**Optiver’s Trading at the Close**

Aditi Balaji, Janhavi Sathe

# Introduction

In this project, we will tackle the Kaggle competition “Optiver - Closing at the Trade”, which involves building a model that can predict the closing price movements for 200 Nasdaq listed stocks using data from the order book and the closing auction of the stock.

The dataset provided by Optiver for this competition contains time series data from the daily Nasdaq Closing Cross Auctions.

In addition to this dataset, we will potentially explore other data sources such as social media mentions, and expert predictions on social media. By tracking sentiments and events in the news, we can get additional context on how the company stock is likely to do. This is an experimental branch of our project, and not included in the main scope.

# Problem Statement

## Problem:

Our problem statement as per the description provided is as follows:

“Stock exchanges are fast-paced, high-stakes environments where every second counts. The intensity escalates as the trading day approaches its end, peaking in the critical final ten minutes. These moments, often characterized by heightened volatility and rapid price fluctuations, play a pivotal role in shaping the global economic narrative for the day.

Each trading day on the Nasdaq Stock Exchange concludes with the Nasdaq Closing Cross auction. This process establishes the official closing prices for securities listed on the exchange. These closing prices serve as key indicators for investors, analysts and other market participants in evaluating the performance of individual securities and the market as a whole.

Within this complex financial landscape operates Optiver, a leading global electronic market maker. Fueled by technological innovation, Optiver trades a vast array of financial instruments, such as derivatives, cash equities, ETFs, bonds, and foreign currencies, offering competitive, two-sided prices for thousands of these instruments on major exchanges worldwide.

In the last ten minutes of the Nasdaq exchange trading session, market makers like Optiver merge traditional order book data with auction book data. This ability to consolidate information from both sources is critical for providing the best prices to all market participants.

In this competition, you are challenged to develop a model capable of predicting the closing price movements for hundreds of Nasdaq listed stocks using data from the order book and the closing auction of the stock. Information from the auction can be used to adjust prices, assess supply and demand dynamics, and identify trading opportunities.”

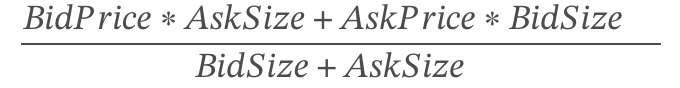
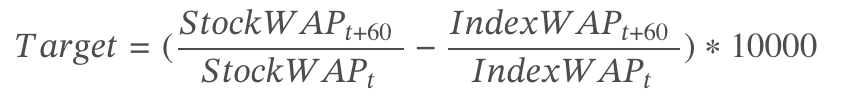
## Dataset used

Our primary dataset will be taken from the Kaggle competition hosted by Optiver:

*Tom Forbes, John Macgillivray, Matteo Pietrobon, Sohier Dane, Maggie Demkin. (2023). Optiver - Trading at the Close. Kaggle.* [*https://kaggle.com/competitions/optiver-trading-at-the-close*](https://kaggle.com/competitions/optiver-trading-at-the-close)

This dataset contains Orderbook/Auction data containing bid/ask price and bid/ask size information for 200 stocks over a period of 481 days. (Not all stocks have data on each day). The size of the dataset is 5237980 rows × 17 columns.

The columns are as follows, quoted directly from the competition host description:

1. stock\_id - A unique identifier for the stock. Not all stock IDs exist in every time bucket.
2. date\_id - A unique identifier for the date. Date IDs are sequential & consistent across all stocks.
3. imbalance\_size - The amount unmatched at the current reference price (in USD).
4. imbalance\_buy\_sell\_flag - An indicator reflecting the direction of auction imbalance.
5. buy-side imbalance; 1
6. sell-side imbalance; -1
7. no imbalance; 0
8. reference\_price - The price at which paired shares are maximized, the imbalance is minimized and the distance from the bid-ask midpoint is minimized, in that order. Can also be thought of as being equal to the near price bounded between the best bid and ask price.
9. matched\_size - The amount that can be matched at the current reference price (in USD).
10. far\_price - The crossing price that will maximize the number of shares matched based on auction interest only. This calculation excludes continuous market orders.
11. near\_price - The crossing price that will maximize the number of shares matched based auction and continuous market orders.
12. [bid/ask]\_price - Price of the most competitive buy/sell level in the non-auction book.
13. [bid/ask]\_size - The dollar notional amount on the most competitive buy/sell level in the non-auction book.
14. wap - The weighted average price in the non-auction book.
15. seconds\_in\_bucket - The number of seconds elapsed since the beginning of the day's closing auction, always starting from 0.
16. target - The 60 second future move in the wap of the stock, less the 60 second future move of the synthetic index.
17. The synthetic index is a custom weighted index of Nasdaq-listed stocks constructed by Optiver for this competition.
18. The unit of the target is basis points, which is a common unit of measurement in financial markets. A 1 basis point price move is equivalent to a 0.01% price move.
19. Where t is the time at the current observation, we can define the target:  
    

In addition to this, we may look at combining the above dataset with the data we will acquire from social media trends corresponding to the dates in the Optiver dataset. We are yet to identify reliable, exhaustive and influential sources for the same. Once identified, we will modify the dataset to include any events that occurred related to a particular stock or industry. We will extract the associated sentiment polarity which will be float values in the closed range -1 to +1, and include it in our analysis.   
The social media data may be acquired using the tool <https://informationtracer.com> developed by Zhouhan Chen, or through web scraping methods.

## Model Selection

We will evaluate the below listed approaches using the Mean Absolute Error as outlined in “Evaluation Procedure” subsection. This list contains several models, and depending on the

**Phase I: Basic Models**

1. Baseline Approach: We assume that there is no useful information about the direction in which a stock moves, and output a predicted value of 0 for all observations.
2. Last Value: This is very similar to the baseline model, except here we predict that the auction price does not change and output the last value available. – This had been planned, however due to the nature of test data provided, this will be more difficult to implement than the expected improvement.
3. Moving Average: In this, the predicted value will be the mean of the previous *N* values. – This had been planned, however due to the nature of test data provided, this will be more difficult to implement than the expected improvement.
4. Linear Regression: Based on previous feedback, we will be implementing a basic linear regression model to predict the closing price.

**Phase II: Boosted Decision Trees**

1. XGBoost (XGB): We will scale our data and apply the XGBoost model using select features such as the closing prices of the last *N* days.
2. LightGBM (LGBM): Similar to the previous, we will implement LightGBM and compare performance with XGB to see if there are any improvements.

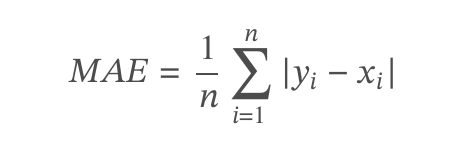
**Phase III: Neural Networks**

1. Recurrent Neural Networks (RNNs): We can implement a very basic RNN to better exploit the time series data.
2. Long Short Term Memory (LSTM): As LSTM is a modified version of RNN that has shown better performance on stock market prediction problems, we will implement LSTM.

For integrating the social media information, we are treating that as an experimental trial. We will use ensemble models to handle the data and see if we can glean anything from additional features.

## Evaluation Procedure

Our evaluation metric is Mean Absolute Error (MAE). The formula is given by:



Where:

*n* is the total number of data points.

*𝑦𝑖* is the predicted value for the 𝑖𝑡ℎ data point.

*𝑥𝑖* is the observed value for the 𝑖𝑡ℎ data point.

## Expected Results

**Phase I: Basic Models**

For financial data, due to the inherent randomness of the data, baseline models also perform well to a certain degree. Based on the results seen so far, we are expecting the error to be of the order 101.

**Phase II: Boosted Decision Trees**

We expected the XGB and LGBM to perform equally well, if not better than the baseline models. As this problem statement is part of a Kaggle competition, we have seen other competitors test their versions of this model with marginal improvements.

**Phase III: Neural Networks**

We are expecting these models to perform the best out of all the models we will test. We are targeting an error rate close to 3, however this is not based on other implementations and is our own target.

