Stock Prices Prediction Using Machine Learning

1st Omar Al Anbasah
Department of Computer Science
Jordan University of Science and Technology
Irbid, Jordan
alanbasah.omar@gmail.

2nd Musa Ibrahim alflat
Department of Physics
University of Jordan
Amman, Jordan
musaalflat@gmail.com

3rd Roaa mamoun sartawi
Department of Artificial Intelligence and Data Science
Zarqa university
Amman, Jordan
roaa.sartawi99@gmail.com

***Abstract*—This research addresses the Kaggle competition "Optiver - Trading at the Close," challenging participants to predict closing price movements for Nasdaq-listed stocks. Leveraging order book and closing auction data, our study explores methodologies to adjust prices, assess supply and demand dynamics, and identify trading opportunities. The findings contribute to advancing financial forecasting models, offering insights into the intricate relationship between market data and trading strategies. This research has broader implications for refining algorithmic trading approaches and understanding the dynamic nature of financial markets.**

***Index Terms*—Machine Learning, Algorithmic Trading Strategies, Financial Markets, Stocks.**

I. Introduction

First, let's talk about stocks. What are stocks, types of stocks, are they always safe to buy, and how their prices are determined. Then, a complete description of the competition will be explained.

A. What are stocks?

A stock, also known as equity, is a security that represents the ownership of a fraction of the issuing corporation. Units of stock are called "shares" which entitles the owner to a proportion of the corporation's assets and profits equal to how much stock they own.

Stocks are bought and sold predominantly on stock exchanges and are the foundation of many individual investors' portfolios. Stock trades have to conform to government regulations meant to protect investors from fraudulent practices. [1]

B. What types of stocks are there?

There are two main types of stock: common shares and preferred shares. Equities are synonymous with common shares because their market value and trading volumes are many times larger than those of preferred shares. Common shares usually carry voting rights that enable the common shareholder to have a voice in corporate meetings and elections, while preferred shares generally do not have voting rights. Preferred shareholders have priority over common shareholders to receive dividends as well as assets in the event of a liquidation. [2]

Common stock can be further classified in terms of voting rights. Some companies have dual or multiple classes of stock with different voting rights attached to each class. In such a dual-class structure, Class A shares may have 10 votes per share, while Class B shares may only have one vote per share. Dual- or multiple-class share structures are designed to enable the founders of a company to control its fortunes, strategic direction, and ability to innovate. [3]

C. Are they always safe to buy?

Although we all might love the idea of investing in risk-free stocks, there's no such thing as a stock that's always safe. Even the best companies can face unexpected trouble, and it's common for even the most stable corporations to experience significant stock price volatility. We saw this during the early days of the COVID-19 pandemic, when many strong companies experienced dramatic drops in stock prices. We have also seen it last into 2023 with rising interest rates, inflation, and continued international conflict. [4]

D. How their prices are determined?

The prices of shares on a stock market can be set in several ways. The most common way is through an auction process where buyers and sellers place bids and offer to buy or sell. A bid is a price at which somebody wishes to buy, and an offer, or ask, is the price at which somebody wishes to sell. When the bid and ask coincide, a trade is made. [5]

E. Competition Description.

In the dynamic realm of stock exchanges, particularly on the Nasdaq, the closing moments of each trading day are crucial. Optiver, a prominent global electronic market maker, specializes in navigating this fast-paced environment, employing technological innovation to trade various financial instruments. The Nasdaq Closing Cross auction in the final ten minutes

determines official closing prices, influencing investors and analysts in assessing market performance. This competition challenges participants to develop predictive models leveraging both order book and closing auction data. The objective is to forecast closing price movements for numerous Nasdaq-listed stocks, aiding market makers like Optiver in adjusting prices, understanding supply and demand dynamics, and identifying trading opportunities. The competition not only addresses real-world data science challenges but also contributes to enhancing market efficiency during the critical closing moments of trading.

II. Background

Stocks hold immense significance for companies listed on exchanges like Nasdaq, serving as a vital financial instrument that facilitates capital formation and business growth. The stock market allows companies to raise funds by issuing shares to the public, providing investors with an opportunity to become partial owners and share in the company's success. For companies listed on Nasdaq, a major global exchange, being publicly traded brings increased visibility, access to a broad investor base, and the ability to attract more capital for expansion, research and development, and other strategic initiatives.

