# Machine Learning Engineer Nanodegree

Capstone Proposal

Egar Garcia

February 21th, 2019

## Stock Price Predictor

### Domain Background

A stock market, also called equity market or share market, is a network of economic transactions, where the stocks of public companies can be bought and sold. The equity market offers companies the ability to access capital in exchange of a portion in the ownership of the company for interested outside parties.

In the stock market, other financial securities like exchange traded funds (ETF), corporate bonds and derivatives based on stocks, commodities, currencies, bonds, etc. can be traded. However, for purpose of this project only the exchange of stocks will be considered.

It is common to use stock market and stock exchange interchangeably, but the stock market is a general superset of the stock exchange. If someone is trading in the stock market, it means that it buys and sells stock/shares/equity on one (or more) of the stock exchange(s) that are part of the overall stock market.

The stock market offers an opportunity to investors to increase their income without the high risk of entering into their own businesses with high overheads and startup costs. On the other hand, selling stocks helps companies themselves to expand exponentially, when a company's shares are purchased it is generally associated with the increased in the company's worth. Therefore, trading on the stock market can be a win-win for both investor and owner.

The stock exchange that are leading the U.S. include the New York Stock Exchange (NYSE), Nasdaq, BATS and Chicago Board Options Exchange (CBOE). The Dow Jones Industrial Average (DJIA) is a price-weighted average of 30 significant stocks traded on the NYSE and the Nasdaq, it is the most closely watched market indicator in the world and it is generally perceived to mirror the state of american economy.

Projecting how the stock market will perform is a very difficult thing to do, there are so many factors involved in the prediction some of them emotional or irrational, which combined with the prices volatility make difficult to predict with a high degree of accuracy. Abundant information is available in the form of historical stock prices, which make this problem suitable for the use of machine learning algorithms.

The following ones are some approaches taken to address the problem of making predictions about how the stock prices will perform:

- ARIMA: It is a very popular statistical method for time series forecasting that takes into account the past values to predict the future values. [1]
- Prophet: It is a time series forecasting library designed and pioneered by Facebook, that is claimed to be extremely simple to implement. [2] [3]
- LSTM (Long Short Term Memory): It is a deep learning approach based in Recurrent Neural Networks (RNN) also used to make predictions in time series. [4] [5]

## Problem Statement

Investment firms, hedge funds and individuals have been using financial models to better understand the market behavior and attempt to make projections in order to make profitable investments and trades.

The purpose of this project is to build a stock price predictor for companies in the DJIA, which takes historical stock data over a certain date range as input, and outputs projected estimates for given query dates.

To address the problem a supervised learning approach is going to be taken, at the end it can be seen as a problem of forecasting in time series, where a generalized function is inferred from the existing data which already contains ground truth.

## Datasets and Inputs

There is an open dataset called "EOD data for all Dow Jones stocks" in Kaggle, which contains the historical data for Dow Jones stocks, this data is updated every trading day. This is link to get the dataset: https://www.kaggle.com/timoboz/stock-data-dow-jones

The dataset contains the historical stock price information for the last 5 years, noticing the the market is closed on weekends and holidays. Each company has a total of 1258 observations, multiplied for 30 companies (the ones considered in the DJIA), resulting on 37740 data points.

The dataset provides one spreadsheet per stock, prefixed with the ticker symbol (also known as stock symbol) of a company, each spreadsheet has the following columns:

- `date`: Trading day
- `open`: Opening price
- `high`: Highest price
- `low`: Lower price
- `close`: Closing price
- `volume`: Number of shares traded
- `unadjustedVolume`: Number of shares traded also considering companies adjustments
- `change`: Change of closing price relative to the day before
- `changePercent`: Change in percent value
- `vwap`: Volume Weighted Average Price
- `label`: Formatted version of the date
- `changeOverTime`: Metric considered to measure the changes bases on weighted dates

The value to be predicted is the closing price, which corresponds to the column `close`. Intuitively the companies themselves and their past closing prices would play a role in future prices, but part of the work in this project aims to figure out if the other columns also useful for performing predictions.

The following is a fragment of how the historical stock looks for the thicker symbol APPL (Apple Inc.):

| date | open | high | low | close | volume | change | changePercent | vwap |
|---|---|---|---|---|---|---|---|---|
| 2014-02-21 | 69.9727 | 70.2061 | 68.8967 | 68.9821 | 69757247 | -0.774858 | -1.111 | 69.4256 |
| 2014-02-24 | 68.7063 | 69.5954 | 68.6104 | 69.2841 | 72364950 | 0.302061 | 0.438 | 69.1567 |
| 2014-02-25 | 69.5245 | 69.5488 | 68.4239 | 68.5631 | 58247350 | -0.72101 | -1.041 | 68.9153 |
| 2014-02-26 | 68.7667 | 68.9492 | 67.7147 | 67.9446 | 69131286 | -0.618575 | -0.902 | 68.1373 |
| 2014-02-27 | 67.917 | 69.4457 | 67.7738 | 69.2999 | 75557321 | 1.3553 | 1.995 | 68.8615 |
| 2014-02-28 | 69.4851 | 69.9671 | 68.571 | 69.1121 | 93074653 | -0.187807 | -0.271 | 69.2731 |
| 2014-03-03 | 68.7417 | 69.6913 | 68.6616 | 69.3117 | 59667923 | 0.199626 | 0.289 | 69.1371 |

## Solution Statement

The solution is the implementation of a stock predictor, with the modules described bellow:

- Training interface: Provided through the usage of a Python object. It accepts a date range (i.e. start and end date) and a list of ticker symbols (p.e. JPM, KO, MCD, etc.). It will use a (Machine Learning) model of stock behavior which should be fed and trained using historical prices.
- Query interface: It accepts a list of dates and a list of ticker symbols, the output is the list of predicted stocks prices for each one of the given stocks in the respective dates. The prediction would be based in the closing price on the respective date. The given ticker symbols in the query must be a subset of the ones used in the training, the dates are aimed to be after the date range used for the training.

