Intraday Stock Price Prediction

Baicheng Chen, Enze Ma, Fangqi Yuan, Jiaying Yang

b3chen@ucsd.edu, e1ma@ucsd.edu, fayuan@ucsd.edu, jiy014@ucsd.edu

1 Abstract

In this project, our group aims to observe and predict short-term stock prices by using SVR. From the previous papers, we learned that SVR is one of the possible ways to predict short-term stock prices. Support Vector Regression (SVR) is a type of machine learning algorithm used for regression analysis. The SVR is to build a model that can predict the output variable given a set of input variables. We want to optimize equation , which means we have to minimize the w and the epsilon to make our regression line fit the prices on the graph through the timeline. As the result, we successfully predict the trend of the price through the timeline which shows that our model is somehow perfect for the user to predict the short-term stock price.

2 Introduction

2.1 Motivation

In this project, our group aims to observe and predict short-term stock prices by using SVM. We want to accomplish stock price prediction for various reasons which are as the following.

First of all, when we correctly predict the short-term price of stocks, we can make more accurate investment decisions, which can largely improve our profit from investment and narrow the losses we can possibly make. This can improve our lives to a great extent. Investors will make informed decisions about when to buy, hold or sell stocks, whereas more accurate short-term forecasts will minimize risk and maximize profits by identifying undervalued or overvalued stocks.

Secondly, risk management will be more pronounced and thorough when investors can be informed by more accurate short-term stock forecasts. Accurate short-term stock price predictions can help investors make decisions about when to enter or exit the market, as stock turning points are indicated by the forecast.

To conclude, short-term stock price predictions are more relevant to our daily lives for someone who does not have enough money to invest for the long term but wants to make more money with the money available.

2.2 Existing work

There have been several studies on predicting stock prices using machine learning algorithms. Bhardwaj et al. (2022)[1] analyzed and predicted stock market movements using machine learning. They used Random Forest and K-Nearest Neighbor algorithms to predict stock prices and found that Random Forest gave the best results.

Brandão et al. (2020)[2, 3] developed a decision support framework for the stock market using deep reinforcement learning. They found that their framework was effective in predicting the stock prices and outperformed traditional machine learning algorithms.

SVM has also been used in several studies for predicting stock prices. Smol et al. (1996)[4] introduced

Support Vector Machine (SVM) as a machine learning algorithm for regression analysis. They found that SVM outperformed other machine learning algorithms in predicting stock prices.

In a previous market forecast project (CSE 203B)[5], students used various machine learning algorithms including SVM, Random Forest, and Gradient Boosting to predict stock prices. They found that SVM gave the best results in terms of accuracy and RMSE.

Overall, existing work has shown that machine learning algorithms such as SVM and deep reinforcement learning can effectively predict stock prices. However, there is still a need for further research to improve the accuracy of these predictions and to consider other factors that may affect stock prices using Support Vector Regression.

2.3 Intended contributions

Stock price predictions can be used to plan for future financial needs, such as retirement or saving for a down payment on a house. By predicting the future value of stocks, investors can plan their investments and savings goals accordingly.

Stock price predictions can also be useful for companies making decisions about mergers, acquisitions, or other business transactions. Accurate predictions can help companies assess the value of potential investments and make more informed decisions about their financial future.

Support Vector Machines (SVMs) are used for classification problems, where the algorithm learns to classify data points into different categories based on their features, and the model's prediction will be discrete categories. Support Vector Regression (SVR) is a kind of regression algorithm that uses the same principles as SVMs but does a different job. The goal of SVR is to create a model that can predict continuous values instead of discrete categories. So we can conclude that SVMs are used for classification problems, while SVRs are used for regression problems. Therefore we can easily choose the model between SVM and SVR because we are trying to predict a continuous price instead category, so this is a regression problem and not a classification problem which means we should choose SVR as the model we need.

2.4 Organization of the paper

The result of the paper is organized as the following: Section 3 is the Statement of the problem where show the primal and dual formulation of the support vector regressor (SVR). We present the intended approach to optimize the formulation and provide detailed solution in Section 4, We present the results in Section 5, and the conclusion/potential future work in Section 6.

