Ensemble Meta-Labeling

Dennis Thumm, Paolo Barucca, and Jacques Francois Joubert

Dennis Thumm
is a quantitative researcher at Hudson and Thames Quantitative Research in London, UK.
dennis@hudsonthames.org

Paolo Barucca
is a lecturer in the Department of Computer Science at University College London in London, UK.
p.barucca@ucl.ac.uk

Jacques Francois Joubert
is the chief executive officer of Hudson and Thames Quantitative Research in London, UK.
jacques@hudsonthames.org

KEY FINDINGS

- The proposed framework is a novel contribution to the evolving field of meta-labeling. It demonstrates how several individual models can be combined to obtain a better generalization performance on metrics for the meta-labeling process.
- The authors demonstrate how ensembles improve the detection and extraction of regimes, leading to higher model robustness.
- The authors show how ensembles increase model efficiency by decreasing the rate of false positives, thus increasing the meta premium.

ABSTRACT

This study systematically investigates different ensemble methods for meta-labeling in finance and presents a framework to facilitate the selection of ensemble learning models for this purpose. Experiments were conducted on the components of information advantage and modeling for false positives to discover whether ensembles were better at extracting and detecting regimes and whether they increased model efficiency. The authors demonstrate that ensembles are especially beneficial when the underlying data consist of multiple regimes and are nonlinear in nature. The authors' framework serves as a starting point for further research. They suggest that the use of different fusion strategies may foster model selection. Finally, the authors elaborate on how additional applications, such as position sizing, may benefit from their framework.

Ensemble learning methods are the best-performing technique for predictive modeling. Yet why are ensembles superior to single models, and how should model architectures for meta-labeling be selected? This article aims to answer these questions by providing practitioners with a framework for selecting suitable meta-labeling models for a given primary strategy. This is the first study to implement the ensemble architectures for meta-labeling suggested by Meyer, Joubert, and Mesias (2022).

The 1990s saw a significant increase in the study of ensemble methods, and during that decade, articles were published on the most popular and widely applied techniques, such as the core bagging, boosting, and stacking techniques (Zhou 2012). Due in part to ensembles' enormous success in machine learning (ML) competitions, such as the Netflix Prize, the use of ensembles increased in the late 2000s. There are two major, related reasons for choosing an ensemble over a single model (Dietterich 2000; Polikar 2006):

- Performance: Compared to a single contributing model, an ensemble can predict events more accurately and perform better overall.
- Robustness: An ensemble narrows the prediction and model performance distribution.

The core tenet of ensemble learning is that merging numerous models allows the mistakes of one model to be compensated for by other models, boosting the ensemble's overall prediction performance over that of any single model (Sagi and Rokach 2018). Improved robustness or reliability in a model's average performance is another significant yet underdiscussed advantage. Performance and robustness are both crucial aspects of ML projects, and practitioners may occasionally favor one or the other of the model's properties.

The forecasting of financial time series to create trading and investment strategies is a challenging problem for quantitative investment teams. The use of ML as an overlay to a primary strategy to help size positions, remove false-positive signals, and enhance strategy metrics such as the Sharpe ratio and maximum drawdown is an appealing proposition. This procedure is called meta-labeling and was introduced by Lopez de Prado (2018) in his book, *Advances in Financial Machine Learning*.

Meta-labeling involves fitting a secondary model to determine whether a primary exogenous model is correct and to size positions accordingly. Binary meta-labels are target variables that indicate whether or not the primary model's forecast was profitable. The probability of a positive outcome can then be used to size positions. Meta-labeling involves a trade-off between recall and precision, which leads to improved model efficiency as measured by an increase in the F1 score (Lopez de Prado 2018).

Meyer, Joubert, and Mesias (2022) provided insightful meta-labeling architectures, and Joubert (2022) suggested creating a framework for choosing model architectures for meta-labeling and investigating the incorporation of ensembles. This is the motivation and research objective of the present article.

Following this introduction, the article begins with an overview of important concepts before summarizing the research methods and goals. Next, the results of our experiments are presented and interpreted. Based on the experimental results, we derive and present a protocol for the ensemble model selection framework for meta-labeling. The article concludes by revisiting the main findings of the investigation, highlighting the contributions of this article and offering suggestions for future studies.

THEORETICAL FRAMEWORK

Ensemble Learning

The phrase *ensemble learning* refers to a range of methods often employed in supervised ML tasks to blend several inducers to arrive at a conclusion. A model (such as a classifier or regressor) is created using a set of labeled examples as input and an algorithm known as an inducer or base-learner. The developed model can then be used to make predictions for new, unlabeled instances. Any ML method can act as an ensemble inducer (e.g., decision tree, neural network, linear regression [LR] model). The aim of ensemble learning is to merge numerous models so that the mistakes of one inducer will be compensated for by other inducers, boosting the ensemble's overall prediction performance over that of a single inducer (Sagi and Rokach 2018).

Model Selection

Model selection is the process of comparing either models of the same type that have been configured with different model hyperparameters (e.g., different kernels in a support vector machine [SVM]) or models of different kinds (such as logistic regression, SVM, *k*-nearest neighbors [KNN]) (Murphy 2012). For instance, when creating a classification or regression predictive model for a dataset, it is impossible to predict beforehand which model will solve the problem best. Therefore, it is necessary to fit a variety of models to the issue and evaluate their performance.

Model selection is the process by which we statistically evaluate and compare potential models before selecting the best one. In contrast, model assessment involves evaluating a model after it has been selected to convey how well it is anticipated to perform in general (James et al. 2013).

Meta-Labeling

The goal of ensemble learning is to enable more accurate predictions than would be possible with any one of the individual learning algorithms. It is a wisdom-of-crowds method that compiles data from several models into a set of remarkably accurate findings.

Although meta-labeling shares certain similarities with an ensemble method called stacking, they are essentially different processes. Stacking involves just two phases: After the other algorithms have been trained using the available data, the combiner algorithm is taught to create a final prediction utilizing the predictions of the other algorithms as extra inputs. It may be viewed as merely incorporating new characteristics in the training set.

Although meta-labeling employs two layers of models, these serve completely different purposes. First, we develop a primary model that provides good recall even if the accuracy is not particularly great. Then we compensate for the low performance by applying meta-labeling to the positives predicted by the primary model, as demonstrated by Joubert (2022).

Meta-labeling is very helpful when attempting to improve F1 results (Lopez de Prado 2018). The key is to train a secondary ML model to utilize the primary model. This leads to improvements in performance parameters, including accuracy, precision, and F1 score, while trading off some recall (Joubert 2022).

METHODOLOGY

To demonstrate how ensemble learning improves the performance metrics during each stage of the meta-labeling process, two separate experiments were conducted:

1. Increasing the informational advantage
2. Modeling for false positives

The methodology utilized closely resembled that of the experiments conducted by Joubert (2022).

Research Aims, Questions, and Hypothesis

The goal of this study was to develop an ensemble learning model selection framework for meta-labeling. To achieve this, we explored different frameworks for combining secondary models. We hypothesized that an ensemble classifier would outperform a single baseline classifier.

