Machine Learning Engineer Nanodegree

Capstone Proposal

Sandeep Paulraj

December 27th, 2017

Proposal

Proposal for stock price estimator.

This proposal also has an associated ipython notebook stock_price_estimator_proposal_v1.1.ipynb that has the initial data exploration and setup.

`stock_price_estimator_proposal _v1.1.ipynb`

Domain Background

Fundamentally, owning stocks is a way to grow wealth. Usually investors buy stocks after some due dilligence and research. If the stock price goes up, the investor has the option of selling the stocks and making a profit. At the same time, several companies shell out dividends to shareholders if they own stock. This too is a way to accumulate wealth. However, traditional buy and hold techniques are no longer in vogue these days. Retail investors find it very tough to enter a particular stock since stock prices do vary and large institutional investors hold sway. High Frequency trading results in significant stock price variation within a trading day. Also fundamental research though still very important is kind of taking a back seat to algorithm based trading.

Machine learning technques are being used to make investment decisions. Machine learning techniques have carved out a niche for themselves in various domains and especially those domains where there is a plethora of data. Stock Prices are a very appealing domain since they provide several stocks and large amounts of data. Over the last few years we have also started hearing more about several hedge funds and investment banks beginning to use these techniques. I intend to investigate some of these techniques to predict stock prices.

To be precise, i would like to be able to predict Broadcom(AVGO) stock price. This company has been in the news lately for its takeover attempt of Qualcomm. Broadcom is also a very large Apple Supplier. Also owning Broadcom stock myself, i want to gauge if i can come up with a model to better gauge future price movement based on available data. As part of this project, i also intend to investigate if Apple stock price movement has an effect on Broadcom stock price. The VanEck Vectors Semiconductor ETF(SMH) is essentially a basket of several semiconductor companies and I also intend to investigate if the sector has an

effect on Broadcom stock price. It is conceivable that an increase in volume of either AAPL stock or the SMH ETF has a positive correlation with Broadcom stock price.

In the Datasets and inputs section below, i show the source of the data and provide links.

Some papers which discuss machine learning techniques are sited below.

Predicting Long term price Movement

Machine Learning for predicting Bond Prices

Problem Statement

Stock Prices fluctuate from day to day and to be precise, fluctuate by the second. Using publicly available data stock price data, i will attempt to predict the adjusted close price of the stock for the following trading day. If we are able to gauge the closing stock price, we might be able to make smart trading decisions based on the predictions. By giving a start date and a finite set of following trading days(in other words a range of trading days), it should be possible to predict the following days adjusted closing stock price. Basically we have a range of trading days and we have to predict the closing adjusted stock prices of the next day immediatley after this range. This is a regression problem and NOT a classification problem.

Datasets and Inputs

It is possible to obtain stock price financial data from various sources. It is also possible to use python api and the yahoo finance library to obtain this data. For some reason, i am having trouble installing the yahoo finance library in python 3.6. So I have decided, to obtain the csv data from the yahoo finance website and read in the dataframe using pandas.

I will obtain Broadcom stock, Apple stock and SMH ETF price data from yahoo and the source is below.

Broadcom(AVGO) stock price

Apple(AAPL) stock price

VanEck Vectors Semiconductor ETF(SMH)

Clicking on the above links will also show the data i will be leveraging. I intend to use 1 year worth of data. During analysis, if I find myself needing more data, it is simple to obtain more data from yahoo by getting 5 years worth of data. The individual pieces of information that i will leverage for each day will be “Open”, “High”, “Low”, “Close”, “Volume” and “Adjusted Close”. 1 year worth of data will provide 252 data points. 5 years worth of data will result in

approximately 1250 data points. The data range of the dataset will be between December 26, 2016 to December 26 2017.

What i will attempt to predict is the adjusted stock price 1 day ahead.

We are dealing with time series data and data has to be handled chronologically. Also the stock price data is continuous is nature.

For the initial exploratory analysis, I read in the data and realize that the dates increase, i.e in the csv file and data frame February 1 will come before February 2. The first thing I do in my notebook and analysis is to reverse this order. I would like to explain the reversal of order: This is done since while visualizing my data, I prefer to see the dataframe with the more recent date on top. I depict this in my data frame that can be seen in the ipython notebook.

In the associated ipython notebook I also have another column where I store the difference between the highest and lowest daily stock price. This may prove to be useful in my analysis as this provides a daily trading range.