Machine learning plays a pivotal role in the realm of stock prediction, offering sophisticated tools to analyze vast amounts of financial data and derive meaningful insights. The unpredictable nature of stock prices, influenced by various market factors and sentiments, makes traditional methods less effective in capturing intricate patterns. Machine learning algorithms, including regression models like CatBoost, can process and learn from historical stock data, identifying complex relationships and patterns that might elude human analysis.

By leveraging machine learning, financial professionals, including market makers and investors on platforms like Nasdaq, can gain a competitive edge in predicting stock movements. These models can analyze historical stock prices, trading volumes, macroeconomic indicators, and other relevant data to generate predictions about future price movements. In the context of the Kaggle competition "Optiver - Trading at the Close," utilizing machine learning techniques allows participants to harness the power of predictive modeling to forecast closing price movements, contributing to improved decision-making in the intense final minutes of trading.

The marriage of machine learning and stock prediction is transformative, offering a data-driven approach to navigate the complexities of financial markets and providing companies, investors, and market participants with valuable insights for informed decision-making. As technology continues to advance, the integration of machine learning in the financial sector, especially in stock prediction, becomes increasingly indispensable for staying competitive and adapting to the fast-paced nature of global financial markets.

In the dynamic realm of financial markets, exemplified by the Kaggle competition "Optiver - Trading at the Close," participants are thrust into the challenging domain of predicting closing price movements for Nasdaq-listed stocks. As the Nasdaq Stock Exchange, a global financial linchpin, hinges its daily closure on the Nasdaq Closing Cross auction in the final ten minutes of trading, the stakes are high for accurate forecasts.

This competition showcases a fusion of cutting-edge technology and machine learning, with participants harnessing advanced algorithms like the CatBoost regression model. This technological synergy mirrors the real-world demands confronted by market makers, such as Optiver, who seamlessly integrate order book and auction data in the critical closing minutes of trading.

As participants grapple with predicting closing price movements, the evaluation framework becomes pivotal. The Mean Absolute Error (MAE), the heart of the competition's assessment, quantifies the average absolute difference between predicted (y_i) and observed (x_i) values, symbolizing the competition's commitment to minimizing discrepancies:

$$MAE = \frac{1}{n}\sum_{i=1}^{N}|y_i - x_i| \tag{1}$$

This metric resonates with the competition's overarching goal of precise predictions, aligning with the intensity of the closing moments of trading on Nasdaq.

Ensuring the competition's integrity is the standardized submission process facilitated by a Python time-series API. This tool not only streamlines submissions but critically prevents models from gaining unfair advantages by "peeking" into future data. Adhering to the provided template in the official notebook, participants contribute to a fair and transparent evaluation, emphasizing the importance of handling time-series data akin to real-world trading scenarios.

This research paper navigates the intricacies of the Kaggle competition, unraveling the symbiotic relationship between machine learning, stock prediction, and the practical demands faced by market participants on platforms like Nasdaq. Through this exploration, the aim is to contribute not only to the competition but also to the broader understanding of leveraging technology and predictive modeling in the ever-evolving landscape of financial markets.

III. Data Description

The 'Trading at the Close' data consists of three distinct phases: the order book, the auction order book, and the combined book. In the order book phase, real-time matching occurs immediately between bid and ask prices. In the auction order book, matching is deferred and collected until the auction concludes. The closing auction price, referred to as the 'far price,' is provided five minutes before the closing cross.

In the combined book phase, the data from both books is amalgamated, resulting in increased accuracy. In this phase, the price is denoted as the 'near price,' and it is provided five minutes before the closing. Additionally, the company furnishes a fair price known as the 'reference price,' calculated as the near price bounded between the best bid and best ask.

The challenge at hand involves predicting the best price during the final 10 minutes of trading.

