Shib Shankar Golder, July 10th, 2018

Stock Price Prediction Using Machine Learning

Domain Background

In today's era, individuals look for investment ideas for building more capital. Surprising, more and more folks get attracted toward stock market for investing their money. But, many of them have a very little or no knowledge on this field. They get attracted towards the dynamicity and consider it as a gamble to earn quick money, some have understanding on the technical side of it and love to play/trade the swing and remaining are more of an institutional investor who believes on company's fundamental and hold it for long term.

Above is the monthly chart analysis of SPY (S&P500), the market is bullish from Aug to Jan but suddenly on Feb a big drop, it become bearish and then started swinging up and down.

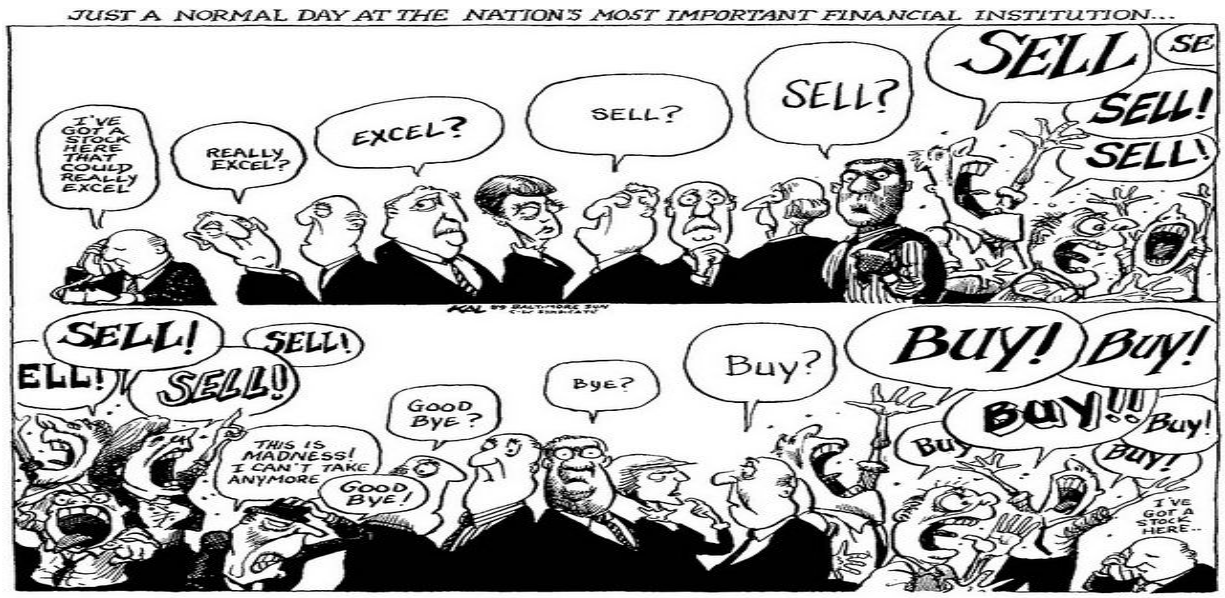
- Why suddenly a big drop happened?
- What triggered this drop?
- What is the trend?
- Can it be predicted in advance?
- Can machine leaning be leveraged to measure the stock price?

Academia Research paper on this subject -

- Ritter, Gordon, Machine Learning for Trading (August 8, 2017). Available at SSRN: https://ssrn.com/abstract=3015609
- http://etd.lib.nsysu.edu.tw/ETD-db/ETD-search/view_etd?URN=etd-0012117-194528
- https://dspace.mit.edu/bitstream/handle/1721.1/105982/965785890-MIT.pdf

Some well-known quotes:

Warren Buffett, *"Risk comes from not knowing what you are doing."*
Peter Sondergaard, *SVP Gartner, 2011: "Information is the oil of the 21st century, and analytics is the combustion engine."*



Humans are inefficient to process all these information flooding from different channels by their own. That's why machine is utilized to handle and process these data with ease. All the top-notch financials companies, Hedge fund corporation, 401K asset Managers investing big time on Analytics, AI and building models for Machine learning for predicting and solving the big puzzles and finding the answers of above questions.

Problem Statement

Our problem statement is simple yet complicated, I am trying to predict the future closing price of a stock to reduce the risk of financial loss. As the stock data is continuous and we are trying to predict the price, hence, this is a classic example of regression
i.e. $y = a0 + a1*x$

In higher dimensions when we have more than one input (x), the line is called a plane or a hyper-plane. The representation therefore is the form of the equation and the specific values used for the coefficients (e.g. a0 and a1 in the above example).

