Machine Learning Engineer Nanodegree

Capstone Proposal

Building a Stock Price Predictor

Andrae Delisser
September 24, 2017

Domain Background

A vast number of studies have been done on how to predict the future prices of stock to provide for the optimal decision-making in trading for investors. One of the first documented method of predicting future prices was in commodities namely rice; that method is called the candlestick charts. The   Candlestick charts are thought to have been developed in the 18th century by Munehisa Homma, a Japanese rice trader of financial instruments. [1] This would be regarded as technical analysis as it dealt only with the price movements and not necessarily the underlying causes of the moves. The contra to this method is fundamental analysis; a method of evaluating a security in an attempt to measure its intrinsic value, by examining related economic, financial and other qualitative and quantitative factors.[2]

In modern times with the advent of computers and the speed and power of processing complex calculations that it provides has increased the probability of a system that could be efficient in predicting stock price movements at least to some acceptable level of accuracy.   Recently some systems using machine learning methods such as artificial neural network achieved better performance than those using only conventional indicators. For this project most the data require classification for that ensemble method of applying algorithms can be quite useful.

.

Machine learning has been used to predict possible future stock prices and also as stock screeners. Having work in Financial Advisory for number of years I have had the desire to analyze a large set of stocks and be able to screen for potential buy and sell opportunities. In this Capstone project I will be Analyzing the Russell 3000 screening for potential stock buys and then using machine learning to predict the price of the Index itself.

Problem Statement

For this project, the task is to predict the Adjusted Close Price of a stock, but for learning purposes and efficiency we will be looking at predicting the Russell 3000 index itself. I will be using five values illustrating movements in the price over one unit of time. The inputs would be trading data for the Russell 3000 over 5 day period, 1 month and 1 year time period The key trading indicators are :

- Open: The starting price for a given trading day.
- Close:The final price on that day.
- High: The highest prices at which the stock traded on that day.
- Low: The lowest prices at which the stock traded on that day.
- Volume: The total number of shares traded before the market is closed on that day.

From these data I will generate new features associated with the price trend by computing ratios between each pair of average price in three different time frames. Example, the ratio between the average price over the past week and over that past year. The volume is another important factor that investors analyze. Similarly, we can generate new volume-based features by computing the average volumes in several different time frames and ratios between each pair of averaged values.

Stock volatility will be taken into consideration as well and as such we will use a standard deviation indicator; in this case we will use standard deviation of close prices in a particular time frame as well as standard deviation of volumes traded. The moving average of returns will also be an indicator I will employ.

Datasets and Inputs

The FTSE Russell 3000 Index .The Russell 3000 Index is a capitalization-weighted stock market index, maintained by FTSE Russell, that seeks to be a benchmark of the entire U.S stock market. It measures the performance of the 3,000 largest publicly held companies incorporated in America as measured by total market capitalization, and represents approximately 98% of the American public equity market. As of 31 March 2017, the stocks of the Russell 3000 Index have a weighted average market capitalization of almost $140 billion; the median market capitalization is nearly $1.7 billion. The index, which was launched on January 1, 1984, is maintained by FTSE Russell, a subsidiary of the London Stock Exchange Group. [3]

I will acquire stock index price and performance data via the Quandl Python API(https://quandl.com/tools/python)

I also be using Yahoo! Finance to obtain historical price data of the index.

Example:work_data=quandl.get(“YAHOO/INDEX_RUA”, start date=”20014-12-01”, end_date=”2017-08-31”)

I will select the index price for the FTSE Russell 3000 for the period August 31, 2016 to August 31,2017. The feature it will have will include: Open price, Close price , High price, Low, price , Volume and Adjusted Price. The generated features will be: moving average prices and volume, standard deviation of prices and volume.

Solution Statement

For this project project I will be using the powerful tool of pandas and its functionalities to analyze the price data. I will then be able to assign a portion of the data for training and testing. I will be using three Machine Learning algorithms: Linear Regression, Random Forest and Support Vector Machine. The Ensemble method is to create a scenario where we get the best result possible the SVM will then even further sharpen this result.

I will be generating 31 sets of features along with the original features; Open, Close , High, Low, Volume and Adjusted close. The 31 sets of features will include the the average price over past 5 days, months and year and the ratio between these time period prices. Price movements represent sentiment and also according to the **efficient-market hypothesis** (**EMH**) a theory in financial economics that states that an asset's prices fully reflect all available information.[4] The average and ratio of the volumes between the time frames will also be a set of feature. By knowing the total volume on a day, you can understand the power of influence on a given stock. The greater the volume, the greater the influence for the price to change. This allows us to identify accumulation and distribution days on a stock chart which can be used to identify current momentum and predict future price movements.[5] The volatility of stock prices are a key indicator for predicting stock prices, I will therefore be examining the volatility ratio(in this case the standard deviation) between the various time frames. The final feature(indicator) will be the Moving average of the returns between the three time frames. Return is the percentage of gain or loss of close price for a stock/index in a particular period. The most effort will be spent on these features as they will be needed for solving the problem.

