# **Machine Learning Engineer Nanodegree**

## **Capstone Project**

Panagiotis Pnevmatikatos

January 10th, 2018

## **I. Definition**

### **Project Overview**

*“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]](#footnote-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]](#footnote-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 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. (<http://acadpubl.eu/jsi/2017-115-6-7/articles/8/12.pdf>). 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%.

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

I will use data from London Stock Exchange 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 csv format (comma separated values) 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 predict Closing Price.

### **Problem Statement**

In this section, you will want to clearly define the problem that you are trying to solve, including the strategy (outline of tasks) you will use to achieve the desired solution. You should also thoroughly discuss what the intended solution will be for this problem. Questions to ask yourself when writing this section:

* *Is the problem statement clearly defined? Will the reader understand what you are expecting to solve?*
* *Have you thoroughly discussed how you will attempt to solve the problem?*
* *Is an anticipated solution clearly defined? Will the reader understand what results you are looking for?*

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.

One of the challenges of this project is that 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 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 on the chronological basis. I will take prices for N days as features and price of the immediately next trading day as a label for these features, after this I will pass some data not using it for training.

I will apply linear regression algorithm to my data and predict the closing price for the next trading day. Having predictions for my test data I will compare them to actual closing prices for the same days and evaluate my algorithm using appropriate metric (see below). I will try to tune my algorithm using feature selection (e.g. vary the number of days before prediction).

### **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.

## **II. Analysis**

### **Data Exploration**

A primary dataset used in this project is a dataset of stock prices for Marks and Spencer Group from London Stock Exchange (code MKS.L). The dataset was downloaded from Yahoo Finance, saved as csv file and converted to Pandas dataframe. It contains 951 data points, from 4/1/2014 to 1/5/2018.

The sample of data:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Date** | **Open** | **High** | **Low** | **Close** | **Adj Close** | **Volume** |
| **0** | 2014-04-01 | 452.000000 | 460.899994 | 452.000000 | 459.799988 | 385.056885 | 7221352.0 |
| **1** | 2014-04-02 | 460.100006 | 472.484985 | 460.100006 | 469.899994 | 393.515045 | 5419474.0 |
| **2** | 2014-04-03 | 469.899994 | 475.100006 | 469.899994 | 471.600006 | 394.938721 | 5786886.0 |
| **3** | 2014-04-04 | 463.600006 | 473.000000 | 460.799988 | 461.899994 | 386.815521 | 8489417.0 |
| **4** | 2014-04-07 | 458.600006 | 460.410004 | 359.200012 | 452.899994 | 379.278534 | 4168587.0 |

Every row is a data point, and columns contain following features:

|  |  |  |
| --- | --- | --- |
| Feature | Format of data | Description |
| Date | Datetime: YYYY-MM-DD | Trading date |
| Open | float 6 decimal places | Price of the stock when market opens on trading date |
| High | float 6 decimal places | The highest price of the stock during trading day |
| Low | float 6 decimal places | The lowest price of the stock during trading day |
| Close | float 6 decimal places | Price of the stock when market closes on trading date |
| Adj Close | float 6 decimal places | Adjusted closing price - closing price of the stock on the trading date that has been amended to include any distributions and corporate actions that occurred at any time prior to the next day's open |
| Volume | float 1 decimal place | Number of shares traded on the trading date |

Statistics for the dataset:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Open** | **High** | **Low** | **Close** | **Adj Close** | **Volume** |
| **count** | 951.000000 | 951.000000 | 951.000000 | 951.000000 | 951.000000 | 951.0 |
| **mean** | 415.092534 | 419.303639 | 410.249037 | 414.812991 | 374.730233 | 6,687,027.0 |
| **std** | 78.815122 | 79.071253 | 78.615665 | 78.639503 | 59.082044 | 4,075,954.0 |
| **min** | 276.000000 | 299.000000 | 255.100006 | 285.200012 | 264.791504 | 400,006.0 |
| **25%** | 339.250000 | 341.959992 | 335.349991 | 338.600006 | 321.249939 | 4,135,072.0 |
| **50%** | 419.899994 | 423.200012 | 414.899994 | 418.500000 | 366.600311 | 5,805,118.0 |
| **75%** | 476.600006 | 481.650009 | 473.333008 | 476.949997 | 415.322662 | 7,907,527.0 |
| **max** | 595.000000 | 600.000000 | 592.500000 | 596.500000 | 520.689880 | 36,639,560.0 |

The count is the same for all the features, which means, that there are no missing values.