# Literature review

There are several prior works for this subject. A few sample implementations that we will take a look at are:

1. As we are adapting this project from the Kaggle competition, we will also be referring to competitive implementations that utilize only the dataset provided here.
   1. Tutorial Notebook for the competition by Optiver:- -   
      <https://www.kaggle.com/competitions/optiver-trading-at-the-close/discussion/441590>
   2. We explored the dataset using the notebook:   
      <https://www.kaggle.com/code/chiangken/introduction-and-explore-data-analysis>
2. ​​Machine Learning Techniques applied to Stock Price Prediction by Yibin Ng, Towards Data Science on Medium, published Jan 28, 2019.  
   <https://towardsdatascience.com/machine-learning-techniques-applied-to-stock-price-prediction-6c1994da8001>   
   This blog gives a broad overview of various methods used for stock market predictions, from the very basic ones such as Last Value and Moving Average predictions to XGBoost and LSTM. We take inspiration from this to try various methods and compare their performance. We expect that XGBoost and LSTM will give the best performance.

The papers below provide latest research on how information from social media can be utilized to computationally predict stock market trends. We will refer to this material only near the end of the project, provided we have time to incorporate the information gathered using the tool “Information Tracer”.

1. S. Qiu and Z. Jia, "Forecasting the Trend of NASDAQ Stock Based on Machine Learning and Tween Analysis," 2022 2nd International Conference on Big Data, Artificial Intelligence and Risk Management (ICBAR), Xi'an, China, 2022, pp. 74-78, doi: 10.1109/ICBAR58199.2022.00022. <https://ieeexplore.ieee.org/document/10108281>
2. Sekioka, S., Hatano, R. & Nishiyama, H. “Market prediction using machine learning based on social media specific features.” Artif Life Robotics 28, 410–417 (2023). <https://doi.org/10.1007/s10015-023-00857-z>
3. Mazhar Javed Awan, Mohd Shafry Mohd Rahim, Haitham Nobanee, Ashna Munawar, Awais Yasin and Azlan Mohd Zain. “Social Media and Stock Market Prediction: A Big Data Approach” <https://www.techscience.com/cmc/v67n2/41320/html> Computers, Materials & Continua, DOI:10.32604/cmc.2021.014253

In addition to this, we hope to utilize some of the prior work from <https://informationtracer.com> in gathering data from social media.

# Technical Approach

Data:   
<https://www.kaggle.com/competitions/optiver-trading-at-the-close/data>

Baseline Code: <https://www.kaggle.com/code/janhavihsathe/optiver-2023-basic-submission-demo/notebook>

