A Short Report on Developing and Evaluating a Long-only Trading Strategy

Group: Zhiqi Ma, Ruoheng Du, Bingqian Lu, Zirui Zhou

Introduction

This report presents the development and evaluation of a long-only day trading strategy for an ETF. The investment plan is constructed using a machine learning approach, in which the model generates buy signals based on the data and pre-defined take-profit and stop-loss conditions. To design the plan with optimal profitability, we evaluate the effectiveness of various machine learning models in generating buy signals.

- The data used for this project included daily open, close, high, and low prices of NYSEARCA: GDX, an ETF designed to replicate the NYSE Arca Gold Miners Index (GDMNTR).
- Three strategies are explored for generating buy signals to optimize trading decisions.
- Four machine learning models—Logistic Regression, KNN, Random Forest, and XGBoost—are employed and rigorously evaluated using key metrics such as AUC and total P&L.
- An ensemble method is implemented to combine outputs from individual models, leveraging their strength to further improve performance.
- The key challenges of developing the strategy are defining reliable buy signals, optimizing model-specific hyperparameters, and ensuring robust out-of-sample performance to effectively simulate real-world trading scenarios.

Methodology

1. Data Preprocessing

Daily price data for the ETF (open, close, high, low) was collected and analyzed to define the stop-loss and take-profit conditions, which were thresholds used to manage risk and lock in profits. For example, if the ETF's open price is $208.24, a stop-loss condition at 0.5% triggers a sell at $207.19, while a take-profit condition at 0.7% triggers a sell at $209.69. These thresholds were dynamically computed for each trading day based on the open price. Based on the threshold, the corresponding profit or loss (P&L) were calculated to evaluate the trading performance and the effectiveness of the method used for generating buy signals.

2. Buy Signal Construction

Three strategies were implemented to generate buy signals:

- **Take Profit Or Stop Loss Strategy**: A buy signal is generated if either the take-profit or stop-loss condition is met during the trading day. This approach assumes entry into the

market if sufficient price movement (upward or downward) is observed, aiming to capitalize on volatility.
- **Daily P&L-based Strategy**: A buy signal is generated if the computed daily P&L is positive. This approach focuses on trades that have demonstrated profitability based on historical price behavior, prioritizing a profit-first perspective.
- **If Not Stop Loss Strategy**: A buy signal is generated if the stop-loss condition is not triggered. This approach assumes that the absence of significant downside risk implies the potential for favorable trading opportunities.

Each strategy represents a unique perspective on managing risk and opportunity, enabling a comparative evaluation of their impact on trading performance.

3. Machine Learning Models

We used four models to predict buy signals, employing the following configurations and hyperparameter tuning:

- **Logistic Regression**: Logistic Regression is simple and interpretable, making it effective for binary classification tasks. Explored penalties (L2), regularization strengths (C), and class weights (None, balanced).
- **KNN (K-Nearest Neighbors)**: KNN is non-parametric and sensitive to the distribution of data points. Tuned the number of neighbors (k), distance metrics (Euclidean, Manhattan), and weighting schemes (uniform, distance-based).
- **Random Forest**: Random Forest is an ensemble model that reduces overfitting by averaging predictions from multiple decision trees. Optimized the number of trees, maximum depth, and minimum samples for splits to ensure balanced bias-variance tradeoff.
- **XGBoost**: XGBoost is a gradient boosting framework known for its efficiency and ability to handle complex patterns. Tuned parameters such as learning rate, maximum depth, subsample ratio, and regularization terms.

Each model was trained on the training dataset and validated on a separate validation dataset to prevent overfitting. The best-performing parameters for each model were selected based on validation performance metrics.

4. Performance Metrics

To evaluate and compare model performance, the following metrics were computed:

- **AUC (Area Under the Receiver Operating Characteristic Curve)**: Measured the model's ability to distinguish between buy and no-buy signals. Provided a threshold-independent evaluation of model performance, making it robust for imbalanced datasets.
- **In-sample P&L**: Calculated as the cumulative P&L over the training and validation datasets. Used to evaluate the model's ability to capture profitable trades on data it was trained and validated on.

