

Title: Meta-Labeling

URL: <https://en.wikipedia.org/wiki/Meta-Labeling>

PageID: 79700295

Categories: Category:Machine learning

Source: Wikipedia (CC BY-SA 4.0).

Meta-labeling , also known as corrective AI , is a machine learning (ML) technique utilized in quantitative finance to enhance the performance of investment and trading strategies, developed in 2017 by Marcos López de Prado at Guggenheim Partners and Cornell University . [1] The core idea is to separate the decision of trade direction (side) from the decision of trade sizing, addressing the inefficiencies of simultaneously learning both side and size predictions. The side decision involves forecasting market movements (long, short, neutral), while the size decision focuses on risk management and profitability. It serves as a secondary decision-making layer that evaluates the signals generated by a primary predictive model. By assessing the confidence and likely profitability of those signals, meta-labeling allows investors and algorithms to dynamically size positions and suppress false positives. [1]

Motivation

Meta-labeling is designed to improve precision without sacrificing recall. As noted by López de Prado, attempting to model both the direction and the magnitude of a trade using a single algorithm can result in poor generalization. By separating these tasks, meta-labeling enables greater flexibility and robustness:

Enhances control over capital allocation.

Reduces overfitting by limiting model complexity.

Allows the use of interpretability tools and tailored thresholds to manage risk.

Enables dynamic trade suppression in unfavorable regimes. [1] [2]

Applications

Meta-labeling has been applied in a variety of financial ML contexts, including:

Algorithmic trading : Filtering and sizing trades to reduce false positives.

Portfolio optimization : Scaling exposure across multiple signals with differing confidence levels.

Risk management : Dynamically disabling strategies in adverse market conditions.

Model validation : Interpreting when and why a model may be underperforming due to regime shifts.

General architecture

Meta-labeling decouples two core components of systematic trading strategies: directional prediction and position sizing. The process involves training a primary model to generate trade signals (e.g., buy, sell, or hold) and then training a secondary model to determine whether each signal is likely to lead to a profitable trade. The second model outputs a probability that is interpreted as the confidence in the forecast, which can be used to adjust the position size or to filter out unreliable trades. [1] [2]

Meta-labeling is typically implemented as a three-stage process: [2] [3]

Primary model (M1): Predicts the direction or label of a financial outcome using features such as market prices, returns, or volatility indicators. A typical output is directional, e.g., $Y \in \{-1, 0, 1\}$, representing short, neutral, or long positions.

Secondary model (M2) : A binary classifier trained to predict whether the primary model's prediction will be profitable. The target variable is a binary meta-label $F \in \{0, 1\}$.

Inputs can include features used in the primary model, performance diagnostics, or market regime data.

Position sizing algorithm (M3) : Translates the output probability of the secondary model into a position size. Higher confidence scores result in larger allocations, while lower confidence leads to reduced or zero exposure.

Stage 1: Forecasting side

Primary model architecture [2]

Figure 1

Figure 1 presents the architecture of a primary model. It focuses on forecasting the side of the trade. Following the example, this model (M1) takes in input data – such as open-high-low-close data and determines the side of the position to take: a negative number is a short position , and positive number is a long position , the range is set between -1 and 1 (the closer it is to -1 or 1 , the stronger the models conviction is). When training the model, the labels are -1 and 1 , based on the direction of forward returns for some predefined investment horizon. The researcher may decide to apply a recall check (τ : "Tau") by setting a minimum threshold that the initial output needs to be to qualify of a short or long position (if the threshold is not met, no side forecast is predicted, leading to closing of any open positions), this leads to the primary model output which is one of three possible side forecasts: -1 , 0 , or 1 .

The primary model also generates evaluation data which can be used by the secondary model, to improve performance of size forecasts. Some examples of evaluation data include rolling accuracy, F1, recall, precision, and AUC scores.

Stage 2: Filtering out false positives

General meta-labeling architecture [2]

Figure 2

Next comes the phase of filtering out false positives , by applying a secondary machine learning model (M2), which is a binary classifier trained to determine if the trade will be profitable or not.

The model takes as input four general groupings of data:

General input data which is predictive of a false positive. For example the last 30 days rolling volatility of the underlying asset.

Evaluation data.

