the concept of threshold selection is introduced, as it constitutes a central element of this thesis.

2.1 Motivations for Forecast Combination

Combining forecasts from several predictive models has become a keystone in modern forecasting practice, particularly in financial applications. Several theoretical and empirical studies have demonstrated the value of combining forecasts, both as a safeguard against model misspecification and as a means of systematically improving out-of-sample performance. Those motivations are collected in this section.

2.1.1 Mitigating Model Uncertainty

The first and perhaps most important reason for combining forecasts is to protect against model uncertainty. As Wang et al. (2023) point out, the true process of data generation is generally unknown and the selection of a single correct model is often impossible. In their review of over five decades of literature, the authors point out that combining forecasts allows practitioners to guard against the risk of choosing a misspecified model by aggregating several plausible alternatives. They note that different models can extract complementary information from the data, particularly when they are built from distinct assumptions, data transformations or learning algorithms.

In this sense, combining forecasts is a practical solution to the fundamental limitation that no single model can consistently dominate across time, data sets or market regimes.

2.1.2 Variance reduction through diversification

The ability of forecast combinations to reduce error variance has been well established. Atiya (2020) provides a geometric intuition, represented in Figure 2.1, showing how forecast combinations can be closer to the true value in Euclidean space than any individual forecast, particularly when the individual models are diverse in structure and have uncorrelated forecast errors. The author shows that the mean squared error of the combined forecast can be strictly lower than that of the best individual model, particularly when the combined models are accurate and diversified.

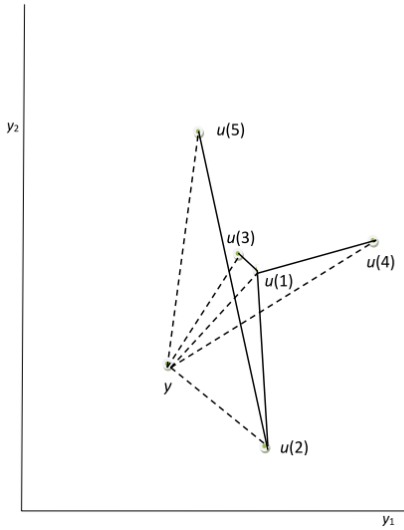


Figure 2.1: Geometrical intuition of forecast combination. *The solid lines between the points (forecasts)* $u(1), ..., u(5)$ *represent the forecast combinations, while the distances along the dotted lines to the true value* y *are proportional to the RMSEs of the corresponding forecasts (Atiya, 2020).*

This corresponds to the formal bias-variance decomposition presented by Timmermann (2006), who explains that while combining forecasts may not substantially reduce bias, it tends to significantly reduce variance, a desirable trade-off in predictive modeling. His analysis further shows that this effect is amplified when model errors are not highly correlated, reinforcing the idea that model diversity is essential to the success of combinations.

2.1.3 Empirical Evidence in Forecasting Competitions and Financial Applications

Empirical results strongly support the theoretical argumentation for the combination previously described. In the M4 competition, a large-scale forecasting challenge in multiple domains, Makridakis et al. (2018) reported that most of the best-performing methods were combinations of simpler models. Of the 17 best-performing methods, 12 were forecast ensembles. The authors point out that hybrid methods (combining statistical and

machine-learning forecasts) have outperformed stand-alone models.

This pattern was reinforced in the more recent M5 Accuracy Competition (Makridakis et al., 2022), which focused on retail sales forecasting using real-world Walmart data. Unlike M4, the M5 dataset included both time series data and structured exogenous variables such as prices, calendar effects and promotions. The best-performing methods were, again, ensemble-based approaches, particularly gradient boosting frameworks like LightGBM. Importantly, this highlighted that combining models trained on complementary information, whether it be time dynamics or exogenous features, leads to substantial gains in out-of-sample performance.

While M5 did not involve financial returns directly, the lessons are transferable: in high-dimensional, hierarchical and noisy environments, ensemble strategies systematically (in a empirical way) outperform individual models. This reinforces their relevance in financial applications, where market data exhibit similar levels of heterogeneity, non-linearity and structural variation.

In the context of stocks returns forecasting, Wolff and Echterling (2022) showed that combining forecasts from various machine learning models led to substantial improvements in predictive accuracy and Sharpe ratios. Their ensemble models consistently outperformed individual models as well as equal-weighted market benchmarks.

2.1.4 Regularization and Structural Stability

Another important reason why forecast combinations tend to work well, especially in finance, is that they help control the complexity of the model. This idea is known as **regularization**. In simple terms, regularization refers to techniques that limit the flexibility or instability of a model, especially when the available data is noisy or limited. Instead of letting the model chase every small variation in the training data (which often leads to overfitting), regularization introduces a form of restraint, for instance, by discouraging large or erratic weights.

In the context of combining forecasts, regularization typically means shrinking the weights of the combined model toward a more neutral or stable configuration, such as equal weights or weights based on past performance. This acts as a safety mechanism: if the data suggests that a certain model should be heavily weighted, the algorithm will still do so, but cautiously and only when the evidence is strong.

Roccazzella et al. (2022) propose a constrained optimization framework for combining forecasts that explicitly uses this principle. They penalize combinations that deviate too far from a chosen reference scheme, such as equal weights or inverse-loss weights. This

penalty, the regularization term, serves to stabilize the solution and reduce the risk of putting too much trust in a single, potentially overfit model. Their empirical results show that such shrinkage-based combinations consistently outperform more aggressive, unconstrained alternatives.

In this sense, combining forecasts does not just improve accuracy through averaging, it also improves reliability by controlling how much we "trust" each individual model. That's a key reason why combination methods often generalize better out-of-sample than any single machine learning model.

2.1.5 Limitations

Combining forecasts is not always **Risk-Free**. While the benefits of forecast combination are well-documented, it's important to acknowledge that combining models doesn't automatically guarantee better performance. In fact, under certain conditions, ensemble strategies can lead to worse predictions than even a single well-calibrated model.

As highlighted by Atiya (2020), combining a high-quality forecast with one that is poorly calibrated or highly biased can lead to degradation in performance, especially if the bad forecast is not down-weighted. The author stresses the importance of model selection and the potential danger of including "dragging" forecasts that reduce the overall forecast quality.

Wang et al. (2023) also remark that correlated forecast errors can limit the benefits of combination. If all models react similarly to a structural break or share the same inputs and assumptions, then combining them may amplify, rather than cancel, their errors.

This highlights the importance of using more refined combination methods. Techniques such as trimmed averages (which exclude the worst-performing forecasts), performance-based weighting, or penalized optimization (as proposed by Roccazzella et al. (2022)) are particularly useful in reducing the impact of poor models and focusing the ensemble on robust contributors.

2.2 Typology of Ensemble Strategies

Forecast combination methods can be classified into a few broad categories, depending on how they group individual forecasts, how weights are assigned and whether or not learning takes place. The subsequents examples are the ones implemented in the empirical study. Moreover, in this thesis, the combination problem is formulated similarly to the optimization problem discussed by Roccazzella et al. (2022), where forecasts are treated analogously to risky assets in a portfolio. Main of the combination strategies here-bellow

are derived from their work. However, the methods developed in Roccazzella et al. (2022) were originally designed for combining point forecasts in a **regression setting**. In this thesis, we extend their framework to the **classification** context, where individual models produce binary predictions. While the mathematical structure of the aggregation remains identical, the definition of forecast "loss" and the interpretation of performance metrics are adjusted accordingly.

Let us denote:

- $\mathcal{M} = \{1, \ldots, m\}$ is the set of all **individual** models,
- $\widehat{\mathbf{Y}} = \left[\widehat{\mathbf{y}}^{(1)}, \ldots, \widehat{\mathbf{y}}^{(m)}\right] \in \mathbb{R}^{N \times m}$ the matrix of forecasts, where each column $\widehat{\mathbf{y}}^{(j)} \in \mathbb{R}^N$ is a vector of N predictions made by model j,
- $\mathbf{w} \in \mathbb{R}^m$ the vector of combination weights, satisfying $w_j \geq 0 \ \forall \ j = 1, \ldots, m$ and $\sum_{j=1}^{m} w_j = 1$. This formulation ensures that:
  - weights can be seen as proportions (of confidence, performance, information),
  - the aggregated result remains bounded within the same scale as the original predictions,
  - the solution is interpretable: each model contributes "up to" its weight.
- $\widehat{\mathbf{y}}^{\text{agg}} \in \mathbb{R}^N$ the combined forecasts vector. All combination methods implemented in this thesis share the same basic structure; the combined forecasts vector is a linear combination of the individual model predictions, i.e.:

$$\widehat{\mathbf{y}}^{\text{agg}} = \widehat{\mathbf{Y}}\mathbf{w} = \sum_{j=1}^{m} w_j \widehat{\mathbf{y}}^{(j)} \tag{2.1}$$

Unlike in regression contexts, an additional step is required in classification: the binarization of the combined forecasts vectors.

Thresholding in classification: In the context of regression-based forecast combinations, no thresholding is needed since the outcome is continuous and directly interpretable. However, in classification problems, the aggregated prediction $\hat{y}_i^{agg} \in [0, 1]$ must be binarized to obtain final class decisions. The default threshold is usually set at $\tau = 0.5$, treating predictions above 0.5 as class 1.

Yet, this choice is arbitrary and may not reflect the optimal decision boundary in real-world datasets, especially when class distributions are imbalanced or when precision and recall need to be balanced. As such, selecting the threshold τ becomes a crucial step in classification ensembles and can significantly impact the final performance. We expand on this idea in Section 2.3.

$$\widehat{y}_i^{\text{class}} = \begin{cases} 1 & \text{if } \widehat{y}_i^{\text{agg}} \geq \tau \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

This post-processing step aligns the continuous ensemble output with the binary nature of the prediction task and ensures consistency across individual and aggregated model decisions.

The ultimate goal is to find a weight vector $\mathbf{w}$ that minimizes forecast error across the N predicted observations. What distinguishes these methods is how the weights are selected. Below, one presents the main strategy families and the methods implemented in this thesis.

2.2.1 Averaging-Based Methods

Simple Average (`SA`)
This method assigns uniform weights to all forecasts:

$$\overline{\mathbf{w}}^E = \mathbf{1}/m, \quad \forall j \Rightarrow \widehat{\mathbf{y}}^{\text{agg}} = \frac{1}{m}\sum_{j=1}^{m} \widehat{\mathbf{y}}^{(j)} \tag{2.3}$$

It is often used as a benchmark and has been shown to perform surprisingly well in many practical applications (Genre et al., 2013; Smith & Wallis, 2009) Its simplicity avoids over-fitting and estimation risk, especially when the number of forecasts is large relative to the available sample size.

Loss-Based Weights (`IL`)
This method assigns weights inversely proportional to past forecast errors: Let ℓ_j be a historical loss for model j. Then:

$$w_j = \frac{\ell_j^{-1}}{\sum_{k=1}^{m} \ell_k^{-1}}, \quad \widehat{\mathbf{y}}^{\text{agg}} = \widehat{\mathbf{Y}}\mathbf{w} \tag{2.4}$$

However, there are various ways to define this historical loss. Since we operate in a classification framework, where models output binary predictions over $\{0, 1\}$, it is necessary to define an appropriate loss function to evaluate model performance and derive combination weights. Several scoring rules can be used in this context: 0-1 loss (binary error), Logarithmic loss (cross-entropy), Brier score, precision, recall or even F1-score.

In this thesis, one chooses the **mean classification error** (0-1 loss) as the proxy for inverse model quality. Specifically, we compute the average error of each model j over a validation set:

$$\bar{e}_j = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\left[\widehat{y}_i^{(j)} \neq y_i\right]$$

The associated inverse-loss weights are then defined as:

$$\overline{w_j}^{IL} = \frac{1/\bar{e}_j}{\sum_{k=1}^{m} 1/\bar{e}_k} \tag{2.5}$$

This weighting scheme favors models that achieve lower average misclassification rates. We adopt the 0-1 loss rather than the others for two reasons. First, our classification problem is nearly balanced across classes (positive class: 48%, negative class: 52%), making the 0-1 error a reliable and interpretable performance measure (Bishop, 2006; Hastie et al., 2009). Second, in subsequent methods (e.g., constrained optimization and shrinkage), we estimate the covariance matrix of model errors, which is naturally defined from the binary error matrix. Using the same loss function for both performance ranking and error structure estimation ensures consistency and coherence across the ensemble framework.

2.2.2 Trimming and Screening-Based Methods

Forecast trimming and model screening methods are based on the idea that not all forecasts in a pool of m candidate models should be included in the final combination. Some may be too noisy, redundant, or clearly suboptimal. This family of methods seeks to identify and retain only a subset of reliable forecasts, thereby improving robustness and potentially reducing forecast error.

Formally, the forecast combination still follows the general structure:

$$\widehat{\mathbf{y}}^{\text{agg}} = \sum_{j=1}^{m} w_j \widehat{\mathbf{y}}^{(j)}, \quad \text{with } w_j = 0 \text{ if } j \notin \mathcal{A}, \quad \sum_{j=1}^{m} w_j = 1, \quad w_j \geq 0 \tag{2.6}$$

where $\mathcal{A} \subseteq \mathcal{M}$ is the set of selected models. The main difference with full-pool approaches lies in the pre-processing step that determines $\mathcal{A}$.

Fixed Proportion Trimming (`T-SA-p`)
This method implements global model trimming, as opposed to pointwise observation-level trimming. It excludes a fixed number p of the worst models based on their average error over a validation set.

Define the set of retained models $\mathcal{A} = \{1, \ldots, m-p\}$ by selecting the $m-p$ models with the lowest mean errors:

$$\mathcal{A} = \text{argmin}_{S \subset \{1,\ldots,m\}, |S|=m-p} \sum_{j \in S} \bar{e}_j \tag{2.7}$$

Assign weights as:

$$w_j = \begin{cases} \frac{1}{|\mathcal{A}|} & \text{if } j \in \mathcal{A} \\ 0 & \text{otherwise} \end{cases} \tag{2.8}$$

Cross-Validated Trimming (`T-SA-cv`)
Same as fixed trimming, this method selects a subset of models by removing a proportion of the worst performers. However, rather than fixing the trimming level p manually, the optimal number of excluded models is chosen via **cross-validation**. Specifically, a range of candidate values $p \in \{1, 2, \ldots, p_{max}\}$ is evaluated on a validation set using the classification loss previously described and the value of p minimizing the **mean classification error** is retained.

This approach increases robustness and adaptiveness, especially when model performance varies across times-series or market regimes.

Model Confidence Set (`T-mcs-`α)
The Model Confidence Set (MCS) procedure, introduced by Hansen et al. (2011), is a formal statistical method used to identify the subset of models that belong to the set of superior forecasts with a given level of confidence α.

Let recall that $\mathcal{M} = \{1, \ldots, m\}$ is the initial set of models. The MCS procedure iteratively tests the null hypothesis of equal predictive ability among models:

$$H_0 : \mathbb{E}[d_{jk}] = 0 \quad \forall j, k \in \mathcal{M}_t,$$

where d_{jk} is the loss differential between models j and k. At each step, the model with the worst performance is removed based on test statistics as $t_{\max}$ and the process continues until the null hypothesis can no longer be rejected at level α.

The resulting set $\mathcal{M}_\alpha \subseteq \mathcal{M}$ is referred to as the *superior set* of models at confidence level α.Then formally:

$$w_j = \begin{cases} \frac{1}{|\mathcal{M}_\alpha|} & \text{if } j \in \mathcal{M}_\alpha \\ 0 & \text{otherwise} \end{cases} \tag{1.9}$$

In our thesis, one implement a simplified bootstrap-based approximation of the Model Confidence Set (MCS) procedure proposed by Hansen et al. (2011). Specifically, we use a moving block bootstrap (to account for serial dependence in forecast errors) to resample the error matrix and compare each model to the current best model using paired t-tests. A model is retained if it is not significantly worse than the best model in at least $1 - \alpha$ of the bootstrap replications. This pragmatic approach, while not strictly equivalent to the full MCS procedure, captures the same intuition of identifying a subset of robust models and performs well in practice (Roccazzella et al., 2022).

2.2.3 Constraint Optimization

The constraint optimization, also called the minimum variance combination strategy treats forecast aggregation analogously to portfolio allocation. The central idea is to minimize the variance of the combined forecast error by exploiting the cross-covariance structure between models, just as in mean–variance portfolio theory where the goal is to minimize portfolio risk (i.e. variance).

Constraint Optimization - Minimum Variance (`CO`)

Let V be the forecast error matrix and $\widehat{\Sigma}$ the empirical covariance matrix of model errors, build as:

$$\mathbf{e}^{(j)} = |\mathbf{y} - \widehat{\mathbf{y}}^{(j)}|, \quad V = \left[\mathbf{e}^{(1)} \quad \cdots \quad \mathbf{e}^{(m)}\right], \quad \widehat{\Sigma} = \frac{1}{N} V^\top V$$

The optimal weight vector is then the solution to the following quadratic program:

$$\mathbf{w}^* = \arg\min_{\mathbf{w} \in \mathcal{W}} \mathbf{w}^\top \widehat{\Sigma} \mathbf{w} \tag{2.9}$$

subject to the simplex constraint:

$$\mathcal{W} := \left\{ \mathbf{w} \in \mathbb{R}^m : w_j \geq 0, \ \sum_{j=1}^{m} w_j = 1 \right\} \tag{2.10}$$

2.2.4 Penalized Constrained Optimization (COP)

The COP framework extends the minimum variance approach by incorporating a regularization term that penalizes deviations from a reference weight vector $\mathbf{w}^{\text{ref}}$. This makes the resulting combination both optimal in terms of variance and robust to estimation errors. One uses the same notation as used in Section 2.2.3.

The optimization problem becomes:

$$\mathbf{w}_\delta^* = \arg\min_{\mathbf{w} \in \mathcal{W}} \left(\mathbf{w}^\top \widehat{\Sigma} \mathbf{w} + \delta \cdot d(\mathbf{w}, \mathbf{w}^{\text{ref}}) \right) \tag{2.11}$$

where:

- $\delta > 0$ is a regularization parameter controlling the strength of shrinkage;
- $d(\cdot, \cdot)$ is a divergence function measuring the distance between $\mathbf{w}$ and $\mathbf{w}^{\text{ref}}$.

Different choices of divergence yield different penalized versions:

- **L2 penalty (ridge)**: $d(\mathbf{w}, \mathbf{w}^{\text{ref}}) = \|\mathbf{w} - \mathbf{w}^{\text{ref}}\|_2^2$
- **L1 penalty (lasso)**: $d(\mathbf{w}, \mathbf{w}^{\text{ref}}) = \|\mathbf{w} - \mathbf{w}^{\text{ref}}\|_1$
- **Elastic Net**: $d(\mathbf{w}, \mathbf{w}^{\text{ref}}) = \alpha \|\mathbf{w} - \mathbf{w}^{\text{ref}}\|_1 + (1 - \alpha) \|\mathbf{w} - \mathbf{w}^{\text{ref}}\|_2^2$

- **Cross-entropy**: $d(\mathbf{w}, \mathbf{w}^{\text{ref}}) = \sum_{j=1}^{m} w_j \log\left(\frac{w_j}{w_j^{\text{ref}}}\right)$

This approach generalizes classical forecast combination by combining both performance optimization and structural regularization. It was formalized and applied to forecast combination by the seminal work of Roccazzella et al. (2022). The reference weight vector $\mathbf{w}^{\text{ref}}$ is typically set as the ones derived from Section 2.2.1:

- $\mathbf{w}^{\text{ref}} = \overline{\mathbf{w}}^{E} \implies$ Equal Weights (i.e. Simple Average);
- $\mathbf{w}^{\text{ref}} = \overline{\mathbf{w}}^{IL} \implies$ inverse Loss-Based.

The COP framework offers a natural regularization mechanism, balancing the flexibility of error-based weighting with the stability of shrinkage toward a structured prior (Roccazzella et al., 2022).

2.2.5 Covariance Shrinkage Optimization (COS)

The COS framework proposes to regularize the combination process not by penalizing the weights directly, but by stabilizing the covariance matrix $\widehat{\Sigma}$ used in the optimization. When the number of observations N is small relative to the number of models m, the empirical covariance matrix of forecast errors may be poorly estimated. The COS method mitigates this by shrinking $\widehat{\Sigma}$ toward a structured target matrix Σ_0, leading a more robust estimate:

$$\Sigma_\lambda = (1-\lambda)\widehat{\Sigma} + \lambda\Sigma_0, \quad \lambda \in [0,1] \tag{2.12}$$

The shrinkage intensity $\lambda \in [0,1]$ governs the trade-off between the empirical covariance matrix $\widehat{\Sigma}$, which may be noisy in small samples and the structured shrinkage target Σ_0. A higher value of λ puts more weight on the prior, improving stability but introducing bias. Conversely, a lower λ allows more flexibility but risks overfitting to estimation noise. In this thesis, λ is selected via cross-validation over a predefined grid, by optimizing classification performance on a validation set. This empirical tuning approach aligns with the practical and data-driven nature of the thesis and reflects the common procedure used in shrinkage-based portfolio optimization.

The final weights are then computed via minimum variance optimization:

$$\mathbf{w}_\lambda^* = \arg\min_{\mathbf{w}\in\mathcal{W}} \mathbf{w}^\top \Sigma_\lambda \mathbf{w} \tag{2.13}$$

Two structured shrinkage targets are considered:

- **COS-E** (Equal weights): the target matrix is set to $\Sigma_0 = \overline{\sigma}^2\mathbf{I}$, where $\overline{\sigma}^2 := \frac{1}{m}\sum_{i=1}^{m}\sigma_i^2$ is a constant and $\mathbf{I}$ the identity matrix. This corresponds to the assumption that all models have equal and uncorrelated forecast errors. In this setting, if the shrinkage were total (i.e.$\lambda = 1$), the solution would be equivalent to assigning equal weights to

all models. The name `COS-E` reflects this implicit connection to the equal-weighted baseline.

- **COS-IL** (Inverse Loss): the shrinkage target is a diagonal matrix $\Sigma_0 = \text{diag}(\sigma_1^2, \ldots, \sigma_m^2)$, where each σ_j^2 is the empirical variance of the classification error of model j. When computed from 0–1 loss, this target reflects heteroskedasticity in model performance. If $\lambda = 1$, the solution tends toward inverse-loss weighting, since minimizing under a diagonal Σ_0 leads to:

$$w_j \propto \frac{1}{\sigma_j^2}$$

which is equivalent to inverse variance or inverse error weighting. Hence, the `COS-IL` label denotes the fact that this target structure approximates the inverse-loss weighting scheme.

2.3 Threshold Selection in Classification-Based Forecast Combinations

While most ensemble forecasting literature has focused on regression tasks, our setting requires combining binary forecasts to predict the direction of future stock returns. In this context, the output of each model $\hat{y}_i^{(j)} \in \{0, 1\}$ typically represents a binary classification output. After aggregation, the combined forecast $\hat{y}_i^{agg} = \sum_j w_j \hat{y}_i^{(j)}$ remains a continuous value in the range $[0, 1]$.

Unlike regression tasks, classification requires transforming these continuous outputs into binary predictions. A typical practice is to set a threshold of $\tau = 0.5$, classifying an instance as class 1 if $\hat{y}_i^{agg} \geq \tau$.This practice comes from logistic regression, where it is assumed that predicted probabilities are calibrated and classes occur in similar proportions. However, these assumptions rarely hold in financial contexts.In stock prediction, datasets tend to be imbalanced, and the costs of mistakes can differ between false positives and missed signals. Consequently, choosing an appropriate threshold τ becomes a critical step. The fixed threshold at 0.5 may be suboptimal and even harmful to real-world performance, especially in a walk-forward evaluation setting.

Some recent studies have emphasized how adjusting thresholds can improve classification when classes are unbalanced (Mohammed et al., 2023). In financial forecasting, some recent works using reinforcement learning ensembles have shown that dynamically adjusting variance thresholds improves trading performance (Shao et al., 2024). However, few studies on financial forecast combination address how thresholds should be set.

In this thesis, we formalize the problem of post-combination threshold selection and propose three practical strategies, in addition to the naïve threshold, each designed to increase the flexibility and robustness of classification ensembles.

- **Naïve Threshold** (`naive`): The default rule assigns class 1 if the aggregated probability exceeds $\tau = 0.5$. While commonly used (Simonis, 2021), this threshold assumes symmetric misclassification costs and balanced data, assumptions that might not hold in real-world financial applications.

- **Direct Optimization** (`OPTI`): We solve for $\tau^* = \arg\max_\tau \mathcal{C}_\mathcal{M}(\tau)$, where $\mathcal{C}_\mathcal{M}$ is a classification metric (e.g., F1-score, accuracy). This approach requires a validation set and directly optimizes the chosen performance measure.

- **Cross-Validated Thresholding** (`CV`): To prevent overfitting due to limited validation data (typically 10% in our walk-forward setup), we implement a TimeSeriesSplit cross-validation. The best τ is selected based on average metric performance across the folds.

- **Quantile-Based Thresholding** (`QTL`): In scenarios where no ground truth is used for threshold tuning, we propose using the empirical distribution of predicted scores. For instance, setting τ at the 75th percentile ensures that only the top 25% of confident predictions are labeled as positive.

Main Takeaways

In this chapter, we clarified the main motivations for adopting ensemble strategies, including their ability to mitigate model uncertainty, reduce variance and improve both regularization and structural stability. We have also emphasized that, despite their attractive properties, ensemble approaches have certain limitations that deserve consideration. In addition, we have explored in greater depth the typology of ensemble techniques selected for this thesis. In particular, we emphasized the relevance of mean-based methods, selection and trimming strategies and more advanced frameworks such as constrained optimization, either with explicit penalty terms or via covariance shrinkage. Finally, we stressed the importance of carefully determining threshold selection procedures. In this regard, we examined four main approaches: a naive method, a direct optimization strategy, a cross-validation procedure, and a quantile-based criterion. Together, these considerations lay a foundation for the subsequent implementation and empirical evaluation of ensemble models in predicting future stock returns.

3 Portfolio Optimization

This chapter addresses the problem of portfolio optimization from a non-traditional angle. Rather than estimating future returns or relying on parametric assumptions, we take a classification-based approach: each year, we train machine learning models; both individual and ensemble-based, to predict whether a stock is likely to outperform or underperform a benchmark over the coming week.

These models are used in a walkforward setting: at the start of each year, they are trained on past data to generate binary predictions for the investment horizon ahead. Based on these predictions, we select a subset of assets classified as likely outperformers.

The focus of this chapter is on how to allocate capital among these selected assets using robust and data-driven methods. Importantly, we do not estimate expected returns, we only compute the empirical covariances based on past returns. Therefore, we restrict our attention to portfolio optimization techniques that either do not require return forecasts or are explicitly designed to handle such classification-based settings.

Moreover, if the reader is not familiar with back-testing metrics used to analyze the performance of portfolios, they may refer to Appendix C for detailed information.

We will review and apply four main allocation strategies:

- Equally Weighted Portfolio (EW)
- Mean-Variance Portfolio (with a focus on its minimum variance form)
- A shrinkage approach
- A filtering approach
- A combined approach

3.1 Equally Weighted Portfolio

The equally weighted portfolio (`EW`) assigns the same weight to each asset in the selected universe. Formally, for a portfolio consisting of N assets, the weights γ_i[1] are given by:

$$\gamma_i = \frac{1}{N}, \quad \text{for } i = 1, \ldots, N$$

This approach does not rely on any statistical estimates of expected returns or covariances. As such, it is often used as a benchmark in empirical studies of portfolio performance. Despite its simplicity, the equally weighted portfolio has shown surprisingly strong out-of-sample performance relative to more sophisticated optimization methods. For instance, DeMiguel et al. (2009) demonstrate that, across a wide range of datasets and investment settings, `EW` often outperforms mean-variance portfolios that rely on estimated inputs.

In our context, the `EW` strategy is applied to the subset of assets classified as likely to outperform the benchmark over the subsequent period, based on machine learning predictions. The use of `EW` on a filtered set of assets allows us to combine the predictive power of classification models with the robustness of a non-parametric allocation scheme.

3.2 Risk-Based Portfolio

3.2.1 Mean-Variance Portfolio

The mean-variance optimization framework, introduced by Markowitz (1952), defines the portfolio selection problem as a trade-off between expected return and risk, where risk is measured by the variance of portfolio returns. The investor selects weights $\gamma \in \mathbb{R}^N$ to solve the following quadratic optimization problem:

$$\begin{aligned} \min_{\gamma} \quad & \gamma^\top \Sigma \gamma \\ \text{subject to} \quad & \gamma^\top \mu = \mu_p \\ & \sum_{i=1}^{N} \gamma_i = 1 \end{aligned} \tag{3.1}$$

where $\mu \in \mathbb{R}^N$ is the vector of expected returns, $\Sigma \in \mathbb{R}^{N \times N}$ is the covariance matrix of asset returns, and μ_p is the target return.

Since the true values of μ and Σ are unknown, they are typically estimated from historical data. Let $R \in \mathbb{R}^{T \times N}$ denote the return matrix, where each row corresponds to a

[1] We use here the notation γ to clearly differentiate those weights from the one used during the combination phase of the forecats

time observation and each column corresponds to an asset. Denote by r_t the t-th row of R, i.e., the vector of returns of all N assets at date t:

$$\hat{\mu} = \frac{1}{T}\sum_{t=1}^{T} r_t \tag{3.2}$$

$$\hat{\mathbf{\Sigma}} = \frac{1}{T-1}\sum_{t=1}^{T}(r_t - \hat{\mu})(r_t - \hat{\mu})^\top \tag{3.3}$$

These estimators are simple and unbiased under the assumption that asset returns are independently and identically distributed. In the context of our work, this estimation is carried out only for the subset of assets that have been classified as likely to outperform the benchmark, using past return vectors for covariance estimation.

Issues with Mean-Variance method, the estimators

Despite its elegant formulation, the mean-variance approach is notoriously sensitive to estimation error, especially in $\hat{\mu}$ and $\hat{\mathbf{\Sigma}}$, as demonstrated in both theoretical and empirical studies.

As discussed by R. O. Michaud (1989), small perturbations in the input estimates can lead to large shifts in the optimized portfolio weights, an issue known as "error maximization". This is particularly problematic in high-dimensional settings, where the number of assets N approaches or exceeds the number of observations T.

Jagannathan and Ma (2003) argue that the sample covariance matrix performs poorly in these settings, resulting in portfolios that are not well diversified and highly unstable out-of-sample. The condition number of $\hat{\Sigma}$ may also be extremely high, leading to numerical instability when inverting the matrix in the optimization routine.

Moreover, DeMiguel et al. (2009) show that these estimation errors frequently result in optimized portfolios performing worse than simple naive strategies such as the equally weighted portfolio. Kan and Zhou (2007) further emphasize that the utility loss caused by parameter uncertainty can be substantial, and that investors may benefit from adopting estimation-robust alternatives.

These limitations highlight the importance of using estimation methods that improve the stability of empirical covariance matrices in practice. In the following sections, we explore two such approaches: shrinkage estimators and eigenvalue filtering techniques.

3.2.2 Shrinkage

As highlighted in the previous section, the sample covariance matrix $\hat{\Sigma}$ can perform poorly in high-dimensional settings due to its sensitivity to estimation noise. To mitigate this issue, shrinkage methods propose combining the sample estimator with a structured and more stable target matrix. The general form of a shrinkage estimator is:

$$\hat{\mathbf{\Sigma}}_{\text{LW}} = \delta\mathbf{\Theta} + (1-\delta)\hat{\mathbf{\Sigma}}$$

where $\mathbf{\Theta}$ is the shrinkage target (e.g., identity matrix, constant covariance matrix), and $\delta \in [0,1]$ is the shrinkage intensity parameter.

Ledoit and Wolf (2004) develop an analytically optimal shrinkage estimator that minimizes the mean-squared error (MSE) between the true covariance matrix and its estimate. Their estimator is particularly well-suited for large-dimensional portfolios, where the sample covariance matrix becomes unstable or even singular. The shrinkage intensity δ is data-driven and adapts to the concentration of eigenvalues in the sample.

Shrinkage is computationally efficient, improves the conditioning of the covariance matrix, and leads to more stable out-of-sample portfolio weights. As emphasized in Ledoit and Wolf (2003), the approach delivers performance close to the true (but unobservable) optimal portfolio. However, although shrinkage techniques improve stability by shrinking noisy sample estimates toward a structured target, they do not fully correct for the distortions present in the eigenstructure of the covariance matrix. This is especially problematic when dealing with a large number of assets. Therefore, we now turn to spectral filtering methods

3.2.3 Filtering

An alternative way to improve the stability of covariance estimates is to employ Random Matrix Theory (RMT), which provides a formal framework for distinguishing noise from signal in large correlation matrices. In practice, the filtering procedure is typically applied to the correlation matrix rather than directly to the empirical covariance matrix $\widehat{\Sigma}$. Let $\mathbf{D}$ denote the diagonal matrix containing the standard deviations of each asset on the diagonal, i.e., $\mathbf{D} = \text{diag}\big(\sqrt{\widehat{\Sigma}_{11}}, \ldots, \sqrt{\widehat{\Sigma}_{NN}}\big)$. Then the empirical correlation matrix is given by

$$\widehat{\mathbf{C}} = \mathbf{D}^{-1}\,\widehat{\mathbf{\Sigma}}\,\mathbf{D}^{-1}.$$

Random Matrix Theory provides a theoretical reference to distinguish noise from signal in such correlation matrices. Specifically, the theory states that for a correlation matrix constructed from N standardized random time series of length T, the eigenvalue distribution converges to the `Marchenko-Pastur` distribution as $N, T \to \infty$, provided that $1 < \frac{T}{N} < +\infty$. Under these assumptions, the limiting density of expected eigenvalues λ is

bounded within the interval $\lambda \in \left[\underbrace{\left(1-\sqrt{\frac{N}{T}}\right)^2}_{\lambda_-}, \underbrace{\left(1+\sqrt{\frac{N}{T}}\right)^2}_{\lambda_+}\right]$.

Therefore, considering our empirical correlation matrix $\widehat{C}$, any eigenvalue that lies between λ_- and λ_+ can be considered as arising from noise. Conversely, any eigenvalue exceeding λ_+ can be interpreted as containing genuine signal. Accordingly, we can decompose the empirical correlation matrix as:

$$\widehat{C} = \underbrace{\widehat{C}^{(n)}}_{\text{noise}} + \underbrace{\widehat{C}^{(s)}}_{\text{signal}},$$

where $\widehat{C}^{(n)} := \sum_{i:\,\lambda_i \leq \lambda_+} \lambda_i \, v_i \, v_i^\top$, with v_i denoting the eigenvector associated with the i-th eigenvalue, and $\widehat{C}^{(s)} := \widehat{C} - \widehat{C}^{(n)}$. In our implementation, we retain only the signal component $\widehat{C}^{(s)}$, effectively projecting the correlation structure into the subspace spanned by the significant eigenmodes. Finally, we recover a filtered covariance matrix:

$$\widehat{\mathbf{\Sigma}}_{\text{filtered}} = \mathbf{D}\, \widehat{C}^{(s)}\, \mathbf{D}.$$

Pioneering work by Pafka et al. (2004) and further extended by Bun et al. (2017) has demonstrated that such spectral cleaning leads to more robust covariance estimators, especially when combined with shrinkage and regularization.

3.2.4 Combination of Shrinkage and Filtering

In our setting, we further regularize the filtered covariance matrix by shrinking it towards the Ledoit–Wolf shrinkage estimator. Concretely, the final estimator is computed as the combination:

$$\widehat{\mathbf{\Sigma}}_{\text{final}} = (1-\delta)\, \widehat{\mathbf{\Sigma}}_{\text{filtered}} + \delta\, \widehat{\mathbf{\Sigma}}_{\text{LW}},$$

where $\widehat{\mathbf{\Sigma}}_{\text{LW}}$ denotes the Ledoit–Wolf shrinkage target and $\delta = 0.3$, fixed arbitrary, controls the balance between the two approaches. This hybrid estimator thus combines the benefits of spectral noise reduction with shrinkage regularization, improving both the stability and the reliability of the covariance estimates used in portfolio optimization. It is this method that is refer as `FILTERING` in the further results.

4 Empirical Methodology

This chapter describes how the models were tested to identify if a stock is likely to outperform or underperform an index and build portfolios based on this predictions. It relies on the theoretical foundations discussed in Chapters 1, 2 and 3. The approach includes four main steps: preparing the dataset, training the models, creating the portfolios, and assessing their performance.

4.1 Dataset Construction

4.1.1 Universe

For this analysis, we focus on the stocks included in the **NASDAQ-100** index. This index represents the 100 largest non-financial companies listed on the NASDAQ Stock Market, ranked by market capitalization. The time horizon of the forecasts extends from January 2007 to December 2024. The NASDAQ-100 is known for its heavy exposure to technology and growth-oriented companies such as Apple, Microsoft, and NVIDIA, which makes it particularly interesting for prediction tasks due to its higher volatility and innovation-driven price dynamics.

We select this index for several reasons. First, the stocks that constitute the NASDAQ-100 are highly liquid, which means it is more reasonable to assume that trading activities based on model predictions would not significantly impact prices. Second, as a large-cap benchmark, it reflects realistic investment constraints faced by institutional investors.

The NASDAQ-100 index is reviewed and rebalanced by NASDAQ every quarter, with constituents added or removed according to predefined eligibility criteria. To build our dataset in a way that reflects these changes over time, we use the actual historical composition of the index as it evolved. Because our out-of-sample predictions cover a full year ahead, we fix the set of constituents to those included in the index at the start of each year.

Given the evolving nature of the index, we were careful to ensure that no stock was included in the dataset before its actual inclusion date in the index, or after its removal, strictly aligning the universe to what would have been known at the time. Additionally, we required that each stock have sufficient historical price data available to be included in the initial training window in which it appears. This condition ensures that models are trained on complete and consistent data.

This methodology avoids **survivorship bias**[1] and reflects the constraints of a realistic, real-time investment strategy. The historical constituents and historical price data are sourced from Bloomberg L.P. (2024). Figure 4.1 shows the evolution of the number of securities included in the index over time.

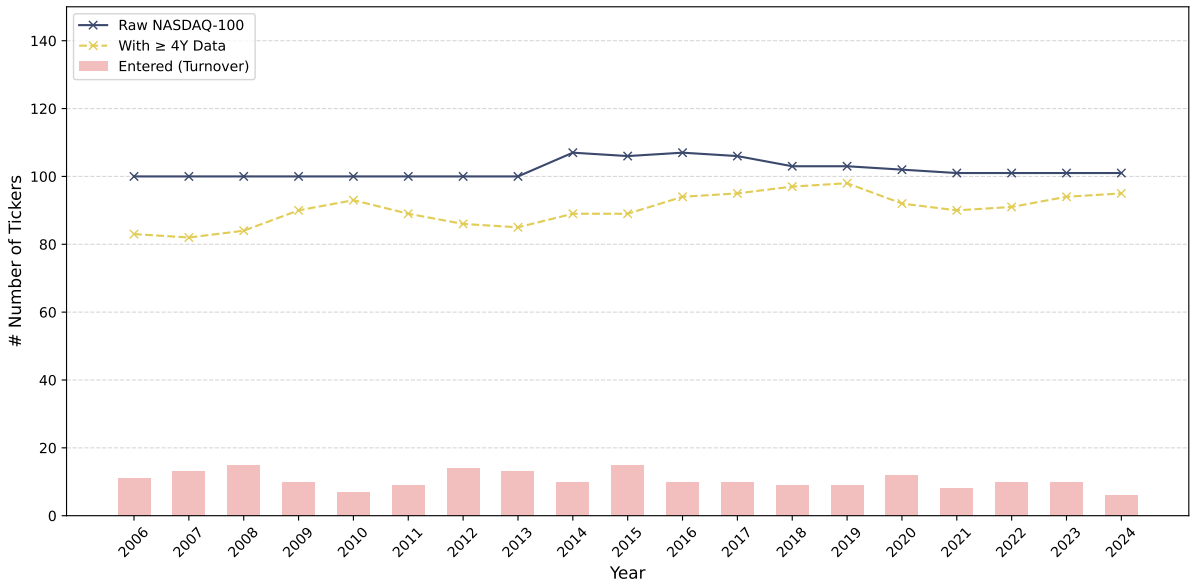


Figure 4.1: Evolution of NASDAQ-100 index composition and annual turnover between 2006 and 2024. *The blue solid line indicates the total number of stocks included in the index each year, while the yellow dotted line shows how many of these stocks had at least four years of historical data at the time of analysis. The red bars represent annual turnover, measured by the number of new stocks entering the index. Over the entire period, more than 80% of the components had sufficient historical data, enabling a solid retrospective analysis. Turnover remains moderate but constant, reflecting the dynamic nature of the index's composition.*

4.1.2 Technical Analysis Features

This thesis focuses solely on features derived from price and volume dynamics, what is commonly referred to as **technical analysis**. This choice is both practical and methodological. First, technical indicators are available for all stocks across time, without requiring point-in-time accounting data. Second, they can capture market behavior such as momentum,

[1] Arises when analyses include only the entities that have "survived" until the end of the observation period, while excluding those that have disappeared (e.g., delisted or failed companies). It leads to an overstimation of performance.

reversals or volatility, which are often hard to quantify using fundamentals alone.

