Fatcent

Saurabh Dewangan, Pradeep Agrawal

Abstract

Stock market investment has grown to represent a significant component of the global economy. People are putting a lot of money into the stock market. While many investors are profiting, others are taking a huge hit owing to a lack of market understanding. With the current advancements in the field of machine learning, the same may be utilized to anticipate stock market values based on past data. Once the Machine Learning model is trained on the historical stock prices data, it can predict the future prices of the stock for which it is trained. Using Machine Learning capabilities, one may automate the process of purchasing and selling stocks, decreasing the need for human engagement. The elimination of human interaction helps in a variety of other ways. It eliminates the possibility of stock purchases on the spur of the moment. A taught Machine Learning model will perform objectively. It will always function to its full capacity.

1. Introduction

We've all heard of the stock market. People are putting a lot of money into the stock market. It's an excellent method to supplement your income. Many individuals are gaining millions of dollars in profits by investing substantially in the stock market these days. However, some investors are losing money as well, owing to their inexperience in investing in stocks and a lack of expertise. As a result, many investors withdraw from the market for the rest of their life. They are frightened of losing additional money, so they quit investing in the stock market.

Investing funds in the stock market is beneficial to the global economy's overall development. This initiative assists the investors described in the preceding paragraph who fall into the latter group. When someone is exposed to anything involving money, they are constantly concerned about the prospect of losing the money. As a result, many people avoid investing in stocks at first.

This project is motivated by various factors, but we will focus on the two most relevant ones. The first is that it will allow neophytes to invest in the stock market without having any prior knowledge of it. When you have a program that does your work, why would you strive to learn it? So, if a Machine Learning model can anticipate stock values, why would someone devote valuable time to market analysis, especially if they are new to it? The second motive is that a Model trained to forecast stock prices can outperform people. It will also be devoid of any emotional, mental, and other impacts that humans have. It will also work indefinitely without tiring.

In light of the foregoing, a computer program (a Machine Learning model in this case) is a highly desirable tool in this day and age. It is not a random program; rather, it is a program that learns and improves on its own as it becomes better and better.

2. Problem Definition

Our goal here is to create a machine learning model which will predict the stock prices of a company based on its historical data with as high accuracy as possible. The problem here involves gathering historical data for different stocks, preprocessing them, and then training the model with different algorithms, and try to figure out the model with best accuracy. Unlike a lot of other other Machine Learning models, which predicts discrete values, our model will predict values which are continuous, so calculating accuracy is not straightforward. It will depend on how closer the predict values of stock prices are when comparing to actual prices.

We are trying to create a model which can predict the stock prices with an accuracy of at least 50 percent. While, as mentioned above, calculating the accuracy of the model is not straightforward, we will check whether the predicted prices of the stock follows the similar trend of the original prices. For this, we will visualize the actual and predicted stock prices in a graph to get the overall idea of how the model is performing.

3. Related Work

According to the (Chen & He, 2018) and (Zhang et al., 2019), there are other solutions for comparable job. The prediction of the stock market may be viewed as a time series forecasting problem, and one of the traditional approaches is the Autoregressive Integrated Moving Average (ARIMA) (Box et al., 2015). ARIMA operates effectively in linear and stationary timeseries, but not in nonlinear and nonstationary data in the stock market. One technique (Pai & Lin, 2005) uses ARIMA and SVM to handle this problem. The idea is that forecasting is made up of a linear and a nonlinear component, and that they can predict the linear component with ARIMA and the nonlinear component with SVM. Furthermore, another strategy (Huang & Wang, 2006) integrates the wavelet basis with SVM, decomposing the stock data using wavelettransformation and predicting with SVM. Following that, the Artificial Neural Network (ANN) was integrated with ARIMA to forecast the nonlinear component of stock price data (Areekul et al., 2009).

The combination of wavelet transformation and ANN revealed that useful characteristics should be retrieved for ANN training (Chandar et al., 2016). Convolutional Neural Networks (CNN) were also utilized to anticipate stock values using the limit order book (Tsantekidis et al., 2017). The number of orders and the price of ten bid/ask orders were converted into a two-dimensional array. Furthermore, certain constructed RNNs were used to forecast stock data (Saad et al., 1998) and (Rather et al., 2015). To anticipate stock prices, financial news and events were extracted and represented as dense vectors (Ding et al., 2015). Furthermore, reinforcement learning is a common strategy for improving trading strategies by combining Q-learning with dynamic programming (Nevmyvaka et al., 2006).