In the context of financial markets, marked by rapid shifts and consequential decisions, the dataset titled "Optiver - Trading at the Close" emerges as a valuable repository, offering comprehensive insights into the intricate proceedings of the NASDAQ stock exchange's daily ten-minute closing auction. The data have 5237980 rows and 16 features and the target, each serving as a distinct lens through which we can gain a deeper understanding of stock trading dynamics, order book intricacies, and temporal considerations. As we embark on a systematic exploration of these columns, we aim to decode the language of the financial markets, with each column presenting a unique facet that contributes to our comprehension of market dynamics. From identifying individual stocks to navigating temporal dimensions and interpreting pricing metrics, these columns form the fundamental components for constructing predictive models. This paper endeavors to meticulously unpack the significance embedded in each column, presenting an in-depth analysis where the precision of data science converges with the pulse of the market. The data features are as follows:

$Stock - id$: A distinctive identifier assigned to each stock. It's important to note that not all stock IDs are present in every time bucket.

$Date - id$: An exclusive identifier for the date, with sequential and consistent numbering maintained across all stocks.

$Imbalance - size$: Represents the amount unmatched at the current reference price, measured in USD.

$Imbalance - buy - sell - flag$: An indicator reflecting the direction of auction imbalance, with values indicating buy-side imbalance (1), sell-side imbalance (-1), or no imbalance (0).

$Reference - price$: The price at which paired shares are maximized, with the sequential minimization of imbalance and distance from the bid-ask midpoint. Alternatively, it can be considered equal to the near price bounded between the best bid and ask prices.

$Matched - size$: Denotes the maximum amount that can be matched at the current reference price, expressed in USD.

$Far - price$: The crossing price maximizing the number of shares matched based solely on auction interest. This calculation excludes continuous market orders.

$Near - price$: The crossing price optimizing the number of shares matched based on both auction and continuous market orders.

$Bid - Ask - price$: The price of the most competitive buy/sell level in the non-auction book.

$Bid - Ask - size$: The dollar notional amount on the most competitive buy/sell level in the non-auction book.

$WAP(WeightedAveragePrice)$: The weighted average price in the non-auction book.

$Seconds - in - bucket$: The duration in seconds since the initiation of the day's closing auction, always starting from 0.

$Target$: The anticipated 60-second future move in the WAP of the stock, subtracted by the 60-second future move of the synthetic index. This information is exclusively provided for the training set.

IV. DATA STRUCTURE

After explaining the data and going through its columns and values, we did some visualization and feature engineering so that we could understand the nature of the data and make it ready for the training and testing phases. These two parts will be explained as follows.

A. Data Visualisation

We applied the most important visualization techniques to understand the data. First, we applied the correlation visualization technique to understand how the features are correlated. The following figure presents the correlation.

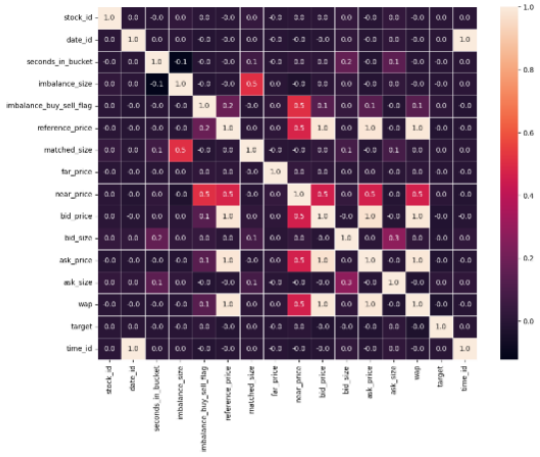

Fig. 1. Features Correlation

As depicted in the figure, a substantial correlation is evident among various pairs of variables. Notably, there exists a significant correlation between imbalance size and matched size, imbalance buy sell flag and near price, reference price and near price, imbalance size and matched size, near price and wap, near price and ask price, near price and bid price, near price and reference price, near price and imbalance buy sell flag, bid price and near price, ask price and near price, as well as wap and near price.

Secondly, we used the subplot technique to understand the fluctuation rate between the reference price and the second in bucket features.

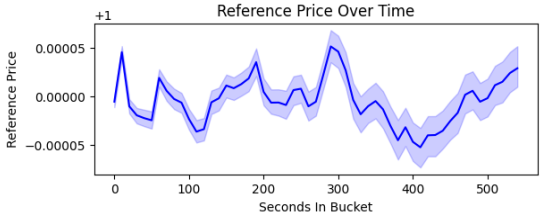

Fig. 2. Reference Price Over Time

It shows a volatile stock with small fluctuations between slightly positive and slightly negative values within a short timeframe (500 seconds). This could be normal behavior

for a high-frequency trading strategy or indicate a period of uncertainty for the stock.