The raw data comes from Bloomberg L.P. (2024). It includes daily close and volume information for each stock. To improve comparability across assets and reduce computational load, we aggregated the data to a weekly frequency. Inspired by the methodology of Wolff and Echterling (2022), we extract the last available price as of each Wednesday to represent the weekly close, and sum the daily volumes from the previous Thursday to that day. This approach helps avoid start-of-week and end-of-week effects, which are commonly observed in financial time series and may bias short-term return signals.

Once the weekly time series are built, a wide range of technical features are computed. These include both classic indicators, such as moving averages, volatility bands or relative strength indices and more statistical ones such as skewness or rolling regression slopes. Table 4.1 describe each of those 34 technical indicators. For more information on the meaning of each, please refer to Appendix D.

Indicator (daily data, pre-resampling)	**Period (Days)**
log(Price / $SMA_{50,100,200}$)	50, 100, 200
log(Price / $EMA_{50,100,200}$)	50, 100, 200
MACD level and signal	-
MACD crossover binary indicator	-
Relative Strength Index (RSI)	14
log(Price / Upper Bollinger Band)	20
log(Price / Lower Bollinger Band)	20
Indicator (weekly data, post-resampling)	**Period (Weeks)**
Weekly return	1
Excess return (vs index)	13 & 26
Lagged stock return	13 & 26 & 52
Price momentum	13 & 26 & 52
Beta (stock vs index)	13 & 26
Rolling return volatility	13 & 26
Return / Volatility (Sharpe proxy)	13
Return-volume correlation	13
Regression slope (linear trend of price)	13 & 26
Weekly RSI	13
Return skewness	13 & 26
On-Balance Volume (OBV)	-
Dollar trading volume	-

Table 4.1: Description of the technical indicators. *Indicators with a specified period are computed using a backward-looking rolling window covering the stated number of weeks. Indicators without a defined period are computed using the available data at the time of observation. All indicators based on daily data are computed prior to resampling the data to a weekly frequency. The computations are performed using the* `TA-Lib` *Python library.*

Most of these features are inspired by previous work in the literature, especially the frameworks presented by Wolff and Echterling (2022) and Kakushadze (2016), who highlight how price-based indicators can effectively signal future return patterns. For missing values, we fill them with the last value known at the time. We end up with a data matrix $\mathbf{X}$ of dimensions $(259,202 \times 34)$, containing all computed technical indicators for the full stock-week observations. However, this matrix is not directly used to train the models. To avoid any form of look-ahead bias and ensure consistency over time, all preprocessing steps, including **standardization** and **dimensionality reduction**, are performed strictly within each rolling training window. Specifically, each feature is standardized in its respective window to maintain temporal comparability. Subsequently, a Principal Component Analysis (PCA) is applied within the same window to reduce the dimensionality of the feature space, while retaining 95% of the original variance. These transformations result in a final matrix $\tilde{\mathbf{X}}$ which is then used as input for each model training window.

4.2 Model Training

To evaluate our stock selection models in a realistic and forward-looking setting, we adopt a walk-forward strategy using rolling five-year windows. At each iteration, the entire window is split chronologically into three segments: 70% is used to train the base models, 10% to calibrate the threshold τ[2] and the several ensemble model methods, and the final 20% is reserved for evaluation. This ensures that all performance metrics are computed on strictly out-of-sample data, while maintaining temporal consistency. This procedure is detailed in Figure 4.2.

During the 70% training phase, we perform hyperparameters optimization for each model using a Randomized Grid Search with 2-fold time series cross-validation. For each model, we sample 10 hyperparameters combinations from a predefined grid and select the one that achieves the highest average accuracy across the folds.

Before model training, each training rolling window is passed through a consistent preprocessing pipeline. All input features are standardized by removing the mean and scaling to unit variance, again based only on the training portion. Finally, we apply Principal Component Analysis (PCA) to reduce the dimensionality of the input space, retaining the number of components that preserve 95% of the total variance. This results in a transformed feature matrix $\tilde{X}$, which serves as input to the machine learning models described in Chapter 1. The window is then advanced by one year, and the process is repeated from 2007 to 2024, resulting in 18 rolling evaluations that mimic real-world model deployment.

[2]The meaning of this threshold is well described in Section 2.3

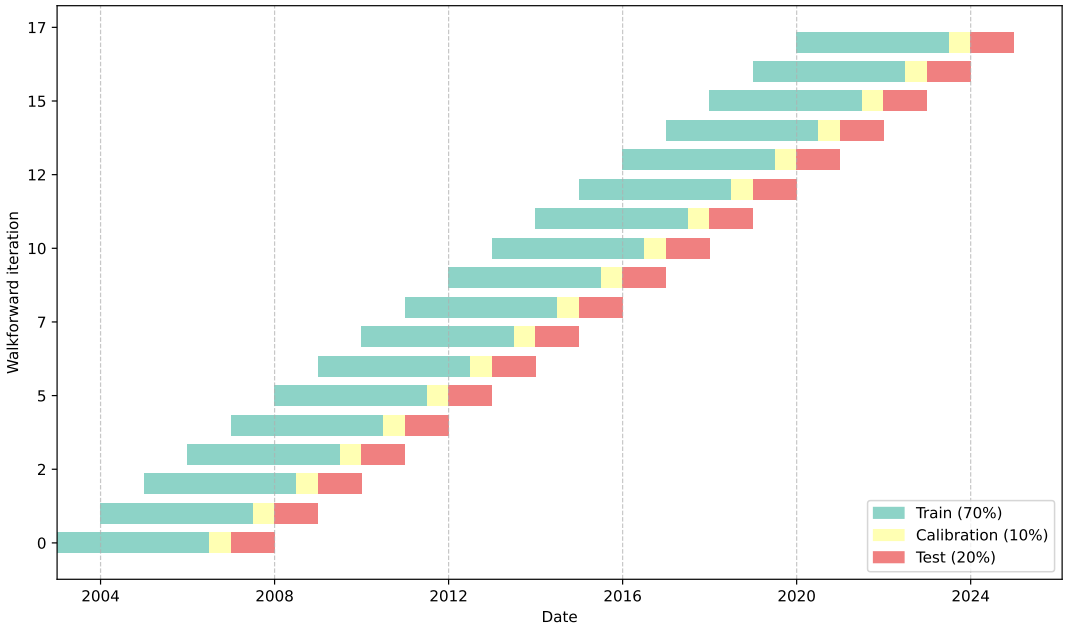


Figure 4.2: Walk-forward strategy for model training. *Each window is split chronologically into 70 % for training base models, 10 % for calibrating the threshold τ and tuning ensemble methods, and 20 % for out-of-sample evaluation.*

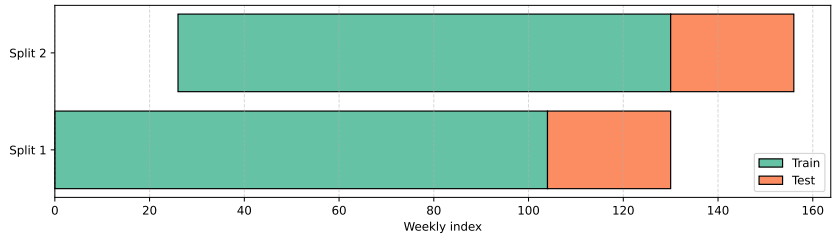


Figure 4.3: Detail of the hyperparameter tuning stage. *A randomized grid search over 10 sampled parameter combinations with 2-fold time-series cross-validation performed on the 70% training segment. Each training fold is composed of 2 years of observations.*

The simulations were executed on a computer equipped with an Intel Core i7 processor. To further enhance computational performance, we leveraged the CuML library to take advantage of the NVIDIA GeForce RTX GPU present in the machine. This GPU acceleration significantly reduced the processing time of machine learning models.

4.3 Portfolio Construction Approach

Once we have trained the stock selection models (individuals and ensembles), we use their predictions for one year. This corresponds to 52 purchase recommendations per model and per ticker over the whole year. Since our models forecast relative out-performance over the next week, we execute trades every Wednesday to both match the prediction horizon and contain transaction costs. The weekly process is as follows:

1. **Load Signals.** We identify the potential of each asset by fitting individual ML classification models exploiting usual technical variables. This leads to binary

variables, identifying whether the net weekly return of each asset is likely up or down. Next, we combine those predictions to form an aggregate up/down forecast for each asset depending on the threshold strategy selected. We Retrieve this week's "buy" or "no-buy" signals for each tickers.

2. **Exit Weak Positions.** Calculate last week's realized returns on all current holdings. Exit any position that did not receive a fresh "buy" signal.

3. **Form Selection.** Define the current investable universe as all stocks flagged "buy" this week, i.e. class 1 (including both survivors and newly recommended names).

4. **Estimate Risk.** Since our models produce only *buy/no-buy* signals and not quantitative return forecasts, we do not have model predicted returns. Therefore, we proxy portfolio risk using historical data: for the current selection, ones collects four years of weekly returns and compute the sample covariance matrix.

5. **Optimize Weights.** Solve a nonnegative weight (we do not permit short-selling) optimization under four schemes:

   - Equal Weight (`EW`): $\gamma_i = 1/N$.
   - Minimum-Variance (`MV`):

   $$\min_{\gamma} \ \gamma^\top \Sigma \gamma \quad \text{s.t. } \mathbf{1}^\top \gamma = 1, \ \gamma \geq 0.$$

   - Shrinkage (`SHRINKAGE`): Replace Σ with a Ledoit–Wolf shrinkage estimator $\hat{\Sigma}$ before solving MV.
   - Filter-Based (`FILTERING`): Replace Σ with a filtered covariance matrix obtained through Random Matrix Theory. Eigenvalues below the Marchenko-Pastur threshold are removed before solving. It is then shrinked to the Ledoit-Wolf estimator at 30% (i.e. $\delta = 0.3$)

6. **Rebalance.** Compare target weights $\gamma^\star$ to current allocations, then trade to adjust positions. We include Interactive Brokers' commission schedule (June 2024): \$0.005 per share (minimum \$1, maximum 1% of trade value).

This procedure is applied on a weekly basis for each individual model and their ensemble combinations, and results are then compared to a passive investment in the `NASDAQ-100` index. Accordingly, we adjust the benchmark performance to account for the fees associated with such a product, specifically using the cost structure of one of the oldest available ETFs, the Invesco EQQQ Nasdaq-100 UCITS ETF (Invesco, 2024).

As for the risk-free rate used in Sharpe and other risk-adjusted metrics, we compute it as the average value of the 3-month Euribor between 2007 and 2024. Lastly, we assume an initial portfolio of \$10 million, which ensures that all simulated strategies are free from liquidity constraints or market impact considerations.

4.4 Performance Evaluation

The performance evaluation approach depends on the specific task being assessed, whether it concerns prediction or portfolio performance. For evaluating the predictive power of classifiers, we rely on standard metrics: **accuracy**, which measures the overall proportion of correct predictions; **precision**, which quantifies how many predicted positives are actually correct; **recall**, which captures how many actual positives were successfully identified; and the **F1-score**, which provides a harmonic mean of precision and recall. Unless otherwise specified, accuracy is used as the main reference metric for comparing classification performance between models (such as hyperparameters optimization...).

For evaluating portfolio performance, we rely on several well-known backtesting metrics that capture both return and risk aspects. Among them, the **Sharpe Ratio**, which measures risk-adjusted return relative to the risk-free rate, the **Annualized Performance Yield (APY)**, which reflects the average yearly growth rate of the portfolio and **annualized volatility**, which captures the standard deviation of returns. In addition, we assess downside risk through metrics such as **Maximum Drawdown**, **Value at Risk (VaR)**, and **Conditional Value at Risk (CVaR)**. Finally, we also consider the **Win Rate**, the proportion of weeks with positive returns, as well as the **average gain** and **average loss** to further analyze the distribution of weekly performance. Together, these indicators offer a clear perspective on the profitability and stability of the tested strategies. Appendix C described in details all the indicators and metrics described here over.

4.5 Robustness Tests

At the end of our results, we also performed several robustness checks to test how sensitive our findings were to changes in the initial assumptions and parameters. Specifically, we varied the training and testing time windows to assess whether the best-performing models remained consistent over different historical periods. We also increased transaction costs to evaluate the impact of higher trading expenses on performance metrics. Finally, we modified the optimization criteria used for hyperparameters selection, replacing the default metric with alternative evaluation measures, to observe whether this influenced the ranking and stability of the models. The detailed analyses of the robustness tests are available in Appendix E.

Main Takeaways

In summary, this chapter presented the empirical methodology used throughout thesis. We described the construction of the dataset based on the dynamic composition of the NASDAQ-100, the extraction of technical indicators and the resampling procedures. We then detailed the rolling walk-forward training process based on a 70%–10%–20% repartition, the model calibration steps, and the portfolio optimization approaches combining

different weighting schemes. Finally, we outlined the metrics used to evaluate both predictive and portfolio performance and how robustness tests were implemented to analyze the sensitivity of our results. Together, these components define a robust framework designed to test machine learning and ensemble strategies in a realistic, out-of-sample setting.

5 Results, Interpretations and Discussions

This chapter presents the main empirical findings of the study.We begin by evaluating the classification performance of each individual model using standard evaluation metrics. We then assess the performance of the ensemble models, starting with the naive threshold strategy. Following that, a dedicated section focuses on the threshold investigation strategy, illustrating how changes in this approach lead to shifts in performance results. The overall goal is to identify which individual and ensemble models generalize best over time and under varying market conditions.

We then evaluate the financial value of these predictions through portfolio construction. Several allocation strategies are tested, combined with different thresholding methods, to determine whether predictive signals can translate into superior out-of-sample returns. All results are benchmarked against the `NASDAQ-100` index, through a passive investment.

Finally, we highlight the best-performing configurations and discuss the broader patterns observed across models and strategies.

5.1 Classification performance

5.1.1 Individual Models

Let's remind all individual models used in this thesis in Table 5.1. Once all models are trained and tuned through the walk-forward validation scheme, we evaluate their classification performance across time. Rather than reporting only aggregated metrics, we provide a *boxplot* representation (Figure 5.2) of the four key evaluation metrics. Models are ordered by decreasing **median** accuracy, allowing for a fair comparison of their typical predictive performance, while preserving the temporal variation in their behavior. This graphical representation enables us to assess both the central tendency and the dispersion

of each model's performance over out-of-sample periods[1].

In addition, it's important to consider class imbalance in the initial dataset. Although the classes are not extremely unbalanced, the proportion of stocks outperforming the benchmark can vary significantly over time, as illustrated in Figure 5.1. For this reason, accuracy alone can be misleading, and complementary measures such as recall and precision are essential to get a comprehensive picture of predictive effectiveness.

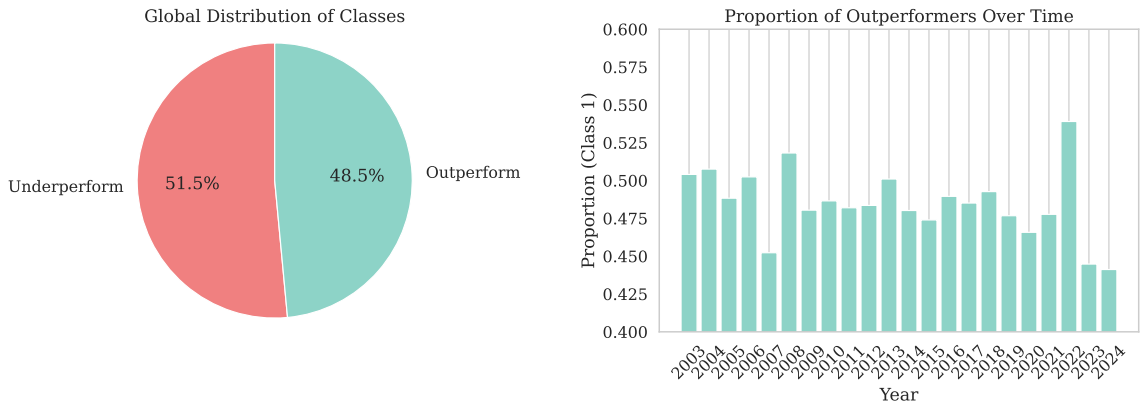


Figure 5.1: Distribution of target classes. *The left pie chart displays the global distribution of the binary target variable across the entire out-of-sample period (test set). The bar chart on the right shows the annual proportion of "Outperform" labels (class 1), revealing temporal variability in class imbalance across years.*

Model Family	Model Abbreviation
Penalized Logistic Regression	Ridge Lasso ElasticNet
Tree-Based Models	DecisionTree RandomForest GradientBoosting XGBoost
Kernel Methods	SVM
Instance-Based Learning	KNN
Neural Networks	MLP

Table 5.1: List of Individual Forecasting Models

Overall, the models exhibit performance in line with the expectations for financial prediction tasks, where signal-to-noise ratios are notoriously low. As shown, most mod-

[1]This approach is particularly relevant in the financial context, where model performance can vary considerably according to market conditions. Observed dispersion is therefore not a defect, but a characteristic that reflects the way in which models adapt, or fail to adapt.

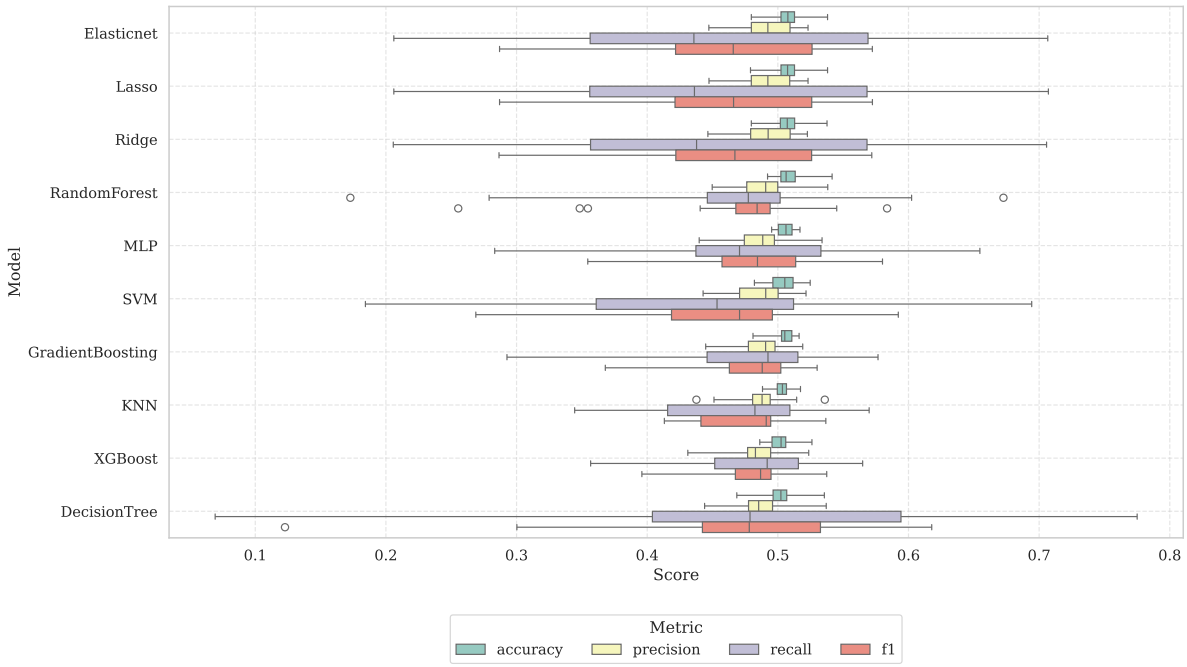


Figure 5.2: Boxplot of classification metrics *for each individual model. The models are evaluated over all out-of-sample periods using a walk-forward validation scheme. This representation highlights both the central tendency and the dispersion of predictive performance across models and time. Indeed, more information about the dispersion of the metrics in front of the years are available in Figure F.1. Moreover, the same data but ordered in decreasing order of the mean accuracy is available in Figure F.2.*

els consistently outperform the random baseline in terms of accuracy, which generally fluctuates around 50%. Our findings about accuracies are in line with the ones in the previous empirical studies of Simonis (2021) and Wolff and Echterling (2022). Notably, `ElasticNet`, `Lasso`, and `Ridge` top the list in terms of median accuracy. However, all of the three display greater dispersion, suggesting sensitivity to regime changes. In contrast, models such as `GradientBoosting`, `KNN`, `XGBoost` and the `MLP` neural network show tighter interquartile ranges across all metrics, indicative of more stable performance over time.

Moreover, it highlights that simpler linear models like `Lasso`, `Ridge`, and `ElasticNet` yield huge dispersion, yet they still achieve better median accuracy than more complex classifier such as neural network (`MLP`). Notably, the `DecisionTree` stands out as the worst performer, with its median scores across metrics barely exceeding random guessing but a wide recall interquartile range that reflects instability across time but also periods with high performance.

A consistent pattern across models is the imbalance between **precision** and **recall**: while precision tends to be around 50% and stable across the years recall remains lower and volatile, meaning many true outperformers are missed. This reflects a conservative selection bias, where models prefer to issue fewer, but more reliable, positive signals. Such behavior is typical in financial prediction tasks, where the risk of overtrading due to false

positives might penalize more than missing some true opportunities.

The **F1-score**, which harmonizes both metrics, provides a more balanced view of model performance. In this regard, `KNN`, `XGBoost`, `GradientBoosting` and `RandomForest` stand out, indicating they offer the best trade-off between precision and recall.

These observations reinforce the importance of using multiple metrics when evaluating financial prediction models. Models with moderate accuracy but high F1-score and low dispersion, such as `KNN` or `XGBoost`, can nevertheless provide very useful and actionable signals. This also justifies the use of ensemble methods to combine the complementary strengths of different classifiers, as one will see in the next section 5.1.2.

Interestingly, model rankings shift slightly when evaluated by mean accuracy rather than median (see F.2). `RandomForest` come out on top in terms of average accuracy performance, suggesting they may generalize more consistently over time, even if their central tendency is lower. However, whether these models, or any others, truly outperform a random baseline remains to be formally assessed (i.e., accuracy $\approx 50\%$, precision $\approx 48.5\%$[2], recall $\approx 50\%$, under balanced class distribution). To that end, we now turn to aggregated predictions and conduct statistical testing to determine whether the observed differences are **significant** across all out-of-sample periods. These results are available in Table 5.2.

The results indicate that all models, excepts `SVM, KNN, DecisonTree` and `XGBoost`, achieve accuracy and precision scores that are statistically significant at conventional levels based on one-sided binomial tests. The McNemar test[3] further supports the conclusion that these models generate prediction patterns that differ from random guessing.

Beyond individual performance, it is also relevant to assess how the models' predictions relate to each other when aggregated over time. It is summarized in Figure F.3. Regularized linear models (`Ridge`, `Lasso`, `ElasticNet`) consistently exhibit near-identical predictions, regardless of the metric used. Tree-based models and neural networks display more heterogeneous behavior, with weaker alignment both with each other and with linear models.This suggests that differences in prediction patterns are primarily explained by model families, rather than by randomness in the data or training process.

[2]Directly linked to the class 1 distribution in Figure 5.1

[3]The McNemar test assesses whether two models make different errors on the same data set, by testing whether the cases in which one succeeds and the other fails are balanced.

Model	Accuracy	Precision	Recall	F1	McNemar p-value (Sig.Level)
RandomForest	**50.89%*****	**49.13%*****	45.64%	47.32%	***
Elasticnet	50.89%***	49.10%***	43.76%	46.27%	***
Lasso	50.89%***	49.09%***	43.77%	46.28%	***
Ridge	50.89%***	49.09%***	43.76%	46.27%	***
MLP	50.58%***	48.84%*	47.91%	**48.38%**	***
GradientBoosting	50.52%**	48.78%*	47.41%	48.08%	**
SVM	50.42%**	48.58%	44.32%	46.35%	*
KNN	50.34%*	48.57%	46.84%	47.69%	-
DecisionTree	50.19%	48.45%	48.14%	48.29%	-
XGBoost	50.18%	48.44%	48.13%	48.29%	-

Table 5.2: Performance metrics and statistical tests for each model, *based on aggregated out-of-sample predictions. Significance stars ('***', '**', '*') indicate results that are statistically significant at the 0.1%, 1%, and 5% levels respectively, based on one-sided binomial tests against a random baseline classifier (accuracy = 50%, precision = 48.5%, recall = 50%). The final column reports McNemar test p-values evaluating whether each model's prediction pattern significantly differs from random guessing.*

5.1.2 Ensemble Models

This section starts by comparing the performance of ensemble models against each individual base model. This provides a first indication of whether the combination strategy can outperform standalone predictors.

We then turn to a broader comparison of all ensemble methods using firstly the **naive strategy** for threshold selection and also using *Simple Average Forecast* (`SA`), a simple equal-weight combination of the predictions, as a baseline to assess the relative performance of more advanced techniques such as trimmed means, loss-based weighting, and constrained optimization. All the combination models used are summarized in Table 5.3 This baseline-centered evaluation allows for a clearer interpretation of gains attributable to each combination strategy.

Importantly, one also emphasizes the role of the **classification threshold**, which is often overlooked in the ensemble learning literature that primarily focuses on regression settings for combination predictions. Unlike in regression, classification performance is highly sensitive to the choice of this threshold τ. In first comparisons, we applied a **naive** default threshold of 0.5 (i.e., assigning class 1 to linear combination results above 0.5), which is commonly used in binary classification. However, as we will show, exploring

alternative thresholding strategies can significantly affect the comparative performance of ensemble methods.

Combination methods	Acronym	Weights computation
Standard methods		
Simple average forecast	`SA`	$\overline{\mathbf{w}}^{E}$
Loss-based weighted average forecast	`IL`	$\overline{\mathbf{w}}^{IL}$
Constrained optimization	`CO`	$\psi(\mathbf{\Sigma})$
Trimmed averages		
Trimmed simple average	`T-SA-p`	$1/\text{card}(\mathcal{M}_p)$ if $i \in \mathcal{M}_p$, 0 otherwise
Trimmed simple average via cross-validation	`T-SA-CV`	$1/\text{card}(\mathcal{M}_p)$ if $i \in \mathcal{M}_p$, 0 otherwise
Trimmed-MCS simple average	`T-MCS-`α	$1/\text{card}(\mathcal{M}_\alpha)$ if $i \in \mathcal{M}_\alpha$, 0 otherwise
CO with shrinkage to $\overline{\mathbf{w}}^{E}$		
COP with L1 penalty	`COP_L1`^{E}	$\arg\min_{\mathbf{w}} \mathbf{w}^\top \Sigma \mathbf{w} + \delta \lVert \mathbf{w} - \overline{\mathbf{w}}^{E} \rVert_1$
COP with L2 penalty	`COP_L2`^{E}	$\arg\min_{\mathbf{w}} \mathbf{w}^\top \Sigma \mathbf{w} + \delta \lVert \mathbf{w} - \overline{\mathbf{w}}^{E} \rVert_2^2$
COP with EN penalty	`COP_EN`^{E}	$\arg\min_{\mathbf{w}} \mathbf{w}^\top \Sigma \mathbf{w} + \delta \left[\alpha \lVert \mathbf{w} - \overline{\mathbf{w}}^{E} \rVert_1 + (1-\alpha) \lVert \mathbf{w} - \overline{\mathbf{w}}^{E} \rVert_2^2\right]$
COP with EN penalty + Shrinkage to 0	`COP_EN`	$\arg\min_{\mathbf{w}} \mathbf{w}^\top \Sigma \mathbf{w} + \delta \left[\alpha \lVert \mathbf{w} \rVert_1 + (1-\alpha) \lVert \mathbf{w} \rVert_2^2\right]$
COP with cross-entropy	`COP_CE`^{E}	$\arg\min_{\mathbf{w}} \mathbf{w}^\top \Sigma \mathbf{w} + \delta \sum_{i=1}^{m} w_i \ln\left(\frac{w_i}{\overline{\mathbf{w}}_i^{E}}\right)$
CO shrinkage with reference $\overline{\mathbf{w}}^{E}$	`COS`^{E}	$\psi(\mathbf{\Sigma}^{E}_{\widehat{\lambda}^*})$
CO with shrinkage to $\overline{\mathbf{w}}^{IL}$		
COP with L1 penalty	`COP_L1`^{IL}	$\arg\min_{\mathbf{w}} \mathbf{w}^\top \Sigma \mathbf{w} + \delta \lVert \mathbf{w} - \overline{\mathbf{w}}^{IL} \rVert_1$
COP with L2 penalty	`COP_L2`^{IL}	$\arg\min_{\mathbf{w}} \mathbf{w}^\top \Sigma \mathbf{w} + \delta \lVert \mathbf{w} - \overline{\mathbf{w}}^{IL} \rVert_2^2$
COP with EN penalty	`COP_EN`^{IL}	$\arg\min_{\mathbf{w}} \mathbf{w}^\top \Sigma \mathbf{w} + \delta \left[\alpha \lVert \mathbf{w} - \overline{\mathbf{w}}^{IL} \rVert_1 + (1-\alpha) \lVert \mathbf{w} - \overline{\mathbf{w}}^{IL} \rVert_2^2\right]$
COP with cross-entropy	`COP_CE`^{IL}	$\arg\min_{\mathbf{w}} \mathbf{w}^\top \Sigma \mathbf{w} + \delta \sum_{i=1}^{m} w_i \ln\left(\frac{w_i}{\overline{\mathbf{w}}_i^{IL}}\right)$
CO shrinkage with reference $\overline{\mathbf{w}}^{IL}$	`COS`^{IL}	$\psi(\mathbf{\Sigma}^{E}_{\widehat{\lambda}^*})$

Table 5.3: List of the considered forecast combination methods. *Minimization is performed over $\mathcal{W}^+$, i.e., the non-negative weight space. For the theoretical framework behind all combination models, the reader may refer to Chapter 2. The notation card$(\cdot)$ denotes the cardinality of a set.* $\mathbf{\Sigma}$ denotes the covariance matrix of the errors, computed trough the 10% combination test set. For the elastic net divergence measure, we set $\alpha = 0.8$ based on Roccazzella et al. (2022). We estimate the penalties δ via 5-fold cross validation both.

The comparative plot in Figure 5.3 clearly shows that many ensemble methods outperform the simple average (`SA`) model in terms of accuracy. Surprisingly (or perhaps not), in terms of F1-score, the `SA` method remains one of the top-performing combination techniques under the naive threshold $\tau = 0.5$. This aligns with findings in the literature suggesting that equally weighted averages often perform as well as, or even better than, more sophisticated methods depending the metric examined (DeMiguel et al., 2009; Goyal & Welch, 2008).

This robustness may be partly due to the limited sample size used for model combination: only about 10% of the sliding window is allocated for weight estimation, corresponding

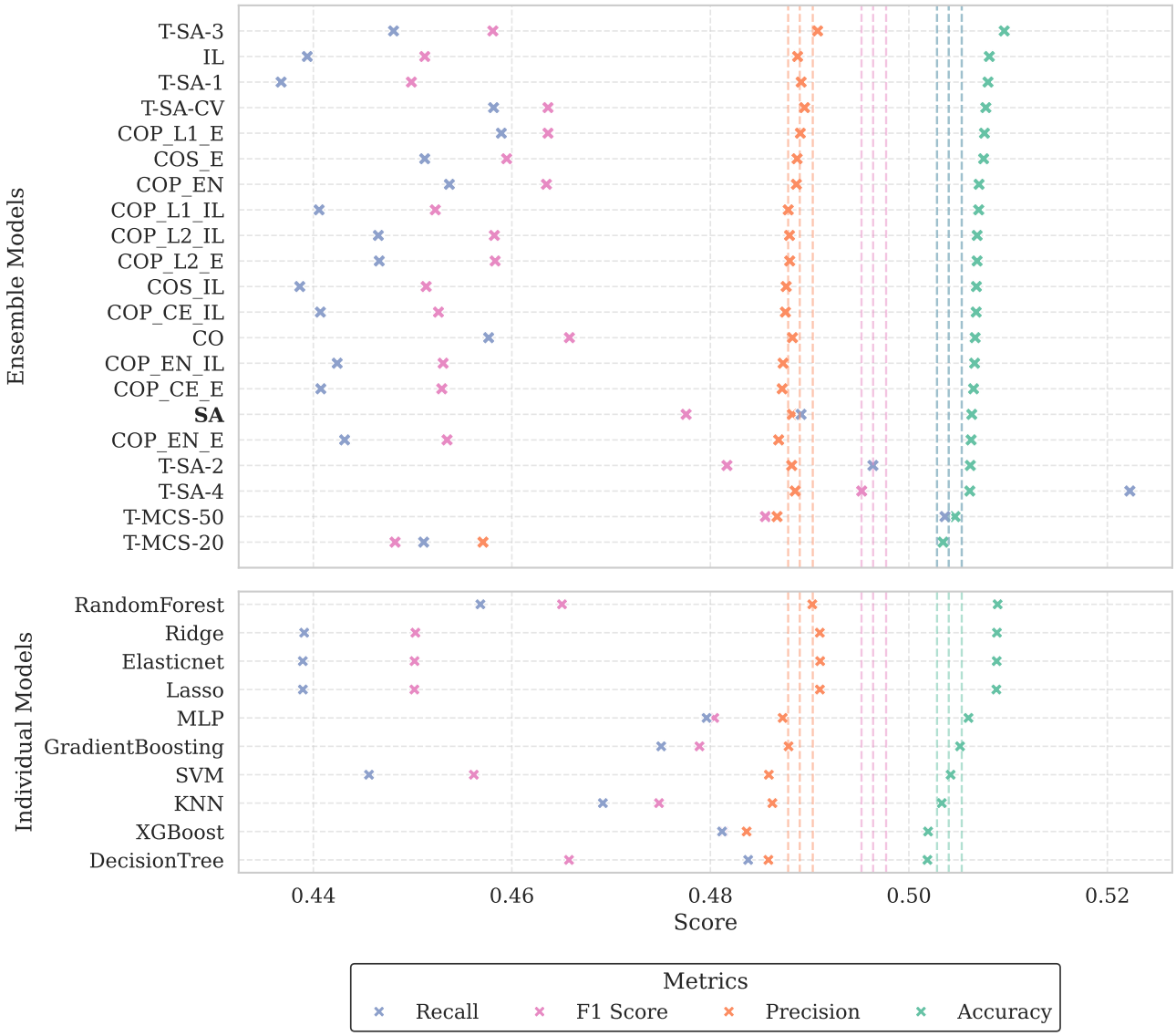


Figure 5.3: Performance of ensemble forecasts. *This plot shows the average performance of all considered ensemble combination methods, evaluated on the aggregated predictions over all walkforward test sets. A* ***fixed naive classification threshold*** *of $\tau = 0.5$ is used. Models are sorted in ascending order based on their accuracy. The dashed vertical lines represent statistical significance thresholds at the 5%, 1%, and 0.1% levels respectively from the left to the right, calculated under the null hypothesis of a random classifier (accuracy = recall = 50%, precision = 48.5%). Note that recall and accuracy share the same statistical threshold in this setting. The SA model is included and serves as a comparative reference, not as a baseline. Individual models performance are also included as a first comparative reference. All detailed metrics are available in Table 5.4*

to around 30 observations per test fold. In such conditions, complex weighting schemes are prone to overfitting or underfitting.

Compared to individual models, `SA` also performs remarkably well, ranking between `Lasso` and `MLP` in terms of accuracy. Interestingly, models like `ElasticNet`, `Lasso`, and

`Ridge` achieve high precision individually, but suffer from very low recall and F1 scores. In contrast, almost all combination models improve both recall and F1 score over these individual models, with one exception: `T-SA-1`, which fails to enhance these metrics.

Model rankings also shift depending on the metric used. For instance, `T-SA-4` does not achieve the highest accuracy, but stands out as the only model to exceed statistical significance threshold (5%, 1%, or 0.1%) across all four metrics, indicating strong balance and robustness. The reader may refer to Figure F.8, F.9, F.10 for other metrics ranking visualizations.

From a statistical perspective, all ensemble models significantly outperform the random baseline in terms of accuracy, capturing the strengths of the best individual models. Precision seems to be more difficult to improve; almost half of ensemble models still achieve a significant level on this measure, although they often lag behind the best individual Machine Learning classifiers in terms of precision.

Overall, a consistent trend emerges: ensemble models tend to trade precision for improved recall, an advantageous shift when false negatives are more critical. This reflects a common trade-off in ensemble learning: gains in recall and generalization may come at the expense of sharper class boundary precision.

To complement this analysis, one now turns to a heatmap comparing each ensemble model directly against the `SA` baseline in Figure 5.4. This visual enables a metric-by-metric inspection and highlights which models most consistently outperform the equal-weight ensemble method.

Overall, most ensemble models struggle to outperform the `SA` benchmark in terms of recall, underlining the surprisingly strong recall of the simple average when used with a naive threshold. Nonetheless, a few models manage to exceed this baseline while maintaining solid results on other metrics such as accuracy and F1-score. One notable exception is `T-MCS-50`, which, although it achieves a clear improvement in recall, does so at the expense of a significant drop in both precision and F1-score.

To complement the analysis of aggregated scores, Appendix F, Figure F.6 shows the year-by-year performance distribution of each ensemble model. This boxplot reveals that, despite strong average results, many methods display substantial variability across walk-forward periods, especially in recall and F1-score. Such dispersion underscores the importance of assessing temporal stability alongside point estimates when evaluating models. Additionally, we find that ensemble predictions are moderately correlated (typically above 0.6), except for the trimmed Model Confidence Set (`T-MCS-`α), which shows lower correlations around 0.45 (see Figure F.5).

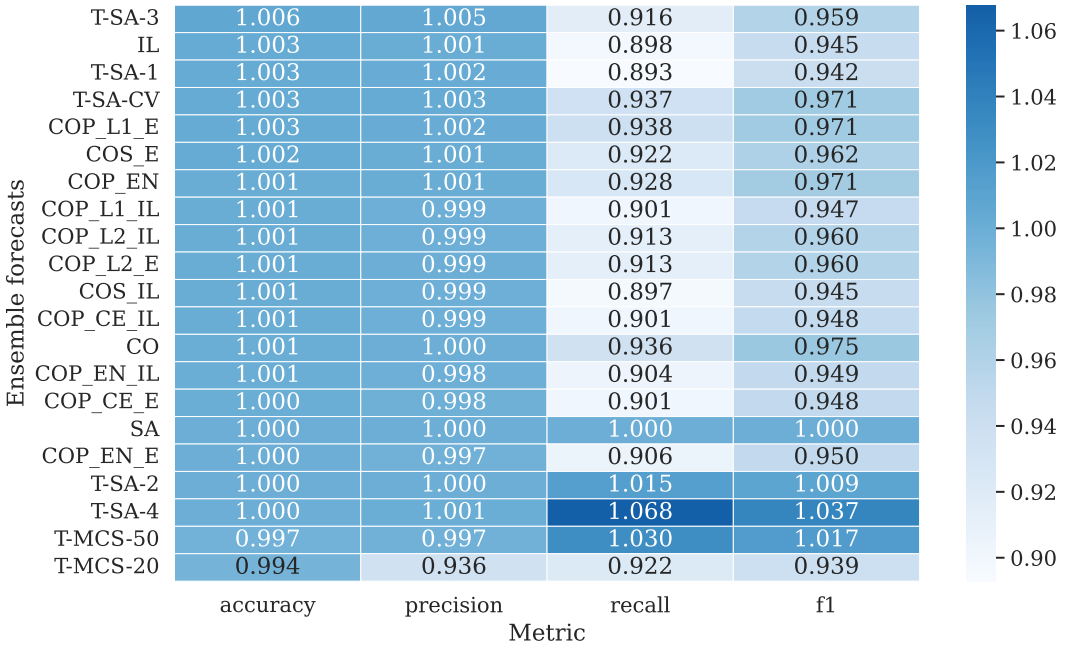


Figure 5.4: Relative performance of ensemble forecasts compared to the SA baseline. *This heatmap displays the average relative score of each ensemble model compared to the simple average forecast (`SA`), A value above 1 indicates that the model outperforms SA on the given metric. All results are computed using a fixed threshold $\tau = 0.5$ over the aggregated predictions from all walk-forward test sets.*

Weights Investigation

To gain an even deeper understanding of the behavior of the ensemble forecasting models, it is relevant to investigate how the different base classifiers are weighted within these combinations. Although some information is inevitably lost due to the need to average weights across multiple years or models, this analysis can still reveal interesting structural patterns. It is also important to note that weight importance does not necessarily correlate directly with final model performance, as the classification threshold is selected only after the weights have been applied. Therefore, the interpretation should remain exploratory rather than strictly causal.