In testing this hypothesis, we aimed to verify the potential of ensemble learning in meta-labeling. One potential benefit of ensemble learning is its ability to handle concept drift from nonstationary distributions (Zhang and Ma 2012). In this context, two key characteristics are the method used to create diversity within an ensemble and the method chosen to combine it (Rokach 2010). Moreover, the base learners need to have high sensitivity and efficient learning ability.

Our approach was to combine several individual models to obtain a better generalization performance. In our experiments, we also compared shallow models

(e.g., logistic regression) with deep models (e.g., neural networks). The correlation strength was another parameter to consider because we aimed to create low-correlated ensemble models to increase the ensemble diversity.

The interplay between bias and variance in ML is unavoidable. The variance decreases as the bias increases. Similarly, the bias will decrease as the variance rises. There is thus a trade-off between these two quantities, and, depending on the algorithms selected and how they are configured, this trade-off will be balanced differently for each particular problem (Kohavi and Wolpert 1996). In light of this trade-off, we aimed to find a balanced configuration.

Our experiments were designed to answer two research questions:

1. Does an ensemble of models increase the information advantage by detecting and extracting regimes more effectively?
2. Does ensemble learning improve model efficiency by reducing false positives, thus increasing the meta premium?

Synthetic Data Generation

An autoregressive process of order three, AR(3), was used to create a linear time series for each experiment.

$$r_{t+1} = \phi_0 + \phi_1 r_t + \phi_2 r_{t-1} + \phi_3 r_{t-2} + a_t \tag{1}$$

where the time series r_{t+1} represents the predicted daily return, and r_t, r_{t-1} and r_{t-2} represent the daily and lagged 1 and 2 returns, respectively. The model coefficients are ϕ_0, ϕ_1, ϕ_2, ϕ_3, and a_t is a white noise series with a mean of 0 and a variance of 0.000212 for the low-volatility regime and 0.000423 for the high-volatility regime. A random sample of IBM's daily returns was used to determine the variance.

In each experiment, a quadruple autoregressive process was used to generate 10,000 sequential daily returns. The quadruple autoregressive process consisted of four separate AR(3) processes. In total, there were four regimes: positive/negative algebraic signs and low/high volatility. After applying the primary model, the dataset was filtered to only include observations that matched what the primary model predicted would be a positive change in direction because the chosen trading strategy was long only. Next, the data were divided 60/40 into a train and test set without being shuffled, meaning that the test set contained only the most recent 40% of the data.

Because we wanted to investigate whether an ensemble was superior to a single model, we introduced different drift and volatility regimes. This was achieved by altering the variance of a_t. In addition to the algebraic sign swap described in the dual autoregressive process of Joubert (2022), a second uniformly random number $[0, 1]$ was generated. If it exceeded a threshold of 0.80, the variance was doubled to simulate a high-volatility regime. Similar to the drift component, this process was repeated every 30 observations.

This added two additional regimes for high and low volatility on top of the existing dual autoregressive process, creating a total of four separate AR(3) processes:

$$r_{t+1} = 0.25r_t - 0.20r_{t-1} + 0.35r_{t-2} + a_{t(low)} \tag{2}$$

$$r_{t+1} = 0.25r_t - 0.20r_{t-1} + 0.35r_{t-2} + a_{t(high)} \tag{3}$$

$$r_{t+1} = -0.0001 - 0.25r_t + 0.20r_{t-1} - 0.35r_{t-2} + a_{t(low)} \tag{4}$$

$$r_{t+1} = -0.0001 - 0.25r_t + 0.20r_{t-1} - 0.35r_{t-2} + a_{t(high)} \tag{5}$$

Underlying Primary Model

The primary model was based on Narro and Caamano's (2020) German power autocorrelation strategy. A simple rule was used to generate an end-of-day buy signal while adding the restriction of long only: If the price difference between yesterday's close and today's close is positive, go long at today's close; otherwise, exit the position (Joubert 2022).

$$\dot{y}_t = \begin{cases} 1, & if \ \Delta r_t > 0 \\ \dot{y}_{t-1}, & if \ \Delta r_t = 0 \\ 0, & if \ \Delta r_t < 0 \end{cases} \tag{6}$$

where Δr_t is the difference in returns between $r_t - r_{t-1}$ and $\dot{y}_t$ is the side of position $\dot{y}_t \in \{0, 1\}$.

Secondary Model

The secondary model's objective was to ascertain whether the primary model's side prediction was accurate and whether a position should be taken. In a binary classification setting, logistic regression can be used as a baseline because it offers good explainability and requires few tuning parameters. A linear classification model such as logistic regression was appropriate because the data generation process was linear (Joubert 2022).

The model used the last three returns, r_t, r_{t-1}, and r_{t-2}, for its features. An additional regime detection indicator was added after the quadruple autoregressive process had been applied. Whether a data point was produced using Equations 2, 3, 4, or 5 was tracked by the regime indicators and flagged accordingly.

To simulate the lag effect of a statistic that uses a window size in its calculation, the regime indicator was then lagged by five observations. By eliminating the mean and scaling each feature to a unit variance based on the training dataset, the features in both the train and test sets were standardized. The target variable was the meta-label—that is, whether the prediction resulted in a gain or a loss.

$$y_t = \begin{cases} 1, \ if \ \dot{y}_t = sign(\Delta r_{t+1}) = 1 \\ 0, \ otherwise \end{cases} \tag{7}$$

where y_t is the meta-label, $y_t \in \{0, 1\}$, and $\dot{y}_t$ is the side forecast from the primary model.

Neither of the experiments included regularization or hyperparameter tuning. This kept the experiments from becoming overly complicated, making it easier to interpret the results.

Although the automated 10-fold stratified cross-validation is generally accepted as the best practice for model selection (Kohavi 1995), it proved challenging to incorporate it for an individually selected pool of classifiers using *K*-nearest neighbor Oracle (KNORA). Therefore, we relied on a standard 60/40 train/test split. The ensemble learning and meta-labeling optimization is trained using the same training set as the individual classifiers. The secondary model is then scored using the test dataset, and those results are reported.

Meta-Labeling Ensembles

In addition to the logistic regression of Joubert (2022), we built ensembles for our secondary model. In creating the ensembles, we faced a trade-off between bias and variance. Low-bias and high-variance models include decision trees, KNN, and SVM. Examples of high-bias and low-variance models include linear/logistic regression and linear discriminant analysis.

Furthermore, models can be selected either statically or dynamically. Static ensemble selection involves averaging all members of the ensemble pool. Static selection criteria are classification and diversity measures, such as accuracy and correlation (Cruz, Sabourin, and Cavalcanti 2018). Dynamic selection aims to select the best combination of classifiers from each data sample because we cannot predict which combiner will produce the best results (Sergio, de Lima, and Ludermir 2016).