Min, max and mean values for the Open, Close, High and Low are very close, but for Adj Close values are significantly lower, which is expected due to the nature of the trading process and of the adjusting closing price definition.

Also, we have 1 NaN value in each column, it is data point 10/6/2017

For better data exploration extra features were created:

df.loc[:,**'Daily Var'**] = df.loc[:,**'High'**] - df.loc[:,**'Low'**]

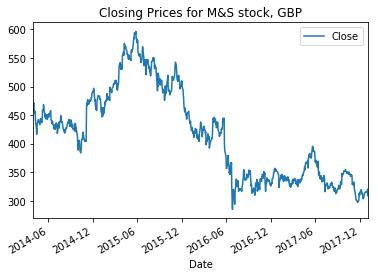
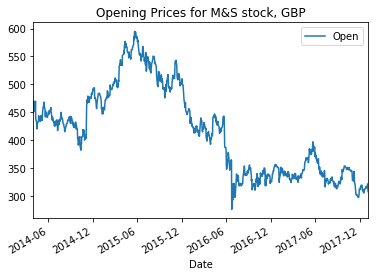
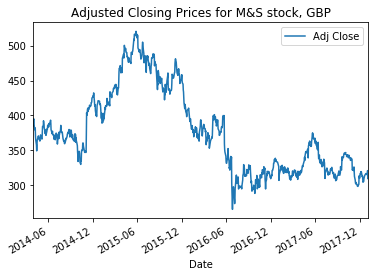
df.loc[:, **'Daily Change'**] = df.loc[:,**'Close'**] - df.loc[:,**'Open'**]

df.loc[:, **'Daily Change %'**] = df.loc[:, **'Daily Change'**] / df.loc[:, **'Open'**] \* 100

I believe features explorations can be done more easily with visualizations.

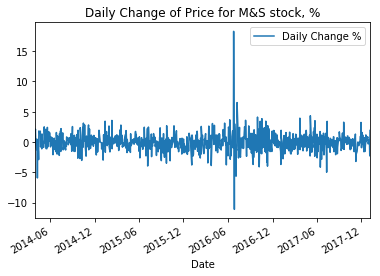
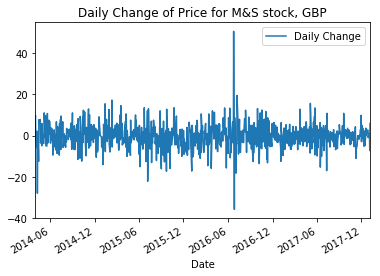
### **Exploratory Visualization**

Let’s explore Open, Close, Adj Close:



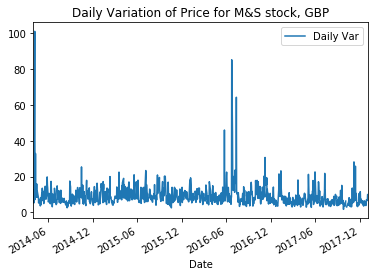
We can see very similar trends for all these prices.

It will be interesting to see also the daily change - difference between closing and opening price:



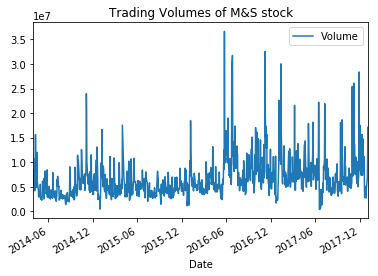
We can see that for the majority of dates price changed less than 20 GBP or less than 5%. Extremely big rise and fall of price within a day we can see the days when Brexit referendum was held, which definitely lead to an abnormal behavior on the stock exchange.