Moreover, we used the subplot technique to observe the fluctuation rate between the imbalance size and the second in bucket.

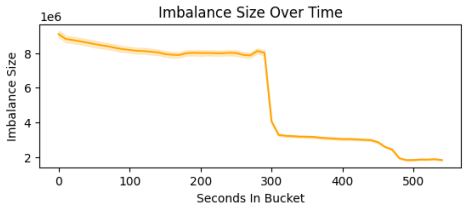

Fig. 3. Imbalance Size Over Time

The graph shows that the size of the imbalance appears to be directly related to the time it takes to process each bucket. The longer it takes to process a bucket, the larger the imbalance grows.

Finally, the legend visualization technique was used to compare the reference price to the bid and ask price.

Fig. 4. Reference price vs ask/bid price

The graph suggests that the price has been relatively stable over the time shown, with some small fluctuations.

B. Feature Engineering

The data needed some feature engineering to be ready for the training and testing phases. Features like reference price, far price, bid price, and ask price were not scaled properly because their standard deviation was bigger than their mean. The standardization technique was used to fix this problem. The following code was used.

```
prices = ["reference_price",'far_price','bid_price','ask_price']
for i in range(len(prices)) :
    mean = df1[prices[i]].mean()
    std  = df1[prices[i]].std()
    df1[prices[i]] = df1[prices[i]].apply(lambda x:(x - mean )/std )
```

Fig. 5. Standardization

Other features like imbalance size, matched size, bid size and ask size had a large value range. The log transformation technique was used to scale these features. The following code was used to scale the data.

```
sizes = ['imbalance_size','matched_size','bid_size','ask_size']
for i in range(len(sizes)) :
    df1[sizes[i]] = df1[sizes[i]].apply(lambda x: np.log(x+1))
```

Fig. 6. Scaling

The most important technique that was used to engineer the data was the Principle Component Analysis. This technique was used to capture information about our data and save it in two components then use it as columns [0, 1]. This technique is applied to reduce the dimensions of large data sets by transforming a large set of variables into smaller ones that still contain most of the information in the large set. The following code was used to apply this technique.

```
# Create an instance of PCA with 2 components
pca = PCA(n_components=2)

pca_df = pca.fit_transform(df1.drop(columns = 'target' ).fillna(0))
pca_df = pd.DataFrame(pca_df)
```

Fig. 7. PCA

The previous techniques were applied to the training set. The same techniques were applied to the testing set to make them ready to choose the perfect model.

V. Models

The selection and implementation of machine learning models are not arbitrary choices but deliberate steps informed by the complexities of the problem at hand. In this section, we navigate through the intricacies of the models used in our research. Here are the models that are used in the research with their results.

A. Linear Regression

```
[ ]  linear_reg = LinearRegression()
     linear_reg.fit(X_train,y_train)
     y_pred_lr = linear_reg.predict(X_test)

[ ]  mae_lr = mean_absolute_error(y_test, y_pred_lr)
     mae_.append(mae_lr)

     mae_lr

     6.1251235429253015
```

Fig. 8. Linear Regression

B. Ridge regression

```
[ ]  reg = Ridge()
     reg.fit(X_train,y_train)
     y_pred_reg=reg.predict(X_test)

     /usr/local/lib/python3.10/dist-packages/sklearn/li
       return linalg.solve(A, Xy, assume_a="pos", overw

[ ]  mae_reg = mean_absolute_error(y_test, y_pred_reg)
     mae_.append(mae_reg)

     mae_reg

     6.128274134138237
```

Fig. 9. Ridge regression

C. Lasso Regression

```
[ ]  clf_la = Lasso()
     clf_la.fit(X_train , y_train)
     y_pred_la=reg.predict(X_test)

[ ]  mae_la = mean_absolute_error(y_test, y_pred_la)
     mae_.append(mae_la)

[ ]  mae_la

     6.128274134138237
```

Fig. 10. Lasso Regression

D. Tweedie regression

```
[ ]  tweed = TweedieRegressor()
     tweed.fit(X_train,y_train)
     y_pred_tweed=tweed.predict(X_test)

[ ]  mae_tweed = mean_absolute_error(y_test, y_pred_tweed)
     mae_.append(mae_tweed)

[ ]  mae_tweed

     6.1624167711263595
```