Figure 5.5 shows which base models most often receive the highest weight in ensembles each year. `RandomForest`, `DecisionTree`, and `KNN` frequently dominate. While the predominence of `RandomForest` is expected given its high average accuracy, the frequent dominance of `DecisionTree` is more surprising, as it generally ranks among the weakest models overall (see Figure 5.2). This may be because `DecisionTree` performs well on the small validation subsets (10% per fold) used to fit weights, or because its performance varies widely, occasionally reaching high scores that the combination schemes capture. Conversely, `ElasticNet` is never dominant in any ensemble-year combination, which highlights how base models with stable but moderate performance may be consistently under weighted.

Model	Accuracy	Precision	Recall	F1 score	McNemar
T-SA-3	**50.97**%***	**49.17**%***	44.67%	46.82%	***
IL	50.82%***	49.01%***	43.78%	46.25%	***
T-SA-1	50.88%***	48.99%**	43.51%	46.09%	**
T-SA-CV	50.78%***	49.01%***	45.69%	47.29%	***
COP_L1^{E}	50.77%***	48.99%**	45.69%	47.29%	***
COS^{E}	50.76%***	48.97%**	44.90%	46.85%	**
COP_L1^{IL}	50.71%***	48.89%*	43.87%	46.24%	***
COP_EN	50.71%***	48.92%**	45.22%	47.00%	***
COS^{IL}	50.70%***	48.86% *	43.67%	46.12%	***
COP_L2^{E}	50.69%***	48.89%*	44.54%	46.61%	***
COP_L2^{IL}	50.69%***	48.88%*	44.53%	46.61%	**
COP_CE^{IL}	50.69%***	48.86%*	43.89%	46.25%	**
COP_EN^{IL}	50.67%***	48.85%*	44.06%	46.33%	**
CO	50.67%***	48.88%*	45.63%	47.20%	**
COP_CE^{E}	50.66%***	48.83%*	43.90%	46.23%	**
COP_EN^{E}	50.64%***	48.81%*	44.16%	46.37%	/
SA	50.64%***	48.92%**	48.77%	48.84%	**
T-SA-2	50.63%***	48.93%**	49.50%	49.21%	***
T-SA-4	50.61%***	48.97%**	**52.09**%***	**50.48**%**	*
T-MCS-50	50.47%**	48.79%*	50.02%	49.39%	**
T-MCS-20	50.35%*	48.51%	44.61%	46.48%	/

Table 5.4: Performance metrics and statistical tests for each combination model, *based on aggregated out-of-sample predictions. Statistical significance stars ('***', '**', '*') indicate results that are statistically significant at the 0.1%, 1%, and 5% levels respectively, based on one-sided binomial tests against a random baseline classifier (accuracy = 50%, precision = 48.5%, recall = 50%). The final column reports McNemar test significance, evaluating whether each model's prediction pattern significantly differs from random guessing. We observe that only the* `COP_EN`^{E} *and* `T-MCS-20` *models are not significant in terms of McNemar test. A* ***fixed naive classification threshold*** *of* $\tau = 0.5$ *is used.*

The averaged perspective in Figure 5.6 complements this analysis by showing the mean weights attributed to each base model by each ensemble strategy across all years. We can clearly see that `ElasticNet` receives always zero weight in the `T-MCS-50` and `T-MCS-20` ensembles, despite achieving strong overall performance. This reinforces the idea that its predictive errors may concentrate in specific parts of the validation set, which then strongly penalize its weight in the allocation process. Interestingly, `Ridge`, `Lasso`, and `ElasticNet` (all regularized linear models) are generally assigned lower weights. This is consistent with their high pairwise correlation (see Figure F.3), suggesting that the ensemble strategies detect redundancy in these forecasts and diversify away from them. It also explains the few dominant positions that one observes in Figure 5.5.

Figure F.11 further examines the relationship between average model weights and standalone predictive performance across several metrics. A positive linear trend appears clearly between average weight and recall, suggesting that ensembles might implicitly favor models that tend to capture more positives. Somewhat counterintuitively, we also

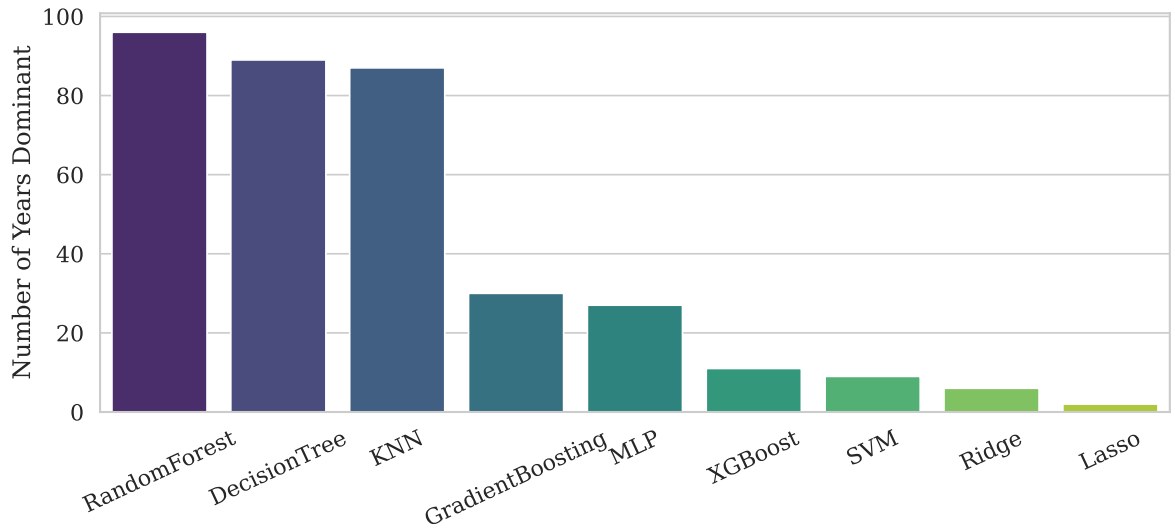


Figure 5.5: Frequency of Top-Weighted Models in Ensembles. *This plot shows how often each individual model received the highest weight (labeled as dominant) across all ensemble combinations and years.*

observe a slightly negative correlation between weight and accuracy, although ensemble models tend to perform better overall (looking at the aggregated predictions) in terms of accuracy and worse in recall. This reveals a trade-off inherent in the way validation sets are constructed, only 10% of the data is used for weight fitting in each fold. This limited view may skew weight allocations, especially when paired with walk-forward validation and small sample sizes. Still, this bias seems partially corrected when evaluated out-of-sample, where ensemble performance is generally more balanced.

Finally, Figure 5.7 visualizes the year-over-year evolution of the average weight given to each base classifier across all ensemble strategies. It highlights how different models gain or lose importance depending on the year, likely reflecting shifts in market dynamics or data structure. The fact that dominant models vary so frequently supports the idea that the ensembles are successfully adapting to changing patterns, capturing temporal dependencies and model relevance over time.

Threshold Investigation for Ensemble Models

In our classification problems, the choice of the threshold τ used to convert the continuous output of ensemble models into binary class labels has a direct and often critical impact on performance (see Section 2.3 for more details). In our context, although the combined predictions are not necessarily probabilistic in nature (as they result from various linear combinations of binary base predictions), they still yield continuous values in the interval $[0, 1]$, which can be interpreted as scores reflecting the strength of support for class 1.

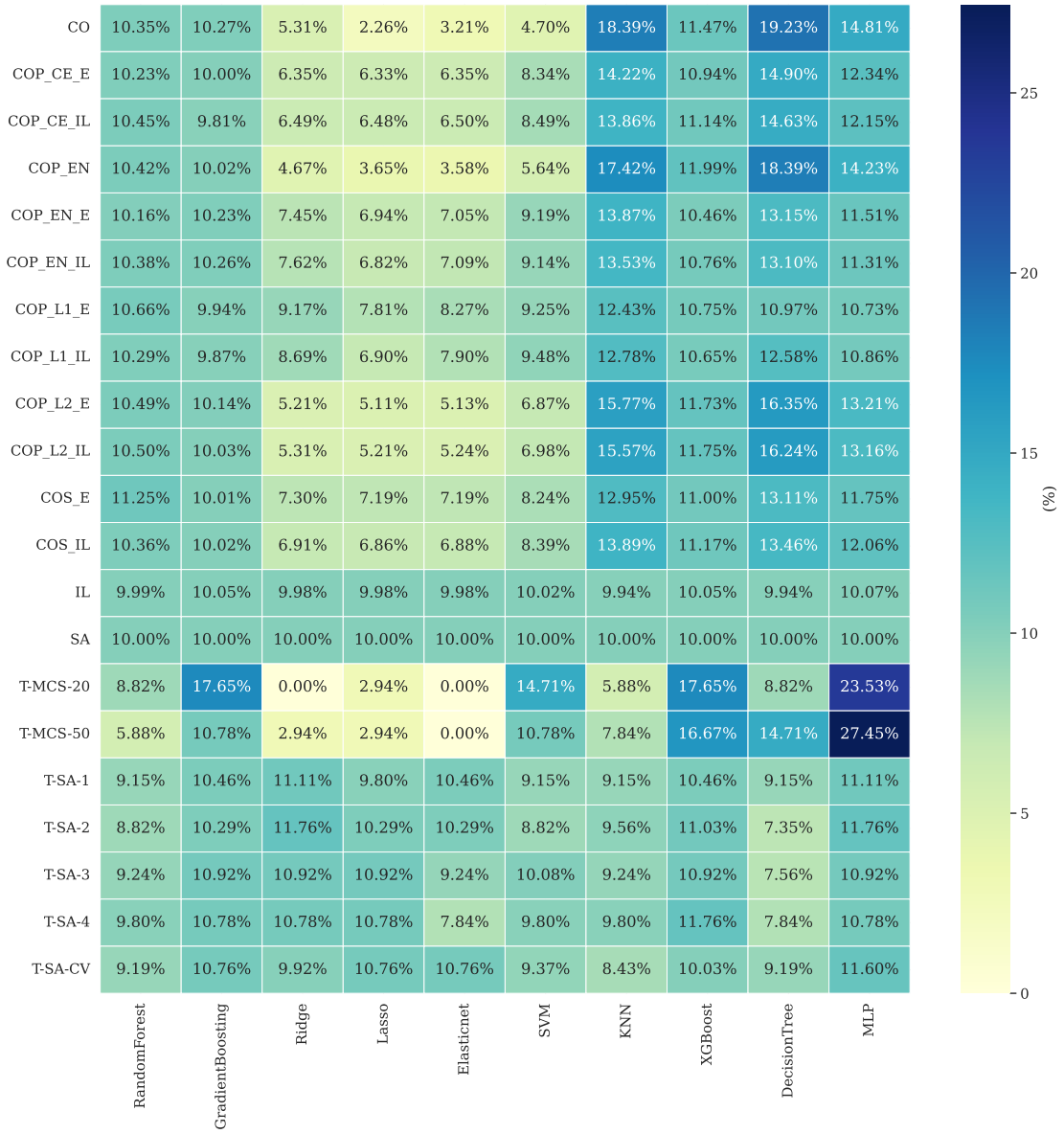


Figure 5.6: Average Weights Assigned to Individual Classifiers by Ensemble Models *This heatmap displays the average percentage of weight allocated to each base classifier by each ensemble strategy across all years. It highlights how some ensemble models tend to consistently favor certain individual learners, such as while others distribute weights more evenly. This view allows for a direct comparison of allocation behavior across ensemble types. A* ***fixed naive classification threshold*** *of* $\tau = 0.5$ *is used.*

As detailed in Section 2.3, instead of relying on the naïve default threshold of $\tau = 0.5$, which may be inappropriate in financial prediction, we evaluate three alternative strategies: optimized (`OPTI`), cross-validated (`CV`), and quantile-based (`QTL`) thresholds. These methods are designed to better adapt the decision boundary to the data distribution and the specific performance objectives (in terms of dedicated metrics).

This analysis aims to quantify the marginal improvement achievable through more refined thresholding rules and to identify which ensemble methods are most sensitive to τ

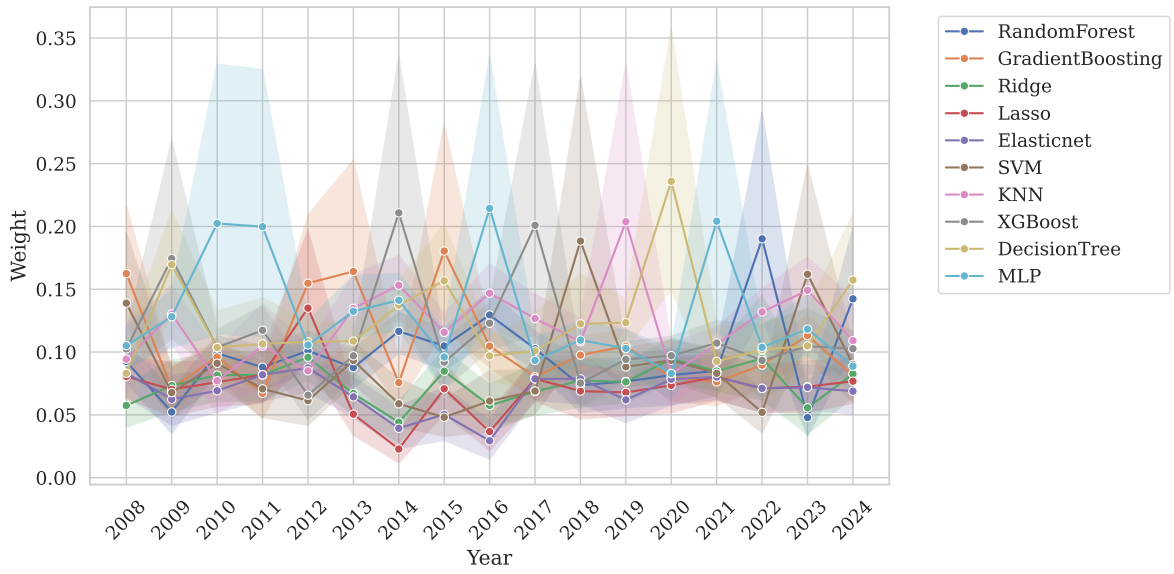


Figure 5.7: Yearly Distribution of Average Ensemble Weights with Confidence Intervals

This line plot shows the average weight assigned to each individual model by ensemble methods for every year, along with confidence intervals reflecting inter-method variability. The figure highlights the strong year-to-year fluctuations in model importance: different classifiers dominate in different years, suggesting that no single model consistently outperforms across time. This pattern likely reflects the low signal-to-noise ratio of financial markets, where model performance can vary greatly depending on shifting market conditions and structural changes in the data.

adjustment.

These findings confirm that threshold selection is far from a trivial detail, it is, in fact, a key lever for improving classification outcomes in ensemble settings. While the naive threshold of $\tau = 0.5$ offers a convenient baseline, our results demonstrate that more informed strategies can lead to meaningful performance gains. Interestingly, each thresholding method carries its own behavioral signature observable in Figure 5.9: quantile-based thresholds consistently favor high-confidence predictions, OPT offers targeted optimization tailored to the metric of interest and CV exhibits greater robustness through distributional smoothing.

Moreover, we observe that `OPT`, `QTL`, and `CV` generally outperform the naive strategy in terms of accuracy and precision. However, this improvement often comes at the cost of significantly reduced recall, and consequently, a lower F1 score. This suggests a tendency to favor high-confidence true positives while neglecting false negatives, which, in a financial context, translates to missing potentially profitable opportunities (i.e., overlooked outperformers).

It is also important to note that both `CV` and `OPT` are intrinsically dependent on the performance metric chosen during optimization. As shown in Figure F.12, optimizing for recall can indeed yield very high recall values, but at the expense of precision and overall

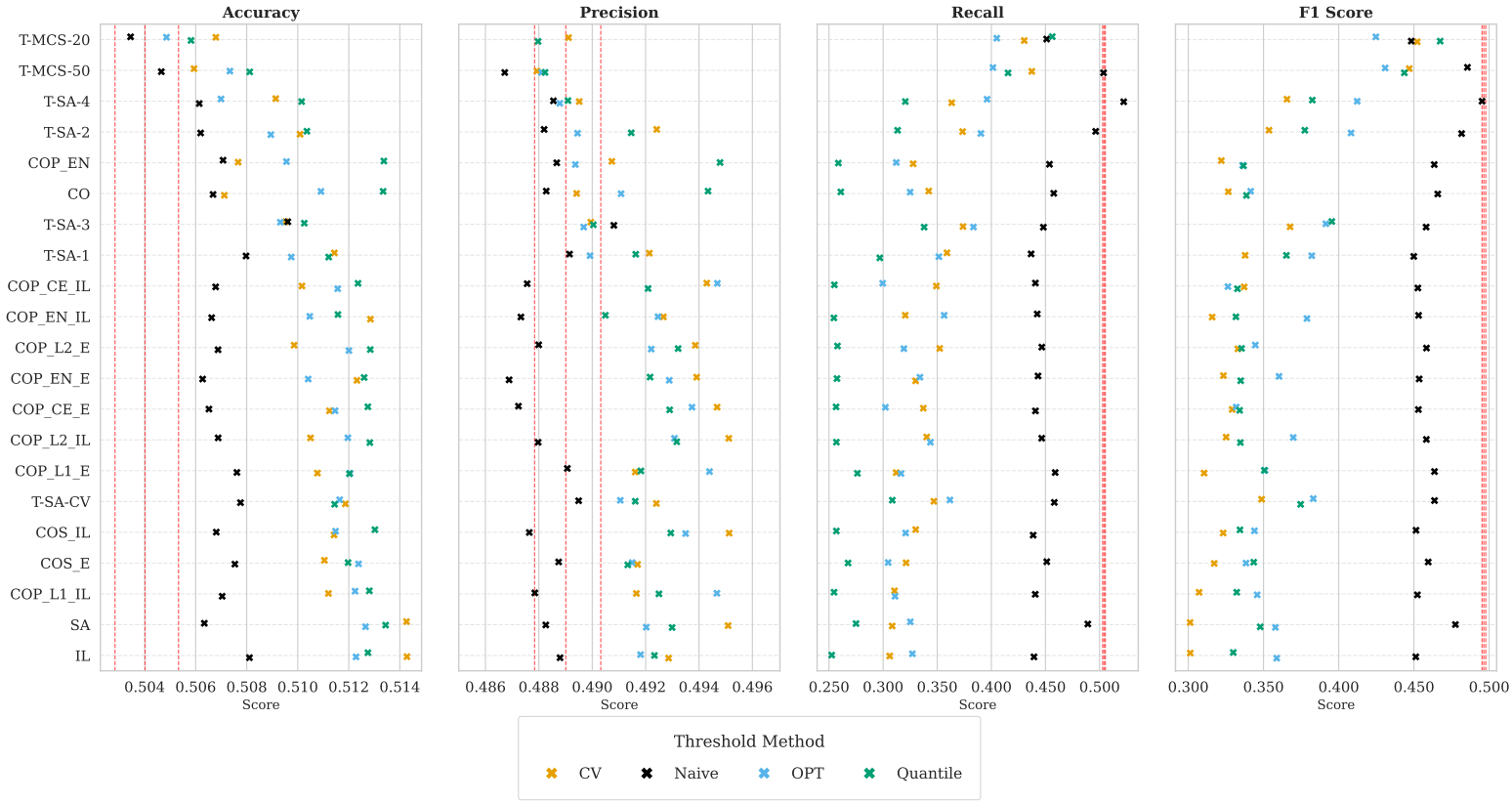


Figure 5.8: Performance of Thresholding Strategies Across Ensemble Models and Evaluation Metrics
This figure compares the performance of four thresholding strategies across all ensemble models and four evaluation metrics. Each cross represents the score obtained by a given ensemble method under a specific thresholding approach.The dashed vertical lines represent statistical significance thresholds at the 5%, 1%, and 0.1% levels (from left to right), computed using a one-sided binomial test under the null hypothesis of a random classifier (i.e., accuracy = recall = 50%, precision = 48.5%). This visualization highlights the potential performance gains from fine-tuning the decision threshold and illustrates how different metrics and ensemble methods respond differently to threshold adjustments. Comparative tables with detailed metric are available in Table G.2 and Table G.3. `CV` *and* `OPTI` *without subscript refer to an accuracy optimization.*

accuracy. This trade-off emphasizes the importance of aligning thresholding strategies with business objectives and evaluation priorities in our portfolios.

In sum, the threshold investigation enriches our understanding of ensemble calibration and highlights a new dimension that is both impactful and often overlooked in financial classification pipelines.

5.2 Portfolio Evaluation and Backtesting

While the previous sections focused on the predictive performance of individual and ensemble classifiers, we now turn our attention to their practical value through the construction and evaluation of portfolios. The central question is whether **better classification translates into superior financial performance**. To answer this question, we explore various

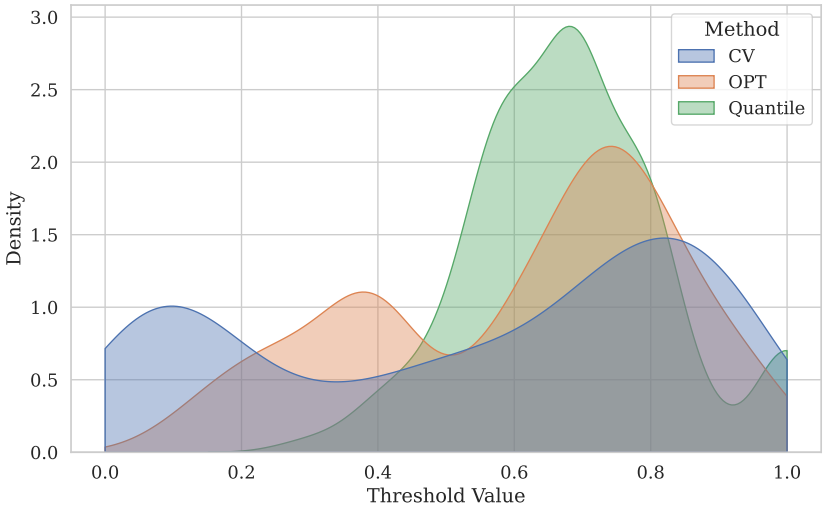


Figure 5.9: Distribution of aggregated threshold values. *The `QTL` method (75th percentile) concentrates around 0.7, reflecting a systematically high threshold selection. The `OPT` method shows higher density near 0.4 and 0.75, with limited presence at the extremes, indicating a preference for intermediate thresholds. In contrast, the `CV` method yields a more uniform distribution, with notable peaks at the extreme values, indicating greater variability in selected thresholds. Obviously, the naive method is not represented here since its distribution is constant.*

portfolio construction techniques, ranging from simple equal-weighted (`EW`) allocations to more sophisticated mean-variance optimization methods (`MV`), including variants based on shrinkage (`Shrinked_MV`) and an hybrid method using eigenvalue filtering with shrinkage (`FILTERED`). Each strategy is applied consistently to all forecast combination models and thresholding approaches to ensure comparability. All portfolios are benchmarked against the Nasdaq-100 index to assess whether model-based allocations can consistently outperform the market.

In total, the analysis involves 21 forecast combination models, 10 individual classifiers, 6 distinct thresholding strategies[4], and 4 portfolio construction methods. This yields a total of $21 \times 6 \times 4 + 10 \times 4 = 544$ distinct portfolio trajectories. To facilitate interpretation and improve readability, we focus primarily on the top 15 portfolios based on Sharpe Ratio. Additional analyses are also presented to evaluate whether certain combination methods, thresholding schemes, or portfolio construction techniques systematically lead to superior performance compared to the `NASDAQ-100` index [5] However, all the Tables with detailed results are available for the reader in Appendix G. Interestingly, the portfolios generated by `MV` and shrinkage-based methods are often very similar, likely because non-negativity constraints in `MV` act as an implicit form of regularization, much like shrinkage itself (Ma

[4]Depending the metric we choose to optimize, generally accuracy but it can be expand to precision or recall for instance

[5]While alternative baselines such as an equally-weighted random selection of stocks could have been considered, our objective here is to assess whether model-driven strategies can outperform the market itself, a feat reportedly missed by over 90% of traders, as discussed in the Introduction.

& Jagannathan, 2003).

Firstly, Figure 5.10 displays the cumulative returns obtained by the individual models under each of the four portfolio allocation strategies. It clearly illustrates that relying solely on individual machine learning models is not enough to outperform the benchmark. This reinforces the financial relevance of combining forecasts through ensemble methods, which aim to aggregate complementary predictive signals and reduce the limitations of any single model, as it is observed hereafter.

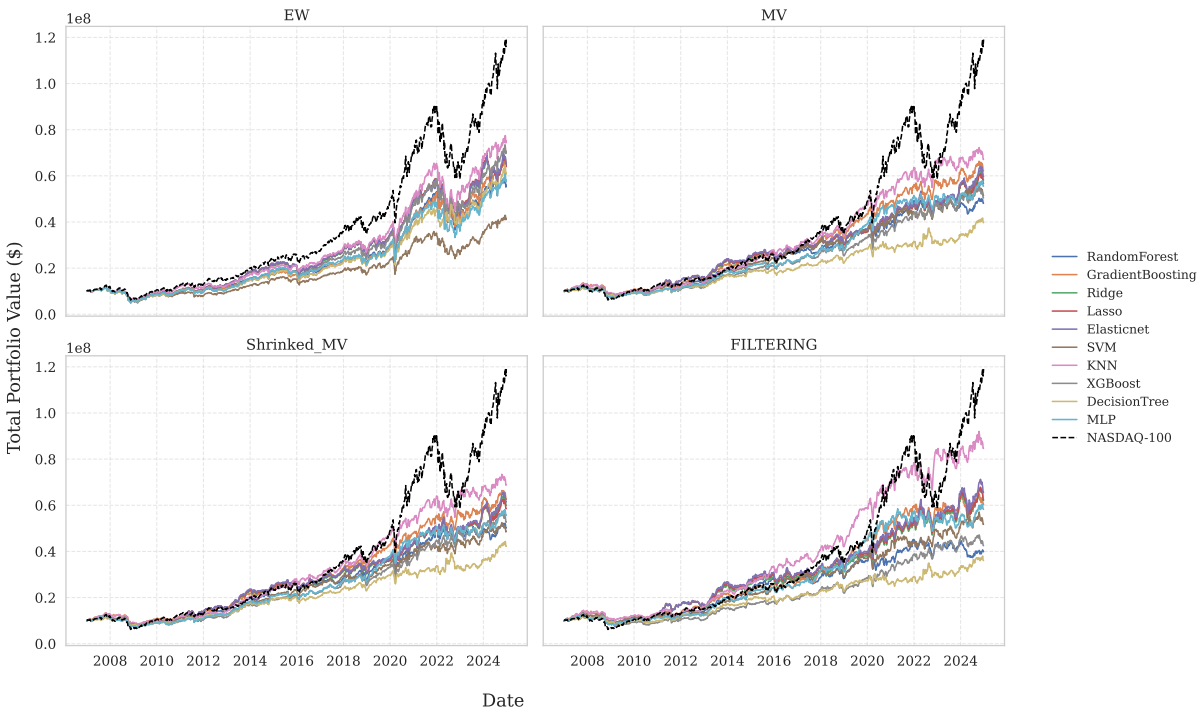


Figure 5.10: Cumulative returns of individual classifier portfolios by allocation strategy *Each subplot shows the evolution of the portfolio value over time for one of the four allocation methods: Equal Weight (EW), Mean-Variance Optimization (MV), Shrinked MV, and Filtering. All individual models are included in each allocation scheme. Results highlight that no individual model under any allocation strategy is able to consistently outperform the Nasdaq-100 benchmark (black dashed line), further reinforcing the need for ensemble-based forecasts to extract financial value. To reminder threshold techniques without any subscripts take the accuracy metric as reference.*

Secondly, looking at the top 15 performing portfolios in Table 5.5, one immediately sees that all of them are generated from ensemble forecasting models. Interestingly, the equal-weighted (`EW`) strategy is the only portfolio construction method that does not appear in this top list, suggesting it may lack the flexibility or responsiveness required to fully exploit the information contained in forecast combinations. Among the optimization methods, the filtering-based allocation clearly dominates, appearing in the majority of top-ranked portfolios.

Portfolio	Ensemble Method	Threshold	SR	APY [%]	Vol. [%]	DD [%]	VaR [%]	CVaR	WR [%]
FILT.	**T-SA-1**	**OPTI$_{PREC}$**	**1.11**	**19.29**	**16.26**	**-29.30**	**-3.19**	**-5.02**	**59.59**
FILT.	COP_EN^{E}	OPTI$_{PREC}$	1.10	19.07	16.24	-29.68	-3.17	-4.98	58.96
FILT.	SA	OPTI$_{PREC}$	1.08	18.34	15.86	-29.64	-3.07	-4.71	57.14
FILT.	IL	OPTI$_{PREC}$	1.08	18.53	16.10	-30.19	-3.16	-4.81	57.36
FILT.	COP_EN^{IL}	OPTI$_{PREC}$	1.00	16.78	15.84	-30.32	-3.24	-4.87	57.78
FILT.	COP_L1^{IL}	OPTI$_{PREC}$	1.00	17.25	16.34	-31.17	-3.42	-4.96	57.68
FILT.	T-SA-3	OPTI$_{PREC}$	0.99	16.75	16.00	-38.91	-3.18	-5.02	58.21
FILT.	T-SA-CV	OPTI$_{PREC}$	0.97	16.80	16.47	-38.90	-3.36	-5.18	58.53
FILT.	IL	OPTI$_{ACC}$	0.97	17.08	16.81	-35.37	-3.28	-5.30	58.21
MV	T-SA-1	OPTI$_{PREC}$	0.95	15.74	15.59	-36.19	-3.09	-5.06	59.17
FILT.	T-SA-3	NAIVE	0.95	15.60	15.54	-38.12	-3.06	-4.95	58.42
FILT.	SA	OPTI$_{ACC}$	0.95	16.73	16.79	-34.69	-3.35	-5.29	57.68
FILT.	T-SA-2	NAIVE	0.93	14.72	14.99	-31.61	-3.03	-4.71	59.06
SHRINK.	T-SA-1	OPTI$_{PREC}$	0.93	15.24	15.61	-36.56	-3.26	-5.16	59.81
MV	COP_EN^{E}	OPTI$_{PREC}$	0.92	15.25	15.68	-37.43	-3.32	-5.08	59.17
INDEX NASDAQ-100			***0.74***	***14.72***	***19.7***	***-51.53***	***-4.66***	***-6.79***	***61.73***

Table 5.5: Top 15 of portfolios ranking by their Sharpe Ratio. *We observe that the first portfolio in term of sharpe ratio is also the one that has the highest APY.*

In terms of ensemble methods, the `T-SA-1` strategy stands out as the TOP-1 portfolio, especially when used with a classification threshold optimized based on precision ($\texttt{OPT}_{\texttt{PREC}}$). Remarkably, the simple average combination (`SA`) still ranks third overall (an observation that aligns with previous findings in the literature suggesting that equal-weighted combinations often perform surprisingly well despite their simplicity (Simonis, 2021)). This once again highlights that more complex weighting schemes do not always lead to better financial outcomes. The `COP_EN` method completes the top three, offering a more refined alternative through shrinkage-based constrained optimization for the ensemble strategy.

Beyond the combination and allocation methods, these results also underscore the crucial role played by the classification threshold. Among the top 8 portfolios, all rely on thresholds optimized according to precision. By contrast, when using a naive threshold of $\tau = 0.5$, the best Sharpe Ratio achieved is only 0.95 (compared to 1.11 when the threshold is optimized) representing a relative improvement of over 17%. This alone justifies deeper attention to threshold calibration in financial classification. One may wonder whether this performance differential could be largely absorbed by transaction costs; this question is further explored in Appendix B, where detailed cost analysis is provided.

Moving beyond individual top-performing portfolios, it is also interesting to evaluate which portfolio construction or thresholding methods tend to outperform the benchmark more frequently across the entire universe of tested portfolios. Observing Figure 5.11 and starting with the Sharpe Ratio, we observe that none of the portfolios using the `EW` allocation method manage to exceed the benchmark consistently. In contrast, portfolios based on the filtering allocation method show much stronger results, with nearly 40% outperforming the NASDAQ-100 in terms of Sharpe ratio.

What's more, we clearly see the benefits of the mean variance strategy to allocate our portfolios. Indeed, every portfolio build with those methods (i.e., `Filtering`, `MV` and `Shrinked_MV`) beat the index in terms of annualized volatility.

However, regarding maximum draw-down, the trend is reversed: `EW`-based portfolios are those that outperform the index most frequently. This may be due to the fact that market indices like the NASDAQ-100 are weighted by market capitalization, which can amplify losses during crises, while equal-weighted strategies diversify risk more uniformly. Although `EW` is less effective in terms of Sharpe ratio and APY, it actually performs best across other risk metrics such as win rate and average gain. This suggests that while `EW` may not generate superior risk-adjusted returns, it tends to offer more robustness in turbulent market conditions.

Figure 5.12 presents a similar analysis, this time focusing on thresholding strategies. While the naive threshold does not yield the highest Sharpe Ratio, it is the most frequently associated with portfolios that outperform the index overall in this metric. However, in terms of APY, the threshold optimized on precision clearly leads, producing the highest proportion of outperforming portfolios. Interestingly, thresholding choice appears to have limited influence on metrics such as maximum drawdown and average loss. That said, optimizing thresholds for recall leads, as expected, to portfolios with higher recall, and consequently slightly improved win rates. Notably, this is the only thresholding strategy that manages to beat the market on win rate, reinforcing the earlier finding that it captures more positive signals, at high costs: lower precision and so lower sharpe ratio and APY, leading to an average gain that never beats the benchmark.

To conclude those results, we perform another similar analysis but focusing on ensemble methods in Figure 5.13. We observe that the ensemble method that tends to beat the benchmark most often is also the `T-SA-1`. Furthermore, deeper analysis are performed such as investigation on portfolio maximum size in Appendix B.

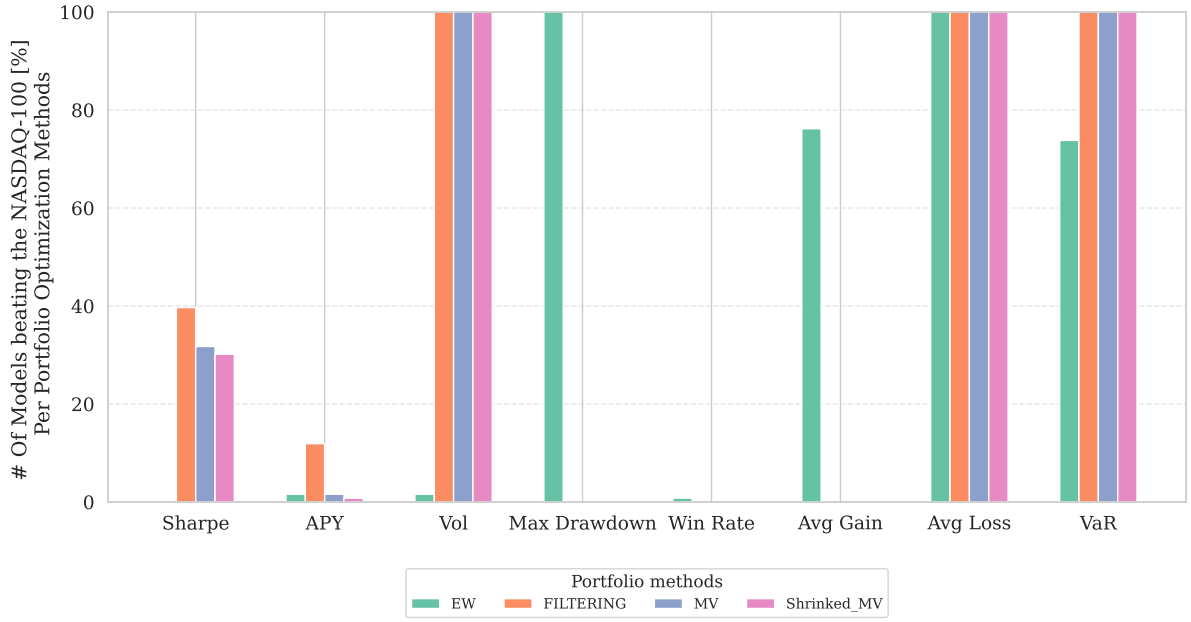


Figure 5.11: Percentage of portfolios beating the Nasdaq-100 by allocation method
This figure compares the percentage of portfolios that outperform the Nasdaq-100 on each back-testing metric, grouped by portfolio optimization method. Filtering clearly stands out for Sharpe Ratio and APY, while Equal Weight performs better on downside-risk measures such as maximum drawdown, average loss, and CVaR. This highlights a trade-off between maximizing returns and minimizing losses across allocation methods.

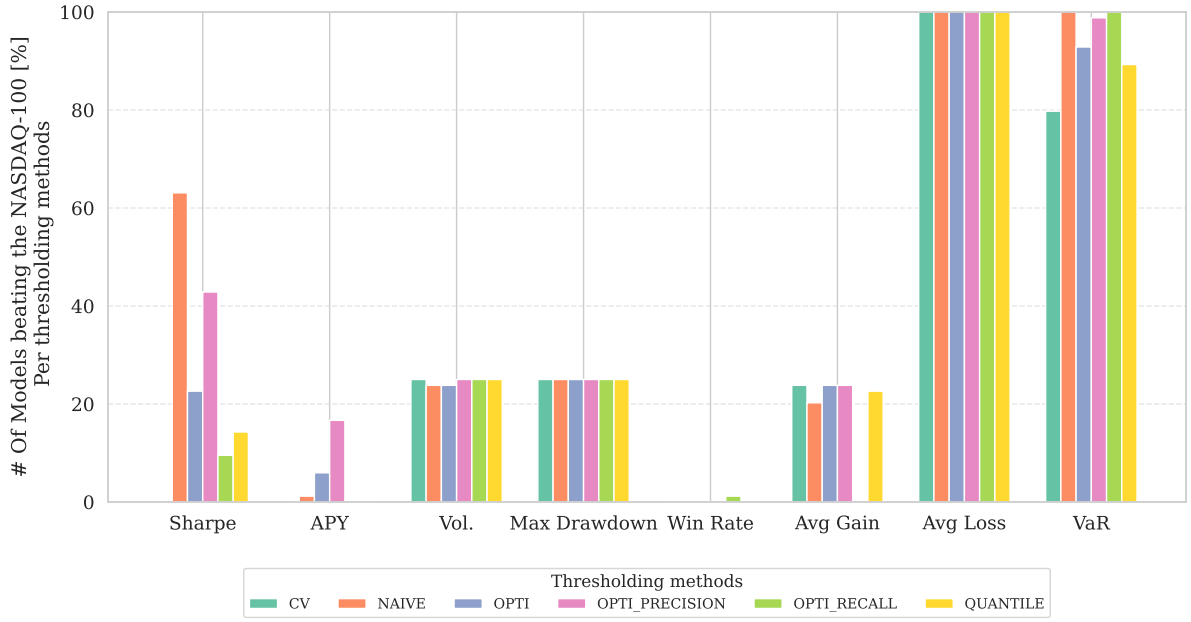


Figure 5.12: Percentage of portfolios beating the Nasdaq-100 by thresholding method
This chart shows the proportion of portfolios that outperform the benchmark according to each evaluation metric, grouped by thresholding strategy. Thresholds optimized on precision (`OPT_PREC`) tend to yield the highest Sharpe and APY values, while naive and recall-optimized thresholds show better win rate and CVaR. These results confirm that threshold calibration plays a key role in financial performance, especially when optimizing for specific risk-return trade-offs. `CV` and `OPTI` without h subscript refer to an accuracy optimization.