Market state and regime data, one may find that macro economic data or clustering the market into regimes may help as specific trading strategies are known to perform better in particular regimes. Example: momentum based strategies perform best in periods with low volatility and strong directional moves.

Primary models initial input which is a value between -1 and 1 . This highlights the strength of the primary models conviction.

The output of the model is a value between -1 and 1 (if using a Tanh function) which will indicate the strength of the conviction that a short or long position is profitable, or it could simply be between 0 and 1 (using a sigmoid function) if one only wanted to know if it made money or not. This output allows filtering out trades that are likely to lead to losses.

One could stop at this point or use the outputs of the secondary model as inputs to a position sizing algorithm (M3) which could further enhance strategy performance metrics by translating the output probability of the secondary model into a position size. Higher confidence scores result in larger allocations, while lower confidence leads to reduced or zero exposure.

Stage 3: Optimizing position sizes

Position sizing methods (M3)

Various algorithms have been proposed for transforming predicted probabilities into trade sizes: [3]

All-or-nothing : Allocate 100% of capital if the probability exceeds a predefined threshold (e.g., 0.5); otherwise, do not trade.

Model confidence : Use the probability score directly as the fraction of capital allocated.

Linear scaling : Rescale the model's probabilities using min-max normalization based on the training data.

Normal CDF (NCDF) : Use a normal cumulative distribution function applied to a z-statistic derived from the predicted probability. [1]

Empirical CDF (ECDF) : [3] Rank probabilities based on their percentile in the training data to ensure relative allocation.

Sigmoid Optimal Position Sizing (SOPS) : [3] Applies a smooth non-linear sigmoid transformation optimized to maximize risk-adjusted returns (Sharpe ratio).

Model calibration

Each machine learning algorithm used in meta-labeling tends to produce outputs with different characteristic distributions; for example, some are approximately normally distributed, whereas others exhibit a pronounced U-shape, concentrating probabilities near the extremes. [4] Due to these varying distributions, simply summing the outputs of different models can inadvertently lead to uneven weighting of signals, biasing trade decisions.

To address this, model calibration techniques are essential to adjust the predicted probabilities towards frequentist probabilities, ensuring that model outputs reflect true likelihoods more accurately. Two common calibration techniques are:

Platt scaling (Sigmoid scaling) : [5] Suitable for correcting S-shaped calibration plots typically produced by models such as support vector machines (SVMs).

Isotonic regression : [6] Fits a non-decreasing step function to probabilities and is effective particularly with larger datasets, though it can sometimes lead to overfitting.

Transforming predictions to frequentist probabilities is crucial as it provides probabilistic outputs that are directly interpretable as the actual likelihood of an event occurring. Such calibration significantly enhances the effectiveness of fixed position sizing methods, reducing maximum drawdowns and increasing risk-adjusted returns. However, calibration has less impact on position sizing methods that directly estimate parameters from the training data, such as ECDF and SOPS, suggesting that calibration is a critical step mainly for fixed methods that rely heavily on raw model outputs.

Notes

The secondary model does not generate additional trading signals; instead, its function is to filter out weaker signals generated by the primary model. Consequently, the performance of the secondary model depends heavily on the accuracy of the primary model, emphasizing the need for a well-constructed initial model.

This approach leverages the tradeoff between precision and recall to determine optimal position sizing, aligning with the Fundamental Law of Active Management, [7] which highlights the importance of increasing the information ratio through either enhanced signal quality (skill) or signal frequency.

This methodology is also fundamentally distinct from employing ensemble methods or stacking techniques within the primary model, as the secondary model targets meta-labels directly rather than supplementary predictive features.

Meta-labeling architectures

Various model architectures exist, each tailored to different aspects and complexities of trading strategy development. [8]

Discrete long and short

Recognizing that factors driving long and short positions can differ significantly, this architecture splits meta-labeling into two specialized secondary models: one optimized for long positions and another for short positions.

Components

Primary model: Generates directional trade signals.

Two Secondary Models:

Long model: Focuses on features relevant to upward market movements.

Short model: Specializes in features indicative of downward market movements.

Separate feature sets may be employed to reflect distinct informational drivers of market rallies versus sell-offs.

Benefits

Improves model fit by addressing fundamentally different informational structures underlying longs vs. shorts.

Enhances predictive accuracy by using targeted features for each direction.