Fig. 11. Tweedie regression

E. Decision Trees

```
[ ]  reg_dt = DecisionTreeRegressor()
     reg_dt.fit(X_train,y_train)
     y_pred_dt = reg_dt.predict(X_test)

[ ]  mae_dt = mean_absolute_error(y_test, y_pred_dt)
     mae_.append(mae_dt)

[ ]  mae_dt

     10.778842887575227
```

Fig. 12. Decision Trees

F. MLPRegressor

```
     mlpreg = MLPRegressor()
     mlpreg.fit(X_train,y_train)
     y_pred_mlp = mlpreg.predict(X_test)

[ ]  mae_mlp = mean_absolute_error(y_test, y_pred_mlp)
     mae_.append(mae_mlp)

[ ]  mae_mlp

     17.234806407867953
```

Fig. 13. MLPRegressor

G. XGBRegressor

```
[ ]  xgbreg=XGBRegressor()
     XGB=xgbreg.fit(X_train,y_train)
     y_pred_xgb=XGB.predict(X_test)

[ ]  mae_xgb = mean_absolute_error(y_test, y_pred_xgb)
     mae_.append(mae_xgb)

[ ]  mae_xgb

     6.185201360374766
```

Fig. 14. XGBRegressor

H. CatBoost Regressor

```
     catboost_regressor = CatBoostRegressor(iterations=100, depth=6, learning_rate=0.1, loss_function='MAE')
     catboost_regressor.fit(X_train, y_train)
     y_pred_catboost = catboost_regressor.predict(X_test)

[ ]  mae_catboost = mean_absolute_error(y_test, y_pred_catboost)
     mae_.append(mae_catboost)

     mae_catboost

     6.09733573078957
```

Fig. 15. CatBoost Regressor

I. Grid Search for catboost regressor model

```
from sklearn.model_selection import GridSearchCV

catboost_regressor = CatBoostRegressor()

# Define the parameter grid for grid search
param_grid = {
    'iterations': [100, 200],
    'depth': [4, 6, 8],
    'learning_rate': [0.05, 0.1, 0.2]
}

# Perform grid search using GridSearchCV
grid_search = GridSearchCV(catboost_regressor, param_grid, cv=3, scoring='neg_mean_absolute_error', verbose=2)
grid_search.fit(X_train, y_train)

# Get the best parameters from the grid search
best_params = grid_search.best_params_
print(f'Best Parameters: {best_params}')

# Make predictions on the test set using the best model
best_catboost_model = grid_search.best_estimator_
catboost_predictions = best_catboost_model.predict(X_test)
```

```
[ ] grid_search.best_params_

    {'depth': 8, 'iterations': 100, 'learning_rate': 0.05}
```

```
mae_catboost_tuned = mean_absolute_error(y_test, catboost_predictions)
mae_catboost_tuned

6.101478588944522
```

Fig. 16. Grid Search for catboost regressor

J. Model Performance

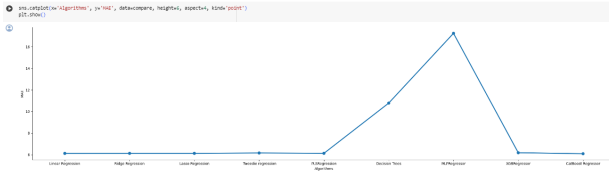

Fig. 17. Model Perfromnace

From the model performance graph, it is clear that the decision trees and MLP regressor didn't perform well. However, the catboost regressor model performed very well compared to the other models.

Based on the performance of the catboost regressor model, we were able to measure the feature importance. The following graph depicts the feature's importance.