In our experiments, we tested three different ensemble frameworks and compared the results to those of a baseline model, for which we used LR. The three ensemble frameworks were (i) a sequential ensemble with a light gradient boosted machine (LightGBM; GBM in the experiments), (ii) a homogeneous dynamic ensemble selection with random forest (DES RF), and (iii) a heterogeneous dynamic ensemble selection with a classifier pool of logistic regression, decision tree classifier, support vector classifier (SVC), naïve Bayes, and a multilayer perceptron (MLP) neural network classifier (DES Pool).

Our choice of model architectures for each of the common ensemble methods—sequential, homogeneous, and heterogeneous—was based on certain considerations. Regarding the sequential ensemble, previous research has relied primarily on LightGBM for meta-labeling; see Mehta (2022) and Nousiainen (2021). LightGBM improves the gradient boosting technique by including a form of autonomous feature selection and concentrating on boosting cases with greater gradients. This may result in a significant increase in training speed and enhanced prediction performance. As a result, LightGBM has become the default approach for ML contests when dealing with tabular data for regression and classification predictive modeling tasks. As such, it, along with extreme gradient boosting (XGBoost), bears some of the responsibility for the rising popularity and broader use of gradient boosting approaches in general.

For our dynamic ensembles we relied on the KNORA. In determining the best ensemble for a given sample, the KNORA concept takes the neighborhood of test patterns into account. To classify a given pattern in the test set, KNORA simply locates the nearest K neighbors for each test data point in the validation set, determines which classifiers correctly classify those neighbors in the validation set, and uses those neighbors as the ensemble (Ko, Sabourin, and Britto 2008).

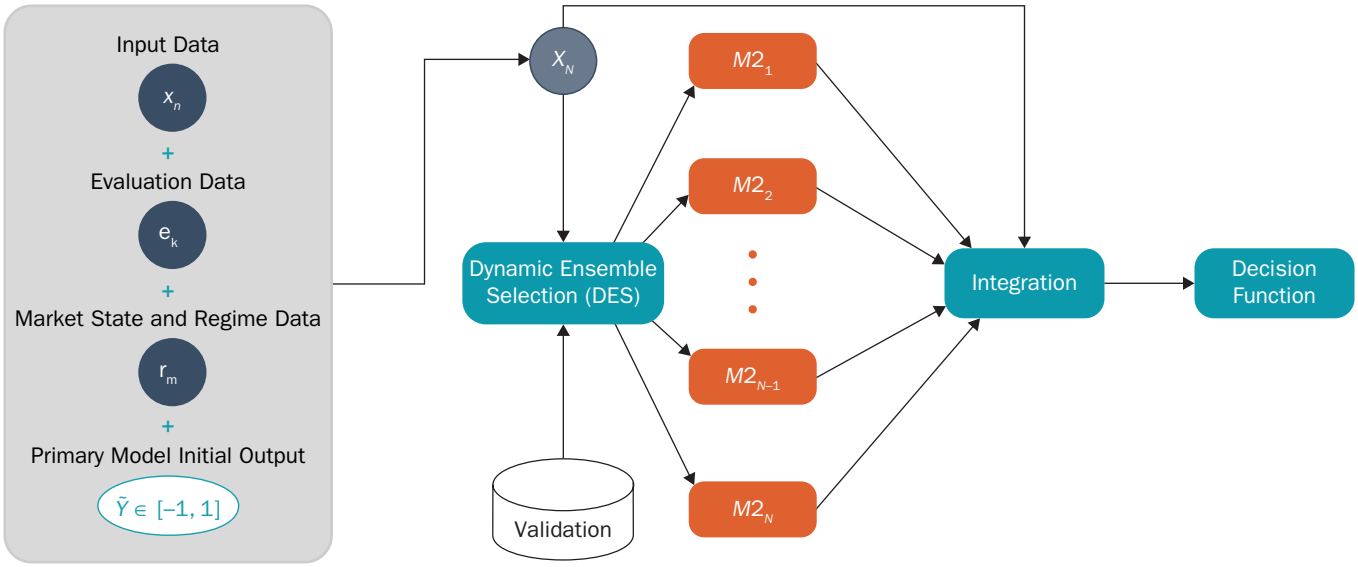
Exhibit 1 provides a flowchart of the dynamic meta-labeling ensemble. X_N denote the feature dataset. $M2_{1,\ldots,N}$ are the meta-labeling classifiers that represent the ensemble of classifiers.

RF (Breiman 2001) is one of the most successful ensemble methods, which is why we chose it for the homogeneous dynamic ensemble. It is an extension of bagging, with the main distinction being the inclusion of randomly selected feature selection. When building a component decision tree, at each stage of the split selection, the RF first randomly selects a subset of features before performing the standard split selection procedure within the chosen feature subset (Zhou 2012).

When choosing a pool of classifiers for a heterogeneous dynamic ensemble, we aimed to cover different areas of the feature space. Boosting and SVMs, two maximum margin methods, produce distinctive distortions in their predictions. They tend to shift probabilities away from 0 and 1, producing a sigmoidal shape. Predictions made using techniques such as naïve Bayes display an opposite distortion, pushing predictions toward 0 and 1. Additionally, techniques like bagged trees and neural nets result in well-calibrated probabilities (Niculescu-Mizil and Caruana 2005).

EXHIBIT 1

Flowchart of the Dynamic Meta-Labeling Ensemble



We chose logistic regression for a high-bias low-variance model, an SVC for a high-variance low-bias model and a naïve Bayes classifier due to its different distribution of outputs, as well as a neural network (MLP) and decision trees to maximize the data exploitation.

Position Sizing

In the experiments, an all-or-nothing algorithm was employed. When a favorable outcome was predicted, 100% of the portfolio was invested; otherwise, the algorithm exited/held a position size of 0%.

$$f_t = \begin{cases} 1, & \text{if } \hat{y}_t > 0.5 \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where f_t is the size of the position at time t, $f_t \in \{0, 1\}$, and $\hat{y}_t$ is the secondary model's prediction of a positive outcome, $\hat{y}_t \in [0, 1]$ (Joubert 2022).

Strategy Creation

By only taking a long position when the primary model advised going long and the secondary model confirmed the prediction with a favorable result, a new trading strategy was developed that filtered out false positives. Applying Equation 8 yielded the size of the position.

RESULTS

Experiment 1: Information Advantage

Does an ensemble increase the information advantage by detecting and extracting regimes more effectively?

EXHIBIT 2

Summary Statistics of the Changes in Strategy Metrics for the 1,000 Trials in Experiment 1: Informational Advantage

Strat Metric	Model	Mean	Median	Std Deviation	Min	Max
Δ Sharpe Ratio^{ip}	LR	1.1989	1.2051	0.4278	**–0.1572**	2.6335
	GBM	1.6390	1.6463	0.3067	**0.6846**	2.7176
	DES RF	1.6379	1.6558	0.3103	**0.5023**	2.6698
	DES Pool	1.6972	1.6956	0.3450	**0.4879**	2.7085
Δ Mean^{ip}	LR	0.0012	0.0011	0.0004	**0.0001**	0.0025
	GBM	0.0014	0.0014	0.0003	**0.0004**	0.0025
	DES RF	0.0014	0.0014	0.0003	**0.0003**	0.0024
	DES Pool	0.0014	0.0014	0.0004	**0.0004**	0.0027
Δ Standard Deviation^{ip}	LR	–0.0070	–0.0070	0.0010	**–0.0111**	–0.0038
	GBM	–0.0060	–0.0059	0.0007	**–0.0087**	–0.0042
	DES RF	–0.0061	–0.0061	0.0007	**–0.0091**	–0.0043
	DES Pool	–0.0060	–0.0056	0.0017	**–0.0120**	–0.0026

NOTES: The differences in Sharpe ratio, mean returns, and standard deviation between the secondary model and primary model are all scaled to an annualized level. ip denotes a metric that is the difference between a secondary model trained only on price data (i.e., the information advantage) and the primary model. The minimum values are boldface since those are the most conservative measures of model performance.

