Data1030 Final Report

Ke Zhang, Data Science Institute

December 10, 2023

Abstract

This is a course project for the course DATA 1030 Hands-on Data Science at Brown University. In this project, we aim to explore machine learning models to forecast the closing price movements of numerous Nasdaq-listed stocks utilizing data from both the order book and the closing auction. GitHub Repository: https://github.com/kzhangaz/DATA1030—Predict-US-stocks-closing-movements

1 Introduction

This machine learning project is focused on a challenging goal: predicting how stock prices will move in the future, which is a regression problem. To do this, we're using historical data, provided by the company Optiver on KaggleOptiver (2023), from the daily ten-minute closing auction, a critical time when market participants come together to make decisions. This period influences the overall mood in the market and is a key moment for sharing important information. Think of the closing auction as a significant window for discovering the true value of a stock, similar to a mirror reflecting the collective perception at the end of each trading day. Understanding this dynamic is crucial for making informed predictions about where stock prices might be headed. Not much previous work was done on this problem.

1.1 Dataset Description

First, we want to explain more about the dataset that we are facing. The dataset revolves around historic data from the daily ten-minute closing auction on the NASDAQ stock exchange, presenting a unique challenge of predicting future price movements for individual stocks relative to a synthetic index composed of NASDAQ-listed stocks. The key features include:

1. imbalance_buy_sell_flag: An indicator reflecting the direction of auction imbalance: 1: Buy-side imbalance, -1: Sell-side imbalance, 0: No imbalance

2. bid/ask_price: Represents the price of the most competitive buy/sell level in the non-auction book.

3. bid/ask_size: Indicates the dollar notional amount on the most competitive buy/sell level in the non-auction book.

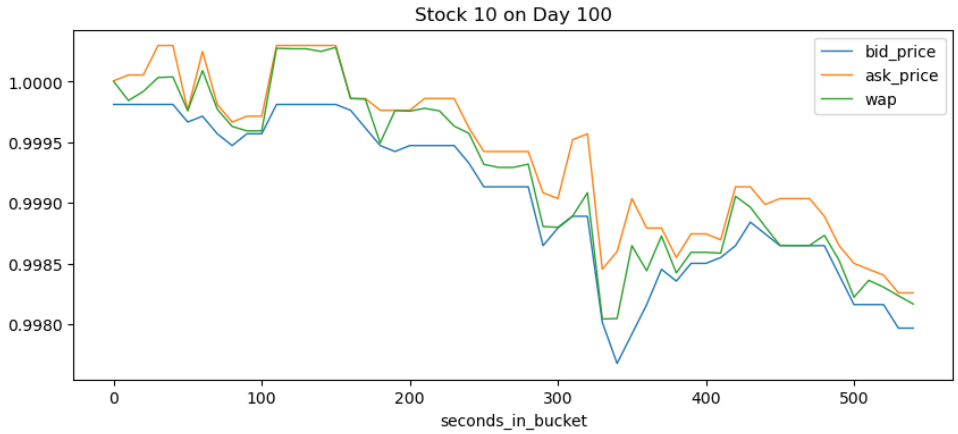

Figure 1: Behavior of stock 10 on day 100

4. wap: The weighted average price in the non-auction book.

5. target: It represents the 60-second future move in the weighted average price (wap) of the stock, less the 60-second future move of the synthetic index. The unit of the target is basis points, equivalent to a 0.01% price move

The dataset encompasses a wealth of information pertaining to 200 distinct stocks, each meticulously documented across more than 20,000 data points. These data points intricately detail the status of each stock on various dates and seconds, specifically capturing its nuances during the daily closing auction period. This characteristic imbues the dataset with a time series nature, underscored by a discernible group structure that encapsulates the dynamic evolution of each stock over time.

2 EDA

To deepen our understanding of the dataset, we conducted an Exploratory Data Analysis (EDA), yielding insightful observations. The following analyses encapsulate key findings from our exploration:

Stock behavior See figure 1. We generated visual representations depicting the bid/ask prices and the weighted average price (wap) of Stock 10 on Day 100 across various time intervals or "seconds in bucket." Our visual inspection reveals a discernible overall decreasing trend in the stock's behavior over this specific time frame.

Upon closer examination, an intriguing observation emerges—the bid/ask prices and wap of Stock 10 exhibit a parallel descending trajectory. This alignment in their directional movements suggests a potential correlation between these key parameters.

