ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2.
Available from http://www.i6doc.com/en/.

Predicting the Closing Cross Auction Results at the NASDAQ Stock Exchange

Manuel Hettich[1], Philipp Bielefeld[1], Crispin Schomers[1],
Sarel Cohen[2] and Tobias Friedrich[1]

1- Hasso Plattner Institute, University of Potsdam, Potsdam, Germany
first.last@hpi.de
2- School of Computer Science, Tel-Aviv-Yaffo Academic College,
Tel-Aviv, Israel, sarelco@mta.ac.il

Abstract. This paper aims to present the results and learnings from our work on the last year's *Optiver - Trading at the Close* Kaggle 2023 challenge. It not only touches the two most widely used approaches in the competition, deep learning models and support vector regression models, but also describes the provided dataset drawn from the NASDAQ stock exchange with many detailed attributes, like the imbalance size and far and near prices, recorded in an interval of one second for the last ten minutes of each trading day. It also describes the constraints of the competition. The presented machine learning model based on the LightGBM engine stood out from the competition by feeding back the revealed target data given for the previous day and was one of the top 5% of all models in the competition.

1 Introduction

On the NASDAQ stock exchange thousands of orders are executed every second, which requires fast and reliable models for bringing buyer and seller together and giving both a competitive price with low spreads compared to other stock exchanges. Especially the last ten minutes of a trading day are characterized by an increased volatility and the closing courses serve as key indicators for many market participants [1]. While predicting the stock market attracts many researchers, as it promises enormous financial gains, it is also one of the most challenging problems. With the increasing computational power over the lasts decade, researchers were especially successful with supervised learning approaches, like artificial neural networks (ANN) or Support Vector Machines (SVN). But most current models still only have an accuracy between 60% and 80% [2], showing that there is still potential for improvement. Therefore, trading companies and stock exchanges, which especially benefit from having the best predictions, sometimes open competitions for finding better models or new approaches. One of these contests is the *Optiver - Trading at the Close* challenge 2023 [1], where we participated in and got in contact with many interesting approaches we want to present in the next chapters.

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2.
Available from http://www.i6doc.com/en/.

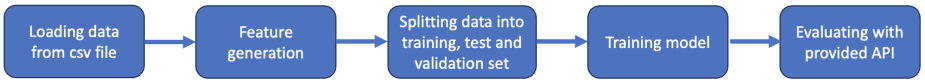

Fig. 1: flow chart model creation and scoring

2 Related Work

Several Studies and concomitant experiments have explored the use of machine learning for predicting the stock market [3]. Even in 1975 Jerry Felsen tried to use Pattern Recognition Techniques for Stock Market Forecasting [4] and found out that it may be superior to the judgment of an human analyst[4, p.594]. Since then many different approaches have been tested. In 1999 Lee and Jo designed an expert system for predicting stock market timing using candlestick charts[5]. They proved that their system has an average hit ratio of 72% but also noted that the lack of automatic machine learning is a limitation of the system and implementing this may enhance the predicting power[5, p.364]. Jasic and Wood proposed an approach for daily stock market indices trades based on neural network predictions[6]. They trained and tested their model with data from the S&P 500, the Dax, the TOPIX and the FTSE in the period 1965-1999[6, p.287]. They yield a profit for trades based on the prediction of the neural network for transaction costs of 0.5% [6, p.296]. Support Vector Machines were used by Das and Padhy for predicting futures prices in the Indian Stock Market[7]. They achieved a mean absolute error of 0.2379 - 0.3887 on the predictions of the Support Vector Machine compared to the actual prices[7, p.25]. A more recent study utilizes another deep learning framework, the Transformer, to predict stock market indices[8]. They trained the model with data from the CSI 300, the S&P 500, the Hang Seng Index and the Nikkei 225 Index[8, p.2]. Wang et. al achieved a superior performance of the Transformer from the perspective of both prediction accuracy and net value analysis compared to other deep learning models and the buy & hold strategy[8, p.7].

3 Methodology

The competition was organised as follows: Optiver, the competition owner, provided a pseudoanonymized dataset and an API which supplied the test data for scoring the model. After all participants handed in their best two models, each model is scored on basis of an, for the participants unknown, test set. After the final scoring, the best 10 teams have to disclose their models, which has unfortunately not yet happened. To have a scored model early on and get used to the Kaggle environment, we decided to build several models early in the competition and improve them iteratively. In order to find a good fitting approach, we split up and each of us focused on a different approach in the early stage, building an own model. After each solution was scored, we decided to focus on the best scoring model to improve it with joint forces. The model creation script

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

itself slightly varies from model to model, but can be roughly split into the steps shown in figure 1. The data was provided in a csv file, which is loaded in the first section. Step two and three are optional and depend on the model. In our best model, which is shown in the flow chart and explained in detail in 4.1, the quality of the model was improved by feeding the model meaningful features, which are created in these two steps. Afterwards, the dataset is split in a train and a validation dataset. The train dataset is needed in the next step to train the model and the validation dataset is used to determine if the model under- or overfit. Last, the solution is scored by calling the provided API which supplies the test data that is used to determine the public score of the submission. One run of the script can take up to nine hours before it is violating the competitions constraints and is killed by Kaggle. Because the computational power put into the training of a model correlates to some extend with the accuracy of an model, it became increasingly challenging to stay just below this threshold.