This was the question that Experiment 1 attempted to address. The primary model used only Δr_t to predict the side; however, this straightforward strategy disregards the fact that r_{t-1} and r_{t-2} are also informative. We produced a more precise forecast by fitting a secondary model that would take these data into account. The meta-premium, which is defined as the difference in trading strategy metrics between the secondary and the primary model, is summarized in Exhibit 2.

Exhibit 2 shows that, in all the trials, the ensemble classifiers had a positive premium across all metrics. However, this was not the case for our baseline logistic regression model. This is emphasized by the values in the minimum column. For the ensemble classifiers, higher expected returns and lower expected standard deviations increased the Sharpe ratio.

Therefore, the results show that meta-labeling is effective when it can use the data more effectively than the primary model. However, because we used a quadruple autoregressive linear process to generate our data, the logistic regression did not perform as well as when the single autoregression was utilized by Joubert (2022).

Different models use the data in different ways (Li, Turkington, and Yazdani 2020); as a result, increasing model diversity may improve performance when the underlying data consist of more regimes and are nonlinear.

Exhibit 3 illustrates the difference in classification metrics between the secondary and primary models (see Exhibit A1 in the Appendix for changes in area under the curve [AUC]). A net increase in performance is indicated by a positive change and a decrease by a negative change. The F1 score significantly increased for all models, which indicates a more effective classifier that can eliminate false positives by sacrificing some recall for greater precision (López de Prado 2020).

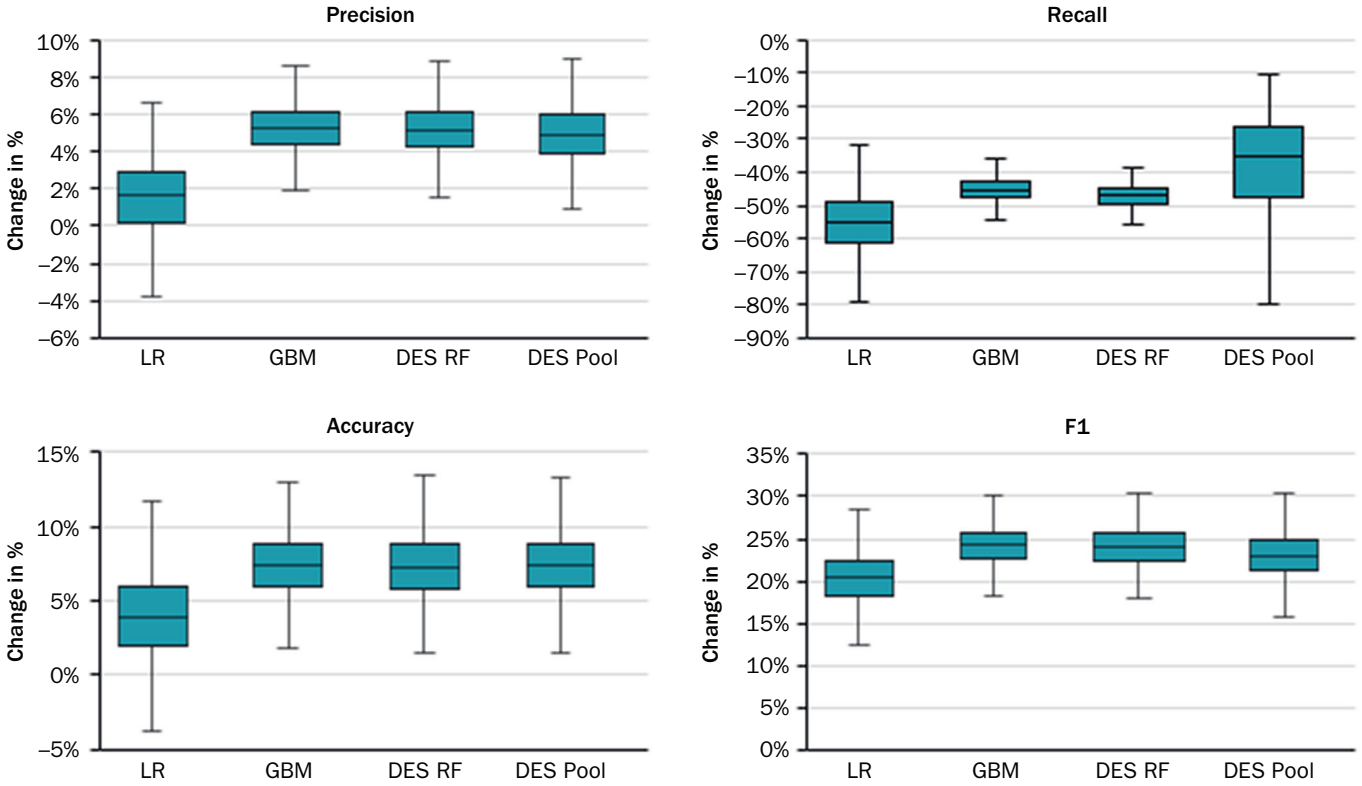
Although the ensembles all performed well, GBM and DES RF appeared to be the most robust models with the least variation in their changes. The large variation in recall displayed by the pool of classifiers (DES Pool) may stem from the diversity of the base classifiers.

Experiment 2: Modeling for False Positives

Experiment 2 aimed to demonstrate how ensembles improve model efficiency by reducing false-positive modeling, thereby increasing the meta-premium.

EXHIBIT 3

Boxplots of the Changes in Classification Metrics for the 1,000 Trials in Experiment 1: Information Advantage



Model evaluation statistics and market state statistics are two useful feature categories for detecting false positives. The former focus on evaluating the recent performance of the primary model; rolling accuracy, F1, recall, precision, and AUC scores are pertinent concepts here. The primary model's confidence level can be measured by adding $\dot{y}$ as a confidence indicator if the primary model is also an ML algorithm (Joubert 2022).

On the other hand, market state statistics demonstrate how differently the primary model is likely to behave under various market regimes. The four distributional moments of mean, variance, skew, and kurtosis, as well as momentum and acceleration, can all be used to represent a market state.

The meta-premiums from the informational advantage and false-positive modelling are shown in Exhibit 4, where they are indicated by *ip* and *fp*, respectively. The fact that the mean of *fp* outperforms *ip* in every situation shows that modeling for false positives enhances the value of the meta-premium. Similar to Experiment 1, we observe that the ensembles outperformed the primary model in terms of Sharpe ratio and mean returns.