In this case, y → Stock Closing Price (final label) and xi → The different technical (Adj.Open, Adj.High, Adj.Close etc.) and fundamental (sentiment, newsbuzz, etc.)

I will try to use RNN (Recurring Neural Network) namely LSTM (Long Short Term Memory) and GRU (Gated Recurrent Unit) to see which one performs best. I will build models using Keras library and try to predict the stock price for most popular FAANG stocks (i.e. FB, AMZN, AAPL, NFLX & GOOGL). Our models will be trained based on historical stock data from Quandl data sets. Will add fundamentals and sentiment analysis to the model for better optimization. Compare the results of both the models, visualize the results and conclude based on the findings.

Datasets and Inputs

I will use the Quandl, Kaggle or google finance dataset for sake for simplicity and ease of use. Quandl python API can be effectively leveraged to load the daily price from Jan 2000 to latest available. These data are very much clean and preprocessed time series data meant for machine learning and making predictions. Quandl also support stock sentiment analysis with fewer sample to play with. I will try to feed this stock sentiment polarity while processing the data. These data are continuous time series data. Quandl support various stocks, ETF etc. as dataset example. My focus would be use FAANG stock for the model. My favorite is Amazon stock. But, will decide the final during the project work.

There are several features which can be leveraged in real work but for simplicity, I will focus on few core inputs namely "Adj. Open", "Adj.High", "Adj.Low", "Adj.Close", "Adj.Volume", "Sentiments" etc. Then, using these input, I will derive few more fundamental attributes for more better accuracy. The data would be divided into 60/20/20 ratio for the training, validation and testing set and the data would be divided chronologically based on time series.

Choose some date then divide the whole set into these 3 buckets or leverage the below idea

A better way to handle the data is to initially choose a small subset of the oldest data to act as your training set. Your model is trained on it and then predicts the next sequential time point. This point is then added to the training data on a rolling basis and the oldest training point is dropped. This method allows you to have a large number of data points for validation/testing and it ensures that the model is always trained on the most up-to-date information before each prediction. Since the data is always handled sequentially, there is no possibility of look ahead bias in the model.

Data source: (Will decide the final data source during the project implementation)

- https://www.kaggle.com/borismarjanovic/price-volume-data-for-all-us-stocks-etfs
- https://docs.quandl.com/docs/python-time-series
- www.google.com/finance/

Solution Statement

Based on my analysis and research, I will try the evaluate the performance of model architecture's using LSTM and GRU. The whole project would be done using Anaconda 2.7 python, more specifically Jupiter notebook for the sake of simplicity. I will leverage Keras for the Tensor flow, Pandas for handling the time series stock data and NumPy for type of transformation. Both the model performance will be measured based on the predicted stock price than the actuals.

Benchmark Model

As stock data is a continuous time series data set and we are trying to predict the price, hence, I decided to use a regression as a benchmark model.

I will use Linear Regression or similar as the benchmark model and then we compare the improvement done using RNN improved model as the final solution. My main aim is to showcase how much improvement is achieved by the RNN model.

Since this is a regression problem, I will use MSE and RMSE to compute the error rate.

Evaluation Metrics

In this project, the model evaluation metrics will be measured using Mean Square Difference & Root Mean Square ($RMSE = \sqrt{\sum{(Y_{actual} - Y_{predicted}) ^2} / n} $) for the predicted verses the actual. And, then both GRU and LSTM model would be compared based on the accuracy than that of the Regression model.

Project Design

I am planning to leverage RNN based model to approach this problem.

Below is the project work flow.

- **Project Setup & Environment**
  - Anaconda python 2.7 based ipython Jupiter Notebook.
  - Github repo
  - ML libraries like Keras, Tensor flow, NumPy, Pandas, Matplotlib etc
- **Data Preparation**
  - Use Quandl API to load the Stock data as it is mostly clean
  - Load the Stock sentiment from Quandl
  - Focus on the FAANG stock for analysis.
  - Data normalization
  - Will leverage Pandas data frame for the data.

    - 60/20/20 Data split for preparing the Training and Testing set or the other aforesaid rolliung approach.
- **Benchmark Model**
    - Creation of the model based on Linear Regression.
    - Add the technical params for prediction.
- **RNN Model**
    - Prepare 2-layer GRU Model to start with.
    - Prepare 2-layer LSTM Model to start with.
    - Feed the necessary Technical param.
- **Optimized RNN Model**
    - Change or add more inputs.
    - BachNormalization for optimization
    - Dropout for regularization
- **Metrics & Result**
    - Visualize the results of Actuals, benchmark and the RNN Models.
    - Analyze and the summarize the result based on the performance and metrics