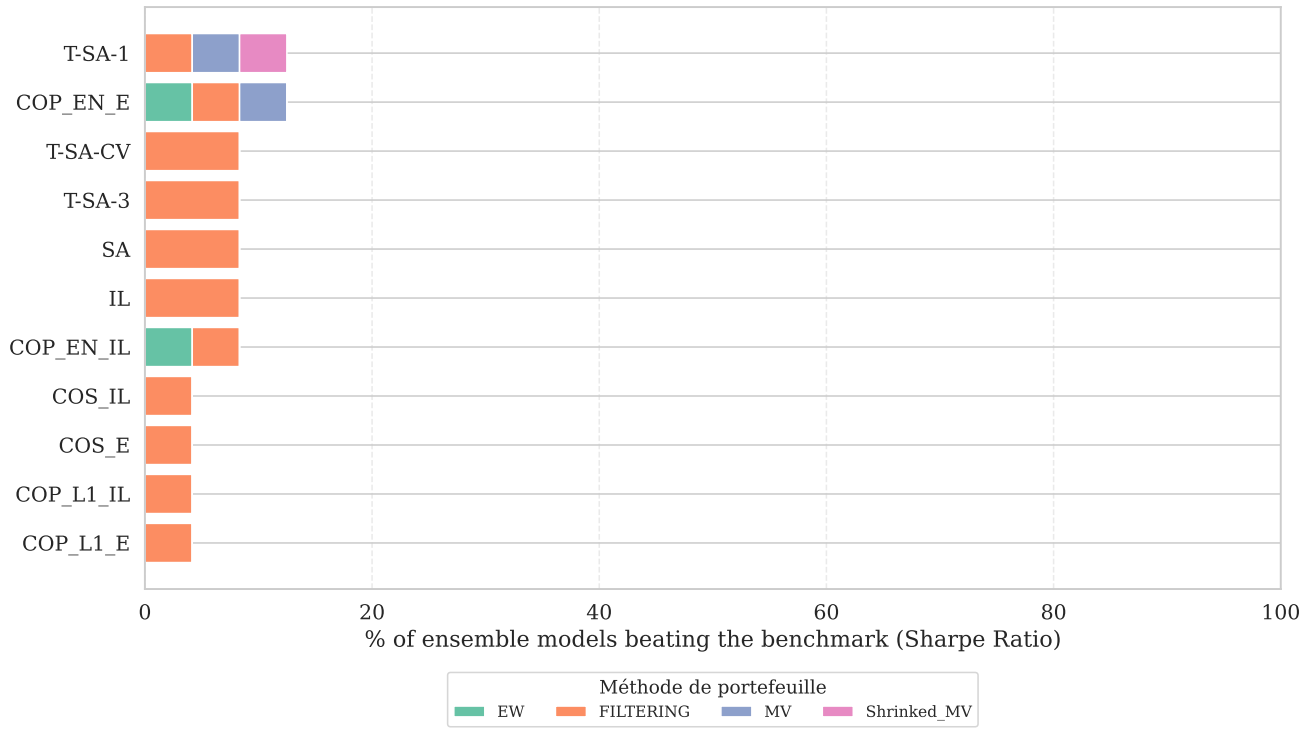


Figure 5.13: Percentage of portfolios beating the Nasdaq-100 by ensemble method
This chart shows the proportion of portfolios that outperform the benchmark according to their sharpe ratio, grouped by ensemble methods and allocations strategies.

Conclusion

The objective of this thesis was to investigate whether the ensemble learning framework proposed by Roccazzella et al. (2022), originally developed for regression-based forecast combinations, could be extended to classification tasks in financial time series. More specifically, we investigated whether these techniques could improve the prediction of future stock returns and lead to the creation of optimal portfolios capable of outperforming standard market indices.

To address this question, Chapter 1 introduced the theoretical background of machine learning classifiers applied to financial data. Chapter 2 presented ensemble learning strategies and adapted the original constrained and penalized optimization methods to the classification setting; These combinations aggregate the binary results of several classifiers and produce a continuous prediction signal. However, in classification problems, an essential but often overlooked step comes next: transforming these continuous signals into binary decisions via a threshold. This conversion determines whether an action is classified as outperforming or not, and therefore whether it enters the portfolio. This nuanced but critical aspect of the methodology has become a central focus of our study. Chapter 3 reviewed portfolio construction frameworks. Given that our classifiers do not produce explicit expected returns, we opted for a simplified mean-variance framework using historical covariances. Chapter 4 detailed the empirical design using walk-forward simulations based on NASDAQ-100 constituents, relying solely on technical indicators. Chapter 5 presented the results: Random Forest stood out among the individual models in terms of average accuracy, but simpler models like logistic regression also performed well, despite more pronounced dispersion. Tree-based and MLP classifiers demonstrated greater stability but lower average performance. Still, only 7 out of 10 classifiers were significantly better than a random model, based on McNemar tests.

Ensemble models provided markedly improved results. Using a naïve threshold of $\tau = 0.5$, all ensemble forecasts achieved statistically significant accuracy. Yet, we observed a consistent trade-off: while **precision increased**, **recall and F1-score often dropped**, indicating that many true positives were being missed. To address this limitation, we explored more sophisticated thresholding techniques, such as optimizing for precision or accuracy or using quantile-based thresholds. These approaches significantly improved

both accuracy and precision across most models. However, they further decreased recall. Conversely, in scenarios where the threshold was optimized for recall, we observed remarkably high recall scores, at the expense of precision. **This duality clearly highlighted thresholding as a critical hyperparameter**, with direct implications for both predictive classification quality and the financial returns of the portfolio that stem from it.

In fact, the threshold selection after forecast combination emerged as a key decision point that can dramatically shift the balance in portfolio performance: backtesting was performed on each of the 544 generated portfolio and none of the individual classifiers produced portfolios that outperformed the NASDAQ-100 benchmark. Conversely, several ensemble-based strategies did, clearly demonstrating the value of forecast combination in portfolio construction. Portfolio using a threshold decision based on an optimization of the precision metrics performed the best in terms of sharpe ratio and APY. The best performing strategy combined a filtering-based covariance estimator with Ledoit-Wolf shrinkage and a trimmed ensemble (T-SA-1) using a precision-optimized threshold. This configuration achieved a Sharpe Ratio of **1.11** and an APY of **19.29%**, compared to **0.74** and **14.72%** respectively for the NASDAQ-100 index.

A meta-analysis of the impact of thresholding strategies and portfolio optimization choices revealed that fewer than **20%** of the generated portfolios managed to outperform the benchmark index in terms of annualized performance. This result confirms the intrinsic difficulty of extracting exploitable signals from noisy financial markets, even when using ensemble learning techniques. Interestingly, the analysis also showed that **different thresholding strategies favor different financial objectives**. For instance, the thresholding method based on precision optimization ($\texttt{OPTI}_{\texttt{PREC}}$) yielded the highest proportion of portfolios that beat the market in terms of APY. In contrast, the naïve threshold ($\tau = 0.5$) resulted in the largest number of portfolios achieving a superior Sharpe Ratio compared to the benchmark. Moreover, we observed that some financial metrics such as **maximum drawdown, volatility** and **average loss**, were only a few affected by the choice of threshold.

Limitations and Directions for Future Researches

Despite these encouraging results, several limitations must be acknowledged. These also suggest interesting directions for future work:

- **Limited input features and individual classifier models.** The models used in this study rely only on technical indicators to generate predictions. While these are widely used in practice and often effective at capturing price action, they ignore other market dimensions. Adding fundamental data such as valuation ratios, earnings revisions, debt levels, as well as alternative sources like sentiment scores from news or social media (Morelle, 2022), could improve the models. Exploring more diverse and expressive models, such as long short-term memory (LSTM) architectures (K. Chen

et al., 2015), could also better capture temporal dependencies.

- **Simplified handling of taxes and real-life constraints.** Although we accounted for transaction fees, we did not model the impact of capital gains taxation, which can significantly reduce net returns. Taxation policies vary greatly across jurisdictions. Moreover, we assumed that it was possible to buy fractional shares, which simplifies portfolio construction but not always reflecting real trading conditions. Enforcing integer share quantities transforms the portfolio optimization problem into a mixed-integer quadratic program, which is `NP-hard` and significantly more complex to solve (Bienstock, 1996).

- **Execution assumptions and slippage.** Throughout the backtests, we assumed trades were executed at the official daily closing price. However, in real markets, especially near the close, execution prices may deviate significantly due to bid-ask spreads and limited liquidity. This may lead to optimistic estimations of both returns and volatility. A more realistic simulation would model execution costs more precisely by using intraday data for instance.

- **Single-path backtesting.** The simulation used in this thesis only followed one historical path, the actual sequence of past market data. As a result, our findings are inherently dependent on this particular realization of events, and may not generalize to future or alternative market conditions. More robust evaluation methods, such as *Combinatorial Purged Cross-Validation* (CPCV) introduced by de Prado (2018), multiple alternative paths to better assess performance stability. Additionally, recent advances in generative modeling, such as GANs introduced by Goodfellow et al. (2014), could allow the simulation of synthetic yet statistically plausible market scenarios, opening new opportunities for scenario-based validation.

- **Hardware and temporal limitations.** The experiments were run on a modern machine (Intel i7 CPU and NVIDIA RTX GPU), which obviously did not exist during the earlier years of our backtesting period. This creates a methodological inconsistency: our models assume access to computational tools and algorithms that were unavailable at the time. The same limitation applies to the benchmark passive investment in the NASDAQ-100. Moreover, this researches was reduced to weekly observations due to computational costs, limiting the potential of technical indicators data.

Overall, this thesis highlights that ensemble learning, when combined with carefully calibrated thresholds and robust portfolio design, can lead to significant improvements in both statistical and financial terms. Among the various design choices, the threshold mechanism stands out as one of the most influential factors, as it directly determines how predictive signals are translated into investment actions, which ultimately impacts portfolio performance.

Bibliography

Atiya, A. F. (2020). Why does forecast combination work so well? [M4 Competition]. *International Journal of Forecasting*, *36*(1), 197–200. https://doi.org/https://doi.org/10.1016/j.ijforecast.2019.03.010

Barbierato, E., & Gatti, A. (2024). The challenges of machine learning: A critical review. *Electronics*, *13*(2). https://doi.org/10.3390/electronics13020416

Bengfort, B., Bilbro, R., & Ojeda, T. (2020). Threshold tuning with yellowbrick [Accessed 2025-06-05].

Bienstock, D. (1996). Computational study of a family of mixed-integer quadratic programming problems. *Mathematical Programming*, *74*(2), 121–140.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Bloomberg L.P. (2024). Bloomberg terminal historical market data [Accessed via Bloomberg Terminal].

Breiman, L. (2001). *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/a:1010933404324

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Wadsworth International Group.

Bun, J., Bouchaud, J.-P., & Potters, M. (2017). Cleaning large correlation matrices: Tools from random matrix theory. *Physics Reports*, *666*, 1–109.

Candelon, B. (2024). L7 – artificial neural network (ann) for forecasting [Lecture slides, reproduction prohibited without explicit permission]. *UCLouvain University*.

Chen, K., Zhou, Y., Dai, F., & Jiang, H. (2015). Lstm neural network for financial time series prediction. *2015 IEEE International Conference on Intelligent Systems Design and Applications (ISDA)*, 747–753.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Clemen, R. T. (1989). Combining forecasts: A review and annotated bibliography. *International Journal of Forecasting*, *5*(4), 559–583.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297.

Deisenroth, M. P., Faisal, A. A., & Ong, C. S. (2020). *Mathematics for machine learning*. Cambridge University Press.

DeMiguel, V., Garlappi, L., & Uppal, R. (2009). Optimal versus naive diversification: How inefficient is the 1/n portfolio strategy? *The Review of Financial Studies*, *22*(5), 1915–1953. https://doi.org/10.1093/rfs/hhm075

de Prado, M. L. (2018). *Advances in financial machine learning*. Wiley.

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, *25*(2), 383–417. Retrieved October 29, 2024, from http://www.jstor.org/stable/2325486

Fan, J., Fan, Y., & Lv, J. (2011). High dimensional covariance matrix estimation using a factor model. *Journal of Econometrics*, *160*(1), 34–52.

Ferson, W. E., & Siegel, L. B. (2003). Estimation risk and portfolio performance. *Journal of Portfolio Management*, *29*(1), 52–60.

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

Genre, V., Kenny, G., Meyler, A., & Timmermann, A. (2013). Combining expert forecasts: Can anything beat the simple average? *International Journal of Forecasting*, *29*(1), 108–121. https://doi.org/10.1016/j.ijforecast.2012.06.004

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. https://www.deeplearningbook.org

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, *27*.

Goyal, A., & Welch, I. (2008). A comprehensive look at the empirical performance of equity premium prediction. *The Review of Financial Studies*, *21*(4), 1455–1508.

Graf, C., Flanagan, D., Wylie, L., & Silver, D. (2020). The Open Data Challenge: An Analysis of 124,000 Data Availability Statements and an Ironic Lesson about Data Management Plans. *Data Intelligence*, *2*(4), 554–568. https://doi.org/10.1162/dint_a_00061

Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies*, *33*(5), 2223–2273.

Hansen, P. R., Lunde, A., & Nason, J. M. (2011). The model confidence set. *Econometrica*, *79*(2), 453–497. https://doi.org/10.3982/ECTA5771

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd). Springer.

Hendry, D. F., & Clements, M. P. (2004). Pooling of forecasts. *Oxford Bulletin of Economics and Statistics*, *66*, 1–19.

Htun, H., Biehl, M., & Petkov, N. (2024). Forecasting relative returns for sp 500 stocks using machine learning. *Financial Innovation*, *10*. https://doi.org/10.1186/s40854-024-00644-0

Huang, W.-H., Nakamori, Y., & Wang, S.-Y. (2005). Forecasting stock market movement direction with support vector machine. *Computers & Operations Research*, *32*(10), 2513–2522.

Invesco. (2024). Invesco eqqq nasdaq-100 ucits etf - accumulation [Accessed June 2024].

Jagannathan, R., & Ma, T. (2003). Risk reduction in large portfolios: Why imposing the wrong constraints helps. *The Journal of Finance*, *58*(4), 1651–1684.

Jaggi, M., & Flammarion, N. (2020). Lecture notes: Machine learning course - cs433 [EPFL, Lausanne]. https://mlo.epfl.ch/

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. *ACM Computing Surveys*, *31*(3), 264–323.

Kakushadze, Z. (2016). 101 formulaic alphas. https://arxiv.org/abs/1601.00991

Kan, R., & Zhou, G. (2007). Optimal portfolio choice with parameter uncertainty. *Journal of Financial and Quantitative Analysis*, *42*(3), 621–656.

Krauss, C., Do, X. A., & Huck, N. (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the sp 500. *European Journal of Operational Research*, *259*(2), 689–702. https://doi.org/https://doi.org/10.1016/j.ejor.2016.10.031

Ledoit, O., & Wolf, M. (2003). Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. *Journal of Empirical Finance*, *10*(5), 603–621.

Ledoit, O., & Wolf, M. (2004). A well-conditioned estimator for large-dimensional covariance matrices. *Journal of multivariate analysis*, *88*(2), 365–411.

Lo, A. (2004). Reconciling efficient markets with behavioral finance: The adaptive markets hypothesis. *Journal of Investment Consulting*, *7*.

López de Prado, M. (2018). *Advances in financial machine learning*. John Wiley & Sons.

López de Prado, M. M. (2020). *Machine learning for asset managers*. Cambridge University Press.

Ma, C., & Jagannathan, R. (2003). Risk and return in portfolio optimization with non-negativity constraints: The impact of estimation error. *Journal of Finance*, *58*(2), 927–971.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, *1*, 281–297.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). The m4 competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, *34*(4), 802–808. https://doi.org/https://doi.org/10.1016/j.ijforecast.2018.06.001

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2022). M5 accuracy competition: Results, findings, and conclusions [Special Issue: M5 competition]. *International*

Journal of Forecasting, *38*(4), 1346–1364. https://doi.org/https://doi.org/10.1016/j.ijforecast.2021.11.013

Markowitz, H. (1952). Portfolio selection. *The Journal of Finance*, *7*(1), 77. https://doi.org/10.2307/2975974

Michaud, R. (1989). The markowitz optimization enigma: Is 'optimized' optimal? *Financial Analysts Journal - FINANC ANAL J*, *45*, 31–42. https://doi.org/10.2469/faj.v45.n1.31

Michaud, R. O. (1989). The markowitz optimization enigma: Is 'optimized' optimal? *Financial Analysts Journal*, *45*(1), 31–42.

Mohammed, T., Khalil, O., & Ibrahim, M. (2023). Threshold optimization in imbalanced classification: An empirical study on credit card fraud detection. *Journal of Big Data*, *10*(1), 1–19.

Morelle, H. (2022). *Portfolio construction with crowd intelligence: A machine learning approach* [Master's thesis]. Louvain School of Management, Université catholique de Louvain [Promotor: Vrins, Frédéric]. http://hdl.handle.net/2078.1/thesis:36531

Pafka, S., Potters, M., & Kondor, I. (2004). Exponential weighting and random-matrix-theory-based filtering of financial covariance matrices for portfolio optimization. *arXiv preprint cond-mat/0402573*.

Pearson, K. (1895). Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, *58*, 240–242.

Roccazzella, F., Gambetti, P., & Vrins, F. (2022). Optimal and robust combination of forecasts via constrained optimization and shrinkage. *International Journal of Forecasting*, *38*(1), 97–116. https://doi.org/10.1016/j.ijforecast.2021.04.002

Shao, Z., Gao, H., Xu, B., & Wang, Y. (2024). Ensemble reinforcement learning through classifier models: Application to stock trading. *arXiv preprint arXiv:2502.17518*.

Simonis, N. (2021). *Stock selection and portfolio optimization: A machine learning approach* [Master's thesis]. Université catholique de Louvain. http://hdl.handle.net/2078.1/thesis:31397

Smith, J., & Wallis, K. F. (2009). A simple explanation of the forecast combination puzzle. *Oxford Bulletin of Economics and Statistics*, *71*(3), 331–355. https://doi.org/10.1111/j.1468-0084.2008.00541.x

S&P Global. (2024, December). SPIVA | S&P Dow Jones Indices — spglobal.com [[Accessed 02-08-2025]].

Timmermann, A. (2006). Forecast combinations. *Handbook of economic forecasting*, *1*, 135–196.

Wang, X., Hyndman, R. J., Li, F., & Kang, Y. (2023). Forecast combinations: An over 50-year review. *International Journal of Forecasting*, *39*(4), 1518–1547.

Wolff, D., & Echterling, F. (2022). Stock picking with machine learning [Available at https://ssrn.com/abstract=3607845].

Yahoo Finance. (2024). Historical stock data [Accessed via Yahoo Finance Python API].

Zhang, G., Zhou, W., & J. Yang, J. (2007). Credit scoring with a hybrid neural network and support vector machine. *Expert Systems with Applications*, *33*(2), 300–312. https://doi.org/10.1016/j.eswa.2006.05.019

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

Appendices

A

Additional Information On Machine Learning Models

Machine learning refers to models that automatically learn patterns in data, including complex relationships often missed by traditional statistical methods. These patterns are used to make predictions or decisions on future outcomes based on new input data.

Machine learning is based on three fundamental concepts: **data**, a **model** and **learning**. As an inherently data-driven discipline, its aim is to extract meaningful patterns from data with a minimum of domain-specific input. A model represents a structured way of describing the relationship within the data - for example, mapping inputs to outputs in a regression. Learning is the process by which the model improves its performance on a task as it is exposed to more data, with the ultimate aim of generalizing well to new, unseen examples (Deisenroth et al., 2020).

Several recent studies have shown that machine learning techniques can outperform classical models in stock return forecasting tasks (Gu et al., 2020). Traditional linear models (such as AR, MA or even ARIMA) often struggle to capture the complex and nonlinear dynamics of financial markets. In contrast, machine learning models can adapt to high-dimensional data and discover hidden, complex, implicit interactions between variables/features, making them well-suited for forecasting tasks under uncertainty.
This chapter presents the classical individual machine learning models used to forecast future stock returns. The prediction issued from those models will further be combined using ensemble techniques. Therefore, the purpose here is not to exhaustively describe each model, but rather to highlight their key differences, role, nature and show how they can[2] be complementary within a forecast ensemble framework.

A.1 Overview of Modeling Paradigms

Before looking at specific machine learning algorithms, it's important to clarify the fundamental modeling paradigms that define how machine learning systems work. Machine

learning methods can be classified into several categories, depending on the nature of the data, the form of relationships they can capture and the type of outcome they aim to predict. This section presents three key distinctions: supervised and unsupervised learning, linear and nonlinear models, and classification and regression tasks.

A.1.1 Supervised vs Unsupervised

Supervised Learning

Supervised learning is a fundamental paradigm in machine learning in which the algorithm is trained on a dataset composed of input–output pairs, denoted as $(\mathbf{X}, \mathbf{y}) = \{(\mathbf{x}_i, y_i) \mid i = 1, \ldots, N\}$ where y_i is the i^{th} output and the vector $\mathbf{x}_i$ is the i^{th} input of dimension D. Each input vector $\mathbf{x}_i$, also referred to as a feature vector, predictor or independent variable, is associated with a known output label y_i also called the response or target (M. M. López de Prado, 2020). The primary goal of supervised learning is to establish a mapping, $f : \mathbb{R}^{D \times N} \to \mathbb{R}^N$ between input and output, enabling the accurate prediction of unseen data in the future.

In this context, learning is performed by minimizing a loss function $L(\hat{y}, y)$, which quantifies the difference between the model's predictions $\hat{y}$ and the actual results y. Common examples of loss functions include mean squared error for regression tasks and cross-entropy for classification. Through an iterative optimization process, the model adjusts its parameters to minimize the loss on the training data. However, the overall goal is not simply to achieve a good fit on the training set (which can lead to overfitting)), but to effectively generalize to unseen data (Barbierato and Gatti, 2024).

This learning paradigm is particularly relevant in financial applications, where historical data (such as firm fundamentals, technical indicators, or macroeconomic variables) serve as input features, and future returns or performance labels (e.g., outperform/underperform) are used as outputs. In this thesis, supervised learning is employed to classify stocks according to their expected future performance, forming the foundation for forecast combination and portfolio construction strategies.

Unsupervised Learning

In the case of Unsupervised learning, we again have access to a data set $\mathbf{X} = \{\mathbf{x}_i \mid i = 1, \ldots, N\}$ but it works only on those input data, with no associated target label, i.e. y_n. The main objective is to discover patterns, groupings or latent structures in the data, without being explicitly asked what to look for. In this context, learning takes place through the model's ability to organize the data on the basis of internal similarities or statistical properties, rather than comparing its results with known output.

Two of the most common approaches to unsupervised learning are **clustering** and **dimensionality reduction**. Clustering involves dividing the data set into subsets (i.e. clusters) so that observations within the same group share common characteristics. More formally, the model attempts to find a partition $C = \{c_1, c_2, ..., c_k\}$ of the data that minimizes the dispersion $\mathcal{D}$ within each cluster. The standard formulation of this objective is

$$\min_{C} \sum_{i=1}^{k} \sum_{x \in c_i} \mathcal{D}(x, \mu_i) \tag{A.1}$$

where μ_i is the centroid of cluster c_i, and $\mathcal{D}(x, \mu_i)$ is a distance metric such as the Euclidean or Mahalanobis distance (Jain et al., 1999; MacQueen, 1967).

The second major application, *dimensionality reduction*, aims to represent complex, high-dimensional data using a reduced number of variables while retaining the essential information. This is particularly useful in financial datasets, where indicators often present high correlations.

In financial research, these techniques have numerous applications. Clustering may be used to identify homogeneous groups of securities based on return profiles (Pearson, 1895), while PCA can be employed to extract unobserved risk factors that influence asset returns (Fan et al., 2011). In this thesis, we use unsupervised learning in to identify latent structures in the feature space (e.g., using PCA) serving as a preprocessing step prior to supervised forecasting models.

A.1.2 Linear vs Non-Linear

Machine learning models can also be distinguished according to their ability to capture the complexity of relationships between inputs and outputs. **Linear models**, such as linear regression or logistic regression, assume a direct and additive relationship between features and the target variable. These models are fast, interpretable and tend to generalize well in high-dimensional spaces with relatively low signal-to-noise ratio. However, they are limited in their ability to capture non-linear interactions or dependencies, which are often present in financial markets (Gu et al., 2020).

In contrast, **non-linear models**, such as decision trees, random forests, growth gradient machines or neural networks, allow for more flexible relationships. It includes interactions between variables themselves. They are therefore better suited to capturing the complex and often non-linear structure of financial data. However, they also tend to be more prone to over-fitting, and generally require more careful tuning and regularization (M. López de Prado, 2018).

A.1.3 Classifier vs Regressor

Another important distinction in modeling concerns the tasks of **regression** and **classification**. Regression involves predicting a **continuous numerical outcome**, such as the future performance of a stock. Classification, on the other hand, involves classifying input data into **discrete categories**.

$$y_i = \mathbf{1}_{\{\cdot\}} := \begin{cases} 1 & \text{if the condition is met} \\ 0 & \text{otherwise} \end{cases}$$

In finance, binary classification problems are often used to predict whether a stock will *outperform* or *underperform* a given benchmark over a defined period (Wolff & Echterling, 2022).

In this thesis, the prediction task is formulated as a binary classification problem: each stock is labeled as 1 if it is expected to outperform a fixed benchmark (e.g., the NASDAQ index) over the next week, and 0 otherwise. This approach is in line with recent literature showing that classification models often outperform regression models in ranking-based portfolio strategies (Gu et al., 2020; Simonis, 2021). Moreover, it mirrors how real-world portfolio managers think: they are less interested in predicting the exact return of a stock, and more focused on ranking assets to identify the best relative performers.

A.2 Supervised Learning Algorithms for Binary Classification

In this thesis, we implement a diverse range of supervised machine learning algorithms for binary classification. The goal is to predict whether a stock will outperform a given benchmark over the next period, based on a set of input features. Below, we describe the core categories of models used, including mathematical formulations where appropriate to better understand the foundation of the model.

A.2.1 Regularized Linear Models

Regularized linear models extend logistic regression to prevent overfitting and improve generalization. This section will briefly summarize the lectures notes of Jaggi and Flammarion (2020). The basic **logistic regression** model assumes a linear relationship between $\mathbf{x}_i$ and y_i and it predicts the probability of a binary outcome using a sigmoid activation:

$$\mathbb{P}(y_i = 1|\mathbf{x}_i, \mathbf{w}) = \sigma(\mathbf{x}_i^T \mathbf{w}) = \frac{1}{1 + e^{-\mathbf{x}^T \mathbf{w}}} \tag{A.2}$$

Our goal with logistic regression is to estimate the weight vector $\mathbf{w}$ that maximizes

the probability that the observed data was generated by the model. Given that our dataset is composed of independent observation pairs $(\mathbf{x}_i, y_i)$, we aim to maximize the likelihood L that these observations were produced by the model. In practice, this is done by minimizing the negative log-likelihood, which provides an estimate for the optimal weights.

$$\mathcal{L}^{LR}(\mathbf{w}) = -\sum_{i=1}^{N} \left[y_i \log\left(\sigma\left(\mathbf{x}_i^\top \mathbf{w}\right)\right) + (1 - y_i) \log\left(1 - \sigma\left(\mathbf{x}_i^\top \mathbf{w}\right)\right)\right] \tag{A.3}$$

We select the vector of weights $\mathbf{w}^*$ such that:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \mathcal{L}^{LR}(\mathbf{w}) \tag{A.4}$$

However, this linear relationship can be extended to a polynomial form to capture potential non-linear dependencies between variables. This augmentation, however, may introduce risks of underfitting or overfitting.

Hence, to control model complexity and to reduce the possibility of over-fitting, we can add a penalty, $\Omega(\mathbf{w})$, to the initial objective function of the Logistic regression (A.3):

$$\min_{\mathbf{w}} \mathcal{L}^{RLR}(\mathbf{w}) = \min_{\mathbf{w}} \mathcal{L}^{LR}(\mathbf{w}) + \Omega(\mathbf{w})$$

We consider three types of regularization:

- **Ridge (L2 Regularization):** $\Omega(\mathbf{w}) = \frac{\lambda}{2}||\mathbf{w}||_2^2$
- **Lasso (L1 Regularization):** $\Omega(\mathbf{w}) = \lambda||\mathbf{w}||_1$
- **Elastic Net (combine both):** $\Omega(\mathbf{w}) = \alpha\lambda||\mathbf{w}||_1 + (1-\alpha)\frac{\lambda}{2}||\mathbf{w}||_2^2$

where λ is a hyperparameter that controls the strength of the regularization that we find through cross-validation. α represents the mixing ratio, how we balance between the L1 and L2 penalties.

A.2.2 Tree-Based Models

Tree-based models are initially non-parametric learners[1] that can capture nonlinear relationships and high-order interactions between variables. They are particularly useful in financial contexts, where the functional form of the relationship between inputs and the target may be unknown or non-linear.

[1] One precises initially because the number of parameters grows with the data and thus the model is non parametric. However, since we cap its size for regularization numerically, it could be considered as a parametric one

- **Decision Tree Classifier** constructs a tree by recursively splitting the feature space. At each internal node, the algorithm selects the feature and threshold that minimizes a given impurity measure, such as the **Gini impurity**:

$$G(t) = 1 - \sum_{k=1}^{K} p_k^2 \tag{A.5}$$

where p_k is the proportion of class k samples in node t. It easures the frequency at which a randomly chosen element would be incorrectly labeled. There is also other impurity measure such as the cross-entropy $\mathcal{H}_m$ (Friedman, 2001).
Decision trees are interpretable but tend to overfit, especially when deep (Breiman et al., 1984)

- **Random Forest** builds an ensemble of decision trees trained on bootstrapped samples of the data. To reduce correlation between trees, a random subset of features is selected at each split. The final prediction is obtained by majority vote among all trees:

$$\hat{y} = \text{mode}(\hat{y}_1, \hat{y}_2, ..., \hat{y}_T) \tag{A.6}$$

Random forests reduce variance and improve generalization compared to a single tree (Breiman, 2001)

- **Gradient Boosting** builds an ensemble of decision trees sequentially, where each tree tries to correct the mistakes of the previous one:

$$F_t(\mathbf{x}) = F_{t-1}(\mathbf{x}) + \eta h_t(\mathbf{x}) \tag{A.7}$$

$$F_M(\mathbf{x}) = F_0(\mathbf{x}) + \sum_{m=1}^{M} \eta_m h_m(\mathbf{x}) \tag{A.8}$$

where $h_t(\mathbf{x_i})$ is the tree fitted to residuals, M the total number of boosting iterations and η is the learning rate. The output is a score, which is passed through a sigmoid function to convert it to a probability and class is by applying a threshold:

$$\mathbb{P}(y_i = 1|\mathbf{x}_i, \mathbf{w}) = \frac{1}{1 + e^{-F_M(x)}} \tag{A.9}$$

- **XGBoost** is an optimized implementation of Gradient Boosting Decision Trees designed for speed, scalability, and performance (T. Chen & Guestrin, 2016). XGBoost builds an ensemble of trees sequentially, where each new tree tries to minimize the residual error, it refines gradient boosting with regularization and sparse-aware

optimizations:

$$\mathcal{L}(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k), \quad \Omega(f) = \gamma T + \frac{1}{2}\lambda \|w\|^2 \tag{A.10}$$

where, $f_t \in \mathcal{F}$ is a decision tree and $\mathcal{F}$ the space of all trees. $l(y_i, \hat{y}_i)$: is the loss function (e.g. log-loss for our classification problem). T is the number of leaves and γ, λ are the regularization parameters.

A.2.3 Support Vector Classifiers

Support Vector Machines (SVMs) are powerful supervised learning models originally introduced by Cortes and Vapnik (1995). In classification settings, the Support Vector Classifier seeks to find the optimal separating hyperplane [2] that maximizes the margin between two classes. Formally, the hard-margin SVC solves:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \quad \text{subject to} \quad y_i \left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq 1 \tag{A.11}$$

Hence, hard-margin assumes that the data is perfectly separable. It has some limitations since it does not tolerate noise or overlap between classes. In practice, soft margins are preferred, allowing some misclassifications in exchange for better **generalization**. It leads to a regularized hinge loss formulation:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{N} \max\left(0, 1 - y_i \mathbf{w}^\top \mathbf{x}_i\right) \tag{A.12}$$

where C is the regularization strength.

With the use of kernel functions (e.g., RBF, polynomial), SVCs can handle non-linear decision boundaries, which is particularly relevant for financial data classification (Huang et al., 2005).

A.2.4 k-Nearest Neighbors (KNN)

The k-Nearest Neighbors (KNN) classifier is a simple, non-parametric algorithm. In the context of classification, the algorithm operates on a simple yet intuitive principle: an unlabeled observation is assigned the most common class among its k closest neighbors in the feature space.

Given a set of labeled training observations, the k-NN algorithm classifies a new data point by computing the distance (typically Euclidean, but other metrics such as Manhattan or Minkowski are also possible) between the new observation and all training samples.

[2] A hyperplane in $\mathbb{R}^n$ is a flat $(n-1)$-dimensional surface defined by $w^T x + b = 0$

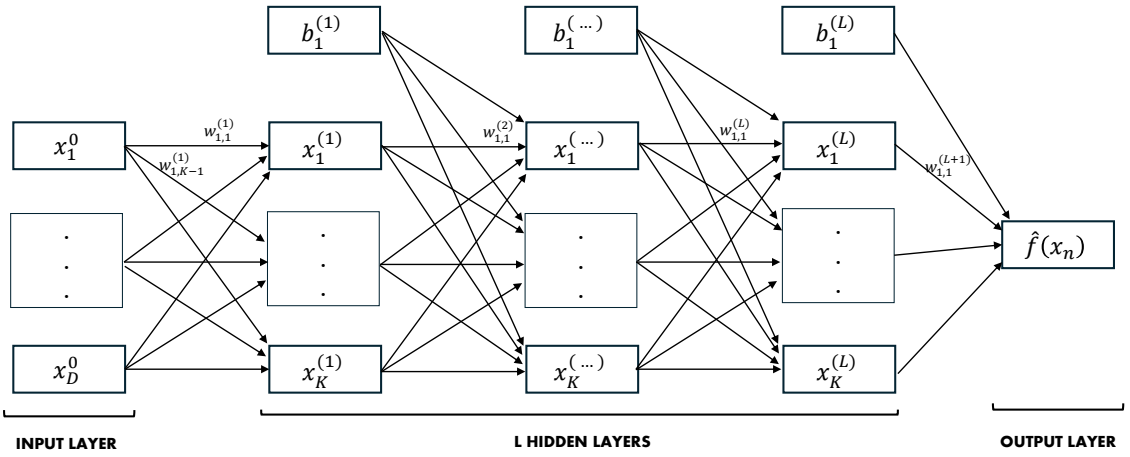
It then selects the k observations with the smallest distances and determines the most frequent class label among them. This label is then assigned to the new data point. Formally:

$$\hat{y}_q = \text{mode}\left\{y_i : \mathbf{x}_i \in \mathcal{N}_k\left(\mathbf{x}_q\right)\right\}, \quad d\left(\mathbf{x}_i, \mathbf{x}_q\right) = \left\|\mathbf{x}_i - \mathbf{x}_q\right\|_2 \tag{A.13}$$

$\mathcal{N}_k\left(\mathbf{x}_q\right)$ represents the ensemble of the k nearest neighbors of x_q. The number of neighbors k is a key hyperparameter and can be selected via cross-validation. A small k makes the classifier sensitive to noise, while a large smooths out local structures. KNN does not require training. As a **lazy learning algorithm**, k-NN does not require a training phase in the traditional sense, but stores the entire dataset and performs computations at inference time, which can be computationally expensive. It is also sensitive to feature scaling. Still, it is used in finance for its intuitive behavior in local prediction tasks (Zhang et al., 2007).

A.2.5 Neural Networks (MLP)

Among the various machine learning models available for classification, neural networks, particularly **multilayer perceptrons** (MLPs), have proven to be highly effective at modeling complex and non-linear relationships between input variables and target labels. In the context of financial forecasting tasks, such as predicting the direction of stock returns, MLPs have the advantage of being universal function approximators, capable of capturing intricate patterns that traditional linear models may overlook Goodfellow et al., 2016.



Structure of the MLP Neural Networks. *Input features of dimension D. We have here L hidden layers of dimension K*

.

An MLP is a type of **feedforward neural network** composed of fully connected layers of interconnected units called **neurons**. Their general structure is illustrated in the figure here over and is divided in three main types of layers.

- **Input layer:** This layer takes in the input raw features of the dataset. Each neuron in this layer represents a different feature.
- **Hidden layers:** These intermediate layers are where most of the learning takes place. Each neuron performs a weighted linear combination of the inputs it receives, adds a bias term $b_k^{(l)}$, and applies a non-linear activation function $\phi(z)$ such as `ReLU`, `tanh`, or `sigmoid`. The output of neuron k in layer l is given by::

$$x_k^{(l)} = \phi\Big(\sum_i w_{i,k}^{(l)} x_i^{(l-1)} + b_k^{(l)}\Big) \tag{A.14}$$

  where $w_{i,k}^{(l)}$ represents the weighted edge from node i in layer $(l-1)$ to the node k in layer l. This stage, where data moves through the network to produce an output, is known as **forward propagation**.
- **Output layer:** For classification tasks, the output layer typically uses a sigmoid activation function to output probabilities associated with each possible class.

A central question is what precisely defines the "right" weights and biases for a neural network. After the network has produced an output, it becomes necessary to assess how close this prediction is to the actual label. This is done through a **loss function** (e.g., Mean Squared Error or Cross-Entropy), which quantifies the prediction error. The process of adjusting the weights and biases to reduce this error is carried out through a method called **backpropagation**. In this process, the initial values of $w_{i,k}^{(l)}$ and $b_k^{(l)}$ are progressively adjusted through optimization, with the goal of minimizing the loss and improving predictive accuracy.

A.3 Unsupervised Algorithms

A.3.1 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is an unsupervised learning technique widely used for **dimensionality reduction**. Its main objective is to transform a set of possibly correlated input variables into a new set of uncorrelated variables, called PRINCIPAL COMPONENTS, which are ordered such that the principal components retain most of the variance present in the original dataset.

In the context of financial modeling, PCA has been used to extract latent factors from stock returns, simplify high-dimensional feature spaces, or reduce noise in large correlation matrices. As pointed out by Simonis (2021), such techniques can be helpful to prevent overfitting when dealing with many input features, particularly in portfolio optimization or predictive stock selection tasks.

More formally, it is done through this process: Let $\mathbf{X} \in \mathbb{R}^{D \times N}$ be a matrix representing N observations and D features, centered such that each column has zero mean. PCA seeks a new basis of orthogonal vectors $\{e_1, e_2, ..., e_d\}$, with $d \leq D$, where the data is projected in order to maximize variance:

$$\underset{w}{\text{maximize}} \quad \mathbb{V}(\mathbf{X}e) = e^T S e \quad \text{subject to} \quad \|e\| = 1 \tag{A.15}$$

Here, $\mathbf{S} = \frac{1}{N}\mathbf{X}^T\mathbf{X}$ is the empirical covariance matrix of the original data set. The solution is obtained by computing the eigenvectors of S; the directions associated with the highest eigenvalues define the most informative axes. The projection of the data along the first K eigenvectors gives a compressed version of the data, while retaining most of its original structure.

In practice, we do not retain all d principal components. Instead, we select a number $K \leq d$ such that a proportion $c\%$ of the total variance in the data is preserved. Using the diagonal entries of $\mathbf{S}$ (or of the singular value matrix in SVD), the value of K is chosen as the smallest integer such that:

$$\frac{\sum_{k=1}^{K} \mathbf{S}_{kk}}{\sum_{n=1}^{d} \mathbf{S}_{nn}} \geq c\%$$

This ensures that the reduced representation retains the most informative structure of the data while significantly reducing dimensionality.

B

Additional Investigations

B.1 Investigation on portfolio composition

It may be interesting to study the allocation features of each portfolio. We conducted an in-depth analysis of the composition of each portfolio based on the portfolio optimization method used.

Turnover represents the frequency and intensity of rebalancing over time; higher turnover indicates that the portfolio requires more frequent adjustments, which can lead to higher transaction costs.

The weekly standard deviation of portfolio weightings reflects the variability of the weightings assigned to each asset from period to period. A higher standard deviation suggests that the allocation is more volatile and that the optimizer frequently transfers capital between positions.

Finally, the *Herfindahl index* defines the overall level of concentration in the portfolio. Lower Herfindahl index values indicate greater diversification among assets, while higher values reveal a stronger concentration of the allocation in a smaller number of positions.

In our results, the turnover rate generally varies between 0.2 and 1.6, with lower turnover levels associated with more stable APY outcomes. The weekly standard deviation of the portfolio weights ranges approximately from 0.005 to 0.04, reflecting varying degrees of allocation volatility. Finally, the Herfindahl Index spans from around 0.05 (highly diversified allocations) to over 0.3 (more concentrated, yet still reasonably diversified, portfolios).