It must be noted, however, that the minimum ΔSharpe^{ip} value is negative for the logistic regression, emphasizing that, occasionally, modeling solely for informational advantage results in worse returns than the primary model. The reason for this is that changing regimes make it challenging for logistic regression to find a good fit. The ensemble models adjust better to regime shifts and contribute a positive value change. In addition, we can see that the minimum of ΔSharpe^{fp} is always a positive value, indicating that even losing strategies become profitable once modeling for false positives is included.

EXHIBIT 4

Summary Statistics of the Changes in Strategy Metrics for the 1,000 Trials in Experiment 2: False Positive Modeling

Strat Metric	Model	Mean	Median	Std Deviation	Min	Max
Δ Sharpe Ratio^{ip}	LR	1.2070	1.2088	0.4346	**–0.6490**	2.5463
	GBM	1.6248	1.6156	0.3156	**0.6319**	2.7969
	DES RF	1.6193	1.6168	0.3070	**0.6537**	2.5135
	DES Pool	1.7111	1.7176	0.3622	**0.6416**	2.8084
Δ Sharpe Ratio^{fp}	LR	1.9479	1.9532	0.3913	**0.4164**	3.0138
	GBM	2.5115	2.5088	0.3173	**1.1891**	3.5037
	DES RF	2.4844	2.4845	0.3103	**1.3669**	3.4843
	DES Pool	2.5805	2.5877	0.3359	**1.5707**	3.6292
Δ Mean^{ip}	LR	0.0012	0.0012	0.0004	**–0.0003**	0.0025
	GBM	0.0014	0.0014	0.0003	**0.0003**	0.0026
	DES RF	0.0014	0.0014	0.0003	**0.0005**	0.0026
	DES Pool	0.0014	0.0014	0.0004	**0.0002**	0.0027
Δ Mean^{fp}	LR	0.0016	0.0016	0.0004	**0.0003**	0.0028
	GBM	0.0019	0.0019	0.0003	**0.0008**	0.0031
	DES RF	0.0019	0.0019	0.0003	**0.0008**	0.0032
	DES Pool	0.0020	0.0019	0.0004	**0.0009**	0.0033
Δ Standard Deviation^{ip}	LR	–0.0070	–0.0070	0.0010	**–0.0107**	–0.0041
	GBM	–0.0060	–0.0060	0.0007	**–0.0087**	–0.0039
	DES RF	–0.0062	–0.0061	0.0007	**–0.0089**	–0.0040
	DES Pool	–0.0066	–0.0066	0.0013	**–0.0106**	–0.0031
Δ Standard Deviation^{fp}	LR	–0.0062	–0.0061	0.0009	**–0.0096**	–0.0038
	GBM	–0.0060	–0.0060	0.0007	**–0.0088**	–0.0038
	DES RF	–0.0061	–0.0061	0.0007	**–0.0084**	–0.0042
	DES Pool	–0.0058	–0.0057	0.0008	**–0.0098**	–0.0035

NOTES: The differences in Sharpe ratio, mean returns, and standard deviation are all scaled to an annualized level. ip denotes a metric that is the difference between a secondary model trained only on price data (i.e., the information advantage) and the primary model. fp denotes a metric that is the difference between a secondary model (trained on both returns data [i.e., the information advantage] and features indicative of false positives [i.e., the false positive advantage]) and the primary model. The minimum values are boldface since those are the most conservative measures of model performance.

Exhibit 5 illustrates the changes to the classification metrics when false positive modeling is included, once more highlighting how all metrics improve compared to using solely the information advantage. The boxplots show a similar pattern of behavior to those in Exhibit 3, in which recall is exchanged for higher precision, F1 score, and accuracy (see Exhibit A1 for changes in AUC). Moreover, the tested ensembles consistently outperformed the baseline logistic regression model.

FRAMEWORK

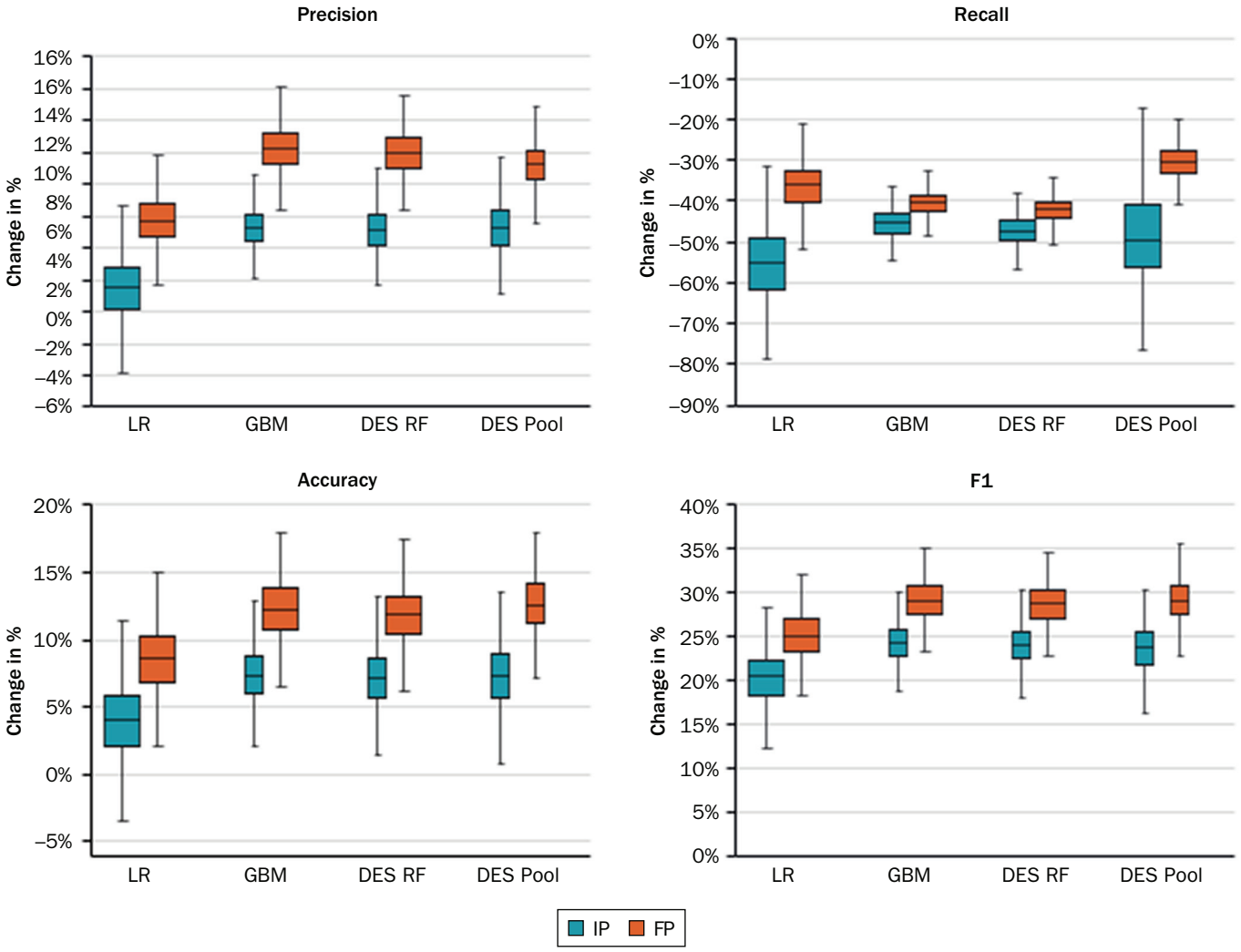
Based on the results of our experiments, we derive the following meta-labeling ensemble model selection framework for a given primary strategy:

1. Choose an ensemble selection technique and method.
2. Select an ensemble creation algorithm.
3. Apply a classifier combination strategy.
4. Define the model selection criteria.
5. Evaluate the ensemble model.