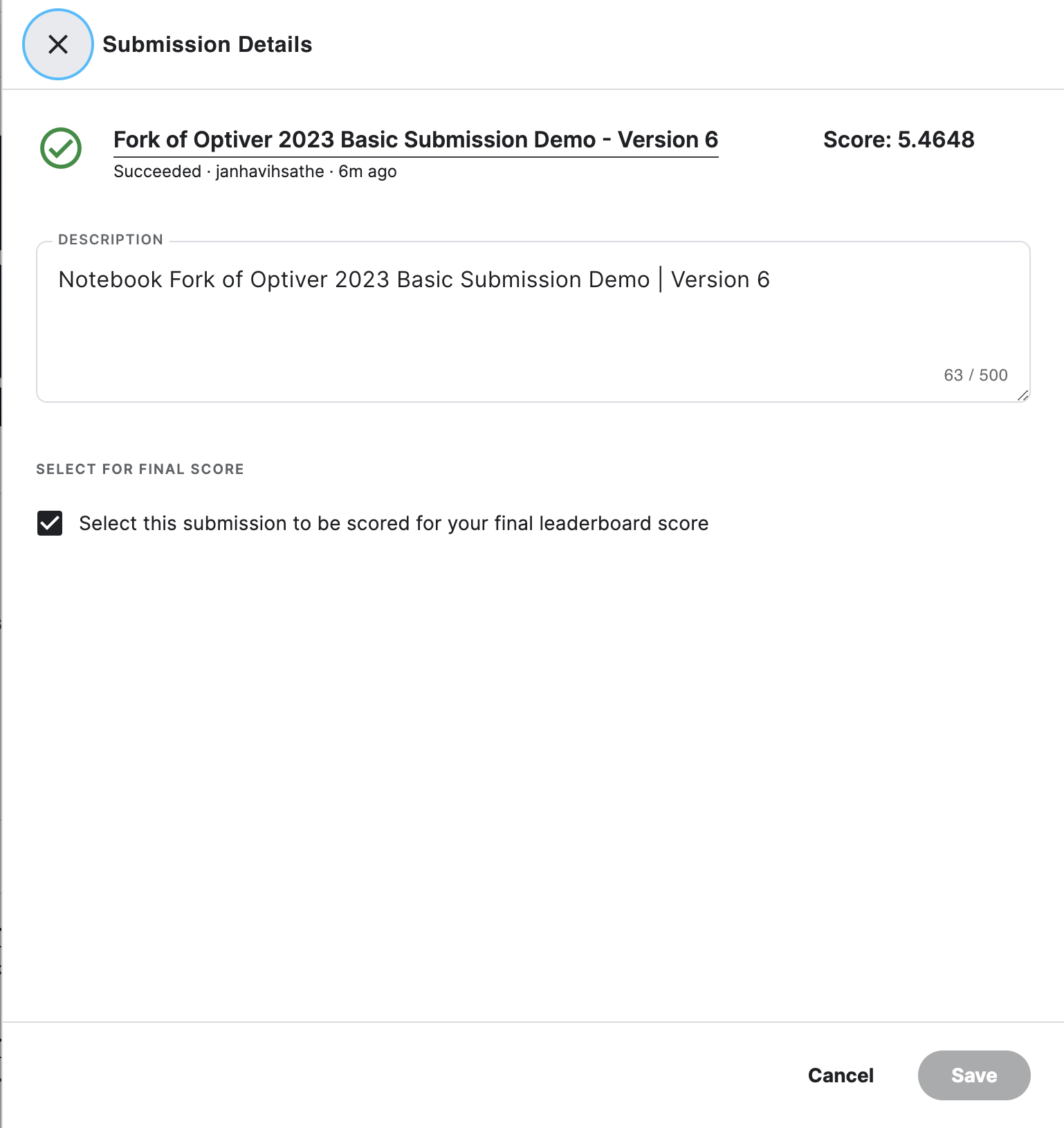
We are splitting our approach into two stages: First is directly implementing modes such as XGBoost, LightGBM, etc. using i.i.d assumption in order to observe the prediction patterns. After the successful implementation of these baseline models, we will start stage 2 of model building which combines the results obtained from the models implemented and exploratory data analysis to engineer and modify the dataset such that its logically and mathematically sound. The final step after rigorous data engineering would be to implement the best model based on least value of mean squared error, which will be then subjected to hyperparameter tuning. In this stage, based on the nature of the dataset, multiple models may be run and ensembled to produce more accurate results.