- **Out-of-sample P&L**: Calculated as the cumulative P&L over the test dataset. Serves as a proxy for real-world trading performance, assessing the model's generalization to unseen data.

These metrics provided a balanced perspective, combining classification accuracy with profitability, ensuring that models were both predictive and actionable in trading scenarios.

5. Ensemble Method

To improve overall performance, an ensemble method was implemented, combining predictions from all four models. The ensemble method was structured as follows:

- **Weight Assignment:** Each model's contribution to the ensemble was weighted based on its out-of-sample P&L. Models with higher profitability were assigned greater weights.
- **Weighted Probabilities**: Each model's predicted probabilities for buy signals were combined using the assigned weights.
- **Threshold Application**: A threshold of 0.5 was applied to the weighted probability to classify buy signals. This ensured consistency with individual model evaluations.

The ensemble method leverages the strengths of individual models while mitigating their weaknesses, resulting in higher cumulative P&L compared to any single model.

Results

1. Individual Model Performance

The table below summarizes the in-sample and out-of-sample performance for each model based on the best strategy (**Daily P&L-based Strategy**):

Model	In-sample P&L	Out-of-sample P&L	AUC Validation	AUC Test
Logistic Regression	152.17	180.14	0.9979	0.9999
KNN	139.87	137.79	0.9424	0.9026
Random Forest	140.15	144.87	0.9571	0.9097
XGBoost	138.33	149.28	0.9451	0.8952

2. Ensemble Performance

The ensemble method resulted in an **Out-of-sample P&L of $167.98**, combining the strengths of individual models. Weights assigned to each model based on their out-of-sample P&L were:

- Logistic Regression: 29.43%
- KNN: 22.51%
- Random Forest: 23.67%
- XGBoost: 24.39%

Discussion

1. Buy Signal Strategies

Among the three buy signal strategies, the **Daily P&L-based Strategy** consistently achieved the highest P&L across all models. This suggests that focusing on immediate profitability leads to better trading outcomes, especially in volatile market conditions.

2. Model Performance

Logistic Regression outperformed other models in both in-sample and out-of-sample P&L, likely due to its simplicity and robustness in binary classification tasks. It effectively captured the relationships between features and buy signals, making it a reliable choice for this strategy.

KNN showed moderate performance, with lower AUC and P&L compared to other models. This may be attributed to its sensitivity to noisy data and the lack of feature scaling, which could affect distance calculations.

Random Forest achieved strong performance due to its ability to handle nonlinearities and reduce overfitting through ensemble learning. It slightly underperformed compared to Logistic Regression in terms of out-of-sample P&L, likely due to overfitting on training data.

XGBoost delivered competitive results, particularly in out-of-sample P&L, reflecting its ability to capture complex patterns in the data. However, its lower AUC suggests that it may have struggled with certain decision boundaries, potentially due to imbalanced classes.

3. Ensemble Method

The ensemble method demonstrated its effectiveness by generating a higher total P&L compared to individual models. By combining model outputs, the ensemble leveraged the strengths of each model while mitigating their weaknesses. This approach highlights the value of diversification in predictive modeling.

Conclusion

This project successfully developed a long-only day trading strategy for an ETF using machine learning models. Key findings include:

- Logistic Regression was the best-performing individual model. However, ensemble methods significantly outperformed individual models by leveraging diverse strengths, achieving a higher total P&L.
- The Daily P&L-based Strategy was the most effective buy signal generation method, reflecting the importance of focusing on immediate profitability in trading decisions.
- Optimizing model weights based on out-of-sample performance improved ensemble results.

The trading strategy can be further enhanced by incorporating additional features into the model, such as technical indicators like RSI, which offers additional insights relating to market trends and results in a more robust and reliable way of generating buy signals. Future work can also explore some of the advanced ensemble techniques, such as stacking or blending, or neural networks and deep learning techniques. These advanced techniques may further enhance the trading strategy's predictive capabilities due to their resistance to high volatility and their ability to intake more complex input data.