.

Machine Learning Stock Price Prediction

Project Midterm Report

Team Members: Augusto Menezes Savaris, Ian Gonzalez Alfonzo

Abstract

The problem of forecasting the stock market has been heavily studied recently, due to the inherent complexity of the data involved. This study explores how three different Machine Learning models (Linear Regression, Neural Network, and Support Vector Machine) perform on the regression problem of predicting the future price of a stock. We study how two hyperparameters (lookback and lookahead, which control how much past information the model is given and how long of a trend in the future the labels are defined) affect the models' error metric - Mean Absolute Error. Based on the experiments' results, we see that the three models have similar error metrics, and the non-linear models could outperform the Linear Regression. Finally, we also do not see significant trends of the lookback and lookahead hyperparameters affecting the final error metric, which seem to be dataset-dependent and need to be experimented with for each particular problem and dataset.

1 Introduction

Stock market data is naturally noisy and nonlinear and understanding the feasibility of using Machine Learning to predict future stock trends or prices has been a very popular topic in recent years. Researchers and practitioners have explored this problem in several different approaches that use different datasets and different ML techniques. This study will focus on predicting future stock prices based solely on past prices and better understanding how feasible this task is.

The dataset used during the completion of this study is of the Brazil Stock Exchange, containing 3397 different stocks and ranging from 1994 to 2020. Each row of the dataset contains the name of the stock, and the open, high, low, and close prices for a given day. Additionally, each row contains the volume of the total transactions recorded in the respective day.

Since this is an exploratory study, we test three different Machine Learning models to understand their tradeoffs in Stock Market Forecast. We use Linear Regression as a baseline model, Support Vector Machines and Artificial Neural Networks due to their ability to approximate highly dimensional, nonlinear data.

The overall expected outcome of this study is not to accurately predict future stock prices - rather, we perform this study to understand if it is possible and to what extent we can predict future stock prices. Many studies have accuracy metrics that are marginally better than random guessing, so this task is highly challenging and we cannot assume or expect strictly positive results. The expected outcome is that by the end of this study, we have a good idea of whether the past stock behavior is any indication of its future price.

2 Related Work

There are several approaches that researchers have tried implementing to forecast the Financial Markets. [1] explores predicting future prices based only on past stock behavior, similarly to this study. The authors studied and compared a Neural Network and a Linear Regression model. Additionally, they explored how changing the number of past days as feature inputs to the models impact performance - an idea similar to what we are studying. The conclusions from this study is that the Neural Network did not outperform the Linear Regression model, and they did not observe a significant trend of how the number of past days inputted to the models affected the performance.

Contrasting this idea, [2] considers factors external to the markets but that might affect them. Specifically, [2] utilizes Sentiment Analysis from Twitter posts to gauge the overall public feeling towards a given stock and add information to the model. However, the authors of the paper did not see an improvement in accuracy by considering the Sentiment Analysis and concluded that more experiments are necessary to fully understand the impact of social media opinions on Stock Market Forecast.

3 Methods

Previous research has focused a lot on the choice of model as the focus to improve performance for Stock Price Prediction. The results have shown marginal improvements over time. However, this study will focus on how the feature extraction and problem framing impact the model's performance.

More specifically, we initially are experimenting with two major hyperparameters, called Lookback and Lookahead. These are defined as follows → Lookback: number of days previous to the prediction that we feed the model as input - for example, if Lookback is 10, we feed the model the data of the 10 days before the prediction day; Lookahead: the target for the regression is the average price for 'Lookahead' days in the future - for example, if Lookahead is 1, then the target will be the price of the following day; if Lookahead is 5, then the target will be the average price for the 5 following days.

We expect that higher Lookahead values will allow us to capture the overall trend of the price and neglect noises in the data. Additionally, we expect that larger Lookback values will give the model more information but will increase the computational cost for training and each inference. Both of these hyperparameters will be the focus of the initial approach to this study.

We utilized data from the Brazilian stock exchange and focused our analysis in one stock (VALE3), with daily prices from 1994 to 2020. Since this data is time-dependent, the test set is not randomly shuffled and selected - it is comprised of all data points belonging to the last year of our dataset, 2020. All features and targets were standardized for a value between 0 and 1, which ensures that the price and volume are in the same scale (which is not the case for the raw data).

This study will utilize the following three models to understand how the problem framing and feature extraction impact them differently. For the scope of the midterm project, we will focus on a simple version of each model without much model-specific hyperparameter exploration.

3.1 Linear Regression

Even though one of the main assumptions when working with financial data is that it is nonlinear, we are fitting and evaluating a Linear Regression model. This will enable us to have a baseline to compare the other models - we expect the SVM and NN to outperform the Linear Regression. Linear Regression minimizes the Residual Sum of Squares metric to define the best weight vector (when possible), and runs inference on new data following this equation:

$$y = w^t x$$

As there is no clear winner as to which is the best metric for reporting the error of a linear regression model, the results for this linear regression model will include mean absolute error, also known as MAE, which will be compared against the MAE of other models.