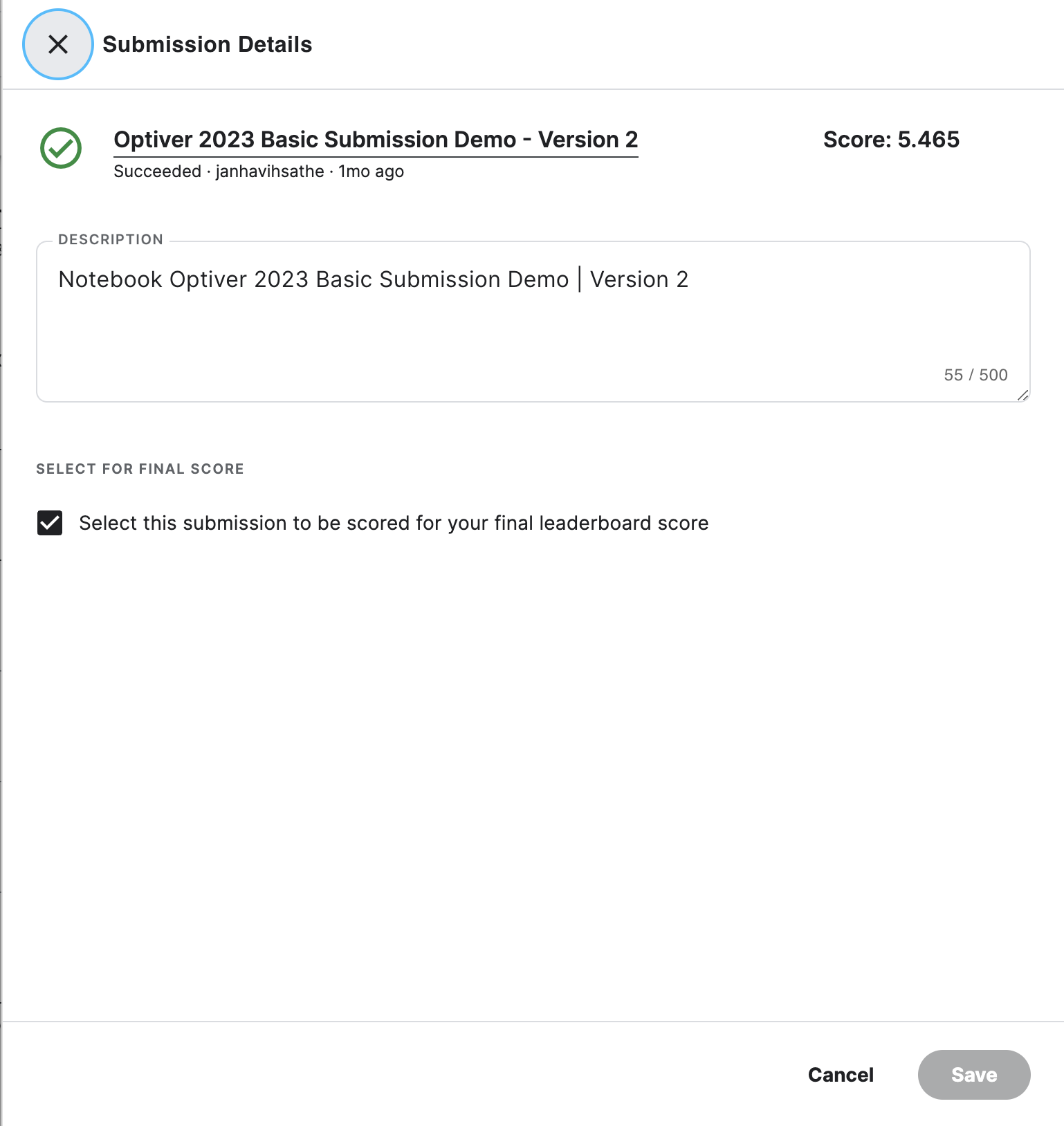
