Machine Learning Engineer Nanodegree

Capstone Proposal

Panagiotis Pnevmatikatos

January 7th, 2018

Proposal

Predicting Stock Prices Project

Domain Background

“Prediction is very difficult, especially about the future”

~~~ Niels Bohr ~~~

The stock market in its early form appeared in France in the 12th century and had a rise in Italy in 13th-14th centuries. Formally, the first company who issued bonds and shares to the general public was Dutch East India Company established in 1602. So, the first formal stock market was Amsterdam Stock Exchange[1]. Now stock markets exist in every developed economy. One of the biggest is the London Stock Exchange, which I will use for this project.

Prediction of stock prices was something that existed from the very beginning of stock markets. And relying only on luck was not enough for people who were trying to get profit. In 1973 Burton Malkiel[2] issued his work A Random Walk Down Wall Street. He argued that you can’t predict stock prices from the historical prices, and financial specialists, predicting the market, actually don’t help or even hurt the profit. Malkiel presented a concept of "random walk" meaning each day's deviations from the central value are random and unpredictable.

Although this work was influential, the attempts of stock predicting did not stop. Nowadays we can pick out 3 general categories of prediction methodologies: Fundamental Analysis (evaluates a company's past performance and its account credibility), Technical Analysis (determines the future price of a stock based on

[1] https://en.wikipedia.org/wiki/Stock_market

[2] https://en.wikipedia.org/wiki/Stock_market_prediction
~~~

the trends of the past price) and Technological Methods (use Data Mining Technologies, Artificial Neural Networks, Machine Learning etc.)

Raut Sushrut Deepak, Shinde Isha Uday, and Dr. D. Malathi from SRM University of India in their academic work Machine Learning Approach in Stock Market Prediction apply Machine Learning and ANN to predict stock values of Bombay Stock Exchange.[3] They came to the conclusion that input data plays an important role in prediction along with machine learning techniques. Using SVM and RBF they reached accuracy up to 89%.

Personal motivation: I was working in the financial industry as a software engineer in Greece, so this area is one of my interest.

Problem Statement

Predicting a stock price is important because having a profit is efficient if we sell the stock at a higher price than we bought it. First, we need to buy a stock which will be rising. Second, in order to achieve maximum profit, we will not sell if the price will continue to go up. And the perfect time to sell is just before the price will go down.

So, the problem is to predict the future price the stock, having historical data for this stock. I will predict the stock price for the next trading day after the last date of my historical data.

Datasets and Inputs

In a very complicated task of predicting stock prices, I believe a dataset plays a very important role and it is important to have as much data as possible for training my algorithm. I decided to use data from London Stock Exchange and to predict stock prices for several companies. Data was obtained from Yahoo Finance.

For the characteristics of the dataset, I looked at the stock data for Marks and Spencer Group (MKS.L). The dataset in excel format contains data for M&S stock from 4/1/2014 to 1/5/2018. There are 951 data points, and for each data point data includes the following: Date, Opening Price, High Price, Low Price, Closing Price, Adjusted Closing Price and Volume. I will use Closing Price as a label for my algorithm.

In this project, I will work with time series data, for this reason, train-test splitting can't be done with functions using shuffle. This would lead to the situation that algorithm would have to predict data from the middle of the dataset, actually being trained on data before and after the predicting data, which is absolutely

[3] http://acadpubl.eu/jsi/2017-115-6-7/articles/8/12.pdf

incorrect. I need to train my algorithm only on the data before the predicting date, as in the real world I will have only these data. I will need to split the dataset manually with respect to the chronological order of our data.

Solution Statement

The solution will be a predicted price for the next trading date after the last date in the training set for the selected stock. This price will be compared to a real observed price for this date for this stock.

Algorithms and techniques that I want to try:

In the pre-processing phase, I will examine abnormalities (zeroes, NaNs, and missing data), I will employ feature engineering.

Tune the regressor with both automated GridSearchCV and manual techniques.

Custom Train/Test set split (the out of the box split algorithm shuffles data and it's not appropriate for our case as mentioned above)

Linear Regression, Ridge Regression and Linear Support Vector Machine (as my benchmark model).

Benchmark Model

As already mentioned, prediction of stock prices is a difficult task. There are people who believe they really can't be predicted, and it is just a guess. Some other people believe that human intuition is the most powerful tool for prediction. Others, again, believe that brokers accumulate knowledge and human intellect works with this accumulated data, figures out trends and gives a prediction without giving a detailed explanation.

In my model, I will try to create a model that works as a third example - finding trend within accumulated data.

I will use an out-of-the-box version of Linear Support Vector Machine algorithm as a benchmark model. I will train and test it on the same data as my primary model, and I will compare the results. Ideally, my final model will outperform the random forests model.

Evaluation Metrics

To quantify the performance we will use a root mean square error (RMSE), which is a frequently used metric to estimate the difference between predicted and observed values. We will use RMSE to evaluate the difference between the predicted stock price for the particular date and actual closing price for this date.

RMSE has some very important advantages for our project. The effect of an error will be proportional to the squared size of this error, so bigger error is more important for this metric, just as bigger errors in predictions are more important in making financial decisions. And small errors have a very small impact. Also squaring the error will ensure that errors for both overestimation and underestimation will be counted instead of neutralized.

Project Design

I will start with Data Exploration. I will check for NaN values, explore summary statistics for the data, visualize the distribution of the data.

I believe linear regression is a good algorithm to use for this problem. I have continuous output data, which, as I believe, depends on several features - our input data. Giving different weights to these features is the way for modeling our output.

Then I will implement train-test splitting with the function I will write. I believe have a very big amount of train data (several years) will not help. It is very unlikely that the stock price will in any way depend on its price 2 or 3 years ago. So I will split the dataset into much smaller pieces, maybe several months or even days, and then I will run a linear regression and obtain the result.

After this, I will try to improve my model by tuning its parameters (e.g. the number of days in the training dataset) and using feature selection. I will compare the results to the initial model.

Finally, I will compare the results to my benchmark.

References:

https://en.wikipedia.org/wiki/Stock_market_prediction
https://en.wikipedia.org/wiki/Stock_market
https://en.wikipedia.org/wiki/FTSE_100_Index#List_of_FTSE_100_companies