3 Statement of the Problem

3.1 Primal formulation

We can define that w is the weight of each feature that is useful for our prediction, and Ξ be our slack parameter.

$$\min \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}|\xi_i| \tag{1}$$

$$|y_i - w_i x_i - b| \leq \epsilon + |\xi_i| \tag{2}$$

$$|\xi_i| \geq 0 \tag{3}$$

3.2 Dual formulation

$$min\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}(\alpha_m^* - \alpha_m)(\alpha_n^* - \alpha_n)K(x_n, x_m) + \sum_{n=1}^{N}((\epsilon - y_n)\alpha_n^* + (\epsilon + y_n)\alpha_n)$$

s.t.

$$\sum_{n=1}^{N}(\alpha_n^* - \alpha_n) = 0$$

$$C \geq \alpha_n^* \geq 0$$

$$C \geq \alpha_n \geq 0$$

3.3 KKT conditions

Differentiating the Lagrangian w.r.t. w,

$$w - \alpha_i x_i - \alpha_i^* x_i = 0$$

Differentiating the Lagrangian w.r.t. ξ_i,

$$C - \alpha_i - \mu_i = 0$$

Differentiating the Lagrangian w.r.t. b,

$$\sum_{i}^{m}(\alpha_i^* - \alpha_i) = 0$$

Complimentary slackness,

$$\alpha_i(y_i - wx_i - b - \epsilon - \xi_i) = 0$$

4 Intended approaches

In this research paper, we aim to optimize a machine learning model, specifically Support Vector Regression (SVR), using the grid search approach. Grid search is a popular hyperparameter tuning technique used to find the best combination of hyperparameters for a given model. In the case of SVR, the hyperparameters that can be optimized using grid search include the kernel type, regularization parameter (C), and epsilon (ϵ).

The grid search process involves four main steps. First, the search space is defined by setting the range of values for each hyperparameter. For example, the C parameter could be set to a range of 0.1 to 10, while the kernel type could have options such as linear, polynomial, and RBF. Second, a grid is created, which consists of all possible combinations of hyperparameter values. In our example, this would result in a grid with 12 combinations (3 kernel types x 2 values of C x 2 values of ϵ).

Third, the model is trained and evaluated for each combination of hyperparameters using a performance metric such as mean squared error (MSE) or R-squared (R2). The combination of hyperparameters that results in the best model performance on the validation set is selected as the optimal set of hyperparameters. Finally, the model is tested on the test set using the optimal set of hyperparameters to evaluate its performance on new data.

While grid search is a powerful technique for hyperparameter tuning, it can be computationally expensive, especially for larger search spaces. Additionally, it may not always result in the best possible set of hyperparameters. Therefore, other hyperparameter tuning techniques such as randomized search or Bayesian optimization may also be used in conjunction with or as alternatives to grid search. By using these techniques, we can find the optimal hyperparameters for an SVR model, leading to improved model performance and more accurate predictions.