Benchmark Model

As benchmark for my model I will be using a A naive forecast. A naive forecast is simply the most recently observed value. In other words, at the time *t*, the *k*-step-ahead naive forecast (Ft+kFt+k) equals the observed value at time *t* (ytyt). Ft+k=yt.[6] In this case a k-means algorithmic model will be used to check the MSE and RMSE results.

Evaluation Metrics

For this project the MSE/RMSE and R^2 of the prediction versus the actual result will be a good representation of how good the models are in being accurate. The MSE the smaller the value the better the regression model. The R^2 indicates the goodness of the fit of the regression model. It ranges from 0 to 1, meaning from no fit to perfect prediction.

The RMSE is calculated mathematically as such[7]:

$$\mathbf{RMSE}_{fo} = [\sum_{i=1}^{N} (z_{f_i} - z_{o_i})^2 / N]^{1/2}$$

Where:

- Σ = summation (“add up”)
- (zfi – Zoi)Sup>2 = differences, squared
- N = sample size.

And the R^2 is calculated mathematically as such:
Step1. Find the correlation(r)

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[n\Sigma x^2 - (\Sigma x)^2][n\Sigma y^2 - (\Sigma y)^2]}}$$

Step 2: *Square the correlation coefficient.[8]*

Project Design

Step 1 Feature Engineering

The first step is to do the feature engineering, here we will be establishing what the predictive variables are. The principal factors used to predict the future prices of the RUA; the Close prices, this includes historical and current open prices and historical performance (High, Low, Volume).

Predicting close price with only these four indicators does not seem promising, and might lead to underfitting. So we need to think of ways to add more features and predictive power. In, machine learning, feature engineering is the process of creating domain-specific features based on existing features in order to improve the performance of a machine learning algorithm. [9]

To ensure underfitting does not occur I will be generating 31 sets of features along with the original features; Open, Close , High, Low, Volume and Adjusted close. The 31 sets of features will include:

- The average price over past 5 days, months and year and the ratio between these time period prices. Price movements represent sentiment and also according to the **efficient-market hypothesis** (**EMH**) a theory in financial economics that states that an asset's prices fully reflect all available information.
- The average and ratio of the volumes between the time frames will also be a set of feature. By knowing the total volume on a day, you can understand the power of influence on a given stock. The greater the volume, the greater the influence for the price to change. This allows us to identify accumulation and distribution days on a stock chart which can be used to identify current momentum and predict future price movements.
- The volatility of stock prices are a key indicator for predicting stock prices, I will therefore be examining the volatility ratio(in this case the standard deviation) between the various time frames.
- The final feature(indicator) will be the Moving average of the returns between the three time frames. Return is the percentage of gain or loss of close price for a stock/index in a

particular period. The most effort will be spent on these features as they will be needed for solving the problem.

Step 2.

Data acquisition and feature generation

I will acquire stock index price and performance data via the Quandl Python API(https://www.quandl.com/tools/python).

I also be using Yahoo! Finance to obtain historical price data of the index.

The Output will be a pandas dataframe object. We will acquire pandas from(ttp://pandas.pydata.org) , this will be used simplify the data analysis on relational or table like data.

Step 3

Define a function to generate features : def generate_features(df) #Generate features for a stock/index based on historical price and performance. Will pass as arguments to df: data frames with columns “Open”, “Close”, “High”’ “Low”, “Volume”, Adjusted Close”. Will then create other functions to handle the data for the features that will be generating such as the standard deviation of of the prices over past week and standard deviation of volumes of past week et al.

Step 4
We will be using three machine learning algorithm namely , Linear Regression, Decision Tree Regression(Random Forest will be used) and Support Vector Machine algorithm(for this case SVR). The Linear regression will explore the linear relationship between observations and target variables and the relationship is represented in a linear equation or weighted sum function. The linear model will be learned from the training data , with the goal of minimizing the estimation error defined as mean squared error(MSE); MSE measures the average squares of difference between the truth and the prediction. The Random forest will be the ensemble learning method that operates by combining multiple decision trees that are separately trained. It will assign the average of regression results from all decision trees to the final decision.[10]

The Support Vector Classification seeks an hyperplane that best segregates observations from different classes.[11] I will be using the the SVR package from scikit-learn. Implementing via : ‘from sklearn.svm import SVR’.

Step 5

Evaluating Regression performance

We mentioned we will me using the MSE , but the square root of the MSE will also be taken. This yields the root mean squared error(RMSE); this will convert the value back to the original scale of the target variable being estimated. We will use the functionality provided by Matplotlib to graphical represent the result the results of the three algorithm used and the actual values of the RUA index over a particular period of time.

References:

[1] https://en.wikipedia.org/wiki/Candlestick_chart

[2] http://www.investopedia.com/terms/f/fundamentalanalysis.asp

[3] https://en.wikipedia.org/wiki/Russell_3000_Index

[4]https://en.wikipedia.org/wiki/Efficient-market_hypothesis

[5]https://www.stocktrader.com/2006/03/21/volume-and-its-meaning/

[6]http://uc-r.github.io/ts_benchmarking#naive

[7]http://www.statisticshowto.com/rmse/
[8]http://www.statisticshowto.com/what-is-a-coefficient-of-determination/

[9]**https://www.packtpub.com/mapt/book/big_data_and_business_intelligence/978178355**

[10]http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[11]http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html