3.2 Support Vector Machine

Although Support Vector Machines are usually used for classification, they can be used for regression as well, using a kind of SVM called Support Vector Regressor (SVR, for short). In this project, we trained our SVR's using two different kernel functions: the Radial Basis Function (RBF) kernel and the polynomial kernel. The C parameter was held constant at 100 and the gamma parameter was left at its default value of 'scale'.

3.3 Artificial Neural Network

Artificial Neural Networks are commonly used for this problem, as they can approximate complex, non-linear data distributions, which are both characteristics of the stock market. We are using a regular ANN, which contains solely Densely connected layers and Dropout layers in its internal structure. We also choose ReLU as the activation function for he output neuron, as this is fundamentally a regression problem. The overall structure of the ANN used for all hyperparameter choices are shown in the image below:

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 5358, 40)          1040
_________________________________________________________________
dense_1 (Dense)              (None, 5358, 20)          820
_________________________________________________________________
dropout (Dropout)            (None, 5358, 20)          0
_________________________________________________________________
dense_2 (Dense)              (None, 5358, 20)          420
_________________________________________________________________
dense_3 (Dense)              (None, 5358, 10)          210
_________________________________________________________________
dropout_1 (Dropout)          (None, 5358, 10)          0
_________________________________________________________________
dense_4 (Dense)              (None, 5358, 1)           11
=================================================================
Total params: 2,501
Trainable params: 2,501
Non-trainable params: 0
_________________________________________________________________
```

We trained the ANN for 25 epochs on each set of hyperparameters, with a batch size of 8 and 0.3 validation split.

4 Preliminary Results

Each numerical entry in the tables below represents the Mean Absolute Error for the validation dataset for the two corresponding hyperparameters, Lookback and Lookahead. In addition, with each table, there is a corresponding graph, showing the performance of the model with the least Mean Absolute Error in its Lookahead-Lookback table.

Before we dive into more details on the performance of each model, it is important to notice that all the graphs below that contain predictions and label prices are normalized and will be between 0 and 1. Additionally, the x axis simply represents the sample number for the test data. Finally, the labels might look different even though we use the same dataset and test data split - and this is caused by different Lookahead parameter values, which average and smooth the target prices.

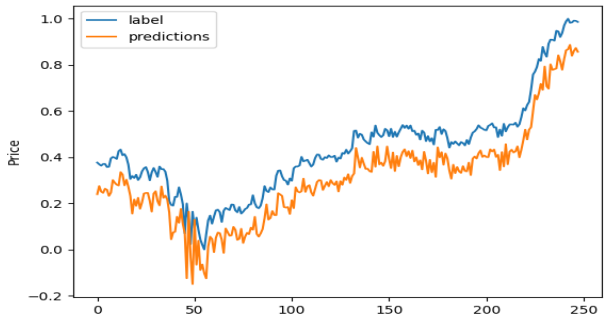
When comparing the models, the Linear Regression models were the worst performing for this study. Although the MAE values for the Linear Regression values were decent, the model seemed wholly unaffected by the changing of the Lookahead and Lookback hyperparameters. The SVR models were better than the Linear Regression models, but only for a specific kernel (RBF) and for specific values of Lookback and Lookahead, as shown in Section 4.2. Here we start to see the importance of varying our hyperparameters. The best performing models were the ANN models, as seen in Section 4.3. They were the most affected by the variation of the Lookback and Lookahead hyperparameters and also led to the model with the best MAE.

4.1 Preliminary Results for Linear Regression

Linear Regression performed better than was expected, since we assumed the non-linearity of the financial data used. We found that the Mean Average Error stayed constant at 0.1 for every combination of Lookahead and Lookback values.

		Lookahead				
		1	5	10	15	20
Lookback	5	0.1	0.1	0.1	0.1	0.1
	10	0.1	0.1	0.1	0.1	0.1
	20	0.1	0.1	0.1	0.1	0.1

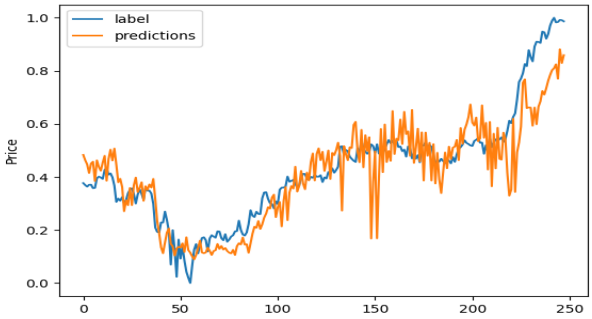
Since they are essentially all the same, the graph for Linear Regression shown below is from the first combination: Lookback=5 and Lookahead=1.



4.2 Preliminary Results for Support Vector Machine

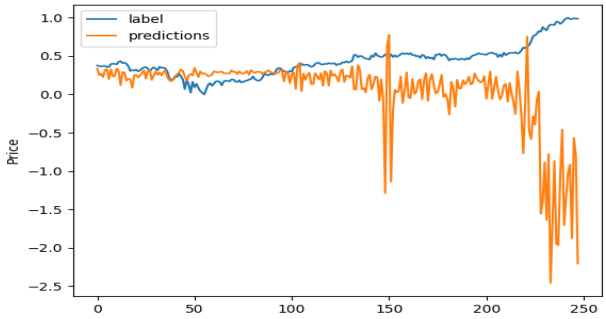
The table and graph below are for the SVR models using RBF as its kernel. The graph is for an SVR model using RBF that had the least MAE of 0.077, with Lookback=5 and Lookahead=1.

