

# Machine Learning Engineer Nanodegree

## Capstone Proposal

Yuichi Hagio  
July 22, 2017

## Proposal

Stock Price Predictor

### Domain Background

Machine learning can be great at predicting the future from the historical data. A machine-learning algorithm can be more accurate on the prediction than conventional trading strategies based on rules set by humans.

Investment firms have adopted machine learning in recent years rapidly, even some firms have started replacing humans with A.I. to make investment decisions.

### Problem Statement

There is no easy way to predict stock prices accurately and no method is perfect since there are many factors that can affect the stock prices (i.e. people's emotion, natural disasters, etc), but I believe that we can predict whether the closing price goes up or down by applying machine learning techniques and algorithm from the historical data set for this project.

### Datasets and Inputs

There are several data sources for the historical stock price data. I can use yahoo finance data set (.csv format) for this project.

The data includes following properties:

- Date
- Open
- High
- Low
- Close
- Adj Close
- Volume

How different is adjusted close price from close price?

Adjusted close price is the price of the stock at the closing of the trading adjusted with the dividends, and the close price is the price of the stock at the closing of the trading. Both values can be same, or not.

Data set is daily historical prices for 10 years (Jul 24, 2007 – Jul 24, 2017), which is 2518 data set for each stock (2518 days of trading).

90% of the data set can be used for training.

10% of the data set can be used for testing.

Download the CSV file for each (GE, S&P 500, Microsoft, Apple, Toyota)

- GE: <https://finance.yahoo.com/quote/GE/history?p=GE>
- Microsoft: <https://finance.yahoo.com/quote/MSFT/history?p=MSFT>
- Apple: <https://finance.yahoo.com/quote/AAPL/history?p=AAPL>
- Toyota: <https://finance.yahoo.com/quote/TM/history?p=TM>
- S&P 500: <https://finance.yahoo.com/quote/%5EGSPC/history?p=%5EGSPC>

## **Solution Statement**

From the data set of yahoo finance. Predict the closing price of a target day based on the historical data up to the previous day of the target day.

I try to use a kind of Recurrent Neural Network (RNN), called Long Short Term Memory (LSTM) from Keras library for benchmark model. RNN is a deep learning algorithm that has a "memory" to remember / store the information about what has been calculated previously.

LSTM networks have memory blocks that are connected through layers, and it can choose what it remembers and can decide to forget, so it can adjust how much of memory it should pass to next layer.

Since I use the time series of data of stock prices and try to predict the price, LSTM looks good fits for this project.

## **Motivation**

I am interested in using Deep Learning and have never used LSTM and Keras library, so this could be great practice to get my hands on it.

## **Benchmark Model**

As a baseline benchmark model, I can use Linear Regression model, and compare the result with the solution model (Deep Learning – LSTM model). As the metric, I can use Root Mean Squared Error (RMSE) and it will show that less RMSE score indicates better prediction. Also, I can visualize the predictions using a plot or a graph, it will be easy to see the result.

## **Evaluation Metrics**

To determine how accurate the prediction is, we analyze the difference between the predicted and the actual adjusted close price. Smaller the difference indicates better accuracy.

I can use Root Mean Squared Error (RMSE) as a metric to determine the accuracy of the prediction. It is a commonly used general purpose quality estimator.

Also, by visualizing the predicted price and the actual price with a plot or a graph, it can tell how close the prediction is clearly.

## Project Design

I'll probably use following tech stack:

- Python 2.7 (Required)
- Numpy
- Pandas
- Sklearn
- Matplotlib / Seaborn
- Tensorflow / Keras
- Python notebook

### Steps

1: Load Data from CSV file and prepare the data for training and testing

- Load datasets downloaded as CSV from Yahoo Finance
- Plot the datasets (Date / Adjusted Closing Price) to see the actual prices
- Split the datasets into training (90%) and testing (10%)

2: Train / test the model, and visualize log the result

- Build baseline Linear Regression model
- Build solution Deep Learning (LSTM) model
- Print and plot the results

3: Improve / Tune up some parameters for improvement and experiment if it needs

- Tune up parameters to improve or experiment
- Print and plot the results

4: Result / Conclusion

- Analyze the result and compare both models
- Conclude the analysis

References:

- <http://fortune.com/2017/03/30/blackrock-robots-layoffs-artificial-intelligence-ai-hedge-fund/>
- <https://seekingalpha.com/article/4083754-superior-portfolio-roi-artificially-intelligent-algorithms>
- <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- <http://machinelearningmastery.com/time-series-prediction-with-deep-learning-in-python-with-keras/>
- <http://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/>
- <http://www.jakob-aungiers.com/articles/a/LSTM-Neural-Network-for-Time-Series-Prediction>
- <https://www.freelancemap.com/freelancer-tips/11865-trend-prediction-with-lstm-rnns-using-keras-tensorflow-in-3-steps>
- <https://medium.com/@TalPerry/deep-learning-the-stock-market-df853d139e02>
- <http://www.naun.org/main/NAUN/mcs/2017/a042002-041.pdf>
- Machine Learning for Trading: <https://www.udacity.com/course/machine-learning-for-trading--ud501>

- Time Series Forecasting <https://www.udacity.com/course/time-series-forecasting--ud980>