Capstone Proposal

Machine Learning Engineer Nanodegree

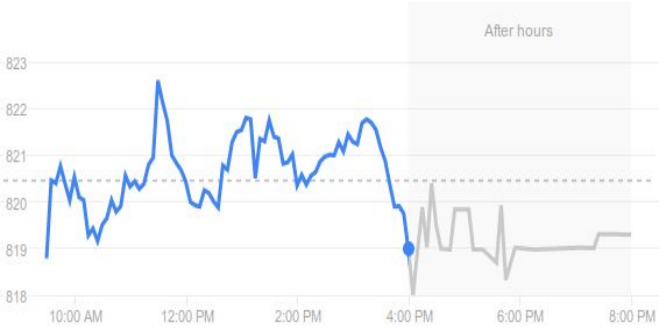
Santiago Bacaro Bravo

August 7th, 2017

Using Machine Learning To Predict Stock Prices

Definition

Mark Twain, the american writer, said: "*OCTOBER: This is one of the peculiarly dangerous months to speculate in stocks in. The other are July, January, September, April, November, May, March, June, December, August, and February.*"[1]. This quote illustrates the volatile nature that the stock market has or has been believed to have. To further elaborate on this point, here's a couple of images taken from Google that show the behaviour of the Alphabet stock both in one year and in one day.



Looking at the graph in the left, it could be possible to think that the data roughly adjusts to a line, but looking at the other graph, the data appears to be completely random. In the stock market world, that's what stock prices usually seem to be: random; and if that were truly the case, stock prices wouldn't be predictable. This project is an approach to predicting stock prices using machine learning.

Now, even if this data is believed to behave randomly, several approaches to predicting it have been developed, ranging from regression, to a combination of SVMs with genetic algorithms[2], to neural networks. And as seen in Udacity's Machine Learning for Trading course, systems to predict stock prices can be developed using a variety of ML techniques, from both supervised and reinforcement learning.

---

[1] Twain, M. The Tragedy of Pudd'nhead Wilson. New York: New American Library, 1964

[2] Choudhry, R., Garg, K. A Hybrid Machine Learning System for Stock Market Forecasting, World Academy of Science, Engineering and Technology 15, 2008

In order to talk about the kind of problem this is and the potential solutions, it's better to look first at the data that will be used. For this project, Google Finance was considered best, because it's simple to use and provides all of the basic data of interest. Sample data obtained from a downloaded CSV file for a given stock looks like this:

Date	Open	High	Low	Close	Volume
7-Aug-17	929.06	931.70	926.50	929.36	1032239
4-Aug-17	926.75	930.31	923.03	927.96	1082267
3-Aug-17	930.34	932.24	922.24	923.65	1202512
2-Aug-17	928.61	932.60	916.68	930.39	1824448
1-Aug-17	932.38	937.45	929.26	930.83	1277734

From this data, it's possible to observe that from all the mess that goes on throughout the day as was seen in the graphs above, only the opening, lowest, highest, and closing prices are returned. However, the only data of interest here, is the close price. All of the other values, except for the date, will be thrown away. Now, there's a couple of ways this problem could be addressed, and the type of the problem is determined depending on which one is chosen. It could be a regression problem if the idea was to predict a precise, continuous value that the stock will take on a given date or date range; it could be a classification problem if the idea wasn't to get an exact price for the stock, but perhaps to determine whether it will go up or down; and it could be a reinforcement learning problem, if the idea was to find a policy to decide if it's best to sell, buy or do nothing with a stock given some state. In this project, despite considering the other approaches to be interesting and potentially as good or even better, the regression approach will be taken. An attempt will be made to predict a precise, continuous value for some given stocks.

For this project, as a benchmark model, the actual stock prices were used. This was made possible by the fact that the project allowed to set some old dates to predict for, thus making it possible for us to compare the predicted values with the real ones.

As an evaluation metric, both mean absolute error (MAE) and R^2 were used. MAE is a good metric for this problem because we care about how far our predictions were from the actual values This metric is calculated as follows:

$$\frac{\sum_{i=0}^{n} |y_i - \hat{y}_i|}{n}$$

Where Yi is the true Y value, Ŷi is the predicted one and n is the number of samples.

On the other hand, R^2 is a good metric, because it portrays the relevance of our features, by giving an insight about how much of the variance in our dependent variable can be explained by our independent variable.

Hence, by applying this metric we'll get a general idea of how far is the average of our predictions from the true values.

The proposed approach to this problem is by using Random Forest regressors.

The flow that is expected to be followed is:

- gather the data
- discard the data that isn't relevant to the project
- Preprocess the remaining, relevant data
- Populate the dataset with relevant indicators for the problem
- Train a model on the dataset
- Predict for unseen data