4 Experiments

Dataset. The "Optiver - Trading at the Close" Kaggle challenge provides participants with a dataset drawn from the NASDAQ stock exchange's daily ten-minute closing auction, aiming to predict future price movements of stocks relative to a synthetic index of NASDAQ-listed stocks. This involves using historical auction data and applying forecasting techniques to predict future price actions. The dataset provided for this competition includes detailed auction data, such as stock identifiers, price imbalances, matched sizes, and weighted average prices, among others. These elements are essential for participants to develop predictive models that can accurately forecast stock price movements in relation to a synthetic index created for the challenge.

Training data: the training data contains the following fields. **stock_id** and **date_id**: Unique identifiers for each stock and trading day, essential for tracking stock performance over time and handling missing data. **seconds_in_bucket**: The number of seconds elapsed since the beginning of the day's closing auction, always starting from 0. The dataset includes data for every ten seconds. The combination of this feature and the date id is used as a timestamp. **imbalance_size** and **imbalance_buy_sell_flag**: Indicate the unmatched share volume and direction of auction imbalance, providing insights into market supply and demand dynamics. **reference_price**, **matched_size**, **far_price**, and **near_price**: Critical for understanding auction pricing dynamics, these fields reflect theoretical equilibrium prices, auction liquidity, and the prices that maximize matched shares considering different market conditions. **[bid/ask]_price** and **[bid/ask]_size**: Offer a glimpse into the competitive buy and sell levels outside the auction, revealing the auction's demand and supply sides. **wap** (weighted average price): Provides the weighted average price in the non-auction book. The wap is calculated as follows: $\frac{BidPrice*AskSize+AskPrice*BidSize}{BidSize+AskSize}$. **target**: Represents the forecast challenge, defined as the 60-second future price movement difference between a stock's WAP and the synthetic index. The tar-

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

get is calculated as follows: $(\frac{StockWAP_{t+60}}{StockWAP_t} - \frac{IndexWAP_{t+60}}{IndexWAP_t}) * 10000$ Where t is the time of the current observation.

Test data: the only difference between the train data and test data is that the test data also includes a "revealed targets" feature. For the data of date i and seconds in bucket t the revealed target feature includes the correct target values for date *i-1* and seconds in bucket t.

4.1 Deep Learning Models

In our primary submission for the Kaggle competition, we developed a sophisticated model leveraging a mix of Python libraries and machine learning techniques to predict the closing price movements of NASDAQ-listed stocks [9]. Utilizing LightGBM for its core prediction engine, our approach centered on comprehensive feature engineering, including advanced statistical aggregations, momentum indicators, price and size imbalances, and liquidity metrics derived from the provided auction and order book data. We implemented custom data handling strategies to manage memory efficiently and ensure the model's responsiveness to the dynamic nature of stock data, addressing potential data leakage with a Purged Group Time Series Split for robust time series validation. Our methodology emphasized the reduction of memory usage, the importance of accurate and nuanced feature generation reflecting market dynamics, and strategic model training to minimize the Mean Absolute Error (MAE) in our predictions. A significant aspect of our approach was the innovative use of revealed targets provided at the start of each date by the competition's API, which offered the actual target values for the preceding date, enhancing our model's accuracy by integrating these real-world data points into our prediction strategy. To achieve this some data transformation of the training data was necessary, because the training data doesn't include the revealed targets as an individual feature. In general we used the target values of date *i-1* and seconds in bucket t as a new feature "revealed_target" for date i and second in buckets t but one special case arose. In case of date zero we don't have any previous target values which could be used as the revealed target feature. To handle this we used as the revealed target feature, the target value of date 0 and multiplied it with a factor of 0.95. We did this to ensure that the value of the revealed target value doesn't have as big of a difference to the target value as if we just set it to zero. We then scaled all input features with the StandardScaler method of the sklearn library[10] and started the training. The application of weighted average predictions and careful feature selection aimed to enhance our model's performance, reflecting a deep engagement with the challenge's objective of predicting relative price movements during the critical closing auction period on the NASDAQ. Notably, our model achieved a public score of 5.2164, ranking us 141th out of 3336 entries (top 5%).