A detailed explanation for each of the five steps is given in the Appendix.

EXHIBIT 5

Boxplots of the Changes in Classification Metrics for the 1,000 Trials in Experiment 2: False Positive Modeling



DISCUSSION

This article has presented a framework to facilitate the selection of models for meta-labeling with ensemble learning. Experiments were conducted on information advantage and modeling for false positives to discover whether ensembles were better at extracting and detecting regimes and whether they increased model efficiency.

The two main reasons to apply ensemble learning are to improve performance and robustness. We established the validity of the trade-off between recall and precision, shown in Joubert (2022), as well as the fact that the use of ensembles in meta-labeling further enhances both classification and strategy metrics.

Among the tested ensembles, the LightGBM and dynamically selected RF delivered the most robust performance. In general, the results showed that ensembles outperform single-base learners when the underlying data consist of many regimes. Thus, the findings support the hypothesis that ensembles should be selected as the models for meta-labeling.

One possible reason for the better performance of ensembles is that different models use the data in different ways, which leads to increased diversity and optimized data exploitation. This benefit is particularly pronounced when the underlying data consists of more regimes and is nonlinear. One explanation for the larger variation in the classification metric scores of the dynamically selected heterogeneous pool of classifiers may be that the individual base learners have a different distribution of outputs and the chosen KNORA-U tries to unite them (see the Appendix for details).

The classifier pools in our dynamic ensembles are built using either well-known ensemble generation techniques, such as bagging or heterogeneous classifiers. The issue with these generation techniques is that they were designed for static combination methods. In other words, they generate the basic classifiers using a global method, and because these strategies examine the issue from a global rather than a local perspective, they do not ensure the participation of local experts (Cruz, Sabourin, and Cavalcanti 2018). As a result, the dynamic selection approaches may be unable to identify competent local classifiers (Souza et al. 2017).

A limitation of this analysis is that it only focused on data that were synthetically generated using a linear autoregressive process with assumptions that are not satisfied in real-world financial data. Moreover, these experiments have not conclusively revealed that meta-labeling model should be selected for a particular given primary model because the choice fundamentally depends on the underlying strategy. Nevertheless, the developed framework serves as a guideline to answer this question on a case-dependent basis.

CONCLUSION

The research objective of this article was to develop a framework for choosing model architectures for meta-labeling and to investigate the incorporation of ensembles. The two questions our experiments aimed to answer were:

1. Does an ensemble increase the information advantage by detecting and extracting regimes more effectively?
2. Does an ensemble improve model efficiency by reducing false positives?

We confirmed our hypothesis that an ensemble framework would outperform a single baseline classifier in meta-labeling, as suggested by Condorcet's jury theorem that a majority decision is more likely to be correct (see Rokach 2010). In Experiment 1, we demonstrated that ensembles increase the information advantage through the improved detection and extraction of regimes. Subsequently, Experiment 2 showed how ensembles improve model efficiency by reducing false positive modelling.

Concerning the ensemble framework, the LightGBM and homogeneous dynamically selected ensembles (with an RF classifier, DES RF) offered the most promising results in our experiments. Due to their great performance and superior robustness, we recommend that practitioners start with these before branching out into heterogeneous pools, depending on the sophistication of their primary model. Although the dynamic ensemble with a heterogeneous pool did prove its potential to further improve the model's performance, it displayed a larger variance in its classification metrics. Nevertheless, because we focused only on synthetic data and a single primary model, the best model for any given underlying strategy is still case dependent.

There are two principal areas where further study is required. The first area deals with model architectures. Model selection remains problem dependent; in our context,

it relies on the underlying primary model for meta-labeling. Because the majority of models concentrate on creating architectures, giving little thought to how to combine them, the base learner's prediction remains unanswered (Ganaie et al. 2021). Therefore, the impact of various fusion strategies, such as specific local information dynamic classifier generation (Cruz, Sabourin, and Cavalcanti 2018), is a potential research topic. Second, in line with Joubert (2022), a future experiment could explore how position-sizing algorithms could be applied to the secondary model's outputs. Each model has a different distribution of probabilities; how these probabilities can be used to maximize returns, reduce drawdowns, and improve the Sharpe ratio (in the context of meta-labeling) is an open research question.

APPENDIX

KNORA

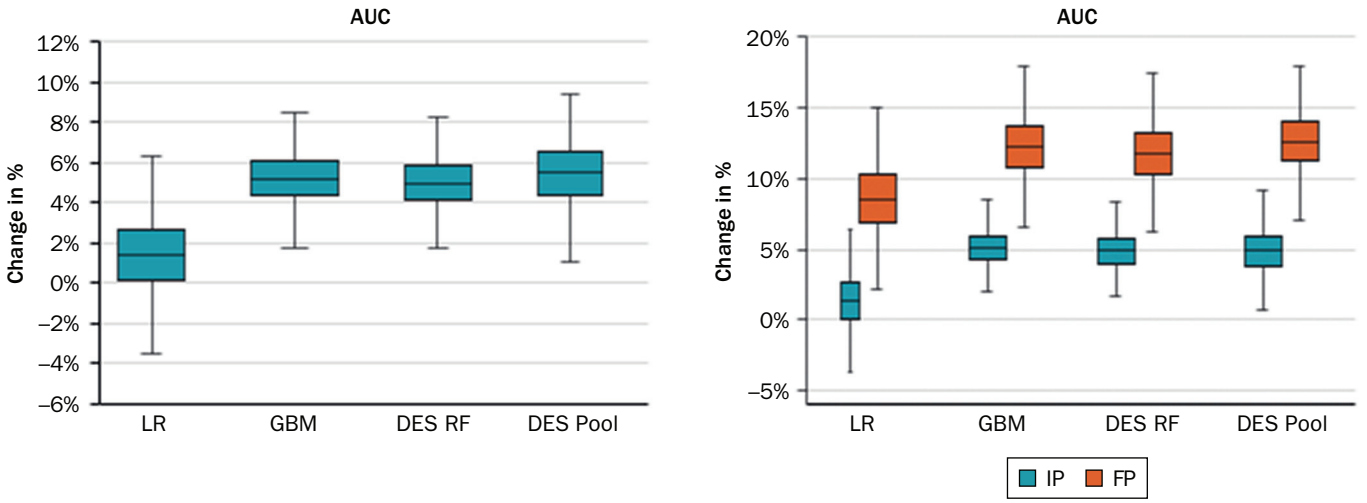
There are two main KNORA schemes (Ko et al. 2008):

- **KNORA-E** (eliminate) chooses all classifiers that produce flawless predictions for the neighborhood of *k* examples. If no classifier achieves an accuracy of 100%, the neighborhood size is decreased by one, and the models are reevaluated. This procedure is repeated until one or more models with perfect performance are identified, after which they are applied to generate a prediction for the new example.
- **KNORA-U** (union) selects all classifiers in the neighborhood that have at least one accurate prediction. The number of accurate predictions in the neighborhood represents the number of votes given to each classifier, and the predictions from each classifier are then combined using a weighted average.