## Benchmark Model

To test the prediction's accuracy/performance different tests sets are going to be reserved for intervals of 1 day, 7 days, 14 days, 28 days, 96 days, etc. These sets would be used for training data which historical dates are before the intervals reserved for testing.

To benchmark the performance of the different methods/models explored or implemented in this project, they can be compared against an initial naive solution which would be produced via linear regression.

## Evaluation Metrics

To measure the performance of the predictions, the predicted value needs to be compared against the real value in the test and/or validation dataset, the predicted value is a (floating point) numerical value, thus using pure accuracy would not be convenient and not provide a useful description about how well the prediction is performing or improving through training iterations.

Instead, a metric that can provide some sort of average of the total error would be more effective, since the purpose of the prediction is to get as value as close as possible to the real one, but it does not need to be the same to be useful. For this reason the Root Mean Square Error (RMSE) is going to be used as evaluation metric for this project.

## Project Design

The phase one of the project would be the **data collection**, data would be extracted from the source mentioned before which contains the historical prices of Dow Jones. However, experimenting with data from other sources or other stock markets is not ruled out and it could be used if convenient. In particular the "EOD data for all Dow Jones stocks" from Kaggle provides the information distributed in several spreadsheet, thus a task when gathering data would be combining them in a single dataset suitable to be used during training and experimentations.

The phase two would correspond to **data analysis**, here the dataset would be examined to find the parameters that have some impact in the prediction power, technics for dimensional reduction and principal component analysis are intended to be applied, at the end the idea is to select a set of parameters significant enough to be used for the training and prediction processes.

The phase there would be the **research** of the applicable machine learning methods, so far the identified candidates are ARIMA, Prophet and LSTM, however in the way more methods can be discovered and incorporated in the experimentation.

The phase four is going to be the **construction of the software components** to perform the experimentation of the different models. For constructing the components, Python running in Jupyter Notebook it is a suitable platform to provide a friendly environment to experiment and incrementally refine a solution for the

problem. In terms of the hight-level architecture of the system, the following are the general modules to be created and incrementally evolved:

- Dataset Retriever: The dataset used for this project has the property of being continuously (daily) updated, which is an advantage since the methods explored/implemented in this project can have feedback in a quicker way. An important module for this project would be in charge of retrieve the most recent data and leave it ready for the module in charge of training.
- Trainer: Once the data set is available and ready to use this module would be in charge of performing the training given the specific parameters (data range and ticker symbols), as a result it would return an artifact which usage would be necessary to make predictions. Different methods are going to be explored for training, then potentially the ML method can be passed as a parameter to this module.
- Predictor: This would be the module in charge of making the predictions once the training phase is done, this module would be fed by the artifact resulted of training and would be capable to perform the predictions to the validation, test or production sets.

The phase five is planned to be the **experimentation**, where different machine learning methods/algorithms/models are going to be implemented and tested using the software components previously described. It is expected to have several iterations in which the performance will be measured and benchmarked, until a solution is developed which can produce satisfactory results, i.e. at least performing better than the naive solution used to benchmark.

The phase six is going to be **the production of a report** in which the results for the different methods are gathered, and the experiences during the project documented. The plan is to make this report in markdown format to be available in a GitHub repository and able to be converted to PDF.

The phase six would be the conception of the **final solution**, where a method or set of methods that performed satisfactorily would be bundled in a final software solution, this solution would transcend Jupiter Notebook to be incorporated in Python scripts able to run standalone, being the idea to be perform the data collection, training and prediction through the command line.

## References


[1] A. O. A. Ayodele A. Adebiyi and C. K. Ayo, "Stock price prediction using the arima model," *International Journal of Simulation Systems, Science & Technology*, vol. 15, no. 4, pp. 105–111, 2014 [Online]. Available: http://ijssst.info/Vol-15/No-4/data/4923a105.pdf

[2] S. J. Taylor and B. Letham, "Prophet: Forecasting at scale," 2017 [Online]. Available: https://research.fb.com/prophet-forecasting-at-scale

[3] R. Ritz, "Using facebook's prophet to predict mongolian stocks," 2018 [Online]. Available: https://medium.com/mongolian-data-stories/using-facebooks-prophet-to-predict-mongolian-stocks-cdf4feabd558

[4] Z. Pang X. and V. Chang, "Stock market prediction based on deep long short term memory neural network," in *Proceedings of the 3rd international conference on complexity, future information systems and risk (complexis 2018)*, 2018, pp. 102–108 [Online]. Available: https://www.scitepress.org/papers/2018/67499/67499.pdf

[5] A. C. M. P. David M. Q. Nelson and R. A. de Oliveira, "Predicting stock prices using lstm," *International Journal of Science and Research (IJSR)*, vol. 6, no. 4, pp. 1754–1756, 2017 [Online]. Available: https://www.ijsr.net/archive/v6i4/ART20172755.pdf