Capstone Project
Machine Learning Engineer Nanodegree

Kevin Robertson
September 18, 2017

Domain Background:

This project originates in the financial industries field. Since the stock market was established in 1817, traders, investors, and everyday people have striven to predict stock prices. With the advent of computers and machine learning algorithms that desire has exploded into a race for the best algorithms.

Predicting stock prices has always been the Holy Grail of the stock market. The New York Stock Exchange (NYSE) is a very powerful institution itself. It has created vast amounts of wealth and destroyed wealth as well. (http://www.investopedia.com/articles/07/stock-exchange-history.asp). Since the NYSE was established and became a dominant force in the world, investors from all over had attempted to predict stock prices. Investing requires a very technical analysis. I believe that an underlying problem with the stock market lies in its unpredictability. For example, if the stock market starts to have a sell off, it can get to a point of a crash. This crash has the potential to wipe out fortunes, retirements, and peoples very livelihoods. I believe with accurate and efficient algorithms, we can predict these downfalls and take action to avoid any type of crash.

The stock market has always been of interest to me. It is a mixture of technical, art, and science rolled up into one big mysterious machine. I am hoping that through this project I can understand a little more about how the market works. There has been much research done regarding stock market prediction, one such approach deals with a recurrent neural network (RNN) architecture called long short-term memory (LSTM). LSTM has emerged as a very effective model for predicting prices. Since the stock market is very complex owing much of its volatility to the emotions of the investors as well, it needs a complex model to represent its intricacies within the system. [1]

Problem Statement:

In this project I will strive to predict stock prices. The problem is it is very difficult to predict future prices of stocks. Stocks are very tangible things. You have prices that go up and down, and this data is available everywhere but the problem and solution seem to be very difficult and it is definitely not an easy prediction to make. Before I have even begun the project I have an intuition that the Volume will play a big role in the data and the ending result. The method with which I will structure this project is regression. The inputs for this will be the normal stock market features, opening price, closing price, highest price during the day, lowest price during the day, volume, in addition to some derived features such as the sharpe ratio and possibly Bollinger bands to determine boundaries for the model.

Datasets and Inputs:

The dataset is the S&P 500 stock dataset found on kaggle.com. It consists of historical data from the last 5 years about all S&P 500 companies. The data I have was gathered through a script found here: https://github.com/CNuge/kaggle_code. The dataset includes 7 features, Date, Open Price, High Price, Low Price, Closing Price, Volume of trades, and the name of the

company. There may possibly be other derived features such as the Sharpe Ratio. The label I will be trying to predict will be a combination of the stock symbol with price, and perhaps a date. There are 600,000+ records in the dataset.

The dataset I have contains three different areas of data. The first is all the stock data for the past 5 years from the 500 companies in the S&P 500. The second file only contains the last years worth of data. The final folder contains a file for each of the different companies in the dataset. As for splitting the data during the pre-processing step I will strip out the last 25% of each company's data in order to test the model to make sure it works. This will be organized by date so as to represent the real world. The validation set will consist of the 25% remaining for each company, and the 50% left will be the training data. This comes to about 315 records for each company for testing, 315 for validation, and about 630 for testing.

Solution Statement:

My objective with this model is to predict the prices of the individual stocks based upon the specific stocks data, my secondary objective will be to predict the overall stock price with training based upon all the stocks. I'll start with choosing one stock in which to predict based upon its individual characteristics, to predict future stock prices. My approach will be to graph the data and use a supervised regression model such as linear regressions to get a preliminary idea of what the data looks like. I will perform this analysis on a few different stocks to compare the different data. One technique I wish to try also is a Support Vector Machine (SVM). This technique has been of interest to me since I learned about it and I would like to push it further to see where it can possibly lead, I also want to see if there is a failure in this type of model.

After going through a preliminary phase with the pre-processing, I am going to put the data through a RNN-LSTM model. Neural networks have been a huge asset to many big companies such as Microsoft, Google, IBM, and Apple. Each are in a race to push the boundaries of what neural networks can do.

Benchmark Model:

Since stock market prices are difficult enough to predict, many people have developed their own methods for doing so. Unfortunately, most of those methods do not work otherwise there would be a lot more rich people. It seems as if the best prediction model is random guessing, because adding in a human element can make the prediction worse when you add in human emotion. I believe the best benchmark model is testing against random guessing. Random guessing in this instance will be in the form of a computer randomization response.

Of course one of the best benchmarks would be to test against the market itself. The whole point of predicting stocks is to see if it is correct plus it could be very lucrative.

Evaluation Metrics:

After some research and some guidance from a reviewer I've determined that the best evaluation metric for preliminary linear regression is Mean Absolute Error (MAE). I decided on this after looking at a few other metrics, and noticing that the MAE is used in regression analysis as well as for time series analysis. My main objective for the metric was to find an accuracy

assessment for both the regression and the time series to be able to compare them more easily. From the Kaggle website MAE is:

> In statistics, the mean absolute error (MAE) is a quantity used to measure how close forecasts or predictions are to the eventual outcomes. The mean absolute error is given by
>
> $$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| = \frac{1}{n}\sum_{i=1}^{n}|e_i|\,.$$
>
> Where
>
> $$AE = |e_i| = |y_i - \hat{y}_i|$$
>
> $$Actual = y_i$$
>
> $$Predicted = \hat{y}_i$$
>
> From: [3]

Project Design:

I will begin by pre-processing data.

This step will include adjusting invalid values such as the infamous NaN value that is common in datasets. I will also begin with some charts and graphs to visualize the data further and what types of paths it takes, if any. As part of the pre-processing step after the visualizations I will run the data through a linear regression model in order to get a preliminary analysis and possible accuracy in which to compare the final model with, as a control benchmark in addition to the, “in the wild” type of data of the actual stock market.

After establishing the linear regression model, the next step is to layout the RNN-LSTM and perform a test run of the data. The test run will help to establish a baseline with which to optimize the model and the different parameters involved. It will also allow for a reevaluation of the data and my choice of model. At that point in the model I’ll be able to judge the appropriateness of the machine learning results and determine whether I should continue on with the optimization of the parameters or end the analysis as a failure and learning experience.

References:

[1] Q. Zhuge, L. Xu, G. Zhang. LSTM Neural Network with Emotional Analysis for Prediction of Stock Price. *Engineering Letters,* 2014, 25:2, EL_25_2_09. http://www.engineeringletters.com/issues_v25/issue_2/EL_25_2_09.pdf

[2] C. Nugent, Kaggle Code, (2017), GitHub repository, https://github.com/CNuge/kaggle_code.

[3] Kaggle.com, https://www.kaggle.com/wiki/MeanAbsoluteError, (2017).