We can see also the daily variance, which will show us the difference between high and low price in the trading day:



We can see rather a similar trend. Most of the daily variations are up to 20 GBP, very high variations over 80 GBP happened on the days of Brexit referendum. But here I can also see an extremely high variation, over 100 GBP, on April the 7th, 2014. Price felt 100 GBP and then rise again. It is hard to find the reason, but maybe it can be connected to the call of the President of Czech Republic for NATO forces to enter Ukraine to prevent eastern expansion.

Finally, the Volume:



Trading volumes change a lot every day, but generally, we can see that they increased last 1,5 year.

### **Algorithms and Techniques**

As mentioned before the signal-to-noise ratio in trading is low. Therefore complicated models would overfit. A linear regression is appropriate for simplicity reasons.

The regression models I use for the predictions are the following:

* Linear Regression Regressor
* Linear Support Vector Machine Regressor
* Ridge Regressor

For tuning the Linear Regression model we will use the **Grid Search** technique.

To evaluate our models we will use as our metrics the **Root Mean Square Error**.

We will also take a look of the

* Root Mean Squared Percentage Error
* Mean Absolute Error
* Explained Variance Score
* Mean Squared Error
* R2 score

For splitting training/test set and because of the nature of the data (time series data) we cannot use the out of the box sklearn's train\_test\_split function which shuffles the data with consequence to loose the influence of the older values to the recent ones. Also, If the data were shuffled, e.g. the close price for 1 Sept 2016 might be in the training set. We might then be asked to predict the close prices for the 7 days after 31 Aug 2016, which would include the price for 1 Sept 2016 which we'd have seen before.

Therefor we would need to develop a custom algorithm for our dataset to split it in train and test set with respect to the chronological order of our data.

### **Benchmark**

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 Linear Support Vector Machine model.

## **III. Methodology**

### **Data Preprocessing**

### During the data exploration we found a very small number of NaN values. We decided to remove those records from the dataset. Also, we observed increased volatility around significant political events like the Brexit referendum. We decided to keep these period in the dataset because they reflect magnified the strong and permanent relation between stockmarket and politics.

Data Exploration has revieled the relation between all of the features of the data set with the Close price. While we made a few experiments with transformed and engineered features we finally decided to use only the Close price because it seemed that the rest of the features because their contribution to our models performance was poor to pay back for the complexity they added.

So, the refined set of features is a set of 10 sequential closing prices.

### **Implementation**

In this section, the process for which metrics, algorithms, and techniques that you implemented for the given data will need to be clearly documented. It should be abundantly clear how the implementation was carried out, and discussion should be made regarding any complications that occurred during this process. Questions to ask yourself when writing this section:

* *Is it made clear how the algorithms and techniques were implemented with the given datasets or input data?*
* *Were there any complications with the original metrics or techniques that required changing prior to acquiring a solution?*
* *Was there any part of the coding process (e.g., writing complicated functions) that should be documented?*

### **Refinement**

In this section, you will need to discuss the process of improvement you made upon the algorithms and techniques you used in your implementation. For example, adjusting parameters for certain models to acquire improved solutions would fall under the refinement category. Your initial and final solutions should be reported, as well as any significant intermediate results as necessary. Questions to ask yourself when writing this section:

* *Has an initial solution been found and clearly reported?*
* *Is the process of improvement clearly documented, such as what techniques were used?*
* *Are intermediate and final solutions clearly reported as the process is improved?*

## **IV. Results**

*(approx. 2-3 pages)*

### **Model Evaluation and Validation**

In this section, the final model and any supporting qualities should be evaluated in detail. It should be clear how the final model was derived and why this model was chosen. In addition, some type of analysis should be used to validate the robustness of this model and its solution, such as manipulating the input data or environment to see how the model’s solution is affected (this is called sensitivity analysis). Questions to ask yourself when writing this section:

* *Is the final model reasonable and aligning with solution expectations? Are the final parameters of the model appropriate?*
* *Has the final model been tested with various inputs to evaluate whether the model generalizes well to unseen data?*
* *Is the model robust enough for the problem? Do small perturbations (changes) in training data or the input space greatly affect the results?*
* *Can results found from the model be trusted?*

### **Justification**

In this section, your model’s final solution and its results should be compared to the benchmark you established earlier in the project using some type of statistical analysis. You should also justify whether these results and the solution are significant enough to have solved the problem posed in the project. Questions to ask yourself when writing this section:

* *Are the final results found stronger than the benchmark result reported earlier?*
* *Have you thoroughly analyzed and discussed the final solution?*
* *Is the final solution significant enough to have solved the problem?*

## **V. Conclusion**

*(approx. 1-2 pages)*

### **Free-Form Visualization**

In this section, you will need to provide some form of visualization that emphasizes an important quality about the project. It is much more free-form, but should reasonably support a significant result or characteristic about the problem that you want to discuss. Questions to ask yourself when writing this section:

* *Have you visualized a relevant or important quality about the problem, dataset, input data, or results?*
* *Is the visualization thoroughly analyzed and discussed?*
* *If a plot is provided, are the axes, title, and datum clearly defined?*

### **Reflection**

The problem of stock market prediction is one which continues to allure researchers every day. In my project I collected data from Yahoo Finance using the pandas datareader, after EDA processes I decided to use Linear Regression to model the stock market prediction. In the data preprocessing phase I engineered model features and, finaly, I implemented 2 models(Linear Regression, Linear Support Vector Machine) and a benchmark model ().As a UI I created a jupyter notebook that loads the model, downloads the latest 10 stock prices and predicts the future price.

Given the powerful toolbox of sklern python library the implementation of the ML regression models were straight forward. Alghough we found challenging the data pre-processing phase. We needed to form the pandas dataframes with the selected features. We also needed to implement the train/test set split logic because we didn’t want to use sklearn’s TimeSeriesSplit method.

GridSearchCV method was very handull in tuning the model parameters but we also had to manual grid search the number of features and the datapoints used for the model training.However, and given that the square of the daily variation of the stock price is slightly larger than RMSE I would suggest to further improvements.

The prediction models developed in this project were based in the linear regression and demonstrate remarkable accurac. However, and given that the square of the daily variation of the stock price is slightly larger than RMSE I would suggest to further improvements.

### **Improvement**

We made several expiraments while we were investigating ways to improve the performance of the models: We included more features (like daily variation, volume, more historical days), we increased the number of training/testing datapoints withouth significant improvement.

However we can still try some improvments like the bellow:

* adding features from the FTSE index and the group of retail stocks index.
* Do more data preprocessing trying to eliminate the Brexit effect by removing the datapoints of the period with the very high market uncertenty. (Of course this would reduce the scope of the project)

Other improvemnts but maybe out of the scope of the project can be

* We can employ LSTM NN model which is also suitable for predicting time series data
* introduce features based on the traders intuition
* use sentiment analysis for the news related to the stock in order to produce features

**Before submitting, ask yourself. . .**

* Does the project report you’ve written follow a well-organized structure similar to that of the project template?
* Is each section (particularly **Analysis** and **Methodology**) written in a clear, concise and specific fashion? Are there any ambiguous terms or phrases that need clarification?
* Would the intended audience of your project be able to understand your analysis, methods, and results?
* Have you properly proof-read your project report to assure there are minimal grammatical and spelling mistakes?
* Are all the resources used for this project correctly cited and referenced?
* Is the code that implements your solution easily readable and properly commented?
* Does the code execute without error and produce results similar to those reported?

References:

Adjusted closing price of the stock:

<https://www.investopedia.com/terms/a/adjusted_closing_price.asp#ixzz53pEXD1WS>

1. <https://en.wikipedia.org/wiki/Stock_market> [↑](#footnote-ref-1)
2. <https://en.wikipedia.org/wiki/Stock_market_prediction> [↑](#footnote-ref-2)