4.2 Comparison to other solutions

One of our competitor's approach employing Support Vector Regression (SVR) with an RBF kernel for stock market predictions represents an alternative strat-

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

egy to the more commonly used models like LightGBM or XGBoost which we based most of our experiments on [11]. By selecting SVR, the competitor aimed to leverage its capability to handle non-linear data patterns, a characteristic often encountered in financial datasets. This choice suggests an exploration into how different algorithms perform under the complex conditions of market data analysis. To address the known computational challenges associated with SVR, particularly for large datasets, the competitor utilized RAPIDS for GPU acceleration, indicating a practical consideration for enhancing the model's efficiency. This approach showcases an exploration of SVR's applicability to financial predictions, highlighting both its potential advantages and the necessary innovations to overcome its limitations.

Another competitor used as ensemble model consisting of four small neural networks[12]. They categorized all features into three groups: price features, size features and other features. These three feature groups are used to train individual neural networks consisting of five to six layer for each group. The outputs of these three models are then utilized as inputs for a fourth neural network model consisting of only three layer whose output represents the final prediction. Notably they achieved a mean absolute error of 4.6106 on the test data with this approach. One special characteristic of this approach is that they used no additional features opposed to many other competitors including our team.

4.3 Ablation study

In our adaptation of a previously published notebook for the Kaggle competition, the incorporation of revealed targets marked a significant enhancement to our model's predictive accuracy. Prior to integrating these revealed targets, our approach, while robust, fell short of breaking into the top 5% of submissions. The revealed targets, which offer actual market outcomes at the start of each trading day, provided a crucial edge by allowing our model to refine its predictions with real-world outcomes. This strategic addition enabled a more precise adjustment to market volatility and trends, directly addressing the limitations of relying solely on historical and derived features.

The scaling of the input features with the StandardScaler method of the sklearn library[10] got us another gain in minimizing the mean absolute error of our model. Without the feature scaling we achieved a mean absolute error of 6.024691 on our validation data. The incorporation of scaled features got us to a mean absolute error of 5.820556 on the validation set, which is an improvement of 0.204135.

5 Conclusions & Further Research

Predicting the stock market remains challenging due to its complex nature, influenced by various factors ranging from central bank policies to company-specific events. Without access to crucial news, predicting individual stock prices or

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

market trends becomes more uncertain. In our recent competition, only technical financial indicators were provided as inputs. Even the best-performing model achieved only a slight improvement over our own, indicating that regardless of the method used, the top 5% of models performed similarly. This suggests that the provided inputs may not be comprehensive enough for significantly better predictions. Some researchers have incorporated additional information, such as indicators from other markets, to gauge global market sentiment just before trading begins. While this was not feasible in our competition due to data constraints, integrating such features or analyzing news sentiment could enhance model precision. Moreover, stock markets exhibit cyclic behavior, with periods of boom, slump, and sideways movements. It's essential to train models on diverse market conditions to improve their resilience and ensure effectiveness across various scenarios. As Warren Buffet famously said, "A long string of impressive numbers multiplied by a single zero always equals zero."

References

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

[2] Surbhi Sharma and Baijnath Kaushik. Quantitative analysis of stock market prediction for accurate investment decisions in future. *Journal of Artificial Intelligence*, 11(1):48–54, 2018.

[3] Troy J Strader, John J Rozycki, Thomas H Root, and Yu-Hsiang John Huang. Machine learning stock market prediction studies: review and research directions. *Journal of International Technology and Information Management*, 28(4):63–83, 2020.

[4] Jerry Felsen. Learning pattern recognition techniques applied to stock market forcasting. *Transactions on System, Man, and Cybernetics*, 1975.

[5] G.S. Jo K.H. Lee. Expert system for predicting stock market timing using a candlestick chart. *Expert Systems with Applications*, 1999.

[6] Douglas Wood Teo Jasic. The profitability of daily stock market indices trades based on neural network predictions: case study for the sp 500, the dax, the topix and the ftse in the period 1965â1999. *Applied Financial Economics*, 2004.

[7] Sudarsan Padhy Shom Prasad Das. Support vector machines for prediction of futures prices in indian stock market. *International Journal of Computer Applications*, 2012.

[8] Shuqi Zhang Qiuhui Zhang Chaojie Wang, Yuanyuan Chen. Stock market index prediction using deep transformer model. *Expert Systems with Applications*, 2022.

[9] Crispin Schomers, Philipp Bielefeld, and Manuel Hettich. Tuned lightgbm - best public score. https://www.kaggle.com/code/crispinschomers/tuned-lightgbm-best-public-score, 2023.

[10] scikit-learn developers. sklearn.preprocessing.standardscaler. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html.

[11] Wilmer E. Henao. Support vector machines through rapids. https://www.kaggle.com/code/verracodeguacas/support-vector-machines-through-rapids/notebook, 2023.

[12] ALINA_CRAM. Optiver_pytorch_regression. https://www.kaggle.com/code/alinacram/optiver-pytorch-regressionCreate-the-model, 2023.