4. Proposed method

To solve this problem, we will use Machine Learning algorithms to data sets from several stocks. Accuracy scores will be computed when the model has been trained. It will involve several steps like data preprocessing, preparing the model, training the model and visualizing the results.

4.1. Data Preprocessing

The initial step in the suggested strategy is data preprocessing. We are receiving historical stock price data from the website data-flair.training. This website includes historical data for the majority of the stocks. Preprocessing data is a critical step. It is absolutely required for data to go through the preparation stage. Because the original data may contain a lot of noise, it is vital to decrease it so that it does not interfere with your results. Furthermore, the data contains several characteristics, some of which are beneficial while others are not. During the preparation step, the data is read from the csv file, and any unnecessary characteristics are either eliminated or ignored (here we are ignoring them). The data values are then standardized. The reason for this is because each column in a csv file has a separate range of values, yet they all have the same influence. So, their values are normalized so that they have the similar impact.

Also, data has to be split into training and test dataset. Training data is used to train the model and test data is used to test the model, here we will be using it to compare our predicted stock prices with the stock prices of test data. While most of the Machine Learning models generally split t:e training and test data in the ration of around 80:20, here we will use different ratios for different stocks where training data will sometimes contains less than 50% of the total data. The reason for doing this is because by doing this, we can predict prices for more duration and compare it with actual prices.

4.2. Analyzing Data

We obtained Tata Consumer Products historical stock data set from the Internet every trading day from October 8th, 2013 to October 8th, 2018. The data originally comprises not only numerous aspects, such as the starting price, closing price, high price, low price, volume, turnover, change rate, and so on, but also the stock's name and code. Figure 1 depicts a portion of the closing price chart. The majority of them will change every five minutes. The first step in accelerating the speed of a program and improving its efficiency is to clean the data. (Nelson et al., 2017) conducted some tests with various characteristics for prediction in order to demonstrate that the experiment would provide a satisfactory result when open, close, high, low, and volume are used as model input features. As a result, we picked the first five characteristics as our input and ignored extraneous data such as their names and stock codes. The data with missing values was then eliminated. Next, because the stock may be stopped one day, using those data for prediction is pointless, and they should be deleted from the data sets as well. Another thing to keep in mind is that, in principle, certain stock data may not change once every five minutes, so we should not consider it repeated data and delete it from the data sets.

Figure 1. Stock price graph of Tata Consumer Products

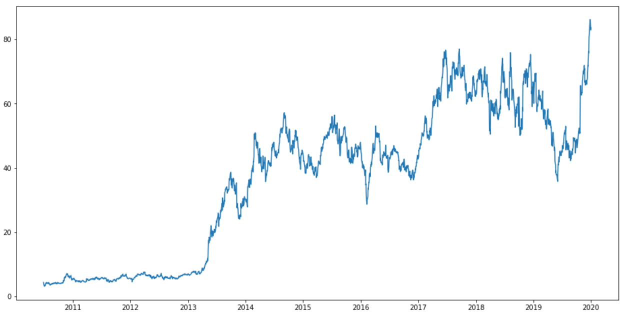


Figure 2. Stock price graph of Tesla

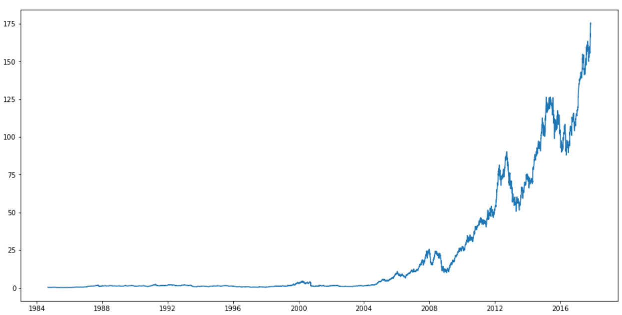


Figure 3. Stock price graph of Apple

4.3. Preparing Model

The following step is to prepare the various Machine Learning models. This is the section where several models and their parameters are defined. The parameters are determined by the type of model. We built an LSTM Sequential model in this stage.

```
lstm_model = Sequential()
lstm_model.add(LSTM(units = 50, return_sequences = True, input_shape = (x_train_data.shape[1], 1)))
lstm_model.add(LSTM(units = 50))
lstm_model.add(Dense(1))
```