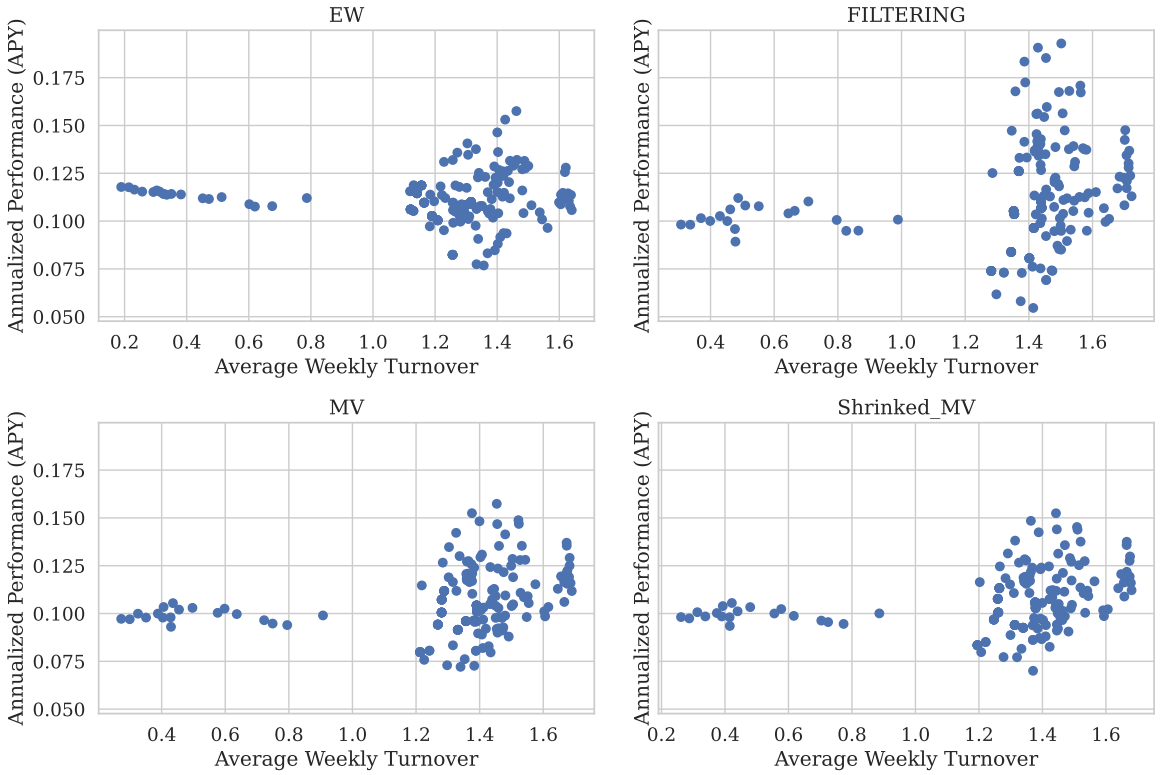


Figure B.1: APY versus average weekly turnover by optimization technique. *This figure shows, for each optimization method, the relationship between average weekly turnover and annualized performance (APY). For turnovers below 1, performances are relatively stable and show limited dispersion. In contrast, when turnover exceeds 1, the variability of returns increases. Although no clear or systematic pattern emerges from the distribution of points, the average APY tends to be slightly higher*

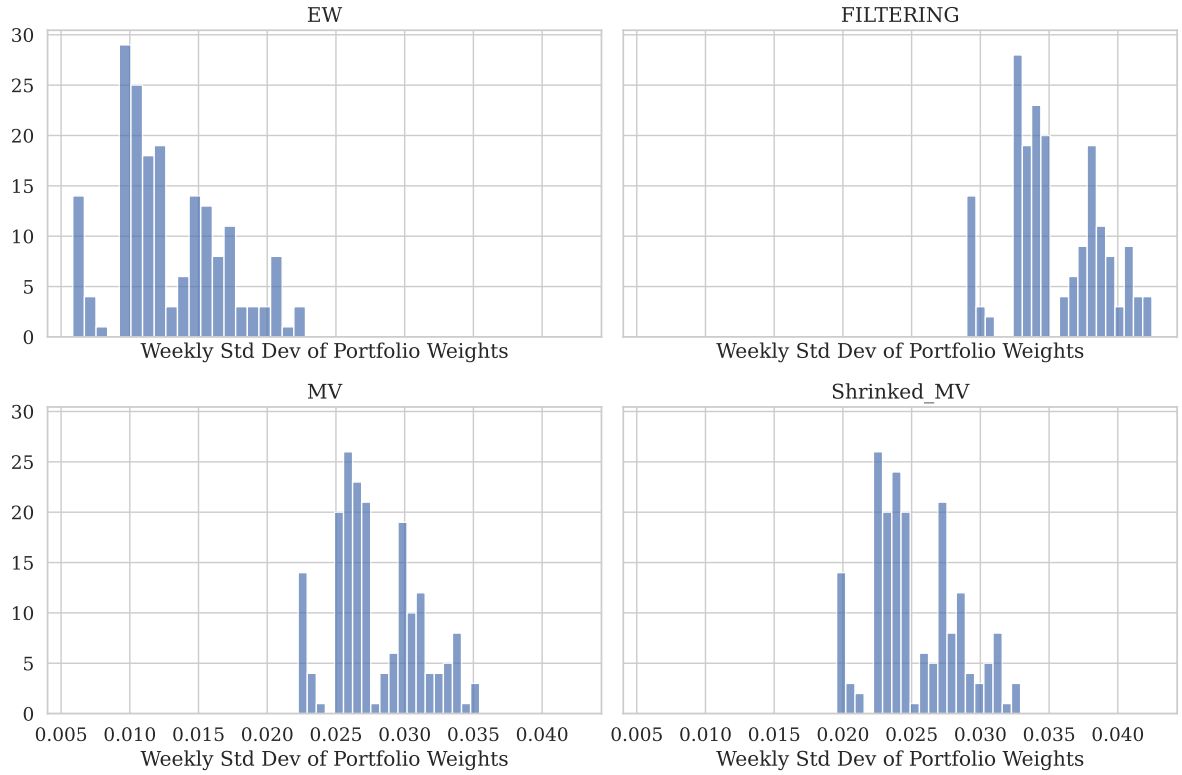


Figure B.2: Distribution of the weekly standard deviation of portfolio weights by optimization technique. *This figure shows the dispersion of asset weights, measured by the weekly standard deviation, for each optimization method. Obviously, Equal Weight portfolios display more homogeneous and less volatile allocations over time, while Filtering and Minimum Variance approaches exhibit higher variability in weight adjustments, reflecting more dynamic reactions to changing market signals.*

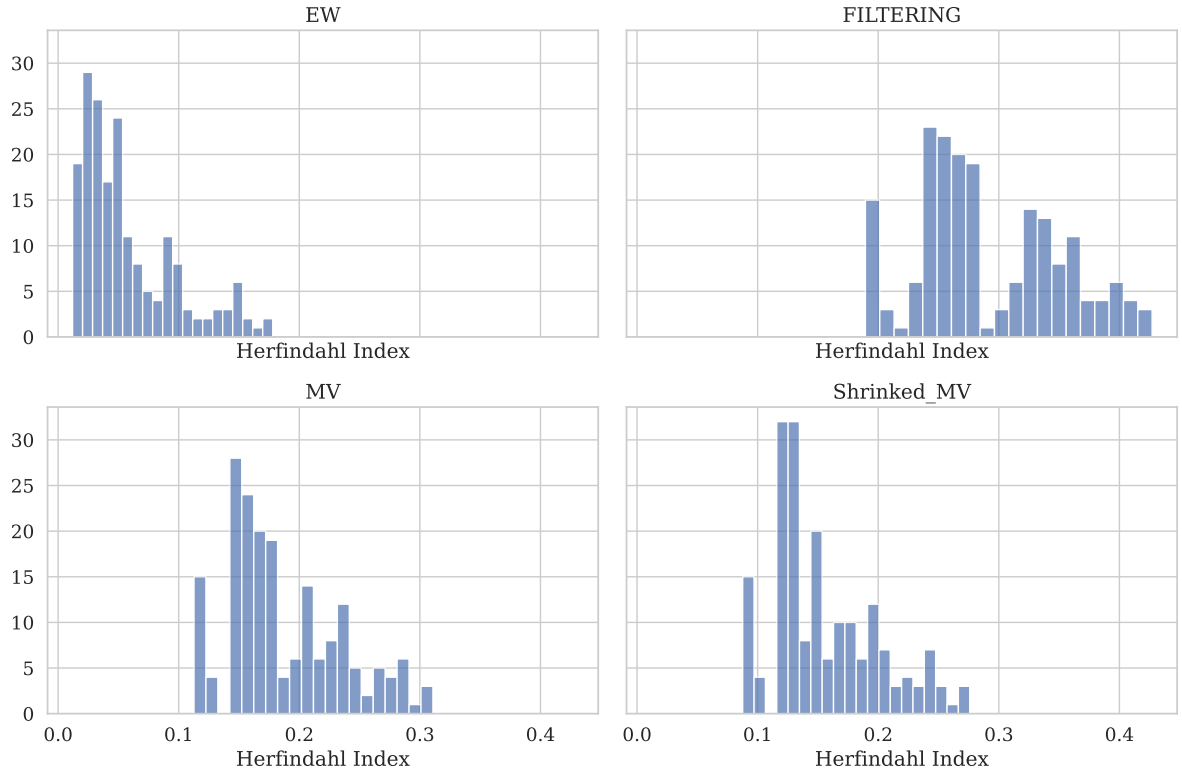


Figure B.3: Distribution of the Herfindahl Index by optimization technique. *It shows the distribution of the Herfindahl Index, which measures the degree of portfolio concentration. Equal Weight portfolios stand out with lower concentration and greater diversification, whereas Minimum Variance and Filtering methods tend to produce more concentrated portfolios, with dominant allocations across a smaller set of assets.*

B.2 Investigation on cost transactions

In this section, we look into how transaction costs affect portfolio performance. It's a fair question to ask whether financial costs linked to trading really matter and as we know, when building and optimizing portfolios, these costs can play a critical role. To get a clear picture, we start by taking the Top 15 portfolios and compare their performance with and without transaction fees.

Overall, the results show that transaction costs do have an impact, but the effect stays relatively small. Take the best-performing strategy, for instance: with the FILTERING approach and T-SA-1 combination, the APY drops from 20.20% to 19.29%, which is a decrease of just 0.91 percentage points. That's about a 4.5% relative drop in APY. As for the Sharpe ratio, it goes from 1.15 to 1.11, a 3.4% decline. So yes, there's definitely a cost, but the reduction in performance stays manageable. What's more, in Figure B.4, we also observe that total transaction costs tend to increase as APY and Sharpe ratio rise. A deeper analysis in Figure B.5 confirms this trend, revealing a positive linear correlation between APY and transaction costs, as well as between the Sharpe ratio and costs.

PortfolioTech	Combination	Threshold	$APY^{Without}$ [%]	APY^{With} [%]	**Δ APY** [%]	$SR^{Without}$	SR^{With}	**Δ SR**
FILTERING	T-SA-1	$OPTI_{PREC}$	20.20	19.29	**-0.91**	1.15	1.11	**-0.05**
FILTERING	COP_EN^{E}	$OPTI_{PREC}$	19.95	19.07	**-0.88**	1.14	1.10	**-0.05**
FILTERING	IL	$OPTI_{PREC}$	19.41	18.53	**-0.88**	1.12	1.08	**-0.05**
FILTERING	SA	$OPTI_{PREC}$	19.17	18.34	**-0.83**	1.12	1.08	**-0.04**
FILTERING	COP_L1^{IL}	$OPTI_{PREC}$	18.07	17.25	**-0.82**	1.04	1.00	**-0.04**
FILTERING	IL	$OPTI_{ACC}$	18.05	17.08	**-0.97**	1.01	0.97	**-0.05**
FILTERING	T-SA-CV	$OPTI_{PREC}$	17.71	16.80	**-0.91**	1.01	0.97	**-0.05**
FILTERING	COP_EN^{IL}	$OPTI_{PREC}$	17.60	16.78	**-0.82**	1.04	1.00	**-0.04**
FILTERING	T-SA-3	$OPTI_{PREC}$	17.64	16.75	**-0.89**	1.04	0.99	**-0.05**
FILTERING	SA	$OPTI_{ACC}$	17.69	16.73	**-0.96**	1.00	0.95	**-0.05**
FILTERING	COS^{IL}	$OPTI_{PREC}$	16.83	15.97	**-0.86**	0.94	0.89	**-0.04**
EW	COP_EN^{E}	$OPTI_{ACC}$	17.49	15.75	**-1.74**	0.80	0.73	**-0.07**
MV	T-SA-1	$OPTI_{PREC}$	16.65	15.74	**-0.91**	1.00	0.95	**-0.05**
FILTERING	T-SA-CV	$OPTI_{ACC}$	16.52	15.63	**-0.89**	0.96	0.91	**-0.05**
FILTERING	COP_L1^{E}	$OPTI_{PREC}$	16.45	15.62	**-0.83**	0.95	0.91	**-0.04**

Table B.1: Impact of transaction costs on portfolio performance. *The comparison between annualized returns (APY) and Sharpe ratios (SR) with and without costs shows that performance decreases slightly. For example, the best-performing strategy without costs (20.20% APY) loses only 0.91 points, which is around a 4.5% drop, while the Sharpe ratio falls by about 3.4%.*

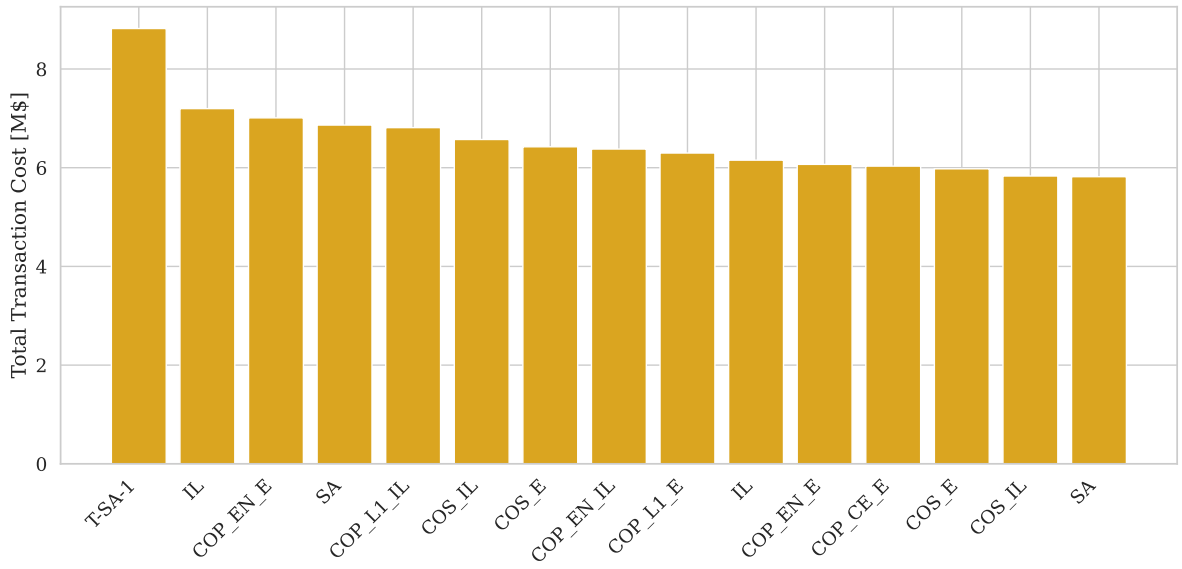


Figure B.4: Top 15 Combinations by Total Transaction Cost *This figure shows the 15 portfolio combinations with the highest total transaction costs, expressed in millions of USD. The T-SA-1 method clearly dominates with the largest cost, highlighting its intensive trading activity.*

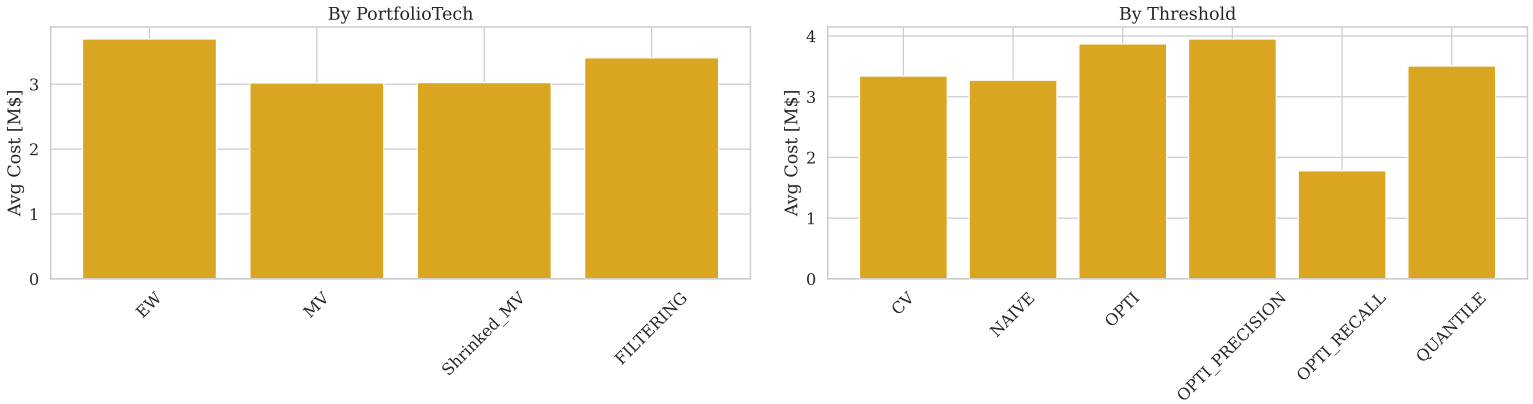


Figure B.5: Average Transaction Costs by PortfolioTech and Threshold – *This figure displays the average total transaction costs (in millions of USD) grouped by portfolio construction methods (left) and thresholding strategies (right). The EW and FILTERING methods appear to incur higher average costs among portfolio techniques, while thresholding based on* $OPTI_{ACC}$ *and* $OPTI_{PRECISION}$ *tends to be more costly than alternatives like* $OPTI_{RECALL}$.

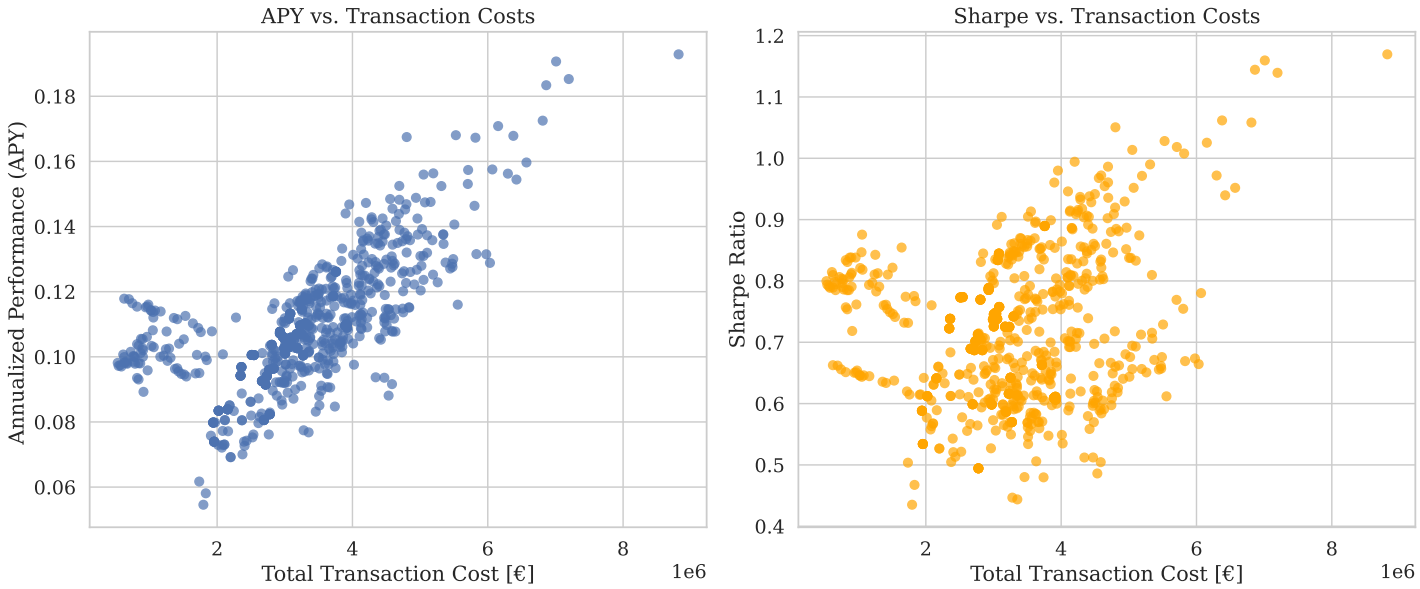


Figure B.6: Relationship between Performance Metrics and Transaction Costs – *The left panel shows a strong positive correlation between annualized performance (APY) and total transaction costs, suggesting that higher-performing portfolios tend to incur higher trading costs. The right panel depicts a weaker but still visible positive relationship between the Sharpe ratio and total costs, indicating that more efficient portfolios often require more active rebalancing.*

B.3 Investigation on maximum portfolio size

One also decided to investigate in more depth the influence of a parameter K, imposing a maximum portfolio size, on financial performance. Indeed, until now, the number of assets in our portfolios was not explicitly limited.

To address this, one identified two possible methods.

First, we could introduce a binary variable and thereby impose an additional cardinality constraint in the optimization problem during portfolio construction. The general formulation of the optimization problem without cardinality constraint is given by:

$$\begin{aligned} \min_{w,z} \quad & w^\top \hat{\mathbf{\Sigma}}\, w \\ \text{s.t.} \quad & \sum_{i=1}^{n} w_i = 1 \\ & w_i \geq 0 \quad \forall i \\ & w_i \leq z_i \quad \forall i \\ & \sum_{i=1}^{n} z_i \leq K \\ & z_i \in \{0,1\} \quad \forall i \end{aligned}$$

However, adding a constraint on the cardinality, such as limiting the number of non-zero weights, would make the problem NP-complete and therefore computationally very difficult to solve in practice.

Consequently, we chose an alternative approach that is simpler and more tractable. After solving the initial optimization problem without cardinality constraints, we select the K assets with the highest weights and re-standardize the weights relative to this new selection. This method allows us to control the effective size of the portfolio while avoiding the complexity associated with mixed-integer programming.

K	Mean Active Assets	Min Active Assets	Max Active Assets
5	4.86	1.93	5
10	9.36	2.67	10
20	17.28	3.97	20
40	28.70	6.02	40
50	32.46	6.19	50
60	35.18	6.28	60

Table B.2: Average, minimum, and maximum number of active assets in portfolios for each value of K.

K	Technique	Combination	Threshold	APY	Sharpe	Mean Active Assets
5	FILTERING	T-SA-1	OPTI_PRECISION	19.180	1.089	4.77
5	FILTERING	COP_EN_E	OPTI_PRECISION	18.960	1.082	4.79
5	FILTERING	IL	OPTI_PRECISION	18.480	1.066	4.78
10	FILTERING	T-SA-1	OPTI_PRECISION	19.320	1.107	9.05
10	FILTERING	COP_EN_E	OPTI_PRECISION	19.070	1.097	9.13
10	FILTERING	IL	OPTI_PRECISION	18.460	1.072	9.03
20	FILTERING	T-SA-1	OPTI_PRECISION	19.290	1.107	16.27
20	FILTERING	COP_EN_E	OPTI_PRECISION	19.060	1.097	16.80
20	FILTERING	IL	OPTI_PRECISION	18.520	1.076	16.14
40	FILTERING	T-SA-1	OPTI_PRECISION	19.290	1.107	25.50
40	FILTERING	COP_EN_E	OPTI_PRECISION	19.070	1.098	27.48
40	FILTERING	IL	OPTI_PRECISION	18.530	1.077	26.07
50	FILTERING	T-SA-1	OPTI_PRECISION	19.280	1.107	28.06
50	FILTERING	COP_EN_E	OPTI_PRECISION	19.070	1.098	30.35
50	FILTERING	IL	OPTI_PRECISION	18.530	1.077	29.21
60	FILTERING	T-SA-1	OPTI_PRECISION	19.280	1.107	29.70
60	FILTERING	COP_EN_E	OPTI_PRECISION	19.080	1.098	32.25
60	FILTERING	IL	OPTI_PRECISION	18.530	1.077	31.47

Table B.3: Top 3 portfolios with the highest APY for each value of K. *It can be clearly observed that the APY and Sharpe ratio do not vary substantially depending on the maximum portfolio size K. This is mainly due to the bias introduced by selecting only the top K highest weights, which already capture most of the financial performance, while the additional assets primarily contribute to reducing the portfolio variance (i.e., the overall risk). However, it is also noticeable that as K increases, the number of active assets converges towards a value around 30, illustrating that the effective diversification tends to stabilize beyond this threshold.*

C

Additional Information On Metrics and Performance Evaluation

Classification Metrics

In order to assess the predictive performance of the models, we compute standard classification metrics. These are based on four quantities:

- True Positives (TP): correct predictions of the positive class.
- True Negatives (TN): correct predictions of the negative class.
- False Positives (FP): instances predicted positive but actually negative.
- False Negatives (FN): instances predicted negative but actually positive.

From these, we define:

- **Accuracy**: proportion of all predictions that are correct. It measures the overall rate at which the model correctly identifies whether a stock will outperform or underperform the benchmark.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Limitation: Accuracy can be misleading when the classes are imbalanced. For example, if most stocks underperform, a model always predicting "underperform" can still achieve high accuracy without ever identifying a winner.

- **Precision**: proportion of predicted positives that are actually positive. It measures how many of the stocks predicted to outperform did indeed outperform the benchmark.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Limitation: Maximizing precision can lead the model to be too conservative and select very few stocks, which may reduce potential profits by missing many true outperformers.

- **Recall**: proportion of actual positives that are correctly identified. It measures the ability of the classifier to find all the stocks that will outperform the benchmark.

$$\text{Recall} = \frac{TP}{TP + FN}$$

Limitation: Focusing only on recall can result in predicting many stocks as outperformers, increasing false positives and lowering the quality of recommendations.

- **F1 Score**: harmonic mean of precision and recall. It provides a balanced metric that accounts for both the ability to detect outperforming stocks and the reliability of these positive predictions.

$$F1 = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Limitation: The F1 score does not take into account true negatives, so it may overlook the overall accuracy and can be sensitive to class imbalance.

We therefore observe that it is essential to report all these measures in order to obtain a complete picture of the classifier's performance. This is particularly crucial because we do not know in advance which type of error, such as a false positive or a false negative, will have the greatest financial impact. For example, missing out on a stock that actually outperforms could result in significant opportunity costs for the portfolio, while incorrectly predicting that a stock will outperform when it does not could result in unnecessary losses. Reporting on all indicators allows us to understand these trade-offs and evaluate the model's behavior in different scenarios.

Example

Consider the following example with 10 predictions:

- Balanced dataset: 5 positives and 5 negatives.

```
y_true = [1,1,1,1,1,0,0,0,0,0]
y_predicted = [1,0,1,1,0,0,1,0,0,0]
```

Here:

$$TP = 3, \quad TN = 4, \quad FP = 1, \quad FN = 2.$$

Metrics:

$$\text{Accuracy} = 0.7, \quad \text{Precision} = 0.75, \quad \text{Recall} = 0.6, \quad F1 = 0.6667.$$

- Imbalanced dataset: 8 negatives and 2 positives.

```
y_true = [1,1,0,0,0,0,0,0,0,0]
y_pred = [0,0,0,0,0,0,0,0,1,0]
```

Here:

$$TP = 0, \quad TN = 7, \quad FP = 1, \quad FN = 2.$$

Metrics:

$$\text{Accuracy} = 0.7, \quad \text{Precision} = 0, \quad \text{Recall} = 0, \quad F1 = 0.$$

Despite the same accuracy, the second classifier fails completely to detect positives, illustrating why accuracy alone is misleading on unbalanced data.

Backtesting Metrics

After selecting stocks, we evaluate the economic performance of the resulting portfolios. Let r_t denote the return of the portfolio in period t for $t = 1, \ldots, T$ and $\bar{r}$ its mean. We define V_0 as the initial value of the portfolio at the beginning of the backtest and V_T as its final value.

- **Total Return:**
$$R = \frac{V_T - V_0}{V_0}$$

- **Annualized Return:** measures the returns obtained over a specific period of time, converted to a one-year period. This scaling allows us to compare the returns of any asset over a given period.
$$AR = \left(\frac{V_T}{V_0}\right)^{1/Y} - 1$$
where Y is the number of years in the backtest.

- **Win Rate:** It measures the fraction of positions that yielded a positive return:
$$WR = \frac{1}{T}\sum_{t=1}^{T} 1_{\{r_t > 0\}}$$

- **Average Gain / Loss:** when we make a profit or suffer a loss, this is given by:
$$AG = \frac{\sum r_t 1_{\{r_t>0\}}}{\sum 1_{\{r_t>0\}}}, \quad AL = \frac{\sum r_t 1_{\{r_t<0\}}}{\sum 1_{\{r_t<0\}}}$$

- **Annualized Volatility:** measures the dispersion of the returns around their mean,

$$\sigma_{ann} = \sqrt{\frac{\sum(r_t - \bar{r})^2}{T-1}} \times \sqrt{K}$$

where K is the scaling factor (e.g., 252 for daily data).

- **Value at Risk (VaR)** is the maximum expected loss of a portfolio over a given time period and confidence interval. Let L_T be the random variable representing the loss of a portfolio over a period of length T. The VaR at confidence level $1-\alpha$ is defined as the smallest number $VaR_{1-\alpha}$ such that:

$$\mathbb{P}\Big\{L_T \leq VaR_{1-\alpha}\Big\} \geq 1-\alpha$$

This means that with probability at least $1-\alpha$, the loss will not exceed $VaR_{1-\alpha}$.

- **Conditional Value at Risk (CVaR)** measures the average loss incurred beyond the VaR threshold. It provides additional information about the tail risk, quantifying how severe the losses can be when they exceed VaR:

$$CVaR_{1-\alpha} = |T|^{-1} \sum_{L_T > VaR_{1-\alpha}} L_T$$

- **Sharpe Ratio:** measures the average returns obtained in excess of a risk-free asset per unit of volatility.

$$SR = \frac{\bar{r} - r_f}{\bar{\sigma}}$$

where r_f is the risk-free rate.

These metrics allow us to quantify not only the returns but also the risks and efficiency of the strategies.

D

Financial Interpretation of Technical Indicators

Indicator	Financial Meaning
log(Price / SMA)	Measures the deviation of the current price relative to its simple moving average; indicates short-term trends or overbought/oversold conditions.
log(Price / EMA)	Similar to SMA but gives more weight to recent observations; captures more reactive price trends.
MACD level and signal	Captures momentum by comparing short-term and long-term exponential moving averages; used to identify bullish or bearish momentum.
MACD crossover	Binary signal indicating whether the MACD line has crossed the signal line, often used as a buy/sell trigger.
RSI	Measures the speed and change of price movements; values above 70 often signal overbought conditions, below 30 oversold.
log(Price / Upper Bollinger Band)	Indicates proximity of price to the upper volatility band; can highlight overextension.
log(Price / Lower Bollinger Band)	Indicates proximity of price to the lower volatility band; can highlight potential undervaluation.
Weekly return	Captures raw return over the past week; basic performance measure.
Excess return (vs index)	Measures outperformance relative to a benchmark index over the period.
Lagged stock return	Historical return over the past periods, used to assess persistence or mean reversion.
Price momentum	Cumulative return over the specified period; positive momentum often signals continuation.
Beta (stock vs index)	Measures the stock's sensitivity to market movements; higher beta implies higher market risk.
Rolling return volatility	Captures recent variability of returns; a proxy for risk or uncertainty.
Return / Volatility (Sharpe proxy)	Simplified risk-adjusted return measure; higher values are preferable.
Return-volume correlation	Indicates whether rising volumes coincide with rising returns; can signal strength of trends.
Regression slope (price trend)	Estimates the linear trend in price; positive slope implies upward drift.
Weekly RSI	Same as daily RSI but computed on weekly data; smoother and longer-term overbought/oversold signals.
Return skewness	Measures asymmetry of return distribution; positive skew can imply frequent small losses and occasional large gains.
On-Balance Volume	Cumulative measure of volume adjusted for price direction; tracks whether volume confirms price trends.
Dollar trading volume	Total trading value; proxies liquidity and investor interest.

Financial Interpretation of Technical Indicators. *Each indicator provides specific insights into price trends, volatility, momentum, or liquidity.*

E

Robustness tests

E.1 Hyperparameters

Portfolio	Ensemble Method	Threshold	SR	APY [%]	Vol. [%]	DD [%]	VaR [%]	CVaR	WR [%]
FILT.	**T-SA-CV**	QUANTILE	**0.88**	**14.03**	**15.21**	**-33.72**	**-3.12**	**-4.83**	**58.85**
FILT.	T-SA-CV	NAIVE	0.87	13.99	15.23	-33.68	-3.11	-4.83	58.74
FILT.	T-SA-CV	OPTI_{recall}	0.87	13.96	15.23	-33.68	-3.11	-4.83	58.74
FILT.	T-SA-CV	CV	0.87	13.94	15.23	-33.72	-3.11	-4.83	58.74
FILT.	T-SA-CV	$\text{OPTI}_{PRECISION}$	0.87	13.93	15.23	-33.72	-3.11	-4.83	58.74
FILT.	T-SA-4	CV	0.86	13.74	15.12	-29.75	-3.06	-4.67	58.10
FILT.	T-SA-CV	OPTI	0.86	13.79	15.20	-33.72	-3.13	-4.82	58.85
FILT.	T-SA-4	QUANTILE	0.86	13.67	15.12	-29.75	-3.06	-4.67	58.10
FILT.	T-SA-4	OPTI	0.86	13.67	15.15	-29.75	-3.06	-4.70	58.10
FILT.	T-SA-4	NAIVE	0.86	13.66	15.15	-29.75	-3.06	-4.70	58.10
FILT.	T-SA-4	OPTI_{recall}	0.86	13.66	15.14	-29.75	-3.06	-4.70	58.10
FILT.	T-SA-4	$\text{OPTI}_{PRECISION}$	0.86	13.65	15.15	-29.71	-3.06	-4.70	58.10
FILT.	T-SA-2	NAIVE	0.85	13.54	15.07	-36.30	-3.13	-4.80	59.17
FILT.	T-SA-2	OPTI_{recall}	0.85	13.52	15.07	-36.30	-3.13	-4.80	59.17
FILT.	T-SA-2	QUANTILE	0.85	13.51	15.07	-36.37	-3.13	-4.80	59.17
	INDEX NASDAQ-100		***0.74***	***14.72***	***19.7***	***-51.53***	***-4.66***	***-6.79***	***61.73***

Table E.1: Robustness test around hyperparameters optimization. *This table shows the top 15 portfolio rankings when hyperparameter optimization is performed using precision score instead of accuracy.*

In this robustness test, we changed the evaluation metric used for hyperparameters tuning from accuracy to precision. This modification had a significant impact on the model selection and final portfolio rankings. Unlike the previous configuration, most portfolios in the top 15 are now based on trimmed simple average (T-SA) combination methods, with T-SA-CV emerging as the best-performing custom ensemble. This suggests that T-SA strategies are particularly consistent and resilient in financial forecasting tasks, even when the optimization criterion is modified.

Despite these changes, one key element remains stable: the FILT.-based portfolio optimization method (FILT.) consistently dominates the top rankings. This highlights its robustness across different model configurations and confirms its contribution to the overall performance of the strategies.

However, it is important to note that, although some portfolios still outperform the benchmark in terms of Sharpe ratio, none of them surpass it in terms of absolute performance (APY). This result suggests that optimizing models based solely on precision may not fully capture the opportunity cost or the magnitude of return differences, especially in a financial context where ranking errors have asymmetric impacts.

E.2 Fees

Portfolio	Ensemble Method	Threshold	SR	APY [%]	Vol. [%]	DD [%]	VaR [%]	CVaR	WR [%]
EW	**CO**	$OPTI_{recall}$	**0.13**	**1.59**	**19.89**	**-56.06**	**-4.67**	**-7.03**	**57.78**
EW	COP_EN	$OPTI_{recall}$	0.07	0.32	19.92	-59.05	-4.78	-7.08	57.78
EW	COP_L2_E	$OPTI_{recall}$	0.02	-0.71	19.90	-55.75	-4.73	-7.06	57.04
EW	COP_L2_IL	$OPTI_{recall}$	-0.05	-2.05	19.93	-59.14	-4.76	-7.08	56.72
EW	COP_CE_E	$OPTI_{recall}$	-0.14	-3.84	19.97	-63.30	-4.83	-7.15	54.80
EW	COP_CE_IL	$OPTI_{recall}$	-0.17	-4.28	19.98	-67.19	-4.80	-7.14	55.01
EW	COS_IL	$OPTI_{recall}$	-0.19	-4.77	19.94	-66.75	-4.76	-7.13	55.44
EW	COP_EN_IL	$OPTI_{recall}$	-0.23	-5.42	19.93	-71.18	-4.83	-7.18	54.80
EW	COP_EN_E	$OPTI_{recall}$	-0.24	-5.63	19.95	-66.79	-4.83	-7.16	54.37
EW	COS_E	$OPTI_{recall}$	-0.26	-6.02	19.93	-74.40	-4.81	-7.16	55.22
EW	COP_L1_IL	$OPTI_{recall}$	-0.30	-6.70	19.89	-76.97	-4.80	-7.19	54.37
Shrinked_MV	CO	$OPTI_{recall}$	-0.32	-3.87	12.79	-54.75	-3.30	-4.44	51.60
MV	CO	$OPTI_{recall}$	-0.38	-4.51	12.79	-60.26	-3.23	-4.41	50.85
EW	COP_L1_E	$OPTI_{recall}$	-0.38	-8.21	19.85	-79.20	-4.83	-7.18	53.30
Shrinked_MV	COP_EN	$OPTI_{recall}$	-0.43	-5.20	12.91	-64.79	-3.34	-4.54	50.53
INDEX NASDAQ-100			***0.74***	***14.72***	***19.7***	***-51.53***	***-4.66***	***-6.79***	***61.73***

Table E.2: Robustness test around fees. *This table shows the top 15 portfolio rankings when fees is changed and constantly is 1% of traded value.*

As expected, the increase in transaction costs has a significant impact on portfolio performance. Given the high frequency of reallocations in our strategy (weekly), these costs accumulate quickly and significantly reduce returns. This robustness test clearly shows that most portfolios underperform the benchmark index and that almost all of them have a negative annualized performance (APY).

These results highlight a major limitation of high-turnover strategies in realistic market conditions. While some models may generate good signals in theory, their profitability

can be entirely offset by frictions such as transaction costs. This underscores the need to directly incorporate cost considerations into model and portfolio design, particularly in cases of frequent rebalancing.

E.3 Time period

Portfolio	Ensemble Method	Threshold	SR	APY [%]	Vol. [%]	DD [%]	VaR [%]	CVaR	WR [%]
FILT.	**T-SA-3**	QUANTILE	**0.96**	**13.34**	**12.86**	**-20.75**	**-2.50**	**-4.16**	**43.18**
FILT.	T-SA-3	CV	0.96	13.32	12.86	-20.75	-2.50	-4.15	43.18
FILT.	T-SA-3	OPTI_{recall}	0.96	13.31	12.86	-20.75	-2.50	-4.16	43.18
FILT.	T-SA-3	NAIVE	0.96	13.31	12.86	-20.75	-2.50	-4.16	43.18
FILT.	T-SA-3	OPTI	0.96	13.31	12.86	-20.75	-2.50	-4.16	43.18
FILT.	T-SA-3	$\text{OPTI}_{PRECISION}$	0.96	13.28	12.86	-20.75	-2.50	-4.15	43.18
FILT.	T-SA-CV	QUANTILE	0.92	12.57	12.65	-19.24	-2.51	-4.09	42.75
FILT.	T-SA-CV	OPTI	0.92	12.56	12.65	-19.24	-2.51	-4.09	42.75
FILT.	T-SA-CV	CV	0.92	12.55	12.65	-19.24	-2.51	-4.09	42.75
FILT.	T-SA-CV	NAIVE	0.92	12.55	12.65	-19.24	-2.51	-4.09	42.75
FILT.	T-SA-CV	OPTI_{recall}	0.92	12.55	12.65	-19.24	-2.51	-4.09	42.75
FILT.	T-SA-CV	$\text{OPTI}_{PRECISION}$	0.92	12.53	12.65	-19.24	-2.51	-4.09	42.75
FILT.	COP_L1_E	OPTI_{recall}	0.91	12.39	12.56	-21.12	-2.71	-4.20	43.60
Shrinked_MV	T-SA-CV	OPTI	0.91	11.67	11.86	-19.04	-2.43	-4.06	44.46
Shrinked_MV	T-SA-CV	QUANTILE	0.91	11.67	11.86	-19.04	-2.43	-4.06	44.56
INDEX NASDAQ-100			***0.64***	***12.48***	***19.98***	***-35.24***	***-4.94***	***-7.15***	***62.02***

Table E.3: Robustness test with 2010–2022 period. *This table shows the top 15 portfolio rankings when the time period is adjusted to 2010–2022. The benchmark index is also shifted accordingly.*

A final robustness test is performed to assess how results change when the period under consideration is modified. Naturally, overall returns and volatilities are influenced by specific market conditions during the new period. However, the main objective is to observe whether the best-performing portfolios remain associated with similar optimization approaches and threshold choices.

Once again, we find that the best-performing portfolios are systematically constructed using the Filtering (FILT.) optimization method, confirming its robustness across all periods. In addition, ensemble models based on combinations of truncated averages continue to dominate the rankings. With regard to threshold selection methods, the results appear to be more diverse among the top 15. It should be noted that precision-based threshold optimization is no longer overrepresented, suggesting that its previous dominance may have been specific to the initial period.