Buy/sell flag analysis We proceeded to construct a histogram illustrating the distribution of imbalance buy/sell flags within the dataset. We observed a nearly equal distribution between the imbalanced flags for buy and sell, while the presence of balanced flags was comparatively less prevalent. Further delving into the temporal aspect, we investigated the evolution of these buy/sell flags over seconds in the bucket. See figure

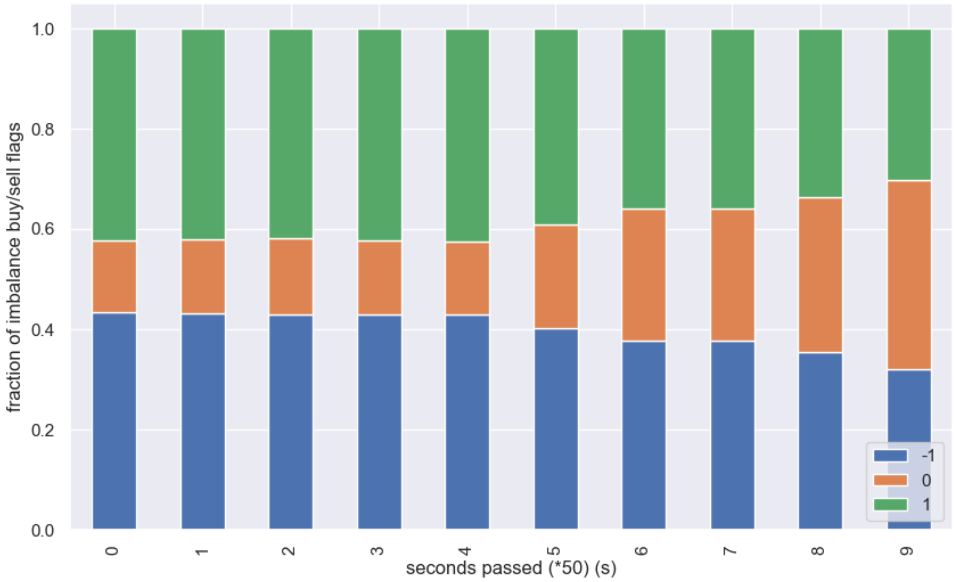

Figure 2: Fraction of imbalance buy/sell flags over seconds passed in the closing period

Figure 3: Bid/ask price and size analysis

2. Notably, we observed a noteworthy surge in the occurrence of balanced flags during the closing period. This temporal pattern suggests an increasing tendency towards equilibrium in the auction dynamics as the closing period unfolds.

Bid/sell price and size correlation analysis In our bid/sell price and size correlation analysis, we constructed correlation plots for bid price and size, ask price and size, as well as bid price and sell price. See Figure 3. A notable observation surfaced during this analysis: the behavior between bid price and bid size mirrors that of ask price and ask size. This synchronous behavior suggests a strong connection between the pricing dynamics and the corresponding size of bids and asks. Intriguingly, we noted that higher bid/ask sizes tend to cluster around the mean bid/ask prices, indicating a concentration of trading interest at these levels. Furthermore, our analysis revealed a robust correlation between bid price and ask price, underscoring the interconnectedness of these critical pricing components. The strength of this correlation emphasizes the influence of bid and ask prices on each other, reflecting the intricate dance between buyers and sellers in the market.

3 Methods

In this section, we discuss our splitting strategy, the data preprocessing, and the machine learning pipeline.

Splitting strategy Given the intrinsic time series nature and the discernible group structure within the dataset, we adopted a deliberate splitting strategy. We opted to partition the data explicitly based on stock IDs, preserving one stock for comprehensive testing while allocating 50 stocks for the combined purposes of training and validation. During the cross-validation phase, 40 stocks were exclusively utilized for training, while the remaining 10 stocks were reserved for validation to rigorously evaluate model performance.

Data preprocessing To deal with missing data, we impute missing data through different strategies. For features with a negligible fraction of missing values, we employed Forward Fill and Backward Fill techniques. In the case of "Far Price" and "Near Price," missing values arose due to Nasdaq recording these variables only from 3:55 pm to 4 pm. To address this, we imputed all values before 3:55 pm with 0 and utilized Forward Fill and Backward Fill for the missing values in the later half of the recording period. Moreover, our data preprocessing incorporated extensive feature engineering. This involved creating lagged variables and deriving features through rolling windows. These engineered features contribute to a more comprehensive representation of temporal patterns and dynamics within the dataset.