Figure 4. LSTM Model

4.4. Training the model

Training the model is one of the simplest parts of the project. After the data has been preprocessed and the model has been prepared, the model is trained until it has converged. Our goal is to forecast whether the stock price will rise or decrease over time. Using these seven parameters, we can calculate the following day's closing price using data. The generator seeks to mine the distributions of the real data and we can retrieve the closing price from the created data, which is why forecasting 7 factors in the next day is possible. The data is divided into two sections for training and testing. For each stock, we select a particular ratio of training to test data.

```
lstm_model.compile(loss = 'mean_squared_error', optimizer = 'adam')
lstm_model.fit(x_train_data, y_train_data, epochs=1, batch_size=1, verbose=2)
```

Figure 5. Training LSTM Model

4.5. Visualizing the results

Since we cannot directly calculate accuracy, it's actually better to visualize the results in a graph. The graph will contain the original stock prices plotted against the predicted prices. If the predicted prices closely follows the actual prices, then we can say that model is working as we are expecting it to. We will plot the graphs for several different stocks and see the results. We will prepare the graph data for different stocks so that we can visualize the results properly.

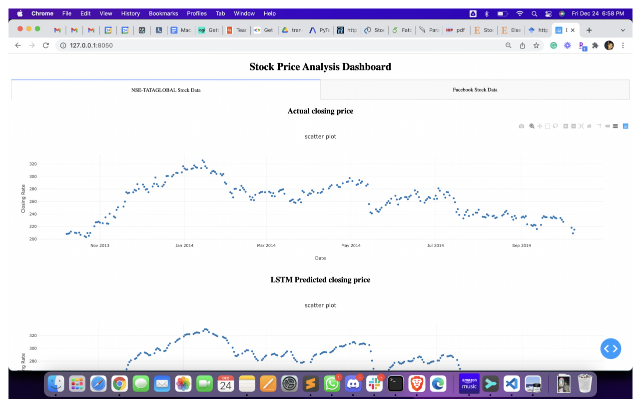


Figure 6. Application Visualization

5. Intuition

Because of the current surge in the field of Machine Learning, we are seeing its use in almost every imaginable discipline. Right now, Machine Learning is all around us, yet we have no idea how crucial it is in our digital lives. YouTube's suggested videos, targeted adverts, and all other web-based recommendations are based on Machine Learning predictions. Not only that, but it is also quite good at forecasting continuous values, which is precisely what we need for predicting stock prices.

As long as there is adequate input data to train the neural network model, machine learning models, particularly neural networks, are excellent tools for forecasting output. A well-trained model will be able to forecast stock prices with sufficient precision to make the entire investment profitable. In order for our model to be lucrative, we do not require a high level of accuracy, such as higher than 90 percent. If the model generates revenues and the accuracy is reasonable, the proposed technique is successful. Since most trained Machine Learning models can make highly accurate predictions, often exceeding 95 percent, we anticipate that as long as our accuracy above 50 percent, we will be able to produce profits, indicating that the suggested strategy is effective.

Many financial calculations and assumptions suggest that the stock market is predictable. In stock investment theory, technical analysis is a process for anticipating the direction of prices based on historical market data. Mean Reversion, a significant assumption, argues that the stock price is transient and tends to go to the average price over time.

6. Experiments

6.1. Models Used

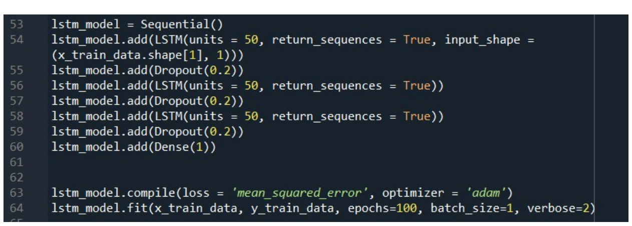
Long Short Term Memory (LSTM), a form of Recurrent Neural Network, and a Linear Regression Model are used to train our Machine Learning model.

The idea behind utilizing LSTM is that, as the name implies, it considers prior information, which is critical for predicting stock prices because they follow a specific pattern that is dependent on previous information. As a result, LSTM may be the ideal model for predicting stock prices.