F

Additional Figures

Figure F.1: Yearly evolution of classification metrics. *The figure shows the yearly out-of-sample performance of each model for four key metrics: accuracy, precision, recall, and F1-score. Each line corresponds to a specific model, allowing a visual assessment of its temporal behavior. While most models exhibit fluctuations over time, metrics such as accuracy and recall appear relatively stable. In contrast, metrics like recall and F1-score show more pronounced volatility, reflecting the challenges of maintaining consistent classification performance in financial time series.*

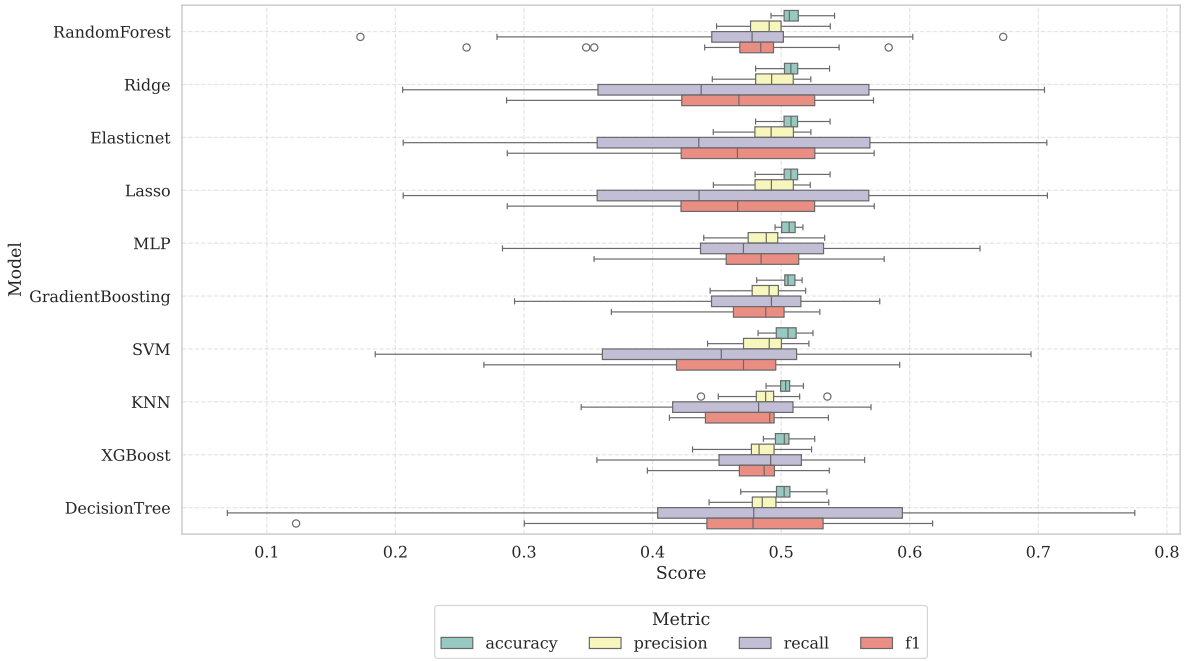


Figure F.2: Boxplot of classification metrics (sorted by mean accuracy) *Same as Figure 5.2, this plot shows the distribution of classification metrics across all out-of-sample years, using a walk-forward validation. Here, models are sorted by decreasing mean accuracy instead of median. This ordering highlights that linear models such as* ***ElasticNet****,* ***Lasso****, and* ***Ridge*** *achieve higher average accuracy than more complex models like* ***MLP*** *or* ***GradientBoosting****, despite having slightly lower median performance.* In this case, and **RandomForest** exhibits the best mean accuracy.

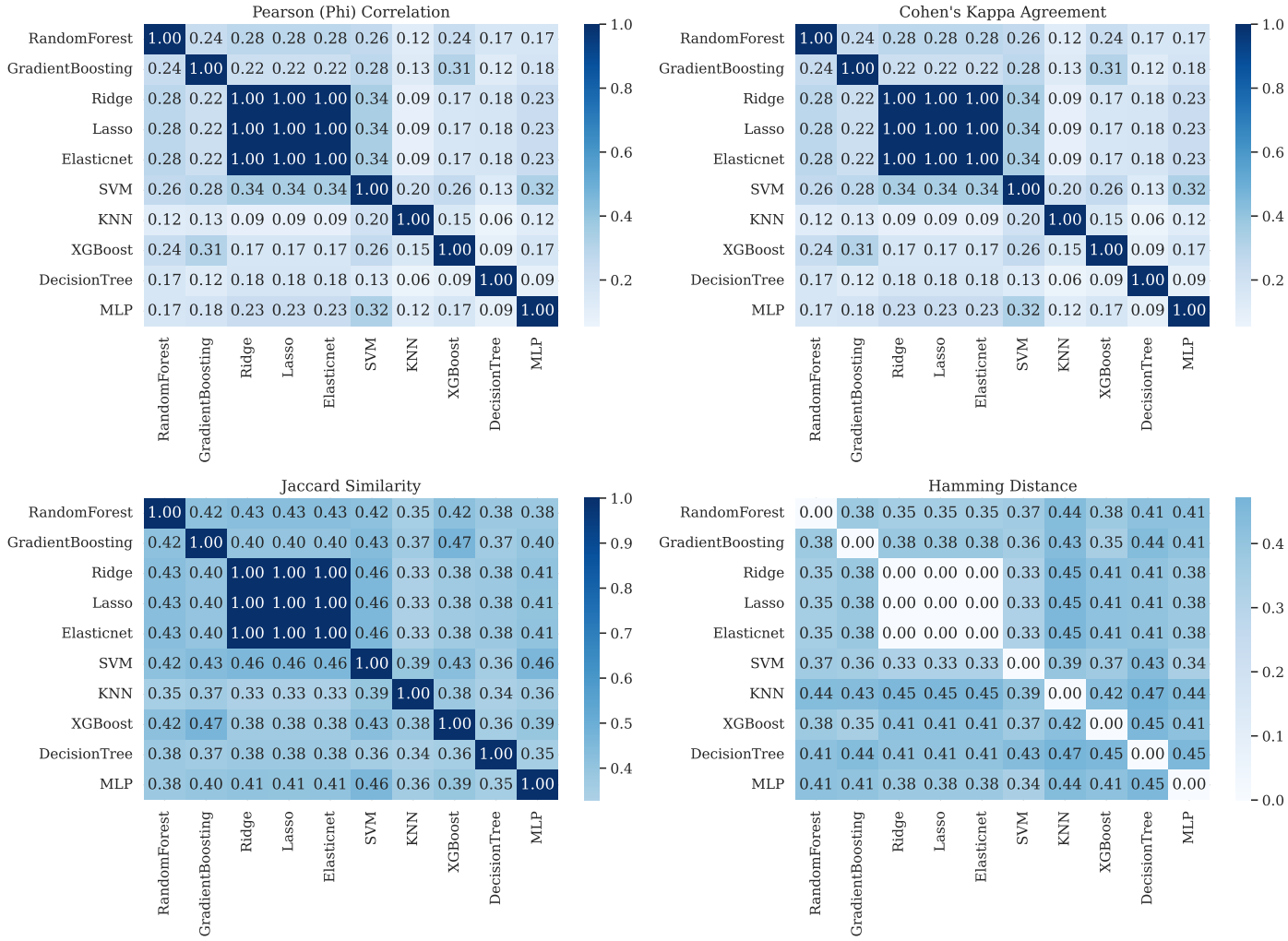


Figure F.3: Similarity between model predictions across all test periods. *The matrices show the pairwise similarity between model outputs using four complementary correlation metrics. Pearson's correlation captures the linear association between prediction vectors, while Cohen's Kappa coefficient adjusts for random agreement. Jaccard similarity focuses on the overlap of positive predictions, and Hamming distance quantifies the overall proportion of divergent predictions. Regularized linear models (`Ridge`, `Lasso`, `ElasticNet`) consistently exhibit near-identical predictions, regardless of the metric used. This is also confirmed when we look at the year-by-year correlation in Figure F.4. Tree-based models and neural networks display more heterogeneous behavior, with weaker alignment both with each other and with linear models.This suggests that differences in prediction patterns are primarily explained by model families, rather than by randomness in the data or training process.*

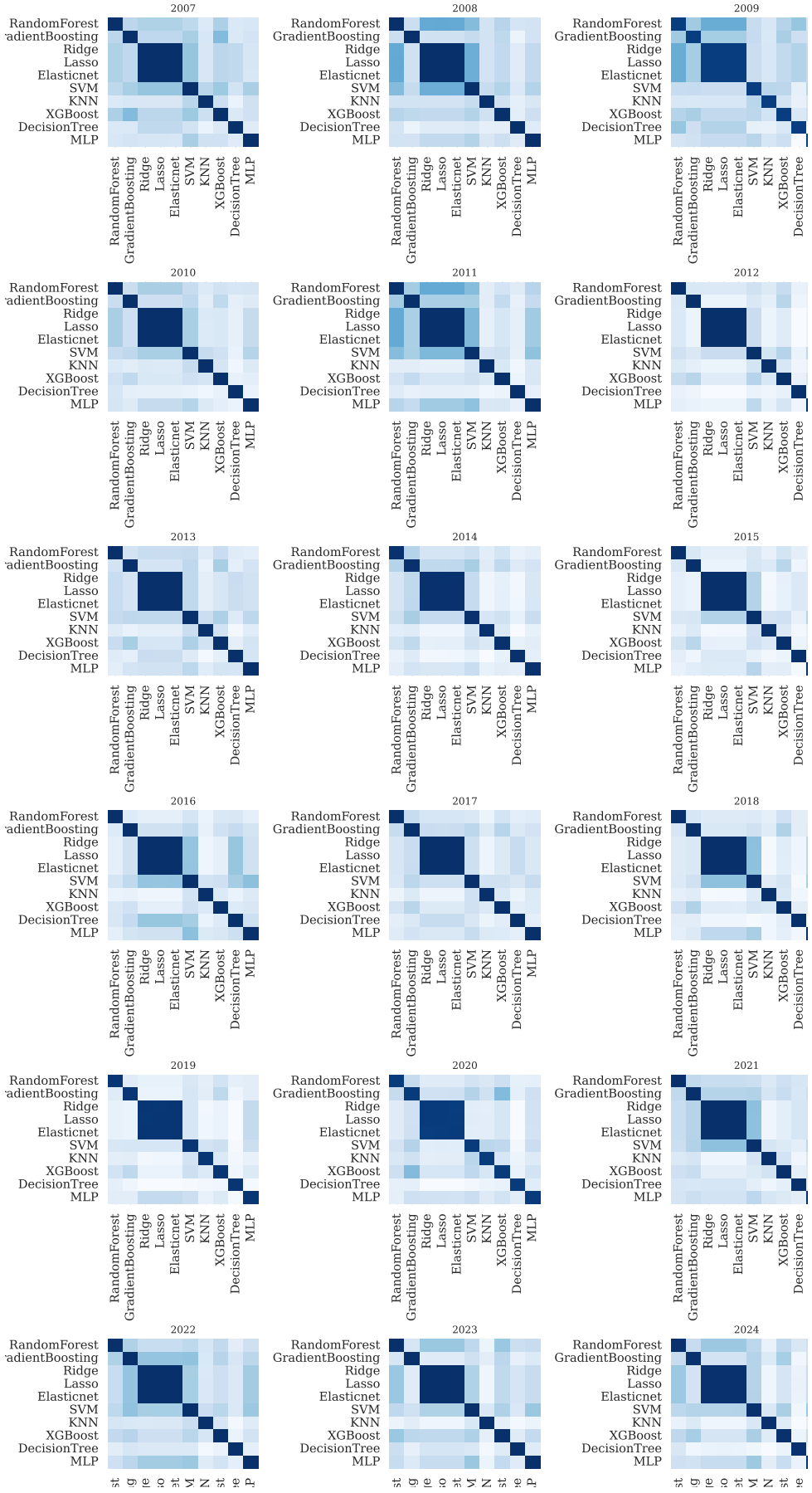


Figure F.4: Yearly Pearson (Phi) Correlation Between Model Predictions *This figure shows the year-by-year correlation (Phi coefficient) between model predictions based on out-of-sample walk-forward validation. The color intensity ranges from light (low correlation, $\phi \approx 0$) to dark blue (high correlation, $\phi \approx 1$)*

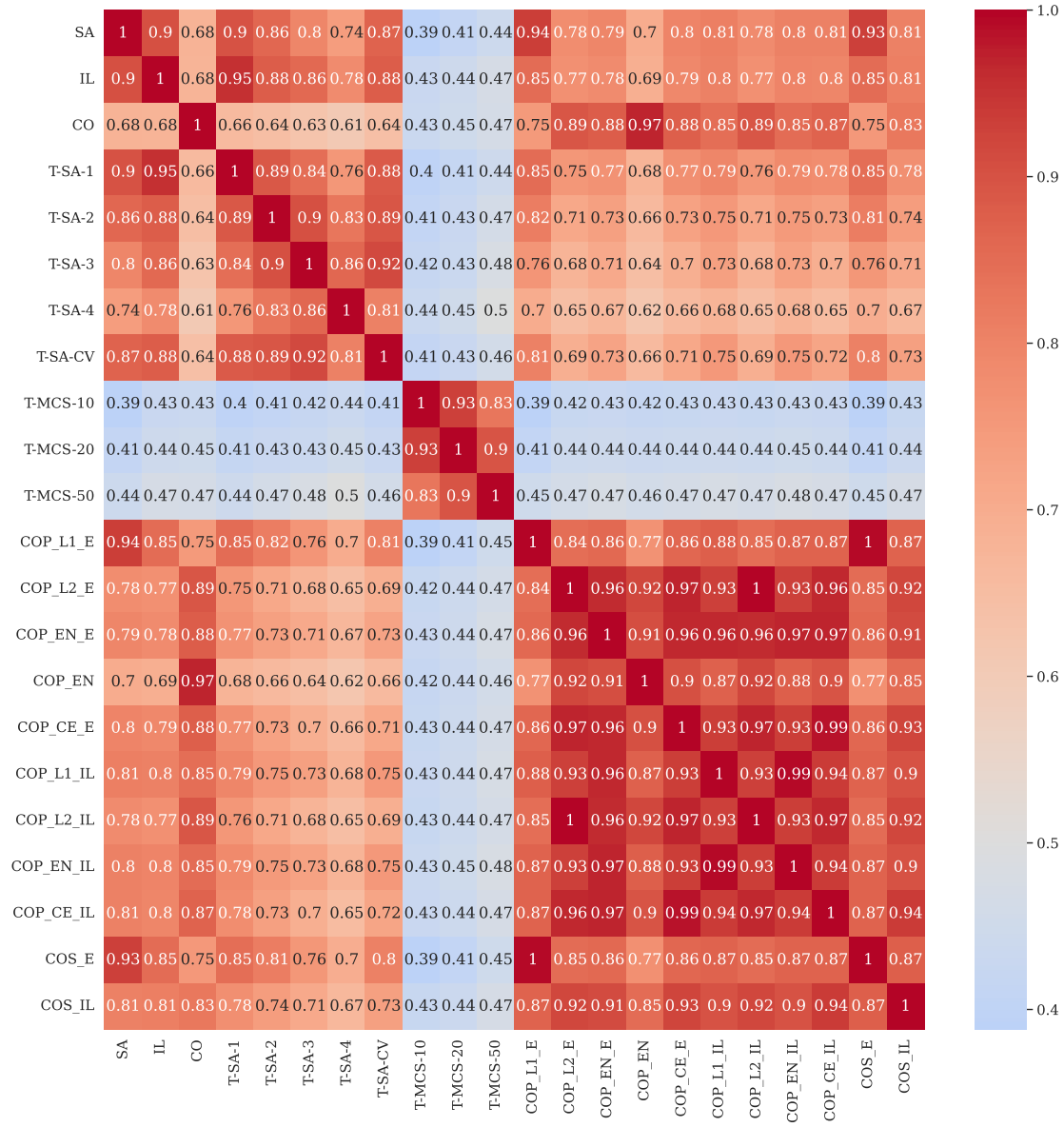


Figure F.5: Correlation between aggregated ensemble forecast predictions *This figure shows the correlation between model aggregated predictions based on out-of-sample walk-forward validation. The color intensity ranges from light blue (low correlation, $\phi \approx 0$) to dark red (high correlation, $\phi \approx 1$)*

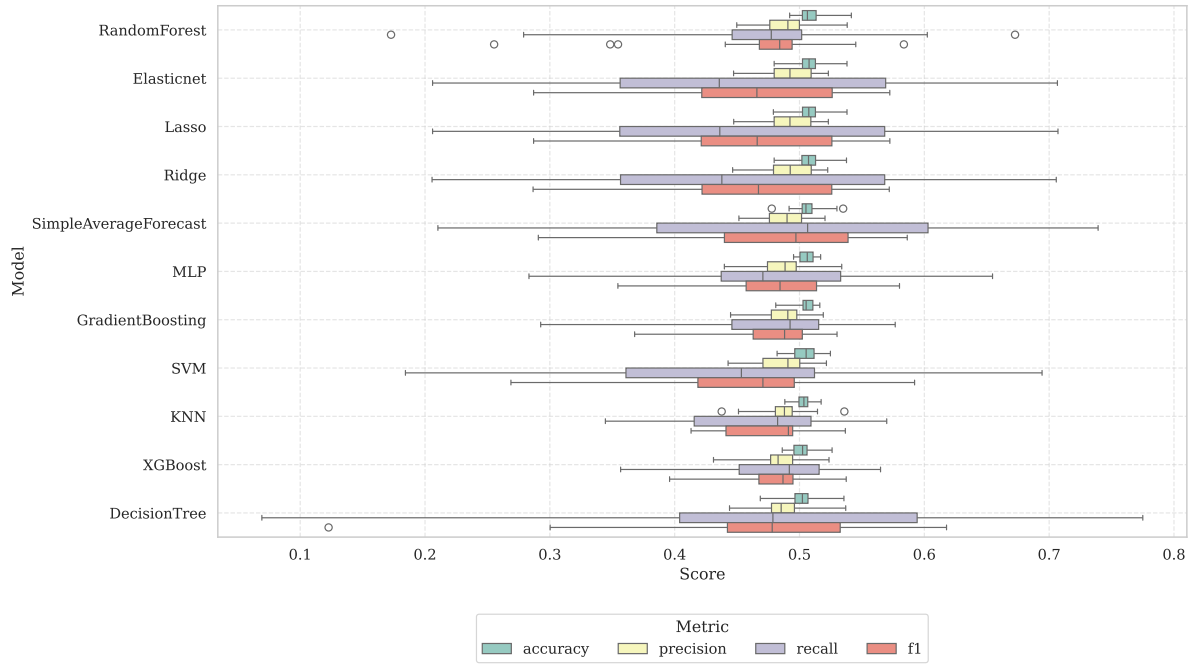


Figure F.6: Distribution of ensemble model performance across years. *This boxplot illustrates the annual dispersion of accuracy, precision, recall and F1-score for each ensemble method. This visualization highlights the greater year-to-year variability for most models. The wider interquartile ranges and presence of outliers suggest that while some methods achieve strong average performance, their stability over time may be limited.*

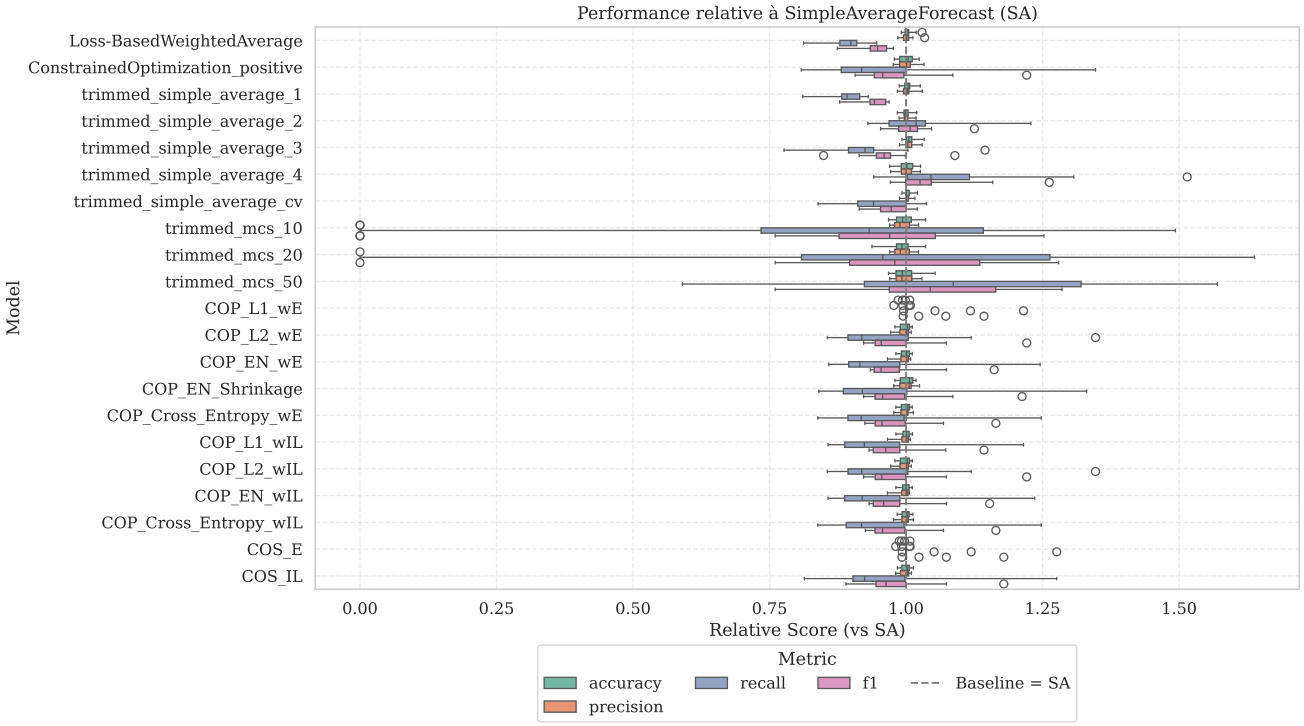


Figure F.7: Relative performance of ensemble and individual forecasts, sorted by precision. *This figure displays the average performance of all ensemble and individual models, sorted in ascending order based on their* ***precision*** *score. Each point corresponds to the score over the aggregated predictions, using a fixed classification threshold* $\tau = 0.5$*. Dashed vertical lines represent statistical significance thresholds (5%, 1%, and 0.1%) relative to a random baseline (accuracy = recall = 50%, precision = 48.5%).*

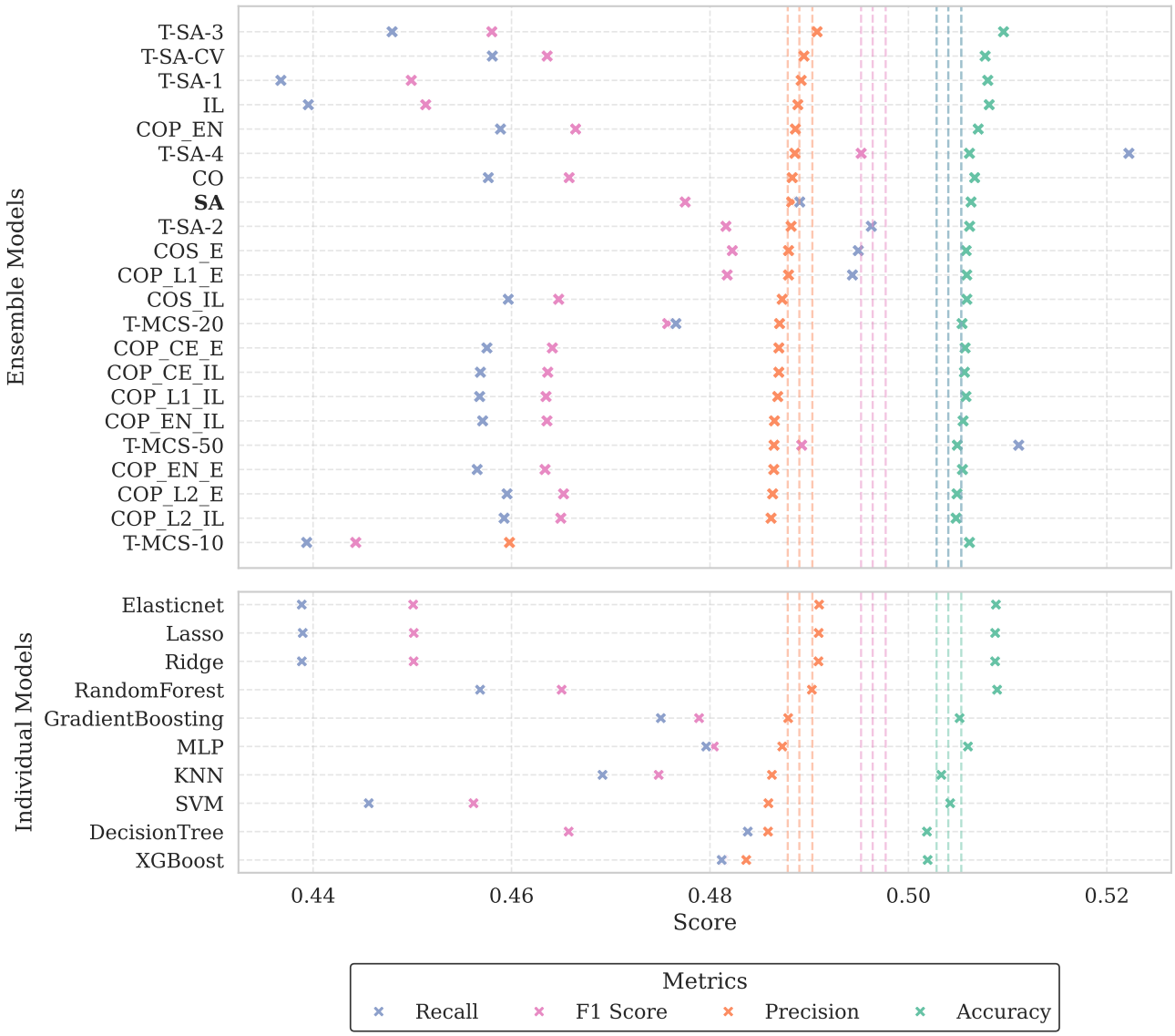


Figure F.8: Relative performance of ensemble and individual forecasts, sorted by precision. *This figure displays the average performance of all ensemble and individual models, sorted in ascending order based on their* ***precision*** *score. Each point corresponds to the score over the aggregated predictions, using a fixed classification threshold* $\tau = 0.5$*. Dashed vertical lines represent statistical significance thresholds (5%, 1%, and 0.1%) relative to a random baseline (accuracy = recall = 50%, precision = 48.5%).*

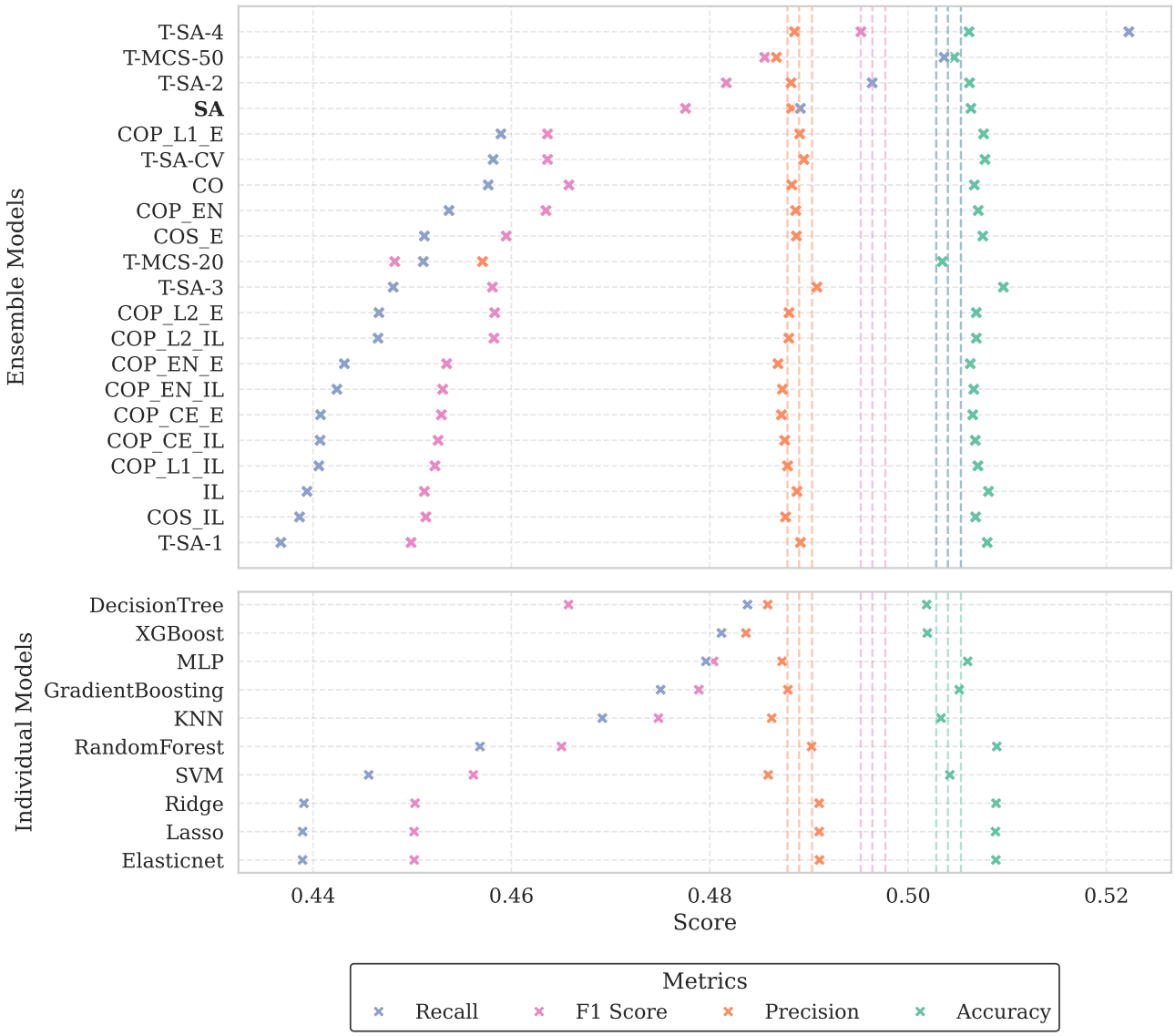


Figure F.9: Relative performance of ensemble and individual forecasts, sorted by recall. *This figure displays the average performance of all ensemble and individual models, sorted in ascending order based on their* ***recall*** *score. Each point corresponds to the score over the aggregated predictions, using a fixed classification threshold* $\tau = 0.5$*. Dashed vertical lines represent statistical significance thresholds (5%, 1%, and 0.1%) relative to a random baseline (accuracy = recall = 50%, precision = 48.5%).*

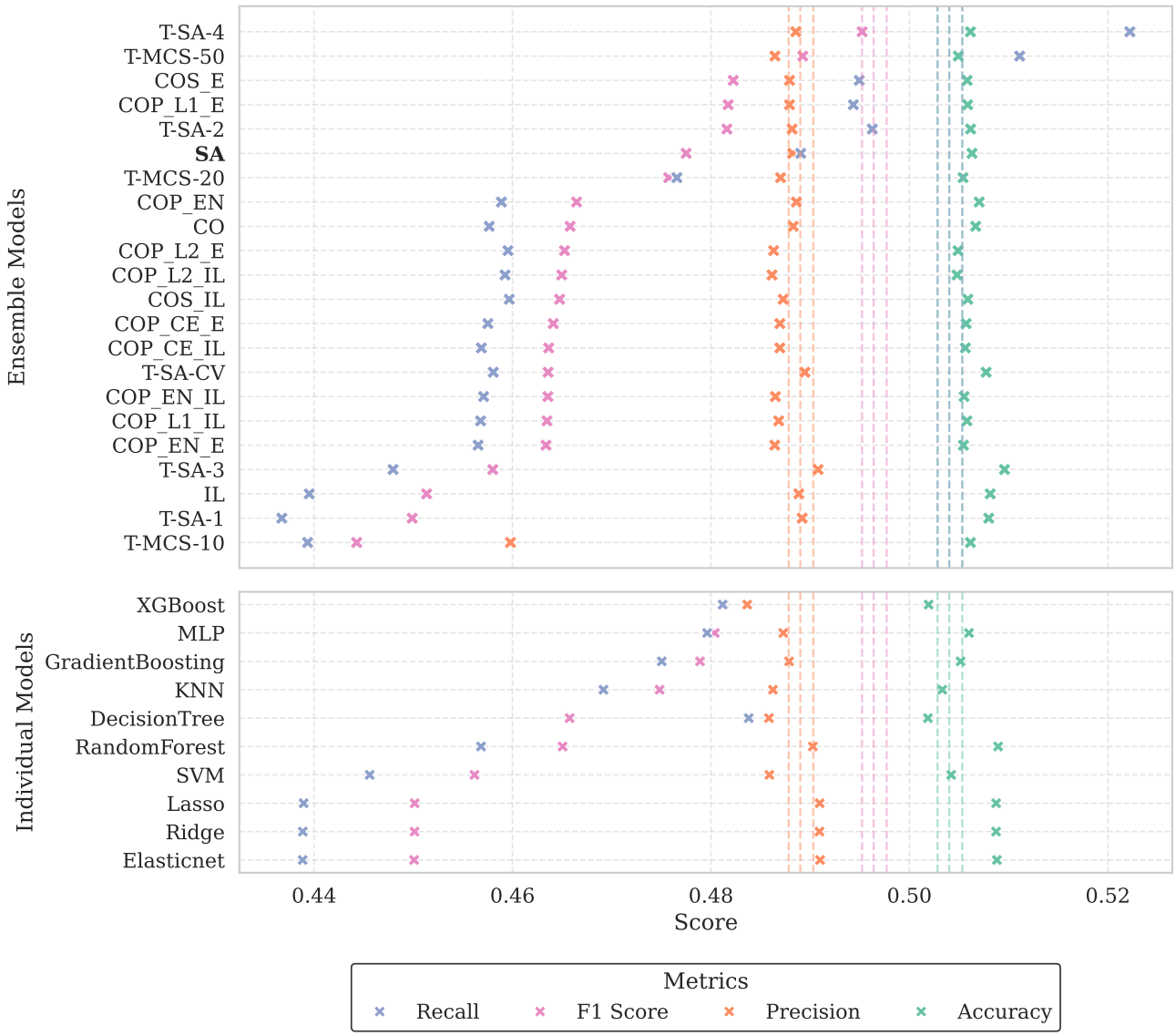


Figure F.10: Relative performance of ensemble and individual forecasts, sorted by F1Score. *This figure displays the average performance of all ensemble and individual models, sorted in ascending order based on their **F1** score. Each point corresponds to the score over the aggregated predictions, using a fixed classification threshold $\tau = 0.5$. Dashed vertical lines represent statistical significance thresholds (5%, 1%, and 0.1%) relative to a random baseline (accuracy = recall = 50%, precision = 48.5%).*

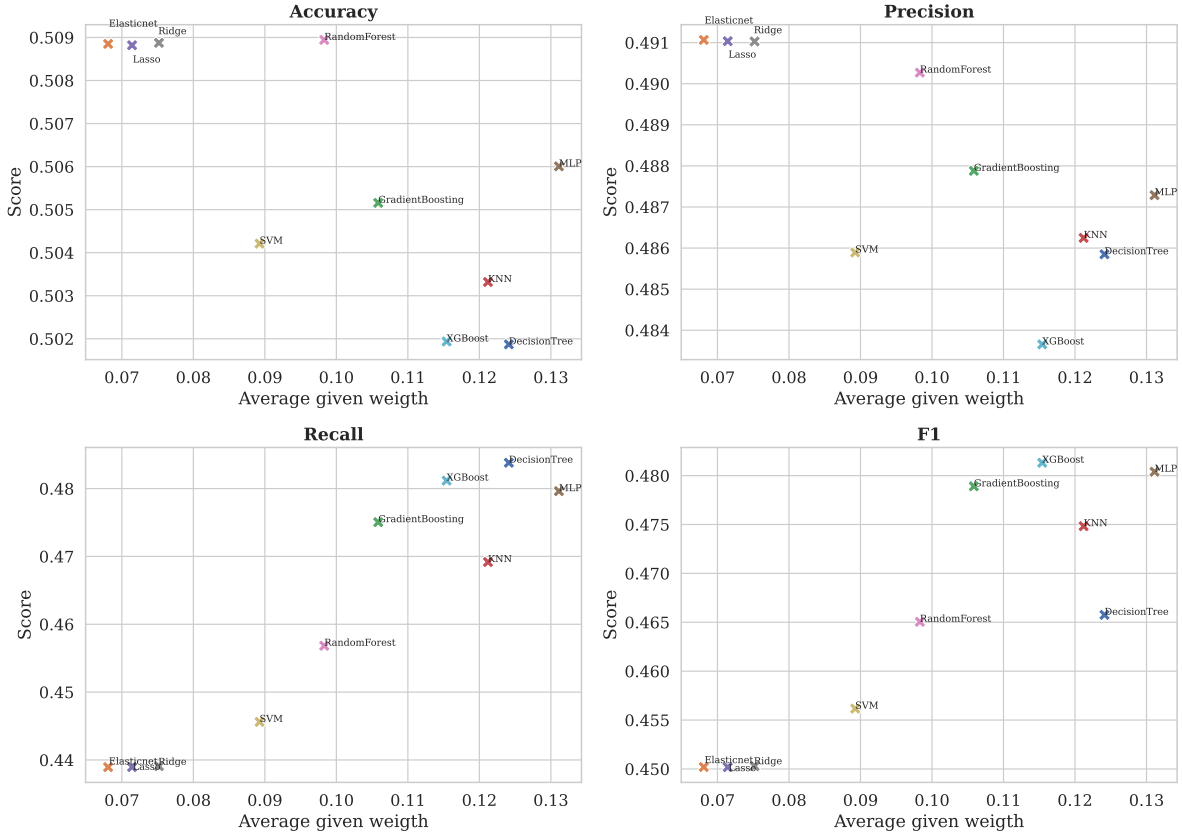


Figure F.11: Relationship Between Average Weights and Individual Model Performance

This set of scatter plots illustrates how the average weight assigned to each base model across all ensemble methods and years correlates with its standalone performance on four metrics: accuracy, precision, recall, and F1-score. While some models like MLP or KNN tend to receive high weights and also achieve strong F1 and recall scores, others such as Ridge or Lasso are consistently under-weighted despite competitive precision. However, since those models were highly correlated, it means that together joins it has a higher weight than average models.

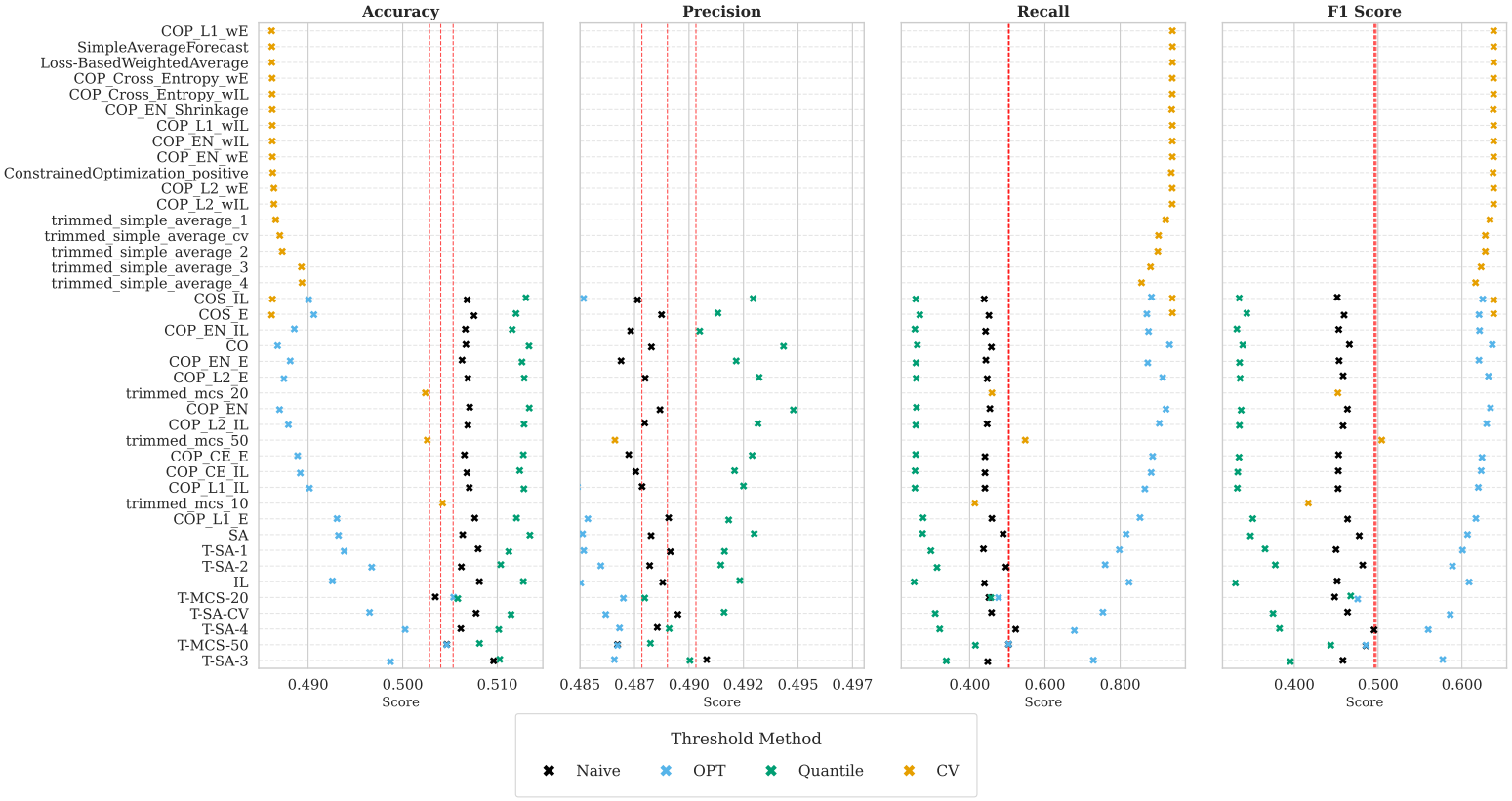


Figure F.12: Performance of Thresholding Strategies Across Ensemble Models and Evaluation Metrics based on Recall optimization

This figure compares the performance of four thresholding strategies across all ensemble models and four evaluation metrics. Each cross represents the score obtained by a given ensemble method under a specific thresholding approach.The dashed vertical lines represent statistical significance thresholds at the 5%, 1%, and 0.1% levels (from left to right), computed using a one-sided binomial test under the null hypothesis of a random classifier (i.e., accuracy = recall = 50%, precision = 48.5%). This visualization highlights the potential performance in recall when we switch the optimization from accuracy to a recall approach.