Machine Learning Pipeline Our forecasting machine learning pipeline encompasses four distinct models: Linear Regression, XGBoost, Random Forest, and K Nearest Neighbors. Following meticulous data splitting and preprocessing, including addressing missing values and feature engineering, we fine-tuned model configurations via hyperparameter tuning using techniques like grid search. A customized cross-validation implementation ensures robust evaluation, employing the Mean Absolute Error metric for model assessment. This comprehensive approach aims to unveil each model's strengths and limitations while optimizing their performance in capturing the intricate dynamics of the dataset. By systematically exploring various machine learning paradigms and employing rigorous evaluation techniques, our pipeline seeks to deliver accurate and reliable predictions for the forecasting challenge at hand.

Other details We chose the metric Mean Absolute Error(MAE) for model assessment since it provides a clear and balanced assessment of the predictive accuracy of models, offering a comprehensive view of their performance without being unduly influenced by specific characteristics of the data or outliers.
The parameters tuned for each model are as follows:

1. Linear Regression: the regularization term (alpha) for Lasso regression. We tried 0.1, 0.01, 0.001, among which 0.01 has the best performance.

2. XGBoost: Maximum depth of a tree (max depth): 3 (best), 5, and 7. Subsample: 0.8 and 1 (best).

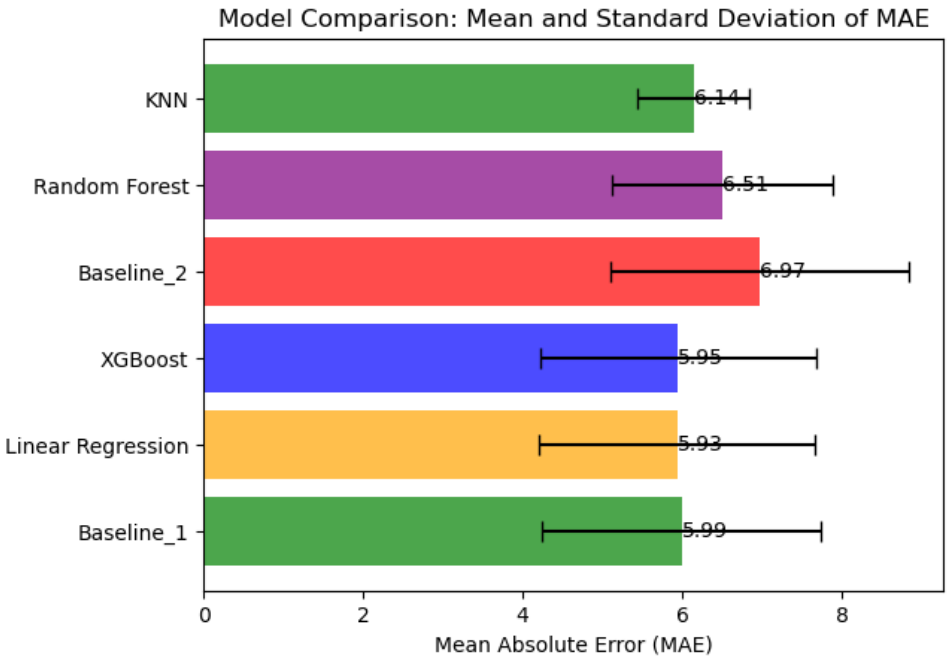

Figure 4: MAE Testscores

3. Random Forest: Maximum depth of the tree (max depth): 3 (best), 5, 10.

4. K Nearest Neighbors: Number of neighbors (n neighbors): 3, 5 (best), 10.

We also iterate through different random states and splitting methods to calculate the standard variance of the test scores to measure the uncertainties due to splitting and due to non-deterministic ML methods (e.g., random forest).

4 Results

test scores Since we are predicting the 60-second future move in the wap of the stock. The baseline model is to predict that the weighted average price remains unchanged, i.e. the target is 0. We have two baseline scores here because we have two sets of train/test data set. From the figure 4, we could see that although the models have improved performance compared to the baseline model, the improvement is not very significant. Further analysis and refinement may be necessary to unlock additional predictive power and enhance model performance in capturing the subtleties of this financial prediction challenge. Among these models, the linear regression model seems to be the most predictive.

feature importance Initially, we generated and scrutinized the permutation importance plot for the Linear Regression model, depicted in Figure 5. This analysis revealed that the most crucial features were engineered variables related to bid size and ask size. The permutation importance method underscores the impact of each feature by assessing the model's performance when the feature values are randomly permuted.