We will also train our model in a Linear Regression Model to examine how stock prices fluctuate linearly, in order to confirm its linear trend and remove all of the wiggle that occurs with stocks on a daily basis. This essentially assists us in following the linear trend of stock prices.

6.2. Model Description

The LSTM model we're employing have many layers. We experimented with varying layer counts, adding dropout layers for much of the training but removing them for others. Aside from that, there were a variety of LSTM and Dropout layer combinations. In the image below, you can see an example code for the model.



```
lstm_model = Sequential()
lstm_model.add(LSTM(units = 50, return_sequences = True, input_shape =
(x_train_data.shape[1], 1)))
lstm_model.add(Dropout(0.2))
lstm_model.add(LSTM(units = 50, return_sequences = True))
lstm_model.add(Dropout(0.2))
lstm_model.add(LSTM(units = 50, return_sequences = True))
lstm_model.add(Dropout(0.2))
lstm_model.add(Dense(1))


lstm_model.compile(loss = 'mean_squared_error', optimizer = 'adam')
lstm_model.fit(x_train_data, y_train_data, epochs=100, batch_size=1, verbose=2)
```

Figure 7. A sample snapshot of LSTM model used for training

We use a basic Linear Regression model for linear regression. The linear regression model in this case is just used to assess the trend of stock prices. We run it over diverse stock data and then analyze the linear trend without all of the stock price swings.

6.3. Predicted Results based on intuition

We anticipate an accuracy of more than 50% based on the model we've developed. While its quiet difficult to calculate the accuracy, since the predicted output is not discrete value, rather its continuous value. So, we'll visualize the results produced by our trained model and see how close the projected values are to the real outcomes and how much it follows the same pattern.

Following the experiment, we would like to determine whether or not our experiment was successful based on the criteria listed above. The fundamental issue that the experiment must address is whether or not the model is practical, i.e. whether or not the model will be able to create profit. This is the single most critical question that must be addressed.

7. Observations

After successfully training the model and generating the results graphically, we can easily observe that the predicted values in the graph closely track the real values.

The blue color in the graph shows the historical stock prices, the orange color represents the stock prices used to compare expected stock prices, and the green color represents the stock prices forecast. The graph plainly shows that the orange and green plots closely follow one other. As a result, we can state that our model is quite effective.

The linear regression graph of Tesla stock shows the overarching general pattern that a stock's price may follow without all of the variations. It is predicted that Tesla stock prices would continue to rise. While it is not very beneficial if you only look at it once, it is valuable in informing us about the overall trend for long-term investing.