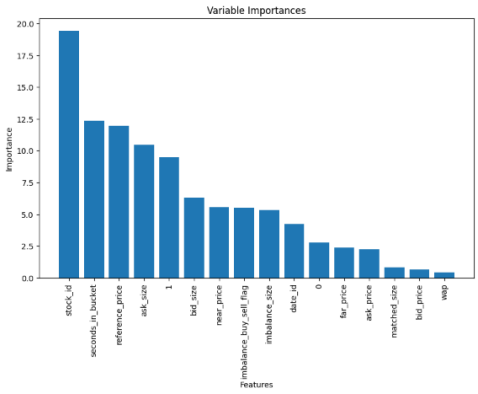

Fig. 18. Feature Importance

VI. Discussion Part

A. Discussion

This research explored the potential of machine learning models in predicting closing price movements for Nasdaq-listed stocks within the framework of the "Optiver - Trading at the Close" Kaggle competition. Utilizing order book and closing auction data, various models were employed to assess their efficacy in forecasting price changes during the critical final ten minutes of trading. By delving into the intricate world of financial data and algorithmic prediction, this study aimed to contribute not only to the competition itself but also to the broader understanding of leveraging technology and modeling in the ever-evolving landscape of financial markets.

B. Key Findings and Their Significance

Our analysis revealed that the CatBoost regression model emerged as the clear frontrunner in terms of predictive accuracy, significantly outperforming other models such as Decision Trees and MLPRegressor. This outcome underscores the effectiveness of gradient boosting techniques in capturing complex relationships within financial data, particularly the non-linear dynamics observed during closing auction periods. As depicted in the model performance graph (Fig. 17), the CatBoost regressor achieved a substantially lower Mean Absolute Error (MAE), demonstrating its superior ability to minimize discrepancies between predicted and actual closing prices.

This finding aligns with the competition's core objective of maximizing precision in closing price forecasts. For market participants like Optiver, the ability to accurately predict price movements in the final moments of trading holds immense value, informing crucial decisions regarding order adjustments, supply and demand assessment, and ultimately, optimizing trading strategies. The success of the CatBoost model in this context highlights the potential of machine learning to empower market makers with data-driven insights, potentially enhancing efficiency and competitiveness within the fast-paced world of financial markets.

C. Limitations and Future Research

While the study yielded promising results, it is crucial to acknowledge certain limitations. The dataset provided by the Kaggle competition encompassed a specific timeframe and a curated selection of Nasdaq-listed stocks. Further research should aim to expand the scope by incorporating data from diverse market conditions and a broader range of stocks, potentially revealing generalizable patterns and enhancing the robustness of predictive models. Additionally, delving deeper into feature engineering techniques and exploring alternative model architectures could potentially unlock further improvements in accuracy andgeneralizability.

Furthermore, ethical considerations surrounding algorithmic trading warrant careful examination. The increasing reliance on machine learning models in financial decision-making necessitates rigorous oversight and regulatory frameworks to address issues such as potential biases, market manipulation,

and systemic risks. Future research should explore these critical aspects to ensure the responsible and ethical application of AI within the financial sector.

D. Conclusion

In conclusion, this research has demonstrated the remarkable potential of machine learning models, particularly CatBoost regression, in accurately forecasting closing price movements for Nasdaq-listed stocks. By harnessing the power of data analysis and predictive modeling, this study has not only contributed valuable insights to the "Optiver - Trading at the Close" competition but also shed light on the transformative potential of technology in navigating the complexities of financial markets. As we move forward, further research endeavors should strive to broaden the scope, refine methodologies, and address ethical considerations to fully unlock the power of AI in shaping a more efficient, informed, and responsible financial landscape.

REFERENCES

[1] Adam Hayes. (2023, Dec, 14). "Stocks: What They Are, Main Types, How They Differ From Bonds. "Investopedia".[https://www.investopedia.com/terms/s/stock.asp]

[2] Adam Hayes. (2023, Dec, 22). "U.S. Securities and Exchange Commission. "What Kinds of Stocks Are There?" "Investopedia."[https://www.investopedia.com/articles/investing/082614/how-stock-market-works.asp].

[3] Adam Hayes. (2023, Dec, 22).U.S. Securities and Exchange Commission. "Description of Capital Stock." Investopedia.[https://www.investopedia.com/articles/investing/082614/how-stock-market-works.asp].

[4] Matthew Frankel. (2023, Nov, 27). "Investing in Safe Stocks and Low-Volatility Stocks". The Motley Fool.[https://www.fool.com/investing/stock-market/types-of-stocks/safe-stocks/].

[5] Adam Hayes. (2023, Dec, 22).U.S. Securities and Exchange Commission. "Description of Capital Stock." Investopedia.[https://www.investopedia.com/articles/investing/082614/how-stock-market-works.asp].