Additionally, we delved into the coefficients of the Linear Regression model, showcased in Figure 6. Notably, bid/ask prices emerged as pivotal determinants of the model's

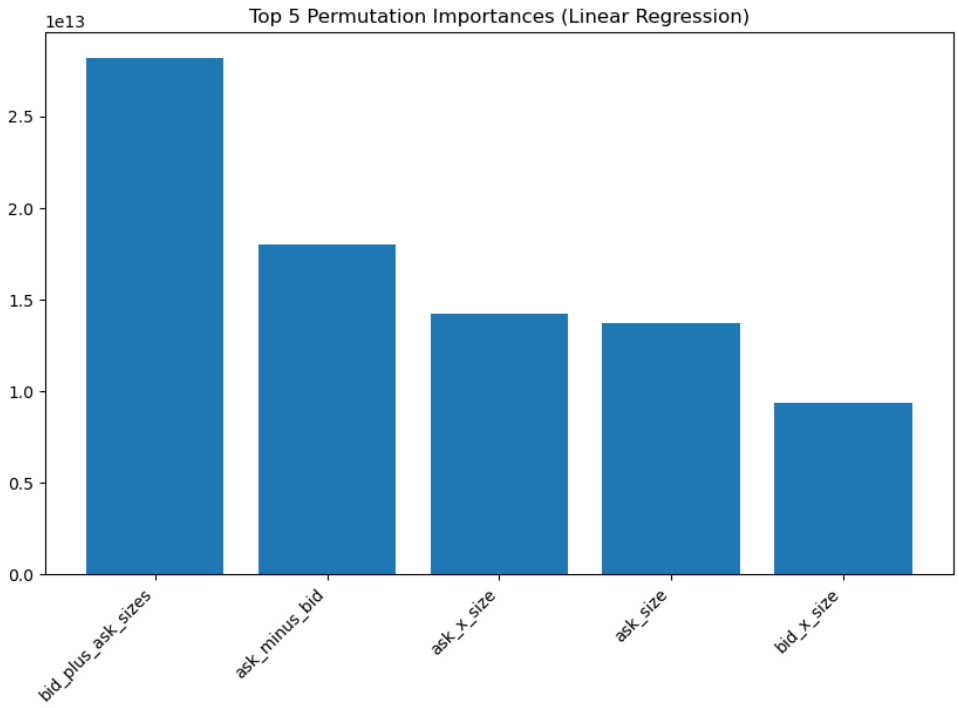

Figure 5: Permutation importance

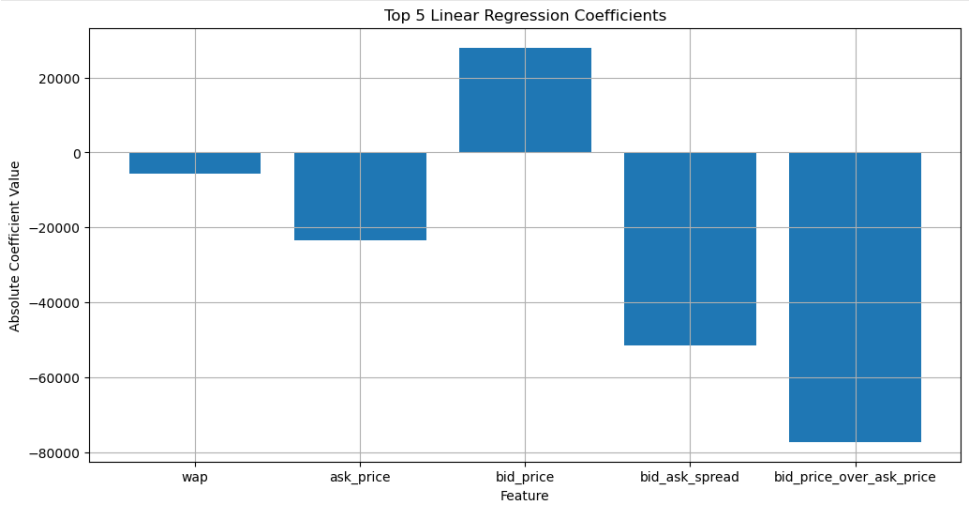

Figure 6: Permutation importance

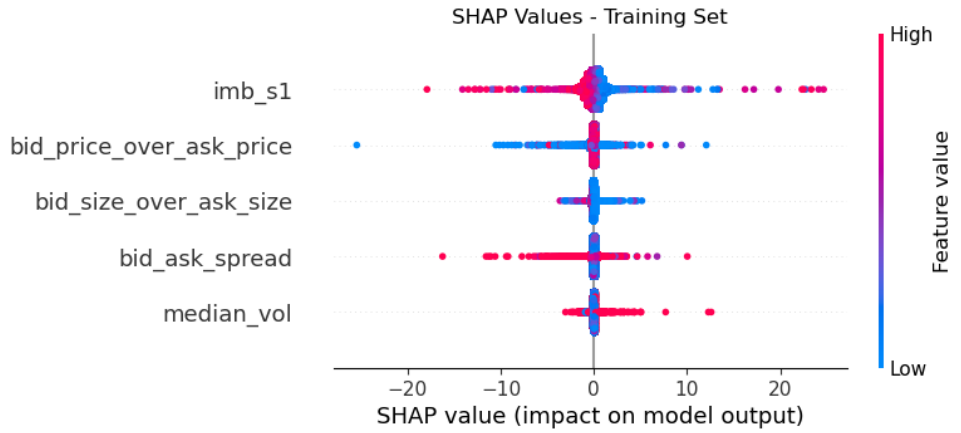

Figure 7: SHAP importance

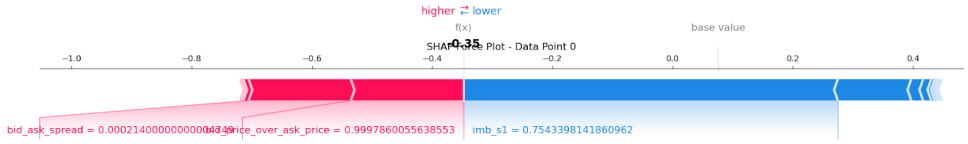

Figure 8: local SHAP importance

predictive performance. The coefficient analysis provides insights into the directional impact of each feature on the target variable, emphasizing the importance of bid/ask pricing dynamics.

Furthermore, our exploration extended to the Shapley Values importance plot, as illustrated in Figure 7. Here, bid/ask size assumed a prominent role in shaping model predictions. Shapley Values provide a game-theoretic approach to attribute the contribution of each feature to the model's output, offering a nuanced perspective on feature importance.

Collectively, these analyses present a comprehensive view of feature importance. The permutation importance highlights the engineered features related to bid/ask size, while the coefficients emphasize the significance of bid/ask prices. Shapley Values provide additional depth, revealing the intricate role played by bid/ask size in shaping model predictions. This multifaceted approach enhances our understanding of the diverse factors influencing the predictive power of the model.

local feature importance See figure 8. For this data point, we can see that imbalance size has a negative impact on the prediction result while bid size over ask size has a positive impact.