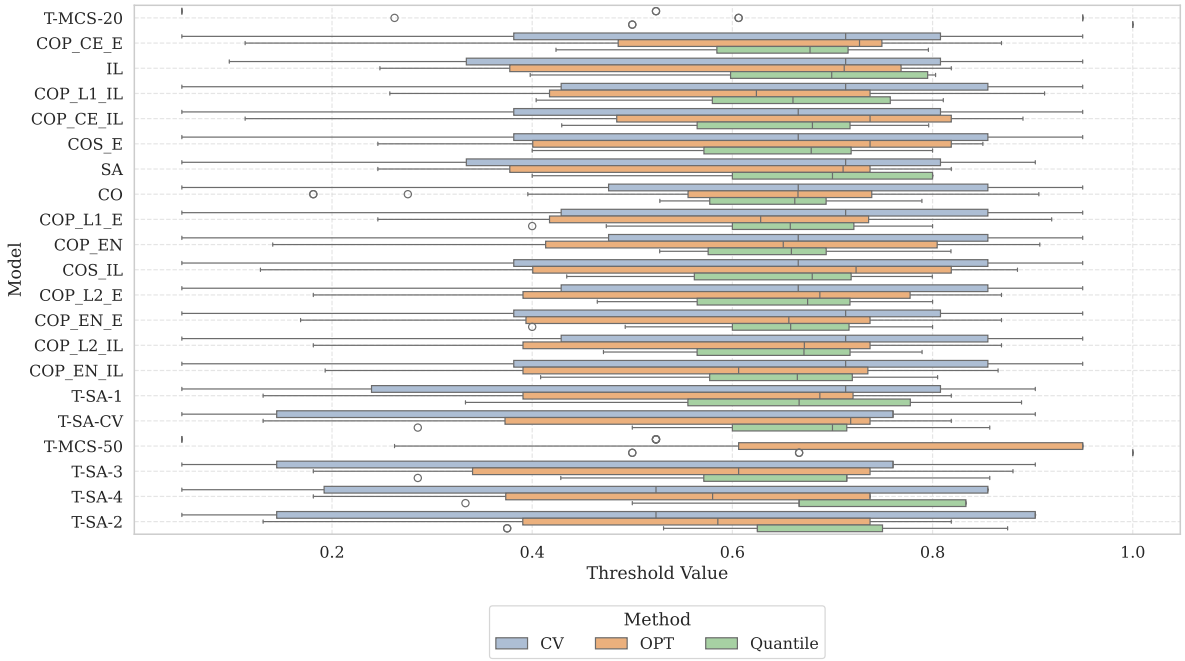


Figure F.13: Distribution of the mean thresholding value by ensemble model *The boxplots illustrate the variability of the threshold selection across methods, highlighting substantial dispersion within and between models. This variability suggests that the choice of thresholding approach can significantly impact the resulting ensemble performance and requires careful calibration.*

G
Additional Tables

model	Predicted **0**	Predicted **1**	% of class 1
XGBoost	43964	40593	48.01
DecisionTree	43964	40593	48.01
MLP	44475	40082	47.40
GradientBoosting	44839	39718	46.97
KNN	45150	39407	46.60
RandomForest	46598	37959	44.89
SVM	47277	37280	44.09
Lasso	48125	36432	43.09
Ridge	48130	36427	43.08
Elasticnet	48139	36418	43.07

Table G.1: Distribution of predicted classes (0 and 1) across models. The last column reports the percentage of predicted positive classes (1) out of the total number of predictions. This table highlights potential class imbalance or bias in model outputs.

	ACCURACY				PRECISION			
model	**NAIVE**	**CV**	**OPT**	**QTL**	**NAIVE**	**CV**	**OPT**	**QTL**
CO	50.67%***	50.71%***	51.09%***	51.33%***	48.83%*	48.94%**	49.11%***	49.43%***
COP_CE^{E}	50.65%***	51.12%***	51.15%***	51.27%***	48.72%	49.47%***	49.37%***	49.29%***
COP_CE^{IL}	50.68%***	51.02%***	51.16%***	51.24%***	48.76%	49.43%***	49.47%***	49.21%***
COP_EN	50.71%***	50.77%***	50.96%***	51.34%***	48.87%*	49.07%***	48.94%**	49.48%***
COP_EN^{E}	50.63%***	51.23%***	51.04%***	51.26%***	48.69%	49.39%***	49.29%***	49.22%***
COP_EN^{IL}	50.66%***	51.28%***	51.05%***	51.16%***	48.73%	49.27%***	49.25%***	49.05%***
COP_L1^{E}	50.76%***	51.08%***	51.20%***	51.20%***	48.91%**	49.16%***	49.44%***	49.18%***
COP_L1^{IL}	50.70%***	51.12%***	51.22%***	51.28%***	48.78%*	49.17%***	49.47%***	49.25%***
COP_L2^{E}	50.69%***	50.99%***	51.20%***	51.28%***	48.80%*	49.39%***	49.22%***	49.32%***
COP_L2^{IL}	50.69%***	51.05%***	51.20%***	51.28%***	48.80%*	49.51%***	49.31%***	49.32%***
COS^{E}	50.75%***	51.10%***	51.24%***	51.20%***	48.87%*	49.17%***	49.15%***	49.13%***
COS^{IL}	50.68%***	51.14%***	51.15%***	51.30%***	48.76%	49.51%***	49.35%***	49.29%***
IL	50.81%***	51.43%***	51.23%***	51.28%***	48.88%*	49.29%***	49.18%***	49.23%***
SA	50.63%***	51.43%***	51.27%***	51.34%***	48.83%*	49.51%***	49.20%***	49.30%***
T-MCS-10	50.34%*	50.68%***	50.49%**	50.58%***	45.71%	48.91%**	45.92%	48.80%*
T-MCS-20	50.47%**	50.59%***	50.73%***	50.81%***	48.67%	48.79%*	48.81%*	48.82%*
T-MCS-50	50.80%***	51.14%***	50.97%***	51.12%***	48.91%**	49.21%***	48.99%**	49.16%***
T-SA-1	50.62%***	51.01%***	50.89%***	51.04%***	48.82%*	49.24%***	48.94%**	49.15%***
T-SA-2	50.96%***	50.95%***	50.93%***	51.03%***	49.08%***	48.99%**	48.97%**	49.00%**
T-SA-3	50.61%***	50.91%***	50.70%***	51.02%***	48.85%*	48.95%**	48.88%*	48.91%**
T-SA-4	50.78%***	51.19%***	51.16%***	51.14%***	48.95%**	49.24%***	49.11%***	49.16%***
T-SA-CV	34.69%	45.81%	36.17%	30.85%	34.88%	46.36%	38.30%	37.47%

Table G.2: Comparison of performance metrics depending on the threshold strategy (1/2)

	RECALL				F1			
model	**NAIVE**	**CV**	**OPT**	**QTL**	**NAIVE**	**CV**	**OPT**	**QTL**
CO	45.77%	34.21%	32.50%	26.09%	46.58%	32.66%	34.15%	33.86%
COP_CE^{E}	44.08%	33.71%	30.22%	25.66%	45.29%	32.92%	33.19%	33.41%
COP_CE^{IL}	44.07%	34.93%	29.96%	25.50%	45.26%	33.71%	32.63%	33.27%
COP_EN	45.37%	32.78%	31.22%	25.86%	46.35%	32.20%	33.61%	33.67%
COP_EN^{E}	44.32%	33.00%	33.41%	25.74%	45.35%	32.34%	36.03%	33.48%
COP_EN^{IL}	44.24%	32.05%	35.64%	25.46%	45.31%	31.59%	37.89%	33.16%
COP_L1^{E}	45.89%	31.21%	31.65%	27.63%	46.36%	31.05%	35.09%	35.05%
COP_L1^{IL}	44.06%	31.05%	31.10%	25.47%	45.23%	30.72%	34.58%	33.22%
COP_L2^{E}	44.66%	35.23%	31.92%	25.79%	45.83%	33.30%	34.46%	33.54%
COP_L2^{IL}	44.65%	34.03%	34.37%	25.68%	45.82%	32.51%	36.98%	33.46%
COS^{E}	45.12%	32.12%	30.47%	26.76%	45.95%	31.72%	33.84%	34.35%
COS^{IL}	43.86%	33.01%	32.08%	25.68%	45.14%	32.32%	34.40%	33.44%
IL	43.94%	30.61%	32.70%	25.25%	45.12%	30.13%	35.88%	32.99%
SA	48.91%	30.84%	32.52%	27.50%	47.76%	30.12%	35.79%	34.78%
T-MCS-10	45.11%	43.04%	40.50%	45.63%	44.82%	45.22%	42.47%	46.74%
T-MCS-20	50.36%*	43.74%	40.15%	41.54%	48.55%	44.69%	43.08%	44.36%
T-MCS-50	43.68%	35.90%	35.15%	29.70%	44.99%	33.78%	38.21%	36.52%
T-SA-1	49.64%	37.34%	39.03%	31.34%	48.17%	35.36%	40.82%	37.74%
T-SA-2	44.81%	37.38%	38.35%	33.79%	45.81%	36.77%	39.14%	39.54%
T-SA-3	52.23%***	36.34%	39.60%	32.05%	49.52%*	36.56%	41.23%	38.24%
T-SA-4	45.82%	34.69%	36.18%	30.85%	46.36%	34.88%	38.31%	37.47%
T-SA-CV	34.69%	45.81%	36.17%	30.85%	34.88%	46.36%	38.30%	37.47%

Table G.3: Comparison of performance metrics depending on the threshold strategy (2/2)

PortfolioTech	Combination	Threshold	SR	APY	Vol.	DD	VaR	CVaR	WR	Av. Ga.	Av. Lo.
FILTERING	T-SA-1	$OPTI_{PREC}$	1.11	19.29	16.26	-29.30	-3.19	-5.02	59.59	1.70	-1.67
FILTERING	COP_EN_E	$OPTI_{PREC}$	1.10	19.07	16.24	-29.68	-3.17	-4.98	58.96	1.70	-1.64
FILTERING	IL	$OPTI_{PREC}$	1.08	18.53	16.10	-30.19	-3.16	-4.81	57.36	1.74	-1.58
FILTERING	SA	$OPTI_{PREC}$	1.08	18.34	15.86	-29.64	-3.07	-4.71	57.14	1.71	-1.54
FILTERING	COP_L1_IL	$OPTI_{PREC}$	1.00	17.25	16.34	-31.17	-3.42	-4.96	57.68	1.72	-1.64
FILTERING	IL	OPTI	0.97	17.08	16.81	-35.37	-3.28	-5.30	58.21	1.73	-1.68
FILTERING	T-SA-CV	$OPTI_{PREC}$	0.97	16.80	16.47	-38.90	-3.36	-5.18	58.53	1.68	-1.65
FILTERING	COP_EN_IL	$OPTI_{PREC}$	1.00	16.78	15.84	-30.32	-3.24	-4.87	57.78	1.66	-1.60
FILTERING	T-SA-3	$OPTI_{PREC}$	0.99	16.75	16.00	-38.91	-3.18	-5.02	58.21	1.65	-1.61
FILTERING	SA	OPTI	0.95	16.73	16.79	-34.69	-3.35	-5.29	57.68	1.74	-1.66
FILTERING	COS_IL	$OPTI_{PREC}$	0.89	15.97	17.16	-32.60	-3.44	-5.45	56.61	1.79	-1.71
EW	COP_EN_E	OPTI	0.73	15.75	21.93	-54.81	-4.50	-7.21	59.49	2.14	-2.34
MV	T-SA-1	$OPTI_{PREC}$	0.95	15.74	15.59	-36.19	-3.09	-5.06	59.17	1.59	-1.62
FILTERING	T-SA-CV	OPTI	0.91	15.63	16.38	-38.90	-3.24	-5.13	58.85	1.65	-1.68
FILTERING	COP_L1_E	$OPTI_{PREC}$	0.91	15.62	16.36	-29.80	-3.44	-5.10	56.93	1.72	-1.65

Table G.4: Top 15 Portfolio by APY ranking

	SR	APY	Vol.	DD	VaR	CVaR	WR	Av. Ga.	Av. Lo.
NASDAQ-100	0.75	14.95	19.70	-51.42	-4.66	-6.79	61.73	1.90	-2.27
COS_IL	0.58	12.27	22.55	-53.23	-4.80	-7.61	56.82	2.19	-2.39
COP_EN_E	0.57	12.01	22.58	-53.37	-4.89	-7.67	57.68	2.17	-2.45
COP_EN_IL	0.55	11.48	22.58	-53.37	-4.94	-7.73	57.46	2.17	-2.45
COS_E	0.54	11.28	22.61	-53.23	-5.16	-7.67	56.93	2.19	-2.42
COP_CE_IL	0.53	10.78	22.56	-54.69	-5.05	-7.61	56.72	2.17	-2.44
T-MCS-20	0.52	9.98	20.50	-58.98	-4.38	-6.92	59.70	1.89	-2.25
T-MCS-50	0.52	9.91	20.47	-57.62	-4.38	-6.87	59.06	1.91	-2.22
COP_CE_E	0.51	10.53	22.62	-58.10	-4.84	-7.60	56.82	2.17	-2.46
T-SA-4	0.51	10.18	21.60	-60.44	-4.58	-7.36	59.06	2.02	-2.39
COP_EN	0.51	10.41	23.00	-57.27	-4.76	-7.82	58.00	2.15	-2.50
CO	0.50	10.40	23.05	-59.51	-4.72	-7.85	58.21	2.15	-2.50
T-SA-1	0.49	9.76	22.08	-58.99	-4.74	-7.46	56.50	2.14	-2.42
COP_L1_E	0.47	9.36	22.66	-53.37	-4.93	-7.83	57.04	2.13	-2.46
COP_L1_IL	0.47	9.37	22.71	-53.92	-4.93	-7.72	57.04	2.14	-2.50
T-SA-2	0.46	9.07	22.11	-56.98	-4.73	-7.50	57.25	2.11	-2.43
IL	0.46	9.16	22.56	-53.23	-4.89	-7.68	55.76	2.19	-2.48
SA	0.44	8.81	22.98	-53.23	-4.89	-7.78	55.54	2.20	-2.49
T-SA-CV	0.43	8.47	22.26	-57.02	-4.62	-7.73	57.25	2.09	-2.43
T-SA-3	0.43	8.32	21.63	-57.02	-4.69	-7.48	59.06	1.99	-2.44
COP_L2_E	0.40	7.74	22.64	-54.13	-4.97	-7.87	56.61	2.11	-2.48
COP_L2_IL	0.40	7.68	22.66	-54.13	-4.96	-7.82	56.72	2.11	-2.49

Table G.5: Backtest Metrics for EW with thresholds selection CV. All data are expressed in % except the Sharpe Ratio.

	SR	APY	Vol.	DD	VaR	CVaR	WR	Av. Ga.	Av. Lo.
NASDAQ-100	0.75	14.95	19.70	-51.42	-4.66	-6.79	61.73	1.90	-2.27
T-MCS-20	0.62	11.56	19.04	-56.13	-4.12	-6.49	56.93	1.84	-2.14
T-SA-4	0.61	11.88	20.41	-56.68	-4.48	-6.97	60.34	1.92	-2.27
T-SA-CV	0.60	11.87	20.44	-57.73	-4.39	-7.01	60.02	1.93	-2.26
T-SA-3	0.60	11.79	20.54	-57.73	-4.50	-6.99	60.02	1.94	-2.29
CO	0.58	11.36	20.39	-56.52	-4.34	-6.99	60.87	1.89	-2.30
T-SA-2	0.58	11.38	20.45	-57.04	-4.57	-7.05	60.02	1.92	-2.27
COP_EN	0.58	11.21	20.41	-56.79	-4.42	-7.00	60.66	1.90	-2.30
SA	0.57	11.04	20.45	-56.40	-4.49	-7.01	59.91	1.92	-2.27
COP_L2_E	0.56	10.96	20.44	-57.03	-4.34	-7.01	59.91	1.93	-2.28
COP_L2_IL	0.56	10.97	20.47	-56.68	-4.34	-7.02	59.81	1.93	-2.27
COS_IL	0.56	10.88	20.59	-58.00	-4.58	-7.12	59.70	1.94	-2.28
T-SA-1	0.56	10.85	20.67	-57.10	-4.61	-7.15	59.70	1.94	-2.30
COP_CE_E	0.56	10.81	20.55	-57.67	-4.45	-7.08	59.38	1.95	-2.26
COP_CE_IL	0.55	10.66	20.57	-57.47	-4.52	-7.09	59.49	1.94	-2.27
T-MCS-50	0.55	10.53	20.24	-56.13	-4.43	-6.92	60.45	1.87	-2.27
COP_EN_E	0.55	10.59	20.53	-57.28	-4.54	-7.05	59.70	1.93	-2.28
COP_L1_E	0.55	10.57	20.53	-57.36	-4.54	-7.11	60.13	1.91	-2.29
COS_E	0.54	10.49	20.49	-57.42	-4.42	-7.07	59.91	1.91	-2.28
IL	0.54	10.52	20.64	-58.20	-4.59	-7.12	59.17	1.95	-2.27
COP_L1_IL	0.54	10.44	20.55	-57.83	-4.46	-7.10	59.59	1.93	-2.27
COP_EN_IL	0.53	10.28	20.53	-57.99	-4.43	-7.07	59.59	1.93	-2.28

Table G.6: Backtest Metrics for EW with thresholds selection NAIVE. All data are expressed in % except the Sharpe Ratio.

	SR	APY	Vol.	DD	VaR	CVaR	WR	Av. Ga.	Av. Lo.
NASDAQ-100	0.75	14.95	19.70	-51.42	-4.66	-6.79	61.73	1.90	-2.27
COP_EN_E	0.73	15.75	21.93	-54.81	-4.50	-7.21	59.49	2.14	-2.34
COP_EN_IL	0.72	15.31	21.66	-54.68	-4.44	-7.16	60.23	2.08	-2.35
SA	0.63	13.00	21.64	-54.68	-4.37	-7.22	57.68	2.12	-2.31
COS_E	0.63	13.15	22.04	-54.68	-4.68	-7.31	58.21	2.15	-2.40
T-MCS-20	0.63	11.82	19.32	-58.98	-4.04	-6.47	56.82	1.86	-2.14
IL	0.62	12.89	21.65	-55.11	-4.47	-7.22	57.68	2.12	-2.32
COS_IL	0.62	13.15	22.29	-54.68	-4.67	-7.51	57.25	2.17	-2.36
COP_CE_E	0.62	12.88	21.98	-58.17	-4.55	-7.24	57.57	2.15	-2.32
T-SA-4	0.62	12.31	20.76	-58.66	-4.57	-6.99	60.02	1.96	-2.29
COP_EN	0.62	13.20	22.86	-57.32	-4.69	-7.68	58.64	2.16	-2.41
COP_L2_IL	0.62	12.60	21.51	-54.13	-4.51	-7.19	58.64	2.07	-2.32
COP_L1_IL	0.61	12.76	22.11	-54.68	-4.42	-7.48	58.32	2.11	-2.36
COP_L1_E	0.61	12.71	22.09	-54.81	-4.43	-7.49	58.64	2.10	-2.38
CO	0.60	12.63	22.22	-55.80	-4.84	-7.46	58.74	2.10	-2.39
T-SA-1	0.60	12.04	21.26	-56.63	-4.38	-7.11	59.06	2.04	-2.32
T-SA-3	0.58	11.51	21.06	-55.56	-4.63	-7.17	59.70	1.98	-2.34
T-SA-CV	0.57	11.52	21.44	-55.56	-4.45	-7.29	58.53	2.05	-2.36
COP_CE_IL	0.57	11.60	21.98	-55.09	-4.56	-7.23	56.40	2.18	-2.34
COP_L2_E	0.55	11.19	21.92	-57.32	-4.66	-7.30	57.36	2.11	-2.35
T-MCS-50	0.55	10.63	20.52	-57.62	-4.37	-6.87	59.38	1.92	-2.23
T-SA-2	0.54	10.63	20.99	-56.22	-4.70	-7.16	59.28	1.98	-2.31

Table G.7: Backtest Metrics for EW with thresholds selection OPTI. All data are expressed in % except the Sharpe Ratio.

	SR	APY	Vol.	DD	VaR	CVaR	WR	Av. Ga.	Av. Lo.
NASDAQ-100	0.75	14.95	19.70	-51.42	-4.66	-6.79	61.73	1.90	-2.27
T-SA-1	0.71	14.64	21.16	-54.63	-4.36	-7.11	58.64	2.08	-2.30
COP_EN_E	0.68	14.06	21.23	-54.88	-4.30	-7.16	57.68	2.10	-2.27
T-SA-3	0.67	13.61	20.91	-55.56	-4.35	-7.09	59.59	1.99	-2.35
IL	0.67	13.76	21.27	-55.98	-4.35	-7.13	57.89	2.11	-2.32
COP_L1_IL	0.66	13.58	21.07	-55.47	-4.38	-7.09	57.57	2.09	-2.26
COP_L1_E	0.66	13.47	21.15	-55.35	-4.31	-7.13	57.57	2.09	-2.28
COP_EN_IL	0.65	13.10	20.98	-55.15	-4.34	-7.09	57.89	2.05	-2.30
SA	0.65	13.20	21.22	-55.47	-4.38	-7.10	57.89	2.09	-2.32
T-SA-4	0.63	12.67	20.80	-58.66	-4.46	-7.01	58.85	2.00	-2.26
COP_CE_IL	0.62	12.86	22.06	-55.59	-4.93	-7.30	55.97	2.22	-2.36
COS_IL	0.61	12.52	21.63	-55.35	-4.49	-7.30	56.50	2.15	-2.30
T-SA-CV	0.61	12.36	21.31	-55.56	-4.39	-7.23	58.74	2.04	-2.34
COS_E	0.60	12.29	21.43	-55.35	-4.39	-7.25	56.61	2.13	-2.28
T-SA-2	0.60	11.90	20.90	-57.41	-4.45	-7.09	58.64	2.01	-2.30
COP_CE_E	0.59	12.22	21.81	-55.91	-4.56	-7.30	55.86	2.17	-2.34
CO	0.59	11.74	20.78	-56.87	-4.56	-7.19	59.38	1.99	-2.35
T-MCS-50	0.57	11.15	20.72	-57.62	-4.38	-6.89	58.96	1.96	-2.23
COP_L2_E	0.54	10.80	21.21	-55.79	-4.37	-7.32	56.61	2.09	-2.32
COP_L2_IL	0.54	10.78	21.21	-54.90	-4.38	-7.31	56.50	2.10	-2.30
COP_EN	0.52	10.24	21.12	-57.77	-4.44	-7.42	57.14	2.03	-2.36
T-MCS-20	0.52	9.98	20.50	-58.98	-4.38	-6.92	59.70	1.89	-2.25

Table G.8: Backtest Metrics for EW with thresholds selection $OPTI_{PREC}$. All data are expressed in % except the Sharpe Ratio.

	SR	APY	Vol.	DD	VaR	CVaR	WR	Av. Ga.	Av. Lo.
NASDAQ-100	0.75	14.95	19.70	-51.42	-4.66	-6.79	61.73	1.90	-2.27
CO	0.61	11.78	19.88	-53.31	-4.45	-6.86	60.98	1.86	-2.25
COP_EN	0.61	11.77	19.88	-53.36	-4.44	-6.85	61.09	1.85	-2.26
COP_L2_E	0.61	11.64	19.90	-53.35	-4.45	-6.87	61.30	1.84	-2.27
COP_CE_IL	0.60	11.60	19.90	-53.48	-4.45	-6.85	61.30	1.84	-2.27
COP_L2_IL	0.60	11.54	19.90	-53.59	-4.44	-6.86	61.30	1.84	-2.27
COS_IL	0.60	11.55	19.92	-54.12	-4.44	-6.87	61.19	1.85	-2.27
COP_CE_E	0.60	11.52	19.90	-53.30	-4.48	-6.86	60.98	1.85	-2.26
COP_EN_IL	0.60	11.43	19.89	-53.56	-4.44	-6.85	61.41	1.84	-2.28
COP_L1_IL	0.60	11.42	19.89	-53.75	-4.44	-6.86	61.94	1.82	-2.32
COP_EN_E	0.59	11.42	19.89	-53.40	-4.44	-6.84	61.19	1.84	-2.27
COP_L1_E	0.59	11.39	19.89	-53.89	-4.44	-6.87	61.73	1.83	-2.30
COS_E	0.59	11.38	19.90	-54.12	-4.44	-6.87	61.41	1.83	-2.28
T-SA-1	0.59	11.26	19.94	-54.12	-4.48	-6.87	61.30	1.84	-2.28
IL	0.58	11.19	19.89	-53.75	-4.44	-6.85	61.41	1.83	-2.28
SA	0.58	11.16	19.90	-54.12	-4.44	-6.86	61.41	1.83	-2.29
T-SA-4	0.58	11.20	20.03	-54.50	-4.54	-6.90	60.45	1.88	-2.25
T-SA-2	0.57	10.88	19.98	-54.75	-4.43	-6.89	60.87	1.85	-2.27
T-SA-3	0.56	10.78	19.99	-54.83	-4.47	-6.90	60.66	1.85	-2.26
T-SA-CV	0.56	10.76	19.95	-54.83	-4.47	-6.88	60.98	1.84	-2.27
T-MCS-50	0.55	10.53	20.24	-56.13	-4.43	-6.92	60.45	1.87	-2.27
T-MCS-20	0.51	9.72	20.23	-56.13	-4.43	-6.94	59.81	1.88	-2.25

Table G.9: Backtest Metrics for EW with thresholds selection OPTI_RECALL. All data are expressed in % except the Sharpe Ratio.

	SR	APY	Vol.	DD	VaR	CVaR	WR	Av. Ga.	Av. Lo.
NASDAQ-100	0.75	14.95	19.70	-51.42	-4.66	-6.79	61.73	1.90	-2.27
COP_L2_IL	0.63	12.80	21.24	-56.70	-4.57	-7.26	60.77	1.97	-2.36
COP_L2_E	0.62	12.56	21.23	-58.90	-4.57	-7.25	60.45	1.98	-2.34
COP_EN	0.57	11.47	21.14	-58.13	-4.47	-7.22	60.77	1.94	-2.37
COS_E	0.57	11.46	21.24	-56.87	-4.53	-7.19	59.28	2.01	-2.31
CO	0.57	11.38	21.07	-59.21	-4.27	-7.23	60.87	1.94	-2.38
COP_EN_IL	0.56	11.45	21.68	-58.17	-4.50	-7.24	59.06	2.03	-2.32
COP_L1_IL	0.56	11.37	21.61	-56.17	-4.72	-7.28	58.74	2.04	-2.29
COP_EN_E	0.56	11.17	21.39	-56.87	-4.70	-7.29	59.17	2.02	-2.32
COP_CE_E	0.56	11.18	21.41	-58.78	-4.72	-7.31	59.06	2.02	-2.31
COP_L1_E	0.55	10.98	21.13	-55.74	-4.73	-7.21	59.70	1.98	-2.33
COP_CE_IL	0.55	11.06	21.35	-57.21	-4.73	-7.29	59.06	2.01	-2.31
T-SA-4	0.55	10.83	20.86	-60.44	-4.54	-7.08	58.64	1.98	-2.24
SA	0.55	10.88	21.23	-58.03	-4.76	-7.28	59.17	2.00	-2.31
COS_IL	0.54	10.77	21.60	-57.18	-4.97	-7.32	59.28	2.03	-2.35
T-SA-2	0.54	10.46	20.81	-57.16	-4.49	-7.15	59.59	1.95	-2.31
T-SA-3	0.54	10.42	20.69	-57.26	-4.62	-7.08	59.28	1.95	-2.28
IL	0.53	10.57	21.38	-56.97	-4.82	-7.27	58.74	2.02	-2.31
T-MCS-50	0.53	10.11	20.43	-57.62	-4.43	-6.90	59.91	1.89	-2.25
T-SA-CV	0.52	10.09	20.68	-57.26	-4.66	-7.10	58.64	1.96	-2.25
T-MCS-20	0.50	9.52	20.44	-58.98	-4.38	-6.93	59.81	1.88	-2.26
T-SA-1	0.50	9.64	21.00	-56.40	-4.55	-7.28	59.38	1.96	-2.33

Table G.10: Backtest Metrics for EW with thresholds selection QUANTILE. All data are expressed in % except the Sharpe Ratio.

	SR	APY	Vol.	DD	VaR	CVaR	WR	Av. Ga.	Av. Lo.
NASDAQ-100	0.75	14.95	19.70	-51.42	-4.66	-6.79	61.73	1.90	-2.27
COP_EN_E	0.61	10.91	17.92	-36.76	-3.51	-6.09	58.00	1.69	-1.87
COP_CE_E	0.61	10.49	17.12	-43.49	-3.55	-5.85	57.04	1.67	-1.83
COP_CE_IL	0.59	10.31	17.44	-37.21	-3.64	-5.94	56.93	1.68	-1.84
T-SA-4	0.59	9.82	16.64	-42.94	-3.39	-5.45	57.04	1.62	-1.70
SA	0.59	10.41	18.02	-36.76	-3.67	-6.04	56.50	1.73	-1.92
IL	0.58	10.29	17.89	-36.76	-3.67	-6.00	56.61	1.72	-1.91
T-SA-1	0.57	9.56	16.73	-48.96	-3.65	-5.52	56.72	1.65	-1.82
COP_L1_IL	0.56	9.89	17.86	-37.15	-3.68	-6.04	56.93	1.73	-1.93
COP_L2_IL	0.56	9.68	17.41	-38.07	-3.55	-5.87	57.89	1.63	-1.89
COP_EN_IL	0.56	9.77	17.87	-36.76	-3.52	-6.18	57.68	1.68	-1.89
COS_IL	0.56	9.77	17.89	-36.76	-3.59	-6.19	57.78	1.67	-1.89
COP_L2_E	0.56	9.58	17.47	-38.07	-3.61	-5.92	58.21	1.63	-1.92
T-SA-CV	0.55	9.23	17.00	-41.04	-3.37	-5.64	57.04	1.61	-1.77
T-MCS-20	0.54	8.05	14.37	-44.20	-2.96	-4.79	58.21	1.40	-1.55
T-SA-2	0.53	8.96	16.91	-42.28	-3.63	-5.61	57.78	1.63	-1.85
COS_E	0.52	9.11	17.95	-36.76	-3.90	-6.29	57.46	1.68	-1.90
CO	0.52	9.21	18.41	-46.09	-3.93	-6.28	57.89	1.69	-1.91
COP_EN	0.51	9.00	18.29	-44.92	-3.97	-6.20	57.89	1.67	-1.91
T-MCS-50	0.51	7.61	14.59	-42.74	-3.22	-4.82	58.42	1.41	-1.59
COP_L1_E	0.51	8.79	17.98	-36.76	-3.61	-6.21	57.14	1.69	-1.92
T-SA-3	0.49	7.96	16.21	-41.04	-3.42	-5.38	57.89	1.55	-1.75

Table G.11: Backtest Metrics for MV with thresholds selection CV. All data are expressed in % except the Sharpe Ratio.

	SR	APY	Vol.	DD	VaR	CVaR	WR	Av. Ga.	Av. Lo.
T-SA-CV	0.83	12.72	14.42	-43.18	-3.20	-4.70	59.28	1.46	-1.51
T-SA-2	0.83	12.66	14.37	-40.21	-3.15	-4.72	59.70	1.44	-1.51
T-SA-3	0.83	12.74	14.65	-43.18	-3.21	-4.83	59.38	1.47	-1.54
T-SA-1	0.80	12.40	14.74	-42.12	-3.15	-4.92	60.13	1.45	-1.58
COS_E	0.80	12.09	14.35	-41.85	-3.15	-4.74	60.13	1.43	-1.56
COP_L1_IL	0.80	12.10	14.42	-42.54	-3.17	-4.73	60.23	1.44	-1.57
COP_L1_E	0.80	12.06	14.40	-41.85	-3.13	-4.79	60.13	1.43	-1.56
SA	0.79	11.89	14.27	-40.58	-3.13	-4.76	59.28	1.43	-1.51
COP_EN_IL	0.79	11.93	14.42	-42.94	-3.17	-4.73	60.23	1.44	-1.58
COP_L2_E	0.78	11.83	14.44	-43.15	-3.15	-4.72	59.91	1.44	-1.56
COP_CE_E	0.78	11.76	14.37	-42.09	-3.14	-4.77	60.23	1.43	-1.57
COP_CE_IL	0.78	11.78	14.43	-42.84	-3.13	-4.78	60.23	1.43	-1.58
COP_L2_IL	0.77	11.74	14.48	-43.21	-3.15	-4.74	60.02	1.44	-1.57
CO	0.77	11.64	14.37	-44.22	-3.15	-4.74	59.06	1.45	-1.52
COP_EN_E	0.77	11.64	14.39	-42.82	-3.16	-4.75	60.13	1.43	-1.57
T-SA-4	0.76	11.47	14.35	-41.49	-3.05	-4.69	58.42	1.45	-1.49
IL	0.76	11.70	14.74	-43.92	-3.16	-4.95	59.81	1.45	-1.58
NASDAQ-100	0.75	14.95	19.70	-51.42	-4.66	-6.79	61.73	1.90	-2.27
COP_EN	0.74	11.18	14.36	-42.79	-3.14	-4.73	58.74	1.45	-1.51
COS_IL	0.73	11.03	14.51	-44.40	-3.15	-4.85	60.34	1.43	-1.61
T-MCS-50	0.56	8.06	13.84	-42.84	-2.97	-4.64	58.42	1.37	-1.52
T-MCS-20	0.54	7.58	13.38	-43.42	-2.86	-4.51	54.69	1.37	-1.48

Table G.12: Backtest Metrics for MV with thresholds selection NAIVE. All data are expressed in % except the Sharpe Ratio.

	SR	APY	Vol.	DD	VaR	CVaR	WR	Av. Ga.	Av. Lo.
IL	0.87	14.88	16.42	-35.36	-3.44	-5.27	58.96	1.63	-1.69
SA	0.86	14.69	16.41	-34.62	-3.44	-5.27	58.64	1.63	-1.68
T-SA-CV	0.82	13.54	15.77	-40.47	-3.27	-5.10	59.28	1.54	-1.65
COP_EN_E	0.79	13.54	16.60	-34.89	-3.45	-5.39	59.38	1.62	-1.71
T-SA-4	0.79	12.43	15.13	-42.48	-3.11	-4.91	57.78	1.52	-1.51
NASDAQ-100	0.75	14.95	19.70	-51.42	-4.66	-6.79	61.73	1.90	-2.27
COS_E	0.75	12.81	16.70	-34.62	-3.59	-5.58	58.64	1.64	-1.77
COP_EN_IL	0.75	12.50	16.24	-34.62	-3.47	-5.38	60.02	1.57	-1.73
COS_IL	0.74	12.85	16.97	-34.62	-3.51	-5.72	58.74	1.63	-1.78
T-SA-1	0.73	11.90	15.82	-41.49	-3.14	-5.17	58.64	1.55	-1.64
COP_EN	0.73	12.80	17.28	-44.92	-3.52	-5.78	59.59	1.63	-1.80
T-SA-2	0.71	11.29	15.31	-40.33	-3.11	-5.10	59.17	1.49	-1.61
T-SA-3	0.70	11.22	15.39	-40.47	-3.27	-5.11	58.74	1.52	-1.63
COP_L2_IL	0.68	10.94	15.61	-38.48	-3.47	-5.19	58.42	1.53	-1.65
COP_CE_IL	0.65	11.08	16.96	-36.13	-3.71	-5.59	57.04	1.68	-1.79
COP_L1_IL	0.63	10.91	17.13	-34.62	-3.75	-5.80	58.10	1.64	-1.79
COP_L1_E	0.63	10.85	17.05	-34.62	-3.86	-5.84	58.96	1.61	-1.82
COP_L2_E	0.63	10.48	16.30	-45.25	-3.51	-5.43	58.21	1.57	-1.74
COP_CE_E	0.62	10.54	16.74	-43.19	-3.62	-5.58	57.14	1.65	-1.73
CO	0.61	10.39	17.07	-44.49	-3.63	-5.83	58.64	1.60	-1.79
T-MCS-20	0.58	8.34	13.85	-44.20	-2.87	-4.62	55.01	1.39	-1.51
T-MCS-50	0.55	8.29	14.65	-42.74	-3.22	-4.83	58.42	1.43	-1.59

Table G.13: Backtest Metrics for MV with thresholds selection OPTI. All data are expressed in % except the Sharpe Ratio.

	SR	APY	Vol.	DD	VaR	CVaR	WR	Av. Ga.	Av. Lo.
T-SA-1	0.95	15.74	15.59	-36.19	-3.09	-5.06	59.17	1.59	-1.62
COP_EN_E	0.92	15.25	15.68	-37.43	-3.32	-5.08	59.17	1.58	-1.64
T-SA-3	0.91	14.68	15.20	-40.47	-3.11	-4.95	58.53	1.54	-1.57
IL	0.90	14.82	15.72	-38.33	-3.21	-5.03	57.46	1.63	-1.59
SA	0.87	14.22	15.52	-37.77	-3.15	-4.97	57.57	1.59	-1.57
T-SA-CV	0.85	14.14	15.82	-40.47	-3.31	-5.14	58.53	1.58	-1.61
COP_EN_IL	0.84	13.47	15.33	-37.48	-3.32	-4.99	58.21	1.55	-1.62
COP_L1_IL	0.79	13.00	15.77	-37.77	-3.43	-5.13	57.36	1.61	-1.63
T-SA-2	0.79	12.35	15.00	-39.22	-3.31	-5.01	58.32	1.52	-1.58
T-SA-4	0.77	12.16	15.06	-42.48	-3.15	-4.88	57.46	1.52	-1.52
COS_E	0.77	12.95	16.20	-37.75	-3.32	-5.39	57.46	1.63	-1.69
COS_IL	0.77	13.08	16.46	-38.13	-3.32	-5.47	57.89	1.63	-1.72
COP_L1_E	0.77	12.58	15.79	-37.75	-3.32	-5.21	57.57	1.59	-1.66
NASDAQ-100	0.75	14.95	19.70	-51.42	-4.66	-6.79	61.73	1.90	-2.27
COP_L2_IL	0.64	10.18	15.61	-41.56	-3.42	-5.23	56.93	1.57	-1.68
COP_CE_IL	0.62	10.73	17.13	-38.93	-3.67	-5.66	55.86	1.72	-1.80
CO	0.61	9.63	15.49	-43.14	-3.64	-5.27	57.25	1.55	-1.66
T-MCS-50	0.60	9.28	14.92	-42.74	-3.22	-4.82	58.10	1.47	-1.58
COP_L2_E	0.56	8.91	15.71	-45.28	-3.46	-5.30	56.18	1.57	-1.70
COP_CE_E	0.55	9.19	16.88	-42.18	-3.56	-5.60	54.90	1.70	-1.78
T-MCS-20	0.54	8.05	14.37	-44.20	-2.96	-4.79	58.21	1.40	-1.55
COP_EN	0.46	7.27	15.97	-47.61	-3.61	-5.52	55.65	1.56	-1.74

Table G.14: Backtest Metrics for MV with thresholds selection $OPTI_{PREC}$. All data are expressed in % except the Sharpe Ratio.