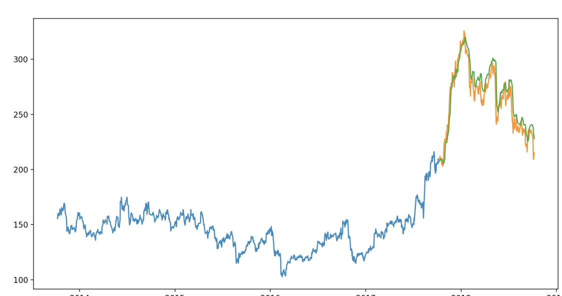


Figure 8. Predicted graph of stock price of Tata Consumer Products

Figure 9. Predicted graph of stock price of Tesla

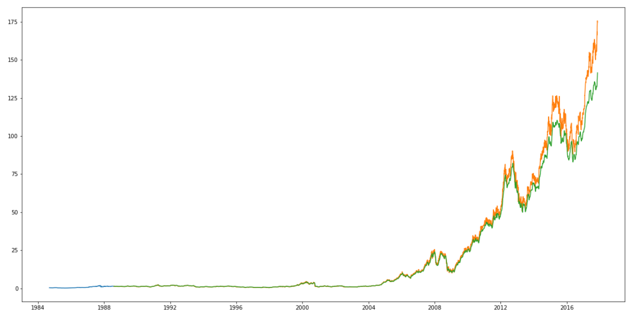


Figure 10. Stock price graph of Apple

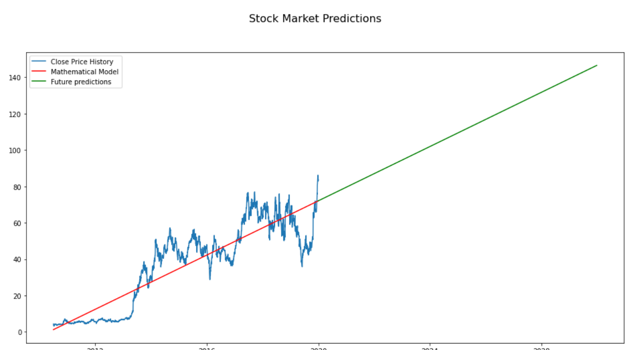


Figure 11. Linear regression graph for stock price of Tesla

Figure 12. Linear regression graph for stock price of TATA shows why sometimes linear regression is not good even for long term

Conclusions

In this paper, we are using LSTM neural network, a type of RNN to predict the stock prices of different stocks, by training the model on their historical prices. We have pre-processed the stock data in order to make our results more accurate.

With the resultant observations discussed in the previous section, we can conclude that our Machine Learning model for stock price prediction is a fully feasible, practical and robust. The results have been evaluated for different stock data and it proves that our model is fully practical, and can be used for real world investment in stock market.

There are also some interesting directions in which this research can go further:

1. Adding real world influences based on the news and headlines of that particular stock.

2. Enhancing the above mentioned feature with the help of trained Machine Learning models, i.e. a separate model will be trained based on news and articles and how it influences the stock prices of the firm.

3. Adding some specific user inputs to better predict the stock prices. For example, if the user is certain that stock prices are going to fall, then based on the user input, the predicted values will change (lower in this case) for better predictions.

References

Areekul, P., Senjyu, T., Toyama, H., and Yona, A. Notice of violation of ieee publication principles: A hybrid arima and neural network model for short-term price forecasting in deregulated market. *IEEE Transactions on Power Systems*, 25(1):524–530, 2009.

Box, G. E., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

Chandar, S. K., Sumathi, M., and Sivanandam, S. Prediction of stock market price using hybrid of wavelet transform and artificial neural network. *Indian journal of Science and Technology*, 9(8):1–5, 2016.

Chen, S. and He, H. Stock prediction using convolutional neural network. *IOP Conference Series: Materials Science and Engineering*, 435:012026, nov 2018. doi: 10.1088/1757-899x/435/1/012026. URL https://doi.org/10.1088/1757-899x/435/1/012026.

Ding, X., Zhang, Y., Liu, T., and Duan, J. Deep learning for event-driven stock prediction. In *Twenty-fourth international joint conference on artificial intelligence*, 2015.

Huang, S.-C. and Wang, H.-W. Combining time-scale feature extractions with svms for stock index forecasting. In *International Conference on Neural Information Processing*, pp. 390–399. Springer, 2006.

Nelson, D. M., Pereira, A. C., and de Oliveira, R. A. Stock market's price movement prediction with lstm neural networks. In *2017 International joint conference on neural networks (IJCNN)*, pp. 1419–1426. IEEE, 2017.

Nevmyvaka, Y., Feng, Y., and Kearns, M. Reinforcement learning for optimized trade execution. In *Proceedings of the 23rd international conference on Machine learning*, pp. 673–680, 2006.

Pai, P.-F. and Lin, C.-S. A hybrid arima and support vector machines model in stock price forecasting. *Omega*, 33 (6):497–505, 2005.

Rather, A. M., Agarwal, A., and Sastry, V. Recurrent neural network and a hybrid model for prediction of stock returns. *Expert Systems with Applications*, 42(6):3234–3241, 2015.

Saad, E. W., Prokhorov, D. V., and Wunsch, D. C. Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. *IEEE Transactions on neural networks*, 9(6):1456–1470, 1998.

Tsantekidis, A., Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M., and Iosifidis, A. Forecasting stock prices from the limit order book using convolutional neural networks. In *2017 IEEE 19th Conference on Business Informatics (CBI)*, volume 1, pp. 7–12. IEEE, 2017.

Zhang, K., Zhong, G., Dong, J., Wang, S., and Wang, Y. Stock market prediction based on generative adversarial network. *Procedia Computer Science*, 147:400–406, 2019. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2019.01.256. URL https://www.sciencedirect.com/science/article/pii/S1877050919302789. 2018 International Conference on Identification, Information and Knowledge in the Internet of Things.