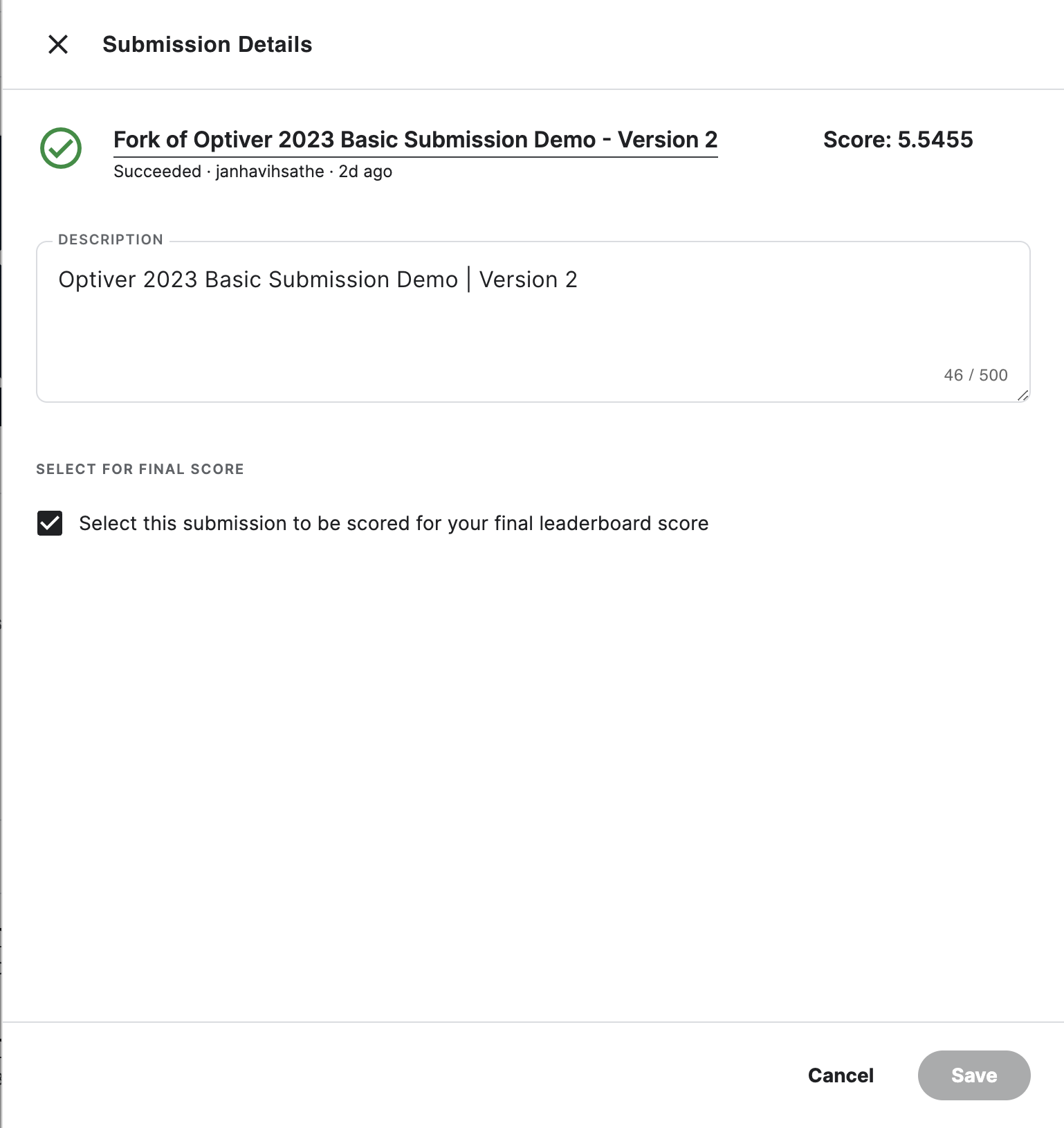