It is important to use standard pandas routines to set up the dataframe. This essentially will result in a more elegant and cleaner final solution. Please take a look at the accompanying notebook to look at all the exploratory analysis

My capstone review provided me with two good sources of how to avoid lookahead bias. I am adding those links here.

9 Mistakes Quants Make

Avoiding Look Ahead Bias in Time Series Modelling

Solution Statement

We are dealing with time series data. Also we fundamentally have a regression problem. This is not a classification problem. We have to predict an actual adjusted stock price; not whether the stock goes up or down. We have to predict the next day's closing stock price. So let us take an example. Say we need to predict the following trading day's closing stock price. We will have various inputs that are available to use such as trading volume, opening price, high price, low price. With this we can use regression techniques to predict the following day's closing stock price. Now, we have already setup our data to know the following day's closing stock price. Thus we will have both actual closing stock price and predicted stock price based on our model. With this we can gauge how well our model is behaving. It is intended that the model will predict stock price withing a +- 5% range.

Benchmark Model

As suggested in the capstone proposal review, I will train and test an out-of-the-box linear regression model on the project data. My final solution should outperform this linear regression model. By using a linear regression model as the benchmark, and training/testing it on exactly the same data as my final solution I will be able to make a clear, objective comparison between the two.

Evaluation Metrics

I will be leveraging sklearn in this project. From sklearn metrics we will have access to an array of metrics from our model. The main evaluation metric I will be using is the root mean squared error. The root mean squared is simple to calculate and can be calulated as shown below. Based on previous projects and experience, i don't think a metric exists to calculate this for us. This has to be derived and is simple to derive as can be seen below.

```
from sklearn.metrics import mean_squared_error
from math import sqrt

rms = sqrt(mean_squared_error(y_actual, y_predicted))
```

Now the difference between actual and prediction prices can be positive or negative. Hence it is important to take the square of the difference. We then have to take the mean of these squared values and finally take the square root.

Project Design

As mentioned previously my project proposal is associated with an ipython notebook.

```
stock_price_estimator_proposal _v1.1.ipynb
```

The above notebook will be a starting point for my final project as well.

Practically speaking a lot of factors go into a stock price. On any particular day, stock price will be somewhat based or related to the index that it is listed on. Then the index itself may and in all likelihood depend on the closing price of other indices. Let us take an example. Say the US has a good trading day and the markets are positive in Asia starting in Japan. All is well until say some bad data originates in Europe. Europe turns negative and this results in a negative start for US the next day. This feeds into a company stock price. Hence, depending on my intermediate trial runs, i will append my training data with data pertaining to Apple and SMH.

After obtaining my data I intend to follow these steps.

- Reverse the rows since as an example February 1 comes before February 2 in the downloaded data. I want it to be the other way round for analysis purposes. To clarify again, the reversal of the rows is just a matter of convenience while depicting the data. I prefer to see recent dates on the top.
- Append a column with trading price range (High - Close)
- Add the following day's stock price. This will be what we are trying to predict.
- Depending on results of various models/scenarios I may and in all likelihood have to append data pertaining to Apple and SMH to the Broadcom data.

I will be leveraging both pandas and numpy in my project.

I will try Linear Regresssion and it is possible for a simple model to provide good results. However, I intend to try other regressors such as SVR(Support Vector Regression), Decision Tree Regressor and Random Forest. Random Forest is a time series algorithm implemented in time series forecasting. Please see a citation below.

Stock Price Prediction using Random Forest

As an example, some of the paramters I will be tuning is the kernel to be used along with Support Vector Regression. Some of the kernels that can be used are 'linear', 'poly', 'rbf' among others. The kernel co-efficient gamma can also be tuned.

My capstone proposal review provided me with very valuable feedback on Cross Validation with Time Series data. I am adding the links provided here.

Cross-validation for time series

Pythonic Cross Validation on Time Series

Using k-fold cross-validation for time-series model selection

sklearn time series split

sklearn: User defined cross validation for time series data

LTSM(Long Short term Memory) network is a type of Recurrent Neural Network. LTSM's have given good results with Time Series Prediction. Hence, I intend to try out LTSM for predicting stock prices in my project. Please see video below.

LTSM Neural Network for Time Series Prediction

For LTSM, I will be using Keras. When I compile the keras model I will have the option to choose an appropriate loss function and optimizer. Again, since RMSE is not standard in keras as well, i will have to code it up myself.

After running various scenarios, I will look at the metrics to gauge how well each model is doing and gauge which is the best model to use for stock price prediction.