Machine Learning Engineer Nanodegree

Capstone Proposal

Dipak Majhi
June 14th, 2017

Proposal

Domain Background

Investment firms, hedge funds, and automated trading systems have used programming and advanced modeling to interact with and profit from the stock market since computerization of the exchanges in the 1970s. Whether by means of better analysis, signal identification, or automating the frequency of trades, the goal has been to leverage technology in order create investment systems that outperform alternatives.

I wanted to use my Machine Learning knowledge to solve a real problem. Since, I have always been interested in investment and trading, I chosen to work under the domain "Investment and Trading" for my Capstone project and Build a Stock Price Indicator to see if it could help in predicting stock prices in the short-term future. For this Capstone Project, I took help from Udacity's courses – "Machine learning for Trading" by Georgia Tech and "Time Series Forecasting".

Problem Statement

Build machine learning models that learn from historical stock price attributes and predict the stock price on a future date (any day after the last training date), I am only predicting the future Adjusted Closing pricing. Since the target variable is a continuous value so this is a regression task.
Building few models such as Linear Regression, KNN Regression, ARIMA which learn from historical stock data, attributes such as: Open, High, Low, Close, Volume, Adjusted Closing and "Adjusted Closing price" n days ahead in future.

Datasets and Inputs

I have downloaded daily historical stocks data from Yahoo Finance [link] between 12-05-2011 and 12-02-2016 for the following stocks along with SNP_500(American Market index, I want to see how these stock trend with respect to market trends explained by SNP_500). ‘FACEBOOK’ went public around mid of 2012, so we don’t have data for ‘FACEBOOK’ prior to that.

The following stocks are considered: [link]

'GOOGLE', 'AMAZON', 'GOLD', 'APPLE', 'FACEBOOK', 'MICROSOFT', 'GENERAL ELECTRIC'

Each of this stock has the following attributes:

Date, Open, High, Low, Close, Volume and Adj Close

There are a total of 1258 data points for all stocks except for ‘FACEBOOK’ where we have a total of 1144 data points.

Solution Statement

The solution would be to build a Python based machine learning regression model for each stock provided by the user, where each model learns from the historical data of the stock between user provided start date and end date, and predicts the ‘Adj Close’ price for the user provided future date which must be greater than the end date. The data between the user provided ‘Start date’ and ‘End date’ would be divided into two parts in the ascending order of the dates, first 70% would be kept for training and the rest 30% for testing. The machine learning regression model must have a R2 score of 0.85 or above on the trained dataset and R2 score of 0.6 and above on the test dataset. Additionally, on an average the predicted prices from all the models must be more or less within +/- 5% of the actual stock prices on the test dataset.

After the model is built on the historical data, it would predict the Adjusted Closing price for a date (only one date) which is n days ahead in future from the last date of training. Please note splitting the datasets has to be done in the ascending order of dates manually, and not using the SKLEARN’s test train split module which randomly split the data. In the stock price prediction world, we must not train a model on future datasets and try to predict the stock prices for past dates (Source - “Machine learning for Trading” Udacity course)

I am also using Non-seasonal ARIMA (is a class of statistical models for analyzing and forecasting time series data) based model. This model is based on three terms:

- AR –Autoregressive – (p), A model that uses the dependent relationship between an observation and some number of lagged observations.

- I – Integrated – (d), The use of differencing of raw observations (e.g. subtracting an observation from an observation at the previous time step) in order to make the time series stationary.

- MA – Moving Average – (q). A model that uses the dependency between an observation and a residual error from a moving average model applied to lagged observations.

Parameters p, d, and q are non-negative integers, p is the order (number of time lags) of the autoregressive model, d is the degree of differencing (the number of times the data have had past values subtracted), and q is the order of the moving-average model. Three items should be considered to determine a first guess at an ARIMA model: a time series plot of the data, the ACF, and the PACF.