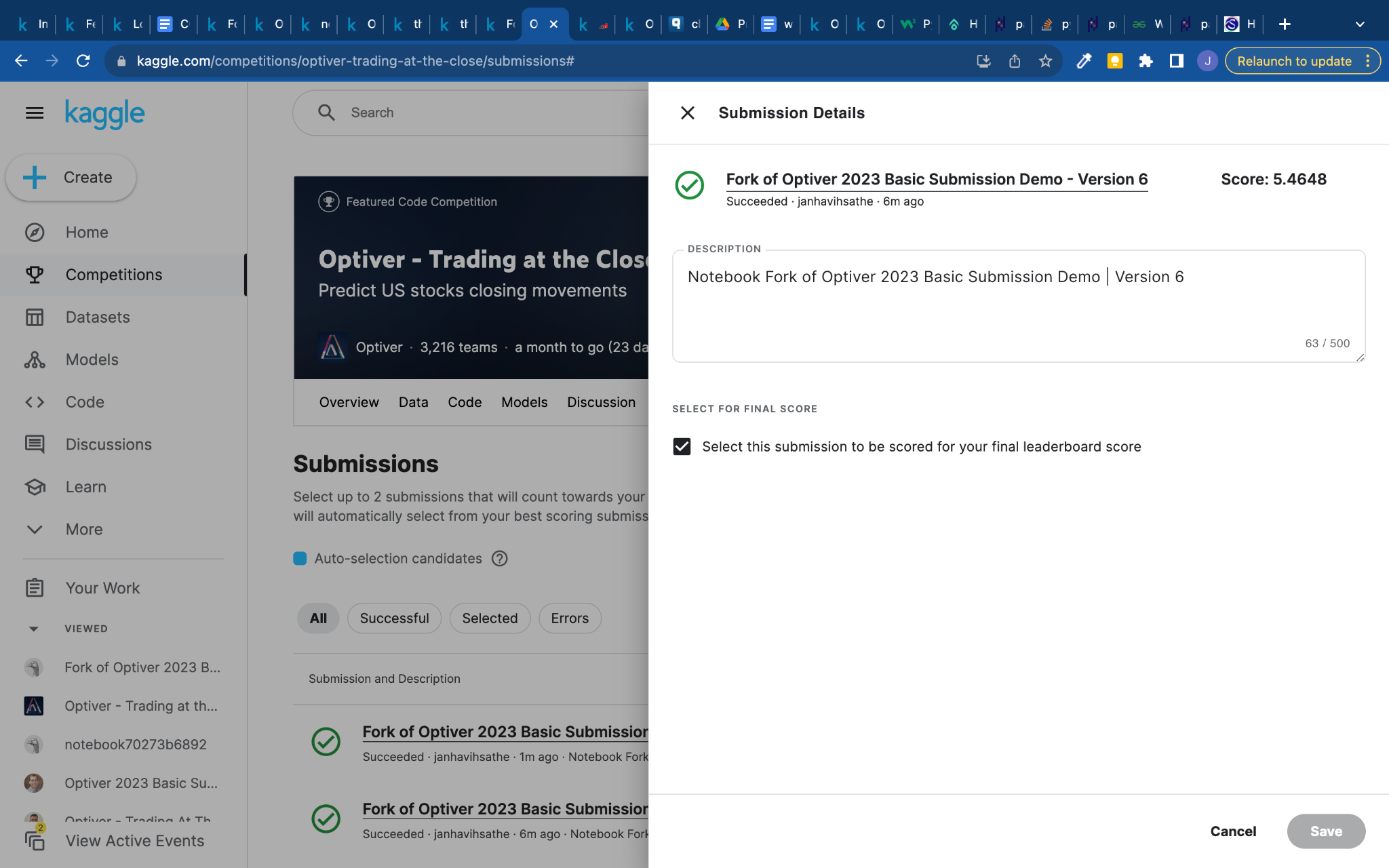
## Baseline Model results