		Lookahead				
		1	5	10	15	20
Lookback	5	0.077	0.077	0.077	0.077	0.077
	10	0.121	0.121	0.121	0.121	0.121
	20	0.119	0.119	0.119	0.119	0.119



The table and graph below are for the SVR models using a polynomial kernel. The graph is for an SVR model using a polynomial kernel that had the least MAE of 0.434, with Lookback=5 and Lookahead=1.

		Lookahead				
		1	5	10	15	20
Lookback	5	0.434	0.434	0.434	0.434	0.434
	10	0.954	0.954	0.954	0.954	0.954
	20	0.589	0.589	0.589	0.589	0.589



As can be seen in the tables and graphs above, the SVR with RBF models performed substantially better than the ones with polynomial kernels. The minimum MAE of 0.077 for RBF is better than the minimum MAE for polynomial kernels of 0.434. In fact, an MAE of 0.434 is higher than even the maximum MAE for RBF, which is 0.121, proving that RBF is the better of the two kernels.

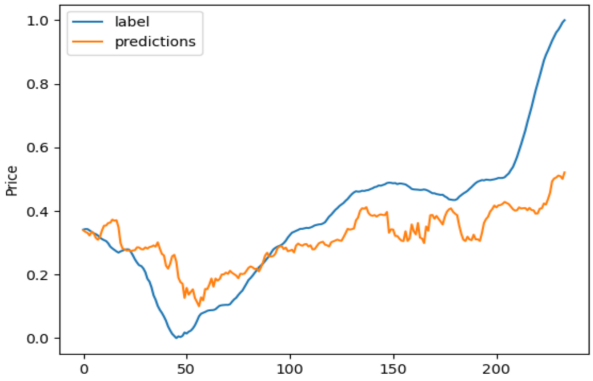
An additional observation is that although changing the Lookahead hyperparameter does not seem to do anything, changing the Lookback hyperparameter value when working with SVR does seem to have an effect on the MAE values. Specifically, for both kernels, a Lookback of 10 is the worst performing and a Lookback of 5 is the best performing, with a Lookback of 20 being in the middle.

4.3 Preliminary Results for Artificial Neural Network

The following table summarizes the results for the Artificial Neural Network and how the choice of the two explored hyperparameters affect the overall model performance.

		Lookahead				
		1	5	10	15	20
Lookback	5	0.1	0.079	0.089	0.111	0.09
	10	0.1	0.095	0.115	0.131	0.09
	20	0.109	0.148	0.1	0.068	0.087

The combination of Lookback=20 and Lookahead=15 showed the lowest Mean Absolute Error, with predictions that compared to the target values closely:



Since there was no clear trend for the Lookahead and Lookback values and the Mean Absolute Error, we believe that the best choice for those parameters is somewhat dependent of the dataset and needs to be experimented with. However, we can see that Lookahead values of 20 showed consistent low errors, as well as Lookback of 5.

All three models have an MAE of at most 0.15, meaning they were at least within 0.15 of the actual price on average. Additionally, we saw the non-linear models outperforming our baseline Linear Regression for certain hyperparameter values as shown above. Thus, it is possible to have an estimation of future stock prices based solely on previous stock price behavior, and the problem framing and amount of past information fed into the model are important factors that affect performance.

5 Future plan

For future studies, we want to explore with different techniques to add information to the models (any techniques we explore will be only based on the stock price, as we do not want external information to affect our analysis). Currently, we are only using the raw price and volume numbers but we want to observe how adding price averages changes the model performance - as the averaged price line will look smoother and more indicative of the general trend, and ignore some of the noise.

Another interesting approach would be to change how the target values are obtained. Instead of considering the raw future price of the stock, we could train the models to predict the difference between the current price and the future price. This way, the model would be directly predicting the future trend of the stock and we could evaluate how this would affect the models' performances and error metrics.

Finally, all of the test results were obtained by using the same stock and a different period of the training data. As a future direction of our work, we want to understand how a model trained on one's stock data generalizes to other stocks. This will give us a better understanding of how much of the underlying price patterns the model is learning.

6 References

[1] Mahdi Pakdaman Naeini, H. Taremian and Homa Baradaran Hashemi, "Stock market value prediction using neural networks," 2010 International Conference on Computer Information Systems and Industrial Management Applications (CISIM), 2010, pp. 132-136, doi: 10.1109/CISIM.2010.5643675.

[2] A. Porshnev, I. Redkin and A. Shevchenko, "Machine Learning in Prediction of Stock Market Indicators Based on Historical Data and Data from Twitter Sentiment Analysis," 2013 IEEE 13th International Conference on Data Mining Workshops, 2013, pp. 440-444, doi: 10.1109/ICDMW.2013.111.