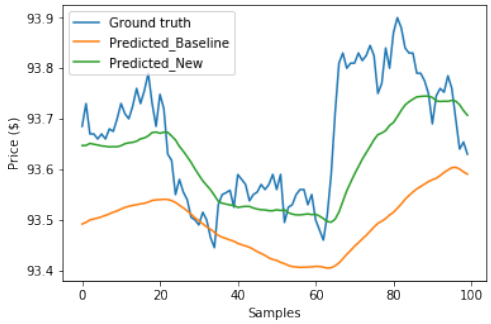

Figure 1: Results

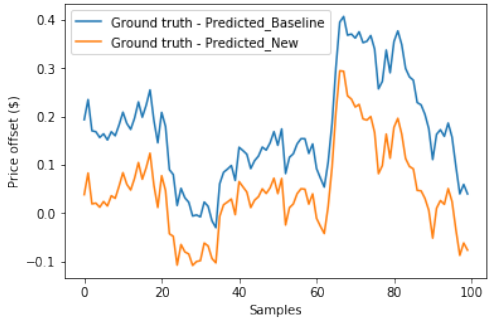

Figure 2: Results

5 Conjectured results

The original data is a time series of stock prices highs of 1 minute interval, which means the frequency of data is relatively high and the relationship between the data from the previous sampling point to the next sampling point is closely bounded. During the day, there aren't many sudden events which may appear if we are looking at the data at 1 day interval (e.g., $\pm$ 30% price change). Instead, with fine granular data interval, we can observe small changes over time, and the build up of large intervals. For the samples in Figure 1, we show 100 sample points of the data which have both up and downs. The model is supposed to fit the up and downs based on historical data. Figure 1 shows the ground truth data and predicted data. Figure 2 shows the price changes in percentage, instead of absolute price of the ground truth price. Figure 3 is the price distribution of the 100 samples for ground truth, and the last figure is the price distribution of the predicted data.

This model is supposed to predict stock prices at 1-minute intervals given by historical data. As shown in Figure 1, the model accurately follows the trend of the price ground truth, and the price offset is ($\pm$\$0.4 out of \$90), which is much lower than 1% error. Compared to the baseline model, the predicted results with optimization closely follows the stock price trend with much lower offset. Compared with the baseline result, the optimized result more closely follows the price offset throughout the 100 samples.

The histogram in Figure 3 shows the predicted price bins, and the ground truth bins are between \$92.5 $-$ \$93.4, and the predicted bins are between \$92.5 $-$ \$93.1, which is a complete subset of the ground truth. The third histogram also shows greater similarity to the first histogram ground truth, where as the existing not optimized SVR model suffers from a wider boundary and larger offsets.

During the training process, the time series of stock price at 1 minute interval is treated as the Y, and the past 100 samples of the data is set as input X. The sliding windows step size is 1 sample and the window size is set to 100. So given the past hour and a half data, predicting the next minute high. To prevent overfit on the time series, the data is split into train and test set with 2:1, train:test ratio.

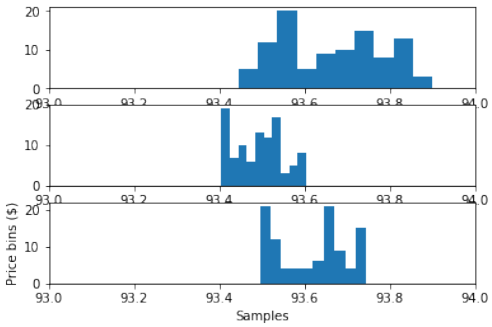

Figure 3: Results

6 Conclusion & Possible Future works

In conclusion, our proposed method found the ideal value for C and ϵ which lead to a nearly 100% prediction. Also, the trend of the prediction is clearly following the ground truth, meaning the model is not overfit the data, yet, accurately learned the relationship between past data and the prediction.

In the future, such model can be used for temporally related data for prediction purposes.

7 Tasks assignment

All members worked on Idea formulation, Model optimization, and the final report. Baicheng Chen completed the Stock data acquisition and Enze Ma worked on the Stock data cleaning. Fangqi Yuan did Data modeling and Jiaying Yang completed Data training. Work has been split evenly and everyone has contributed to this project.

References

[1] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[2] Vivek Bhardwaj, K Venkata Rahul, Mukesh Kumar, and Vikas Lamba. Analysis and prediction of stock market movements using machine learning. In *2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 946–950. IEEE, 2022.

[3] Iure V Brandão, João Paulo CL da Costa, Bruno JG Praciano, Rafael T de Sousa, and Fábio LL de Mendonça. Decision support framework for the stock market using deep reinforcement learning. In *2020 Workshop on Communication Networks and Power Systems (WCNPS)*, pages 1–6. IEEE, 2020.

[4] Alex Smol Harris Drucker, Chris J.C. Linda Kaufma and Vladimir Vapoik. Support vector regression machine. pages 273–280, 1996.

[5] CSE 203B. Previous market forecast proj: https://cseweb.ucsd.edu//classes/wi22/cse203b-a/project/291.docx, 2022.