Code: <https://drive.google.com/file/d/1d0smsRQAZQGuFU6TKqZ5EbDoRhqRx94Z/view?usp=sharing>

Due to the inherent randomness of the stock market, the baseline models give an overall low mean square error. By always defaulting to 0, we got an error of 5.465% and by always defaulting to 1, we got an error of 5.5455%. By calculating the exact mean and predicting that value always, we get an error of 5.4648%

For this kind of basic model, by always predicting a value close to the mean, we get an overall low error, however the drawback is that the individual values may not always be close to the actual value.

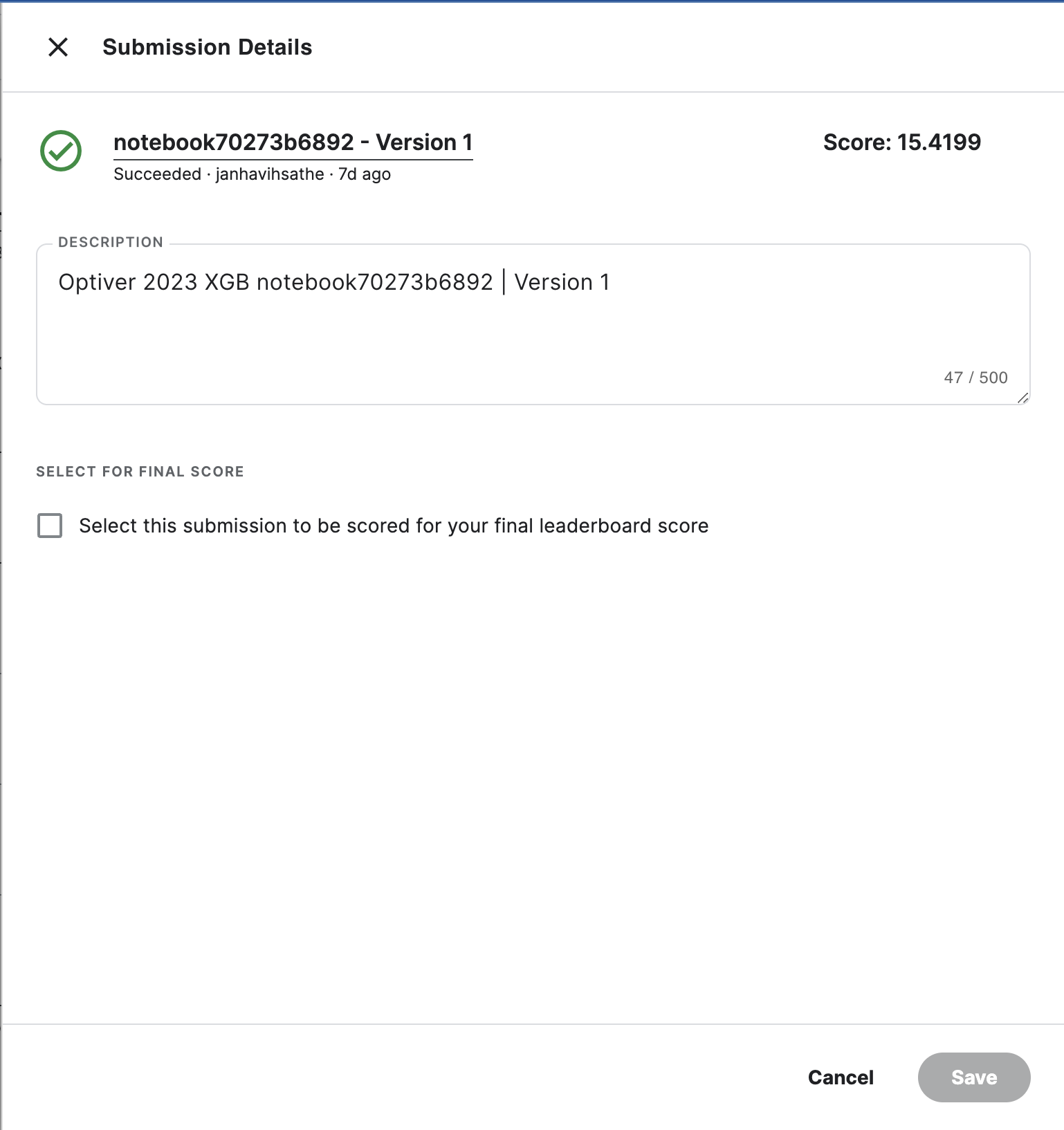




## XGBoost Model

Code: <https://colab.research.google.com/drive/1V7KK69pA0qfrjdMBntk3ZKAE02SIYz8N>

As part of the first stage, we have implemented XGBoost, which gave an MSE of 15.42%



# Results so far

Based on the submission linked above, our results have a minimum error rate of 5.4648% calculated with approximately 94% of the test data. The final results will be based on the other 6%, so the final standings may be different.

For the current phase, we will also be implementing LightGBM and Linear Regression. We will refer to this implementation of Linear Regression to deduce the Synthetic Index values - <https://www.kaggle.com/competitions/optiver-trading-at-the-close/discussion/442851>

And this implementation for LightGBM: <https://www.kaggle.com/code/verracodeguacas/high-speed-predictions-no-gpu>

In the next phase, we will implement RNN and LSTM to better utilize the sequential nature of the data. We expect these to have better results.