	SR	APY	Vol.	DD	VaR	CVaR	WR	Av. Ga.	Av. Lo.
COS_E	0.76	10.54	13.00	-34.98	-2.94	-4.22	58.64	1.34	-1.39
NASDAQ-100	0.75	14.95	19.70	-51.42	-4.66	-6.79	61.73	1.90	-2.27
COS_IL	0.75	10.34	12.88	-34.22	-2.94	-4.16	58.74	1.33	-1.39
COP_L1_E	0.74	10.30	13.05	-33.54	-2.94	-4.22	58.85	1.33	-1.40
COP_L1_IL	0.74	10.20	13.03	-35.60	-2.98	-4.21	58.96	1.32	-1.41
SA	0.74	10.25	13.11	-34.98	-2.97	-4.22	58.42	1.35	-1.40
COP_L2_E	0.73	9.99	12.81	-33.09	-2.91	-4.13	58.42	1.32	-1.37
COP_CE_E	0.72	9.99	12.96	-33.37	-2.98	-4.18	58.00	1.34	-1.38
IL	0.72	10.04	13.12	-35.60	-2.98	-4.22	58.32	1.35	-1.41
COP_L2_IL	0.72	9.79	12.82	-34.54	-2.91	-4.15	58.21	1.32	-1.37
CO	0.71	9.72	12.74	-33.36	-2.89	-4.10	58.64	1.30	-1.38
T-SA-1	0.71	9.97	13.15	-36.79	-2.98	-4.29	58.42	1.34	-1.41
COP_EN	0.71	9.71	12.82	-34.89	-2.92	-4.13	58.64	1.31	-1.39
COP_CE_IL	0.71	9.79	12.99	-34.54	-2.94	-4.20	57.78	1.35	-1.38
COP_EN_IL	0.71	9.80	13.02	-35.56	-2.93	-4.19	58.53	1.33	-1.40
T-SA-4	0.69	9.90	13.54	-37.63	-3.06	-4.41	57.68	1.38	-1.41
T-SA-2	0.69	9.65	13.30	-37.77	-2.98	-4.35	58.64	1.34	-1.43
COP_EN_E	0.68	9.30	12.95	-34.45	-2.94	-4.20	58.42	1.32	-1.40
T-SA-CV	0.67	9.48	13.31	-37.47	-3.02	-4.35	58.32	1.34	-1.42
T-SA-3	0.67	9.40	13.36	-37.47	-3.02	-4.36	58.00	1.35	-1.41
T-MCS-50	0.56	8.06	13.84	-42.84	-2.97	-4.64	58.42	1.37	-1.52
T-MCS-20	0.50	7.29	13.92	-43.42	-2.96	-4.69	57.89	1.37	-1.52

Table G.15: Backtest Metrics for MV with thresholds selection OPTI_RECALL. All data are expressed in % except the Sharpe Ratio.

	SR	APY	Vol.	DD	VaR	CVaR	WR	Av. Ga.	Av. Lo.
COP_L2_IL	0.82	13.70	16.01	-43.00	-3.26	-5.22	59.38	1.59	-1.65
COP_L2_E	0.82	13.55	15.93	-43.75	-3.16	-5.17	59.59	1.57	-1.65
COP_CE_IL	0.79	12.91	15.69	-42.33	-3.13	-5.06	58.74	1.57	-1.62
COP_CE_E	0.77	12.50	15.60	-43.48	-3.12	-5.01	58.53	1.57	-1.61
NASDAQ-100	0.75	14.95	19.70	-51.42	-4.66	-6.79	61.73	1.90	-2.27
COP_EN	0.75	12.23	15.79	-43.72	-3.31	-5.16	59.38	1.55	-1.65
COP_EN_E	0.74	12.11	15.82	-43.68	-3.33	-5.18	58.10	1.58	-1.62
SA	0.74	11.94	15.58	-46.76	-3.17	-5.08	58.10	1.55	-1.58
COP_L1_IL	0.73	11.91	15.79	-37.97	-3.42	-5.11	57.78	1.59	-1.61
CO	0.72	11.64	15.61	-46.61	-3.37	-5.19	59.06	1.54	-1.65
T-SA-4	0.72	11.53	15.47	-46.27	-3.24	-5.10	59.17	1.51	-1.62
COP_EN_IL	0.71	11.55	15.66	-39.82	-3.38	-5.09	57.78	1.58	-1.62
COP_L1_E	0.71	11.29	15.45	-41.22	-3.12	-5.05	57.04	1.57	-1.56
IL	0.70	11.59	16.14	-43.54	-3.30	-5.30	57.68	1.60	-1.63
COS_IL	0.69	11.18	15.74	-44.00	-3.28	-5.16	58.32	1.55	-1.63
COS_E	0.67	10.61	15.36	-42.69	-3.12	-4.99	57.57	1.54	-1.59
T-SA-1	0.65	10.34	15.50	-47.10	-3.12	-5.15	59.38	1.49	-1.65
T-SA-2	0.64	10.11	15.42	-47.23	-3.21	-5.16	58.64	1.49	-1.62
T-SA-CV	0.63	9.85	15.26	-45.35	-3.07	-5.01	57.68	1.50	-1.57
T-SA-3	0.63	9.82	15.24	-45.35	-3.16	-5.05	57.78	1.50	-1.58
T-MCS-50	0.55	8.20	14.58	-45.62	-3.18	-4.87	59.06	1.40	-1.59
T-MCS-20	0.49	7.21	14.38	-44.20	-3.22	-4.82	58.00	1.39	-1.56

Table G.16: Backtest Metrics for MV with thresholds selection QUANTILE. All data are expressed in % except the Sharpe Ratio.

	SR	APY	Vol.	DD	VaR	CVaR	WR	Av. Ga.	Av. Lo.
NASDAQ-100	0.75	14.95	19.70	-51.42	-4.66	-6.79	61.73	1.90	-2.27
COP_EN_E	0.63	11.18	17.90	-37.19	-3.59	-6.15	59.28	1.66	-1.94
COP_CE_E	0.62	10.75	17.33	-44.03	-3.60	-5.97	58.10	1.66	-1.90
COP_CE_IL	0.60	10.55	17.61	-38.29	-3.67	-6.05	58.21	1.66	-1.92
T-SA-4	0.59	10.01	16.80	-43.29	-3.47	-5.60	57.89	1.61	-1.75
IL	0.59	10.50	17.98	-37.18	-3.75	-6.11	57.36	1.71	-1.96
COS_IL	0.58	10.30	17.93	-37.18	-3.68	-6.25	58.21	1.67	-1.91
SA	0.58	10.48	18.32	-37.18	-3.75	-6.20	57.36	1.73	-1.98
T-SA-1	0.58	9.77	16.88	-49.55	-3.70	-5.64	56.61	1.67	-1.82
T-MCS-20	0.58	8.62	14.42	-43.64	-3.14	-4.83	58.10	1.41	-1.53
COP_EN_IL	0.58	10.14	17.90	-37.19	-3.60	-6.25	59.06	1.66	-1.96
COP_L2_E	0.56	9.69	17.73	-38.79	-3.65	-6.05	58.53	1.63	-1.95
COP_L2_IL	0.56	9.66	17.70	-38.79	-3.60	-6.00	58.21	1.64	-1.93
COP_L1_IL	0.55	9.85	18.19	-37.94	-3.75	-6.21	57.89	1.71	-1.99
COS_E	0.55	9.65	18.00	-37.18	-3.95	-6.35	58.32	1.67	-1.94
T-SA-CV	0.54	9.21	17.11	-41.07	-3.41	-5.78	58.10	1.59	-1.83
T-SA-2	0.54	9.22	17.15	-42.82	-3.70	-5.78	57.89	1.64	-1.87
T-MCS-50	0.54	8.16	14.63	-42.43	-3.20	-4.87	58.10	1.42	-1.57
CO	0.53	9.39	18.42	-46.14	-3.93	-6.34	58.10	1.70	-1.93
COP_L1_E	0.51	9.06	18.21	-37.19	-3.73	-6.32	58.42	1.67	-1.99
COP_EN	0.51	9.11	18.34	-45.42	-3.99	-6.29	58.10	1.68	-1.93
T-SA-3	0.51	8.26	16.44	-41.07	-3.50	-5.55	58.74	1.55	-1.81

Table G.17: Backtest Metrics for Shrinked_MV with thresholds selection CV. All data are expressed in % except the Sharpe Ratio.

	SR	APY	Vol.	DD	VaR	CVaR	WR	Av. Ga.	Av. Lo.
T-SA-CV	0.84	12.84	14.41	-42.93	-3.17	-4.77	59.70	1.45	-1.52
T-SA-3	0.83	12.77	14.65	-42.93	-3.24	-4.91	59.49	1.46	-1.54
T-SA-2	0.82	12.46	14.38	-40.14	-3.15	-4.80	60.66	1.41	-1.55
T-SA-1	0.80	12.40	14.73	-42.20	-3.03	-4.99	59.59	1.46	-1.56
COS_E	0.79	11.93	14.33	-42.27	-3.19	-4.76	60.02	1.43	-1.55
COP_L1_IL	0.79	11.98	14.41	-43.00	-3.20	-4.78	60.02	1.44	-1.56
SA	0.79	11.85	14.24	-40.86	-3.17	-4.78	59.28	1.43	-1.50
COP_L1_E	0.79	11.88	14.38	-42.27	-3.20	-4.82	59.81	1.43	-1.55
COP_EN_IL	0.78	11.78	14.40	-43.32	-3.23	-4.78	60.13	1.43	-1.58
COP_L2_E	0.78	11.74	14.39	-43.25	-3.21	-4.77	59.81	1.43	-1.55
T-SA-4	0.77	11.64	14.29	-41.54	-3.00	-4.71	59.06	1.44	-1.50
COP_CE_E	0.77	11.67	14.35	-42.49	-3.21	-4.81	60.13	1.42	-1.56
COP_CE_IL	0.77	11.67	14.41	-43.39	-3.21	-4.82	60.02	1.43	-1.57
COP_L2_IL	0.77	11.65	14.43	-43.48	-3.21	-4.78	59.81	1.43	-1.55
COP_EN_E	0.76	11.55	14.38	-43.36	-3.23	-4.79	60.02	1.43	-1.56
CO	0.76	11.52	14.33	-44.02	-3.24	-4.78	59.70	1.42	-1.54
IL	0.76	11.71	14.73	-43.83	-3.15	-5.01	59.70	1.45	-1.57
NASDAQ-100	0.75	14.95	19.70	-51.42	-4.66	-6.79	61.73	1.90	-2.27
COP_EN	0.74	11.07	14.31	-42.84	-3.25	-4.77	58.85	1.44	-1.51
COS_IL	0.73	11.07	14.47	-44.59	-3.19	-4.88	60.55	1.42	-1.61
T-MCS-50	0.59	8.51	13.87	-43.03	-3.05	-4.68	58.21	1.38	-1.50
T-MCS-20	0.57	7.98	13.40	-43.55	-2.86	-4.55	55.01	1.36	-1.48

Table G.18: Backtest Metrics for Shrinked_MV with thresholds selection NAIVE. All data are expressed in % except the Sharpe Ratio.

	SR	APY	Vol.	DD	VaR	CVaR	WR	Av. Ga.	Av. Lo.
IL	0.85	14.52	16.45	-35.93	-3.43	-5.39	59.81	1.60	-1.73
SA	0.84	14.39	16.44	-35.16	-3.43	-5.39	59.59	1.60	-1.72
T-SA-CV	0.80	13.13	15.76	-40.27	-3.24	-5.21	60.55	1.51	-1.71
COP_EN_E	0.80	13.76	16.63	-35.41	-3.46	-5.47	60.55	1.60	-1.76
T-SA-4	0.79	12.46	15.17	-42.53	-3.00	-4.99	59.17	1.49	-1.56
COP_EN_IL	0.77	12.90	16.30	-35.16	-3.49	-5.44	60.23	1.57	-1.73
NASDAQ-100	0.75	14.95	19.70	-51.42	-4.66	-6.79	61.73	1.90	-2.27
COS_E	0.74	12.74	16.74	-35.16	-3.60	-5.65	59.49	1.62	-1.81
COS_IL	0.73	12.71	17.04	-35.16	-3.52	-5.78	58.85	1.63	-1.79
T-SA-1	0.72	11.73	15.88	-41.31	-3.32	-5.28	59.06	1.54	-1.66
COP_EN	0.71	12.53	17.42	-45.43	-3.58	-5.89	60.23	1.62	-1.85
T-SA-2	0.70	11.22	15.39	-40.38	-3.07	-5.25	59.49	1.49	-1.63
T-SA-3	0.70	11.20	15.46	-40.27	-3.31	-5.23	59.49	1.50	-1.67
COP_L2_IL	0.69	11.14	15.71	-38.97	-3.47	-5.28	59.91	1.50	-1.72
COP_CE_IL	0.66	11.36	17.08	-36.72	-3.71	-5.69	58.00	1.67	-1.84
COP_L1_IL	0.65	11.20	17.21	-35.16	-3.75	-5.89	59.59	1.61	-1.86
COP_L1_E	0.64	11.05	17.09	-35.17	-3.88	-5.89	60.02	1.58	-1.86
COP_CE_E	0.64	10.91	16.89	-43.56	-3.60	-5.69	58.42	1.63	-1.79
COP_L2_E	0.63	10.45	16.42	-45.50	-3.54	-5.54	58.96	1.56	-1.79
T-MCS-20	0.61	8.88	13.89	-43.64	-2.86	-4.65	55.12	1.40	-1.50
CO	0.60	10.27	17.16	-44.25	-3.66	-5.93	58.53	1.61	-1.81
T-MCS-50	0.58	8.81	14.70	-42.43	-3.20	-4.88	58.21	1.44	-1.57

Table G.19: Backtest Metrics for Shrinked_MV with thresholds selection OPTI. All data are expressed in % except the Sharpe Ratio.

	SR	APY	Vol.	DD	VaR	CVaR	WR	Av. Ga.	Av. Lo.
T-SA-1	0.93	15.24	15.61	-36.56	-3.26	-5.16	59.81	1.57	-1.66
COP_EN_E	0.90	14.85	15.61	-37.85	-3.34	-5.16	60.02	1.55	-1.68
T-SA-3	0.89	14.40	15.27	-40.27	-3.15	-5.06	59.91	1.51	-1.64
IL	0.86	14.25	15.73	-38.86	-3.34	-5.15	58.53	1.59	-1.65
SA	0.85	13.81	15.50	-38.48	-3.27	-5.07	58.64	1.55	-1.62
T-SA-CV	0.82	13.58	15.82	-40.27	-3.35	-5.25	59.49	1.55	-1.67
COP_EN_IL	0.82	13.14	15.34	-38.17	-3.34	-5.06	58.53	1.54	-1.64
COP_L1_IL	0.77	12.66	15.81	-38.48	-3.45	-5.23	58.10	1.58	-1.67
T-SA-4	0.77	12.13	15.12	-42.53	-3.06	-4.97	58.64	1.50	-1.57
NASDAQ-100	0.75	14.95	19.70	-51.42	-4.66	-6.79	61.73	1.90	-2.27
T-SA-2	0.75	11.82	15.08	-39.64	-3.37	-5.14	58.42	1.51	-1.61
COP_L1_E	0.75	12.19	15.78	-38.55	-3.39	-5.28	57.57	1.58	-1.66
COS_E	0.74	12.32	16.20	-38.55	-3.42	-5.44	57.89	1.60	-1.71
COS_IL	0.73	12.40	16.45	-38.82	-3.42	-5.52	58.32	1.61	-1.75
T-MCS-50	0.63	9.78	14.95	-42.43	-3.20	-4.87	57.68	1.49	-1.56
COP_CE_IL	0.63	10.84	17.20	-39.29	-3.80	-5.74	57.57	1.68	-1.88
COP_L2_IL	0.61	9.73	15.61	-42.34	-3.54	-5.32	57.89	1.54	-1.73
CO	0.59	9.38	15.51	-43.35	-3.56	-5.33	58.10	1.53	-1.71
T-MCS-20	0.58	8.62	14.42	-43.64	-3.14	-4.83	58.10	1.41	-1.53
COP_CE_E	0.56	9.42	16.95	-42.46	-3.72	-5.70	56.08	1.67	-1.84
COP_L2_E	0.54	8.66	15.73	-45.34	-3.58	-5.38	57.25	1.54	-1.75
COP_EN	0.44	7.00	16.01	-47.68	-3.72	-5.61	56.40	1.54	-1.78

Table G.20: Backtest Metrics for Shrinked_MV with thresholds selection $OPTI_{PREC}$. All data are expressed in % except the Sharpe Ratio.

	SR	APY	Vol.	DD	VaR	CVaR	WR	Av. Ga.	Av. Lo.
COS_E	0.76	10.56	13.02	-35.51	-2.93	-4.27	59.91	1.31	-1.43
COS_IL	0.76	10.39	12.88	-34.78	-2.94	-4.22	59.59	1.31	-1.42
NASDAQ-100	0.75	14.95	19.70	-51.42	-4.66	-6.79	61.73	1.90	-2.27
COP_L1_E	0.74	10.34	13.05	-34.05	-2.96	-4.27	59.70	1.31	-1.43
COP_L2_E	0.74	10.07	12.83	-33.64	-2.92	-4.19	59.38	1.30	-1.40
SA	0.73	10.23	13.12	-35.51	-2.93	-4.28	59.28	1.33	-1.43
COP_L1_IL	0.73	10.12	13.05	-36.18	-2.93	-4.27	59.70	1.31	-1.43
COP_CE_E	0.73	10.03	12.97	-33.99	-2.96	-4.24	59.06	1.32	-1.41
CO	0.72	9.81	12.76	-33.93	-2.93	-4.15	59.59	1.28	-1.41
COP_L2_IL	0.72	9.85	12.83	-35.11	-2.92	-4.21	59.28	1.30	-1.40
IL	0.72	10.00	13.14	-36.18	-2.91	-4.29	59.17	1.33	-1.43
COP_EN	0.71	9.75	12.83	-35.50	-2.94	-4.19	59.59	1.29	-1.42
COP_CE_IL	0.71	9.85	12.99	-35.16	-2.96	-4.26	59.06	1.32	-1.42
COP_EN_IL	0.71	9.80	13.03	-36.16	-2.90	-4.25	59.28	1.31	-1.42
T-SA-1	0.71	9.88	13.18	-37.25	-2.91	-4.35	58.96	1.33	-1.43
T-SA-4	0.70	10.01	13.52	-37.86	-3.04	-4.47	59.06	1.35	-1.45
T-SA-2	0.68	9.63	13.29	-38.17	-2.92	-4.42	58.85	1.34	-1.44
COP_EN_E	0.68	9.34	12.95	-35.09	-2.96	-4.25	59.17	1.30	-1.42
T-SA-CV	0.68	9.55	13.31	-37.73	-3.05	-4.42	58.85	1.33	-1.43
T-SA-3	0.67	9.46	13.34	-37.73	-3.05	-4.43	59.17	1.32	-1.45
T-MCS-50	0.59	8.51	13.87	-43.03	-3.05	-4.68	58.21	1.38	-1.50
T-MCS-20	0.53	7.72	13.95	-43.55	-3.06	-4.73	58.00	1.38	-1.52

Table G.21: Backtest Metrics for Shrinked_MV with thresholds selection OPTI_RECALL. All data are expressed in % except the Sharpe Ratio.

	SR	APY	Vol.	DD	VaR	CVaR	WR	Av. Ga.	Av. Lo.
COP_L2_IL	0.82	13.75	16.03	-43.75	-3.21	-5.29	60.23	1.57	-1.68
COP_L2_E	0.82	13.58	15.95	-44.77	-3.17	-5.24	60.34	1.55	-1.68
COP_CE_IL	0.79	12.98	15.74	-43.18	-3.20	-5.13	59.59	1.55	-1.65
COP_CE_E	0.78	12.76	15.64	-44.14	-3.15	-5.08	59.38	1.55	-1.64
NASDAQ-100	0.75	14.95	19.70	-51.42	-4.66	-6.79	61.73	1.90	-2.27
SA	0.74	12.06	15.65	-47.05	-3.36	-5.18	58.53	1.54	-1.60
COP_EN_E	0.74	12.18	15.88	-44.16	-3.41	-5.27	59.38	1.55	-1.67
COP_EN	0.74	12.13	15.84	-44.84	-3.31	-5.24	60.66	1.51	-1.71
COP_L1_IL	0.73	11.93	15.86	-39.27	-3.45	-5.18	58.96	1.56	-1.65
T-SA-4	0.73	11.68	15.53	-46.50	-3.31	-5.18	59.38	1.51	-1.63
COP_EN_IL	0.72	11.67	15.73	-40.67	-3.43	-5.17	59.17	1.55	-1.67
CO	0.72	11.62	15.66	-47.06	-3.32	-5.28	60.23	1.51	-1.69
COP_L1_E	0.71	11.32	15.51	-41.81	-3.21	-5.14	58.32	1.54	-1.61
IL	0.70	11.60	16.19	-44.25	-3.39	-5.38	58.53	1.58	-1.66
COS_IL	0.69	11.22	15.81	-44.20	-3.35	-5.25	59.38	1.53	-1.67
COS_E	0.68	10.89	15.43	-42.90	-3.15	-5.07	58.64	1.52	-1.62
T-SA-1	0.64	10.22	15.50	-46.89	-3.14	-5.25	59.91	1.47	-1.68
T-SA-2	0.64	10.17	15.44	-46.89	-3.31	-5.26	59.06	1.48	-1.63
T-SA-3	0.64	10.04	15.27	-45.47	-3.22	-5.15	57.68	1.51	-1.57
T-SA-CV	0.63	9.87	15.29	-45.47	-3.14	-5.13	57.14	1.51	-1.55
T-MCS-50	0.58	8.72	14.58	-44.93	-3.14	-4.90	59.06	1.40	-1.58
T-MCS-20	0.52	7.71	14.42	-43.64	-3.17	-4.85	58.00	1.40	-1.55

Table G.22: Backtest Metrics for Shrinked_MV with thresholds selection QUANTILE. All data are expressed in % except the Sharpe Ratio.

	SR	APY	Vol.	DD	VaR	CVaR	WR	Av. Ga.	Av. Lo.
NASDAQ-100	0.75	14.95	19.70	-51.42	-4.66	-6.79	61.73	1.90	-2.27
T-SA-1	0.66	11.34	16.96	-46.69	-3.59	-5.52	55.01	1.77	-1.76
SA	0.66	12.03	18.28	-34.54	-3.98	-6.09	54.58	1.86	-1.84
IL	0.66	11.95	18.15	-34.54	-3.98	-6.00	54.80	1.85	-1.84
COP_CE_E	0.64	11.29	17.51	-37.01	-3.67	-6.02	56.40	1.75	-1.85
COP_CE_IL	0.64	11.32	17.86	-33.12	-3.77	-6.05	55.97	1.78	-1.84
COP_L1_IL	0.61	11.10	18.38	-33.86	-4.01	-6.21	55.76	1.84	-1.92
T-SA-4	0.60	10.38	17.27	-39.06	-3.41	-5.62	56.18	1.73	-1.75
COP_EN_E	0.59	10.74	18.56	-35.61	-3.91	-6.32	56.61	1.77	-1.88
COP_L2_E	0.58	10.35	18.11	-33.91	-3.64	-6.13	56.40	1.73	-1.86
T-SA-2	0.58	9.89	17.31	-38.74	-3.52	-5.71	56.40	1.73	-1.83
COP_L2_IL	0.57	10.13	17.98	-33.91	-3.58	-6.03	56.93	1.71	-1.89
T-SA-CV	0.57	9.76	17.46	-39.53	-3.43	-5.78	56.40	1.70	-1.81
COP_EN_IL	0.53	9.49	18.62	-35.61	-3.94	-6.41	55.65	1.80	-1.88
COS_IL	0.53	9.47	18.60	-33.46	-3.97	-6.42	56.29	1.76	-1.89
CO	0.50	9.22	19.32	-41.44	-3.89	-6.56	56.40	1.82	-1.95
COP_L1_E	0.50	8.96	18.69	-35.61	-3.98	-6.42	55.12	1.81	-1.89
T-MCS-50	0.50	7.62	15.05	-37.91	-3.18	-4.77	55.65	1.54	-1.57
COS_E	0.48	8.51	18.65	-34.54	-4.00	-6.58	55.76	1.77	-1.89
COP_EN	0.47	8.53	19.09	-40.21	-3.93	-6.44	55.97	1.79	-1.91
T-MCS-20	0.46	6.92	14.83	-43.82	-3.18	-4.77	55.97	1.50	-1.56
T-SA-3	0.45	7.42	16.73	-39.55	-3.45	-5.57	56.61	1.63	-1.78

Table G.23: Backtest Metrics for FILTERING with thresholds selection CV. All data are expressed in % except the Sharpe Ratio.

	SR	APY	Vol.	DD	VaR	CVaR	WR	Av. Ga.	Av. Lo.
T-SA-3	0.95	15.60	15.54	-38.12	-3.06	-4.95	58.42	1.63	-1.57
T-SA-2	0.93	14.72	14.99	-31.61	-3.03	-4.71	59.06	1.55	-1.55
T-SA-CV	0.91	14.55	15.21	-38.12	-3.02	-4.80	58.21	1.60	-1.55
COP_L1_E	0.89	14.17	15.12	-36.95	-3.15	-4.82	58.74	1.57	-1.56
CO	0.88	14.15	15.26	-40.55	-3.21	-4.76	55.97	1.66	-1.48
COP_L1_IL	0.88	14.03	15.22	-38.66	-3.27	-4.81	59.28	1.56	-1.60
COP_CE_E	0.87	13.90	15.15	-37.76	-3.27	-4.78	58.53	1.58	-1.56
COS_E	0.87	13.90	15.19	-36.97	-3.24	-4.79	58.53	1.58	-1.57
T-SA-1	0.87	14.28	15.68	-32.72	-3.10	-5.03	58.64	1.61	-1.61
COP_L2_E	0.85	13.70	15.35	-38.92	-3.27	-4.75	56.72	1.63	-1.52
COP_CE_IL	0.85	13.57	15.26	-39.34	-3.27	-4.81	58.10	1.60	-1.58
SA	0.85	13.31	14.95	-33.37	-3.06	-4.75	58.10	1.56	-1.54
COP_L2_IL	0.84	13.63	15.41	-39.28	-3.27	-4.78	56.82	1.63	-1.53
COP_EN_IL	0.84	13.49	15.29	-39.94	-3.27	-4.81	58.42	1.58	-1.58
COP_EN_E	0.84	13.42	15.25	-39.52	-3.26	-4.79	57.89	1.59	-1.56
COP_EN	0.83	13.32	15.25	-37.37	-3.24	-4.80	56.61	1.62	-1.51
IL	0.79	12.94	15.65	-40.01	-3.20	-5.06	58.85	1.58	-1.64
T-SA-4	0.79	12.51	15.15	-39.87	-3.03	-4.78	57.78	1.56	-1.54
COS_IL	0.79	12.67	15.38	-41.03	-3.26	-4.91	57.57	1.61	-1.59
NASDAQ-100	0.75	14.95	19.70	-51.42	-4.66	-6.79	61.73	1.90	-2.27
T-MCS-50	0.50	7.31	14.25	-38.03	-3.15	-4.63	55.86	1.47	-1.50
T-MCS-20	0.43	6.17	13.82	-40.39	-2.95	-4.54	53.41	1.43	-1.54

Table G.24: Backtest Metrics for FILTERING with thresholds selection NAIVE. All data are expressed in % except the Sharpe Ratio.

	SR	APY	Vol.	DD	VaR	CVaR	WR	Av. Ga.	Av. Lo.
IL	0.97	17.08	16.81	-35.37	-3.28	-5.30	58.21	1.73	-1.68
SA	0.95	16.73	16.79	-34.69	-3.35	-5.29	57.68	1.74	-1.66
T-SA-CV	0.91	15.63	16.38	-38.90	-3.24	-5.13	58.85	1.65	-1.68
T-SA-4	0.83	13.87	16.05	-39.42	-3.09	-5.00	57.78	1.63	-1.59
COP_EN_E	0.79	13.81	17.05	-31.80	-3.49	-5.58	57.89	1.71	-1.70
COS_IL	0.78	13.92	17.43	-32.33	-3.48	-5.77	57.14	1.74	-1.75
T-SA-1	0.78	13.09	16.32	-41.43	-3.24	-5.15	58.21	1.64	-1.67
COS_E	0.77	13.73	17.29	-34.69	-3.52	-5.74	57.14	1.76	-1.76
NASDAQ-100	0.75	14.95	19.70	-51.42	-4.66	-6.79	61.73	1.90	-2.27
COP_EN	0.75	13.75	17.97	-40.53	-3.50	-5.95	58.21	1.75	-1.81
COP_EN_IL	0.74	12.86	16.89	-34.69	-3.57	-5.59	58.53	1.67	-1.73
T-SA-3	0.74	12.27	16.00	-38.90	-3.31	-5.25	58.42	1.59	-1.67
T-SA-2	0.72	11.82	15.95	-37.50	-3.21	-5.12	57.68	1.59	-1.61
COP_L2_IL	0.68	11.19	16.14	-35.90	-3.44	-5.36	57.25	1.61	-1.65
COP_L2_E	0.65	11.06	16.65	-42.07	-3.57	-5.52	57.04	1.66	-1.74
COP_L1_IL	0.65	11.46	17.62	-34.69	-3.76	-5.90	57.25	1.73	-1.81
COP_CE_IL	0.64	11.26	17.40	-33.03	-3.65	-5.68	55.65	1.77	-1.78
COP_L1_E	0.64	11.24	17.55	-34.66	-3.79	-5.96	57.14	1.72	-1.80
COP_CE_E	0.61	10.44	17.08	-39.53	-3.63	-5.61	55.44	1.74	-1.72
CO	0.59	10.53	17.96	-41.59	-3.54	-5.98	56.29	1.75	-1.79
T-MCS-50	0.56	8.71	15.17	-37.91	-3.18	-4.79	55.65	1.58	-1.57
T-MCS-20	0.49	7.29	14.36	-43.82	-3.00	-4.63	53.20	1.49	-1.54

Table G.25: Backtest Metrics for FILTERING with thresholds selection OPTI. All data are expressed in % except the Sharpe Ratio.

	SR	APY	Vol.	DD	VaR	CVaR	WR	Av. Ga.	Av. Lo.
T-SA-1	1.11	19.29	16.26	-29.30	-3.19	-5.02	59.59	1.70	-1.67
COP_EN_E	1.10	19.07	16.24	-29.68	-3.17	-4.98	58.96	1.70	-1.64
SA	1.08	18.34	15.86	-29.64	-3.07	-4.71	57.14	1.71	-1.54
IL	1.08	18.53	16.10	-30.19	-3.16	-4.81	57.36	1.74	-1.58
COP_EN_IL	1.00	16.78	15.84	-30.32	-3.24	-4.87	57.78	1.66	-1.60
COP_L1_IL	1.00	17.25	16.34	-31.17	-3.42	-4.96	57.68	1.72	-1.64
T-SA-3	0.99	16.75	16.00	-38.91	-3.18	-5.02	58.21	1.65	-1.61
T-SA-CV	0.97	16.80	16.47	-38.90	-3.36	-5.18	58.53	1.68	-1.65
COP_L1_E	0.91	15.62	16.36	-29.80	-3.44	-5.10	56.93	1.72	-1.65
COS_IL	0.89	15.97	17.16	-32.60	-3.44	-5.45	56.61	1.79	-1.71
T-SA-2	0.89	14.74	15.81	-32.61	-3.27	-5.06	58.32	1.63	-1.63
COS_E	0.88	15.44	16.84	-32.60	-3.42	-5.36	56.08	1.78	-1.68
T-SA-4	0.82	13.76	16.06	-39.43	-3.29	-5.01	57.78	1.62	-1.61
COP_L2_IL	0.80	13.50	16.30	-40.22	-3.44	-5.18	56.72	1.71	-1.71
NASDAQ-100	0.75	14.95	19.70	-51.42	-4.66	-6.79	61.73	1.90	-2.27
COP_L2_E	0.70	11.65	16.29	-45.92	-3.55	-5.28	56.08	1.68	-1.72
COP_CE_IL	0.68	12.17	17.70	-40.51	-3.78	-5.78	55.22	1.84	-1.83
CO	0.64	10.65	16.24	-39.52	-3.52	-5.30	57.78	1.64	-1.78
T-MCS-50	0.60	9.55	15.55	-37.91	-3.23	-4.83	55.44	1.62	-1.58
COP_CE_E	0.59	10.15	17.43	-42.18	-3.66	-5.64	53.52	1.83	-1.79
COP_EN	0.46	7.53	16.62	-43.38	-3.62	-5.61	55.76	1.65	-1.85
T-MCS-20	0.46	6.92	14.83	-43.82	-3.18	-4.77	55.97	1.50	-1.56

Table G.26: Backtest Metrics for FILTERING with thresholds selection $OPTI_{PREC}$. All data are expressed in % except the Sharpe Ratio.

	SR	APY	Vol.	DD	VaR	CVaR	WR	Av. Ga.	Av. Lo.
COS_E	0.80	11.21	13.16	-29.06	-2.84	-4.10	58.00	1.40	-1.41
T-SA-1	0.78	11.03	13.32	-29.06	-2.84	-4.16	57.46	1.42	-1.40
COP_L1_IL	0.77	10.81	13.19	-30.63	-2.84	-4.07	57.57	1.40	-1.40
COP_L1_E	0.77	10.78	13.22	-26.30	-2.83	-4.09	57.78	1.40	-1.41
COS_IL	0.76	10.61	13.09	-27.92	-2.84	-4.01	57.68	1.40	-1.40
NASDAQ-100	0.75	14.95	19.70	-51.42	-4.66	-6.79	61.73	1.90	-2.27
SA	0.75	10.54	13.31	-29.04	-2.84	-4.11	57.14	1.42	-1.41
IL	0.73	10.40	13.35	-30.64	-2.87	-4.11	57.04	1.43	-1.41
COP_CE_E	0.73	10.26	13.15	-25.78	-2.79	-4.01	57.36	1.40	-1.40
COP_L2_E	0.73	10.15	13.09	-25.44	-2.84	-3.97	57.46	1.39	-1.40
COP_L2_IL	0.72	10.01	13.09	-28.22	-2.87	-4.01	57.36	1.39	-1.40
COP_CE_IL	0.71	10.00	13.19	-28.26	-2.84	-4.05	57.04	1.41	-1.40
CO	0.71	9.82	12.96	-25.30	-2.81	-3.94	57.46	1.38	-1.40
COP_EN	0.71	9.81	13.04	-28.25	-2.85	-3.99	57.14	1.39	-1.40
T-SA-2	0.70	10.06	13.62	-29.86	-2.87	-4.30	57.36	1.42	-1.44
T-SA-4	0.69	10.08	13.95	-30.00	-2.98	-4.36	57.89	1.44	-1.49
COP_EN_IL	0.69	9.58	13.19	-30.67	-2.86	-4.08	56.50	1.41	-1.38
T-SA-CV	0.66	9.49	13.71	-30.06	-2.86	-4.28	57.36	1.42	-1.45
T-SA-3	0.66	9.50	13.74	-30.06	-2.87	-4.32	57.89	1.41	-1.48
COP_EN_E	0.64	8.92	13.13	-28.25	-2.88	-4.07	56.93	1.38	-1.41
T-MCS-50	0.50	7.30	14.25	-38.03	-3.15	-4.63	55.86	1.47	-1.50
T-MCS-20	0.40	5.81	14.31	-40.36	-3.18	-4.69	56.18	1.44	-1.55

Table G.27: Backtest Metrics for FILTERING with thresholds selection OPTI_RECALL. All data are expressed in % except the Sharpe Ratio.

	SR	APY	Vol.	DD	VaR	CVaR	WR	Av. Ga.	Av. Lo.
COP_L2_IL	0.82	14.76	17.56	-44.98	-3.44	-5.59	58.96	1.73	-1.76
COP_L2_E	0.79	14.25	17.48	-45.81	-3.44	-5.55	58.42	1.73	-1.74
COP_CE_IL	0.79	13.68	16.93	-44.81	-3.41	-5.36	57.04	1.73	-1.66
COP_EN	0.76	13.44	17.20	-45.54	-3.52	-5.49	57.68	1.72	-1.70
NASDAQ-100	0.75	14.95	19.70	-51.42	-4.66	-6.79	61.73	1.90	-2.27
COP_CE_E	0.75	13.02	16.82	-46.53	-3.41	-5.34	56.82	1.72	-1.65
COP_L1_IL	0.75	12.80	16.65	-32.05	-3.45	-5.27	56.72	1.72	-1.66
SA	0.72	12.32	16.88	-46.69	-3.52	-5.42	57.36	1.70	-1.71
CO	0.71	12.27	16.83	-48.96	-3.49	-5.51	57.68	1.68	-1.70
IL	0.70	12.38	17.50	-44.26	-3.51	-5.58	57.14	1.74	-1.74
COP_EN_E	0.69	12.10	17.18	-46.68	-3.58	-5.56	56.18	1.75	-1.68
COP_L1_E	0.69	11.71	16.64	-42.51	-3.46	-5.33	56.29	1.70	-1.65
COP_EN_IL	0.69	11.73	16.70	-37.34	-3.57	-5.33	56.40	1.73	-1.69
T-SA-4	0.68	11.52	16.74	-44.44	-3.48	-5.45	57.57	1.65	-1.69
COS_IL	0.66	11.30	16.99	-47.48	-3.40	-5.48	56.29	1.71	-1.67
COS_E	0.64	10.83	16.63	-45.04	-3.51	-5.31	56.50	1.68	-1.67
T-SA-2	0.63	10.67	16.78	-49.02	-3.58	-5.49	57.89	1.63	-1.72
T-SA-CV	0.60	9.96	16.46	-43.88	-3.34	-5.31	56.82	1.64	-1.68
T-SA-1	0.60	10.12	16.85	-49.24	-3.50	-5.50	56.72	1.67	-1.71
T-SA-3	0.58	9.50	16.34	-43.87	-3.31	-5.33	56.82	1.62	-1.67
T-MCS-50	0.48	7.39	15.38	-45.04	-3.27	-4.98	56.72	1.52	-1.62
T-MCS-20	0.37	5.46	14.76	-43.82	-3.23	-4.84	55.86	1.47	-1.58

Table G.28: Backtest Metrics for FILTERING with thresholds selection QUANTILE. All data are expressed in % except the Sharpe Ratio.

Abstract:

Over the past few decades, it has proven extremely difficult to consistently outperform financial markets. For instance, about 90% of active traders and fund managers fail to beat the major benchmark indices over extended horizons. At the same time, the availability of financial data has exploded and machine learning (ML) methods have advanced significantly. This convergence offers a promising opportunity to revisit traditional approaches to return forecasting and portfolio construction. Drawing on empirical data from large-scale forecasting competitions and recent research on ensemble learning, this thesis examines whether combining multiple machine learning models can improve short-term stock return forecasting and enhance portfolio performance.

Specifically, this thesis develops a set of ML classifiers to predict whether individual stocks will outperform their benchmark index over the following week. The models are trained using a walk-forward approach over a long historical period (January 2007-December 2024), allowing for continuous re-estimation to adapt to changing market conditions. Particular emphasis is placed on threshold selection strategies, which transform the continuous results from the combination into binary signals. These thresholds are optimized according to various criteria and are found to have a significant influence on trading performance.

The predicted signals are integrated into a portfolio construction framework using various weights allocations techniques. Of the 544 portfolios constructed, all those that outperformed the benchmark did so using ensemble machine learning strategies. The best performing strategy combined a mean-variance portfolio allocation with a filtering-based covariance estimator and Ledoit-Wolf shrinkage, as well with a trimmed ensemble of ML models whose threshold was optimized to maximize precision. This configuration results in a Sharpe ratio of 1.11 and an APY of 19.29%. By comparison, its universe index achieved a Sharpe ratio of 0.74 and an APY of 14.72% over the same period. Moreover, 11 of the 15 best-performing portfolios used dynamic threshold techniques explicitly designed to maximize classification precision.

Overall, this research demonstrates empirically the practical potential of ensemble learning and adaptive thresholding in financial forecasting and portfolio management.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
Louvain School of Management

Place des Doyens, 1 bte L2.01.01, 1348 Louvain-la-Neuve
Boulevard Emile Devreux 6, 6000 Charleroi, Belgique
Chaussée de Binche 151, 7000 Mons, Belgique

www.uclouvain.be/lsm