We used the KNORA-U scheme in our experiments.

EXHIBIT A1

Boxplots of the Changes in AUC for the 1,000 Trials in Experiments 1 and 2



Framework

Based on the results of our experiments, we derive the following meta-labeling ensemble model selection framework for a given primary strategy:

1. Choose an ensemble selection technique and method.

   Practitioners must first choose either a static or a dynamic ensemble selection technique. Static ensemble selection involves averaging or weighting all members of the ensemble pool. In dynamic selection, the best combination of classifiers is selected dynamically from each data sample.

   Next, one of four ensemble methods must be chosen. Here, the choice is between sequential and parallel ensembles and homogenous and heterogenous ensembles. Parallel ensembles can be composed of either the same base learners (homogenous) or different base learners (heterogenous). Sequential ensembles are always homogenous. Homogenous ensembles work well with larger datasets, whereas heterogenous ensembles are suitable for small datasets.

   It is important to carefully consider ensemble selection techniques and methods as not all of them are compatible. For instance, sequential methods generate base learners based on previous data dependencies and will not work with current dynamic selection techniques.
2. Select an ensemble creation algorithm.

   Two interrelated questions must be addressed when creating an ensemble system: First, how will the individual classifiers (basic classifiers) be created? Second, how will they vary from one another? Popular algorithms are boosting for sequential ensembles, bagging for homogeneous ensembles, and stacking for heterogenous ensembles.

   Breiman's bagging, also known as bootstrap aggregating, was one of the earliest ensemble-based techniques. Not only is it one of the most intuitive and easiest to use, but it also performs remarkably well (Breiman 1996). Different training data subsets are randomly selected—with replacement—from the entirety of the training data using bootstrapped clones of the training data. Consequently, bagging creates more diversity. Each training data subset is used to train a distinct classifier of the same type. Subsequently, the classifiers are merged depending on the choices with the most votes. For every given instance, the ensemble choice is the class chosen by the majority of classifiers. Bagging is particularly interesting when only a tiny dataset is available. To ensure that there are enough training examples in each subset, relatively large chunks of data (between 75% and 100%) are pulled into each (Polikar 2006).

   In 1990, Schapire demonstrated how to transform a weak learner—an algorithm that creates classifiers that perform only slightly better than random guessing—into a strong learner, an algorithm that creates classifiers that can accurately classify all but a small subset of instances (Schapire 1990). Known as *boosting*, the algorithm is regarded as one of the most significant advances in ML (Polikar 2006). Boosting works by resampling the data and then combining it through majority voting; it thus produces an ensemble of classifiers, just like bagging. The similarities stop there, however. In boosting, resampling is used to provide the most insightful training data for each subsequent classifier. Boosting essentially produces three weak classifiers. A random subset of the training data is used to train the first classifier, C_1. Given C_1, the most informative subset is the training data subset for the second classifier, C_2. This means that only half of the training data used to train C_2 is correctly classified by C_1, and the other half is misclassified. Instances in which C_1 and C_2 disagree are a constraint on the third classifier, C_3. A three-way majority vote is used to combine the three classifiers (Polikar 2006). Schapire proved that the error of this ensemble of three classifiers is limited to less than the error of the best classifier in the ensemble, assuming each classifier has an error rate of less than 0.5. For a two-class

issue, the highest error rate we can expect from a classifier is 0.5 because an error rate of 0.5 would be equivalent to random guessing. Boosting thus merges three weaker classifiers to form one stronger classifier. Boosting techniques may then be used recurrently to construct a powerful classifier in the strict meaning of Probably Approximately Correct (PAC) learning (Polikar 2006).

Stacking is a strategy for achieving the best generalization accuracy (Wolpert 1992). Using a meta-learner, this technique aims to determine which classifiers are trustworthy and which are not. Stacking is often used to merge models generated by different inducers. For each tuple in the original dataset, a tuple is produced as part of the meta-data set. However, the strategy utilizes the classifiers' projected classifications as input characteristics rather than the actual input attributes. The aim characteristic remains unchanged from the original training set. Each of the basic learners is the first to classify an instance, and the classifications from a meta-level training set are then used to train a meta-classifier. This classifier's final prediction incorporates all the preceding ones. The original dataset should be split into two halves. The second subset is used to build the base-level classifiers, whereas the first subset is used to build the meta-dataset. Consequently, the meta-classifier predictions capture the performance of the core learning methods. Using base-level classifier output probabilities for each class label may improve stacking performance. In this case, the number of classes is multiplied by the number of input characteristics in the meta-dataset (Rokach 2010).

3. Apply a classifier combination strategy.

   Once an ensemble has been built, a strategy is employed to combine the classifiers. Combination strategies can be categorized either as trainable and nontrainable rules or rules that apply to class labels versus class-specific continuous outputs (Polikar 2006).

   In trainable combination rules, the combiner's parameters, commonly referred to as *weights*, are chosen using a different training algorithm. An example of this is the expectation-maximization algorithm employed by the mixture-of-experts model. Dynamic combination rules, which are typically instance specific, are another example of combination parameters created by trainable rules. In contrast, nontrainable rules require no additional training beyond that necessary to produce the ensemble. An example of a rule that cannot be trained is weighted majority voting, given that the parameters are immediately available upon the creation of the classifiers (Polikar 2006).

   In the second taxonomy, some combination rules only require the classification choice (i.e., one of ω_j, $j = 1, \ldots, C$), whereas other rules require the continuous-valued outputs of separate classifiers. They can estimate class conditional posterior probabilities $P(\omega_j|x)$ if (i) they are correctly normalized to add up to 1 for all classes and (ii) the classifiers have been trained with sufficiently dense data. These numbers often indicate the degree of support each class receives from the classifiers. Although the adequately dense training data criterion is seldom reached in practice, several classifier models, such as the MLP and radial basis function networks, provide continuous-valued outputs that are often understood as posterior probabilities (Polikar 2006). Examples of classifier combination strategies include algebraic combiners, voting methods and decision templates.