Overall, bid size and ask size emerge as the most important features, with a noteworthy observation that weighted average price (wap) exhibits relatively lower importance. Surprisingly, regardless of the model employed, the dataset demonstrates a limited predictive power, shedding light on the challenges inherent in forecasting within this financial context.

5 Outlook

Looking ahead, further enhancements in both predictive power and interpretability could be pursued. Feature engineering remains a potent avenue, where the creation of more sophisticated features derived from a deeper understanding of market dynamics may unveil hidden patterns. Hyperparameter tuning offers an opportunity to fine-tune model configurations for optimal performance. Exploring more comprehensive models, potentially beyond those initially considered, could capture complex relationships within the data. To improve interpretability, employing model-agnostic techniques, such as SHAP values or LIME, can provide deeper insights into individual predictions. Engaging in collaborative discussions within the Kaggle community fosters knowledge exchange and diverse perspectives. Weak spots in the current approach may lie in the complexity of market behaviors and the challenge of predicting short-term movements. Improving this model could involve incorporating advanced techniques such as ensemble methods, neural networks, or time series modeling. Additionally, collecting supplementary data, such as market news sentiment or macroeconomic indicators, may enrich the model's understanding of contextual factors impacting stock movements. Regular model reassessment and iteration, coupled with a dynamic approach to feature engineering and advanced modeling techniques, would contribute to an evolving and more effective forecasting framework.

References

Optiver (2023). Optiver trading at the close dataset. `https://www.kaggle.com/competitions/optiver-trading-at-the-close/data`. Accessed on: 2023.12.09.