Machine Learning Engineer Nanodegree
Capstone Proposal

Stock Price Prediction using Machine Learning

Samarth Mothakapally
December 7, 2020

Domain Background

Predicting stock prices has been a widely popular domain since the establishment of the trading exchanges in 1792. Accurate predictions help in minimizing the risks. Since the last few decades, conventional statistical methods, like time-series and multivariate analysis, were being used for making such predictions. In the last few years, with the advent of cloud computing and several improvements in running the machine learning algorithms efficiently at scale, machine learning methodologies, including artificial neural networks, genetic algorithms (GA), and fuzzy technologies, have been used to predict stock prices.

In the paper "Support Vector Machines for Prediction of Futures Prices in Indian Stock Market" [1], authors discuss Back Propagation Neural Network and Support Vector Machine techniques to construct the prediction models for forecasting the five major futures indices of Indian markets. They found the normalized mean squared error (NMSE) to be in the range of 0.929 to 1.152 while using SVM, which is pretty decent.

In the paper "A deep learning framework for financial time series using stacked autoencoders and long-short term memory" [2], authors present a novel deep learning framework where wavelet transforms (WT), stacked autoencoders (SAEs), and long-short term memory (LSTM) are combined for stock price forecasting. The performance is reported across a year's worth of data across various types of markets, and reports Mean absolute percentage error (MAPE), correlation coefficient (R), and Theil's inequality coefficient (Theil U). They predict the S&P 500 index in the developed market, with an average value of MAPE and Theil U of WSAEs-LSTM as 0.011 and 0.007.
It looks like ML and Deep learning methods yield good results in stock prediction.

I have always been intrigued by the stock market and have been mainly concentrating on fundamental analysis. Leveraging ML to build a model to predict stock prices will enable me to understand stock movements better if the technical analysis does have any substantial correlation with a company's fundamentals.

Problem Statement

For this project, I will build a Machine learning model that will predict the stock price using historical time series data. The model will take the daily trading data over a specific date range as input and out the prediction for the given query dates.

I will approach the project as a regression problem, such that we get the prediction value of the stock price. I would be using r-square and RMSE to evaluate the performance.

Datasets and Inputs

In this project, I will use the stocks in the S&P 500 as inputs and targets. Each stock will have the following

a. Open: price at which a stock started trading when the opening bell rang
b. High: the highest price at which a stock traded during the day
c. Low: the lowest price at which a stock traded during the day
d. Close: the price of an individual stock when the stock exchange closed for the day
e. Volume: the total number of shares traded in security during the day.
f. Adjusted Close: amends the stock's closing price to reflect that stock's value after accounting for any corporate actions (split, dividends, offering)

The data will be obtained from publicly available sources, mainly yahoo finance, and alpha vantage. I plan to use the data from 2010 to 2020. I plan to scrape Wikipedia for getting all the ticker symbols and leverage several python packages to get the required data.

The Adjusted Close would be the target variable, and the remaining 5 variables will be used as the input variables. We would have 500 * 6 = 3000 data points per day. In total, we would have approximately 3000 * 253 * 10 = 7,590,000 data points, where 253 is the approx. Trading days in a year and 10 is the number of years considered.

I plan to have a data split of 80:20; 80% of the training data and the remaining 20% for testing the performance. Since we would be using the time Series data, I will be dividing the data based on chronological order taking ~ the first 7 years for training and the remaining for testing.

Solution Statement

I would like to use Deep LSTM (Long-Short Term Memory) Neural Network architecture leveraging multidimensional time series stock market data to predict the stock's adjusted close price. I would be using Keras and TensorFlow for model building and predictions. I plan to use a sequence length of 5 to 10 days as the LSTM network's timestep (hyperparameter). The forecast will be the one-step-ahead stock price of the target stock. Hence after the model is trained, the user can choose the input period, and the model can predict the next date's adjusted close price. The solution will be evaluated based on the r-square score and RMSE.

Benchmark Model

I would use a linear regression model as the benchmark. The benchmark model will use precisely the same input as the LSTM network model. This will provide benchmark performance for the LSTM.

Evaluation Metrics

As this is a regression problem, the metrics used would be R-squared, and root mean squared error (RMSE).
R-squared is a statistical measure of how close the data are to the fitted regression line.
R-squared = Explained variation / Total variation
In general, the higher the R-squared, the better the model fits your data.