AR model: Identification of an AR model is often best done with the PACF**.** the theoretical PACF "shuts off" past the order of the model. The phrase "shuts off" means that in theory the partial autocorrelations are equal to 0 beyond that point

MA model: For an MA model, the theoretical PACF does not shut off, but instead tapers toward 0 in some manner. A clearer pattern for an MA model is in the ACF. The ACF will have non-zero autocorrelations only at lags involved in the model.

Udacity Free Course: Time Series Forecasting:

- https://www.udacity.com/course/time-series-forecasting--ud980
- https://onlinecourses.science.psu.edu/stat510/node/62
- https://onlinecourses.science.psu.edu/stat510/node/49
- https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average
- http://www.inertia7.com/projects/time-series-stock-market-python

Benchmark Model

The benchmark scores for the models: R2 Score of 0.6 on test data (unseen data) and a R2 Score of 0.85 on train data. Additionally, on an average the predicted prices for all the models must be more or less within +/- 5% of the actual stock prices.

Evaluation Metrics

R2 score [link] on the train and test datasets are calculated to measure the performance of the model. Best possible score is 1. Besides R2 score, I am also calculating the variation of the predicted stock price compared to the actual prices in the test dataset. R2 Score is used to test the performance of all three models – Linear Regression [link], KNN Regression [link] and ARIMA [link]

In statistics, the coefficient of determination [link], denoted R2 or r2 and pronounced "R squared", is a number that indicates the proportion of the variance in the dependent variable that is predictable from the independent variable. The coefficient R^2 is defined as (1 - u/v), where u is the regression sum of squares ((y_true - y_pred) ** 2). sum () and v is the residual sum of squares ((y_true - y_true. mean ()) ** 2). sum ().
Best possible score is 1.0 and it can be negative (because the model can be arbitrarily worse). A constant model that always predicts the expected value of y, disregarding the input features, would get a R^2 score of 0.0.

Project Design:

I would start-off with various financial analysis of the historical data such as plotting a comparative analysis of the various stocks prices, plotting the normalized stock prices, plotting daily returns, plotting Bollinger bands, calculating cumulative returns, creating portfolio stocks, plotting portfolio statistics, Optimizing the Portfolio by optimizing the Sharpe ratio, scatter plots. Also, co-relation between my portfolio daily return and SNP_500 daily return, Beta value to see how reactive is my optimized portfolio compared to the market, alpha value to see how well it performs compared to the market.

The next part would be to build machine learning models to predict 'Adj Close' price for a future date as discussed above. Please note that the reason for doing the financial analysis on the historical is to kind of depict that we can do the same analysis on the Predicted Stock prices. Although the goal of this project is just to predict the 'Adj Close' price for a stock for a particular future, to build a real investment strategy we need to perform financial perform analysis on the predicted stocks such as maintaining an optimized portfolio.

Workflow stages:

Generally, a solution workflow goes through seven stages :

- Question or problem definition.
- Acquire training and testing data.
- Wrangle, prepare, cleanse the data.
- Analyze, identify patterns, and explore the data.
- Model, predict and solve the problem.
- Visualize, report, and present the problem solving steps and final solution.
- Supply or submit the results.

Here are Steps I would follow:

1. Plot data¶
   -Plot selected data
   -Normalize data
   -Rolling mean
   -Rolling Standard Deviation
   -Bollinger Bands
   -Daily Return

2. Analysis:
   -Cumulative return
   -Creating portfolio
   -Plotting portfolio statistic

3. Optimizing the Portfolio by optimizing or minimizing the negative sharp ratio

4. Plotting optimal portfolio statistics

5. Scatter plot, co-relation between my portfolio daily return and SNP_500 daily return, Beta value to see how reactive is my optimized portfolio compared to the market, alpha value to see how well it performs compared to the market

6. Plotting histograms

7. Stock price prediction interface – Linear Regression

8. Stock price prediction interface – KNN regression

9. Stock price prediction - ARIMA