4. Define the model selection criteria.

   Selection criteria are used to estimate the competence of classifiers. For static ensembles, diversity and classification accuracy are the most popular selection criteria (Cruz, Sabourin, and Cavalcanti 2018). First, the use of a variety of inductive biases contributes significantly to the superior performance of ensemble models (Deng et al. 2013). To achieve the desired predictive performance, the participating inducers should be sufficiently diverse (Sagi and Rokach 2018). Second, the individual inducer's predictive performance must be at least as good as a random model (Sagi and Rokach 2018). Applied to meta-labeling, different base learners increase the information advantage and reduce false positives.

In dynamic selection, on the other hand, a single classifier or an ensemble is selected to classify each unknown sample. The purpose of a dynamic selection approach based on a pool of classifiers is to discover a single classifier or an ensemble of classifiers capable of categorizing a given query. The argument in favor of a dynamic selection approach is that each base classifier is an expert in a distinct section of the feature space. The purpose of this method is to choose the best classifiers in the neighborhood of the query (Ko, Sabourin, and Britto 2008).

5. Evaluate the ensemble model.

When selecting a model, there are often several tuning parameters and distinct learning algorithms to choose from. Model selection is the process of picking the optimal algorithm and setting its parameters, and in order to do so, we must estimate the learner's performance. In this context, an empirical approach entails creating experiments and comparing models using statistical hypothesis testing (Zhou 2012).

Because training errors, the mistakes that the learner makes on the training data, favor complex learners over learners who generalize well, it is not a good idea to estimate a learner's generalization error rate by their training error rate. A fully evolved decision tree, for example, may have zero training mistakes but may perform poorly on untried data due to overfitting. The correct technique is to compare learners' performance against a validation set. It should be emphasized that the labels in the training set and validation set must be known before commencing the training process in order to generate and fine-tune the final learner once the model has been selected (Zhou 2012).

In practice, the training and validation sets are usually obtained by dividing a given data set into two halves. To prevent erroneous validation set estimations, the attributes of the original dataset should be retained as much as possible in the splitting process. As an extreme example, the training set could contain only positive cases, and the validation set only negative occurrences; this must be avoided. When the original dataset is randomly split, the class proportion should be maintained in both the training and validation sets—this procedure is known as stratification or stratified sampling (Zhou 2012).

Hence, we compare learners against a validation set (train-test split) and use cross-validation where possible but avoid shuffling time-series data to avoid leakage. In addition to common model evaluation metrics (for classification: accuracy, precision, recall, and F1 weighted score), strategy metrics should also be included (expected returns, standard deviation, and Sharpe ratio).

REFERENCES

Breiman, L. 1996. "Bagging Predictors." *Machine Learning* 24 (2): 123–140.

——. 2001. "Random Forests." *Machine Learning* 45 (1): 5–32.

Cruz, R. M., R. Sabourin, and G. D. Cavalcanti. 2018. "Dynamic Classifier Selection: Recent Advances and Perspectives." *Information Fusion* 41: 195–216.

Deng, H., G. Runger, E. Tuv, and M. Vladimir. 2013. "A Time Series Forest for Classification and Feature Extraction." *Information Sciences* 239: 142–153.

Dietterich, T. G. 2000. "Ensemble Methods in Machine Learning." In *International Workshop on Multiple Classifier Systems*, 1–15. Berlin, Heidelberg: Springer.

Ganaie, M. A., M. Hu, A. K. Malik, M. Tanveer, and P. N. Suganthan. 2021. "Ensemble Deep Learning: A Review." arXiv 2104.02395.

James, G., D. Witten, T. Hastie, and R. Tibshirani. 2013. *An Introduction to Statistical Learning*, Volume 112. New York: Springer.

Joubert, J. F. 2022. "Meta-Labeling: Theory and Framework." *The Journal of Financial Data Science* 4 (3): 31–44.

Ko, A. H., R. Sabourin, and A. S. Britto, Jr. 2008. "From Dynamic Classifier Selection to Dynamic Ensemble Selection." *Pattern Recognition* 41 (5): 1718–1731.

Kohavi, R. 1995. "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection." *Ijcai* 14: 1137–1145.

Kohavi, R., and D. H. Wolpert. 1996. "Bias Plus Variance Decomposition for Zero-One Loss Functions." *ICML* 96: 275–283.

Li, Y., D. Turkington, and A. Yazdani. 2020. "Beyond the Black Box: An Intuitive Approach to Investment Prediction with Machine Learning." *The Journal of Financial Data Science* 2 (1): 61–75.

Lopez de Prado, M. 2018. *Advances in Financial Machine Learning,* 1st ed. Hoboken, NJ: Wiley.

Man, X., and E. Chan. 2021. "The Best Way to Select Features? Comparing MDA, LIME, and SHAP." *The Journal of Financial Data Science* 3 (1): 127–139.

Mehta, M. 2022. "Machine Learning for Efficacy Improvements in Automated Decision-Making in Financial Trading: Using Sigtech Platform." Helsinki, Finland: Arcada University of Applied Sciences.

Meyer, M., J. F. Joubert, and A. Mesias. 2022. "Meta-Labeling Architecture." *The Journal of Financial Data Science* 4 (4): 10–24.

Murphy, K. P. 2012. *Machine Learning: A Probabilistic Perspective*. Cambridge, Massachusetts: MIT Press.

Narro, J., and M. Caamano. 2020. *Systematic Trading in Energy Markets*. London: Risk Books.

Niculescu-Mizil, A., and R. Caruana. 2005. "Predicting Good Probabilities with Supervised Learning." In *Proceedings of the 22nd International Conference on Machine Learning*, 625–632. Bonn, Germany: ACM.

Nousiainen, P. 2021. "Exploration of a Trading Strategy System Based on Meta-Labeling and Hybrid Modeling Using the Sigtech Platform." Arcada University of Applied Sciences, Helsinki, Finland.

Polikar, R. 2006. "Ensemble Based Systems in Decision Making." *IEEE Circuits and Systems Magazine* 6 (3): 21–45.

Rokach, L. 2010. "Ensemble-Based Classifiers." *Artificial Intelligence Review* 33 (1): 1–39.

Sagi, O., and L. Rokach. 2018. "Ensemble Learning: A Survey." *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8 (4): e1249.

Schapire, R. E. 1990. "The Strength of Weak Learnability." *Machine Learning* 5 (2): 197–227.

Sergio, A. T., T. P. de Lima, and T. B. Ludermir. 2016. "Dynamic Selection of Forecast Combiners." *Neurocomputing* 218: 37–50.

Souza, M. A., G. D. Cavalcanti, R. M. Cruz, and R. Sabourin. 2017. "On the Characterization of the Oracle for Dynamic Classifier Selection." In *2017 International Joint Conference on Neural Networks (IJCNN)*, 332–339. Anchorage, AK: IEEE.

Wolpert, D. H. 1992. "Stacked Generalization." *Neural Networks* 5 (2): 241–259.

Zhang, C., and Y. Ma, eds. 2012. *Ensemble Machine Learning: Methods and Applications*. New York, NY: Springer Science & Business Media.

Zhou, Z.-H. 2012. *Ensemble Learning: Foundations and Algorithms.* Boca Raton, FL, USA: CRC Press.