Root-mean-squared-error is the average deviation of the prediction from the true value, and it can be compared with the mean of the true value to see whether the deviation is large or small. We can then use the RMSE to measure the spread of the y values about the predicted y value. In general, the lower the RMSE, the better the model fits your data.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

$$\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n \text{ are predicted values}$$
$$y_1, y_2, \ldots, y_n \text{ are observed values}$$
$$n \text{ is the number of observations}$$

Project Design

In this final section, summarize a theoretical workflow for approaching a solution given the problem. Provide a thorough discussion for what strategies you may consider employing, what analysis of the data might be required before being used, or which algorithms will be considered for your implementation. The workflow and discussion that you provide should align with the qualities of the previous sections. Additionally, you are encouraged to include small visualizations, pseudocode, or diagrams to aid in describing the project design, but it is not required. The discussion should clearly outline your intended workflow for the capstone project.

The project is expected to have the following stages:

1. Obtaining Data:
    - I plan to get the data using yahoo finance and alpha vantage API, and publicly available packages. I plan on using at least 10 years of stock market trading data summarized at the day level.
2. Data Prep:
    - I would clean up the raw data to improve its quality as I anticipate some inconsistencies in the data due to the following

i. different listing dates (A specific Company may have been listed for less than 10 years)
ii. Trading halts due to several reasons like circuit breakers
iii. Ticker stops trading in the same name due to mergers or acquisitions

3. EDA:
    - I will explore the structure of the data and try to identify the potential relationship between various variables

4. Feature engineering:
    - I will finalize the features and, if required, may create new features that could be used for the model building.
    - I will normalize the data using min-max scaler if feasible. otherwise, will use more traditional approaches of subtracting the mean and dividing by the standard deviation
    - I will create train and test data sets
    - I will create timesteps of the data. Plan to use 5 to 10 days as a timestep.
    - I will get the data into a format which could be used for the model downstream
    - Depending on the model's performance, I could potentially use other stocks or indexes to enrich the features for a specific stock.

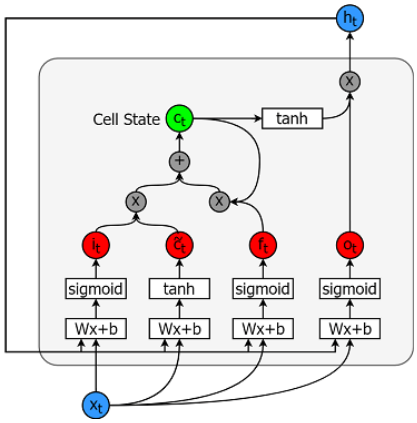
5. Model Building:
    - I plan to build three models; regression benchmark model, vanilla LSTM model, and stacked LSTM model
    - For each stock, a separate model would be built to predict the next day's adjusted close price. For this project, few stocks will be selected from the S&P 500 index.
    - The networks will have at least three layers, each with a hidden states dimension of 150-25. Each layer will use Relu as an activation function, and each layer will have a dropout layer to avoid overfitting. The output layer will have a dimension of 1 and a linear activation function.

6. Model Validation:
    - I will run predictions on the built models using the testing data which was not used for training.
    - I will use R-square and RMSE to check the model performance of all the models.
    - Once a successful model is built, the user can choose which timestep (5 – 10) days of data need to be used as input, and the model will predict the next date's adjusted close price.

7. LSTM[3]:

   Long Short-Term Memory models are extremely powerful time-series models. They can predict an arbitrary number of steps into the future. An LSTM module (or cell) has 5 essential components to model both long-term and short term data.

- Cell state (ct) - This represents the internal memory of the cell, which stores both short term memory and long-term memories
- Hidden state (ht) - This is output state information calculated w.r.t. current input, previous hidden state, and current cell input, which you eventually use to predict the future stock market prices. Additionally, the hidden state can decide to only retrieve the short or long-term or both types of memory stored in the cell state to make the next prediction.
- Input gate (it) - Decides how much information from current input flows to the cell state
- Forget gate (ft) - Decides how much information from the current input and the previous cell state flows into the current cell state
- Output gate (ot) - Decides how much information from the current cell state flows into the hidden state, so that if needed, LSTM can only pick the long-term memories or short-term memories and long-term memories



Reference

1. Das, Shom Prasad, and Sudarsan Padhy. "Support vector machines for prediction of futures prices in Indian stock market." International Journal of Computer Applications 41-3-2012
2. Bao, Wei, Jun Yue, and Yulei Rao. "A deep learning framework for financial time series using stacked autoencoders and long-short term memory." PloS one 12-7-2017
3. Datacamp.com