Sequential meta-labeling (SMLA)

The SMLA introduces multiple layers of secondary models. Each secondary model's inputs include previous secondary models' outputs and evaluation statistics. This iterative process incrementally improves the model's accuracy.

Components

Primary model: Predicts initial trade direction.

Sequential secondary models, where each subsequent model:

Receives features and performance evaluation statistics from the primary model.

Includes output and statistics from preceding secondary models.

May utilize diverse ML algorithms to capture different feature relationships (e.g., logistic regression followed by support vector machines).

Final predictions reflect accumulated insights and error-corrections from preceding models.

Benefits

Progressive improvement and adaptive error correction.

Enhanced robustness and accuracy through layered, heterogeneous modeling.

Conditional meta-labeling (CMLA)

The CMLA partitions data based on specific market states or regimes, applying specialized secondary models tailored to these conditions. It explicitly recognizes that trading strategy performance varies significantly across different market conditions.

Components

Primary model: Provides base directional signals.

Condition-specific Secondary Models:

Activated based on predefined conditions (e.g., volatility regimes, economic environments).

Utilize unique condition-relevant features tailored to each specific market scenario.

Outputs merged into final decision function.

Benefits

Improved model performance in varied market environments by capturing specific regime characteristics.

Enhanced interpretability regarding strategy effectiveness under different conditions.

Ensemble meta-labeling

Ensemble methods combine multiple model predictions to achieve better performance than individual models by balancing bias and variance. Two prominent ensemble architectures are:

1. Bagging meta-labeling

Employs Bootstrap Aggregation (bagging), training multiple secondary models on bootstrapped samples of the data to mitigate variance and overfitting.

Components

Primary model: Generates initial directional signals.

Multiple secondary models:

Each trained independently on bootstrap-sampled subsets of data.

Typically uses simpler models (e.g., linear discriminant analysis, single-layer perceptrons, or decision trees).

Predictions combined via majority voting or weighted aggregation.

Benefits

Significantly reduces overfitting through variance reduction.

Robust against noisy financial data and unstable model training conditions.

2. Boosting meta-labeling

Sequentially trains secondary models where each model aims to correct the mistakes of the preceding model. Particularly effective at addressing bias and under-fitting.

Components

Primary model: Provides the initial trade signals.

Sequentially Trained Secondary Models:

Each model focuses specifically on correcting the previous model's prediction errors.

Models are homogeneous (usually of the same type, e.g., decision trees in gradient boosting).

Final output combines sequential error corrections into a single enhanced prediction.

Benefits

Reduces bias, improving predictive accuracy.

Efficient at capturing complex, non-linear feature interactions missed by simpler architectures.

Inverse meta-labeling

Inverse meta-labeling reverses the standard process by first identifying important features from secondary models to refine and improve the primary model. This iterative improvement cycle helps create more effective primary models before applying meta-labeling.

Components

Primary model: Provides base directional signals.

Initial secondary model:

Evaluates feature importance related to trade profitability (meta-labels).

Generates insights into crucial predictors of profitable trades.

Adjusted primary model:

Uses newly identified critical features from the secondary model.

Re-trained to enhance recall and reduce false positives upfront.

Revised secondary model:

Applied again after primary model refinement to further enhance precision.

Benefits

Enables systematic identification and incorporation of informative features.

Improves trade quality and recall at the primary modeling stage, increasing effectiveness of subsequent meta-labeling.

Performance

Empirical studies using synthetic data and simulated trading environments have demonstrated that meta-labeling improves strategy performance. Specifically, it increases the Sharpe ratio, reduces maximum drawdown, and leads to more stable returns over time. [3] [2]

Open-source code for experiment replication

The following GitHub repositories link to open-source code to replicate the experiments which show how meta-labeling improves the performance statistics of trading strategies.

Theory and Framework: Using synthetic data and building a meta-labeling example.

Model Architecture Diagrams.

Ensemble Techniques and Meta-Labeling.

Position Sizing and Model Calibration

References

Further reading

López de Prado, M. (2020). *Machine Learning for Asset Managers* . Cambridge University Press. ISBN 9781108883658.

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

Meyer, M., Barziy, I., & Joubert, J.F. (2023). "Meta-Labeling: Calibration and Position Sizing". *Journal of Financial Data Science* . 5(2): 23–40.