Portfolio Optimizer

August 30, 2017

1 Machine Learning Engineer Nanodegree

1.1 Capstone Proposal

By Giordanni Piguing

1.2 Domain Background

Financial models have been used for years to better understand market behavior and make portfolios profitable. Years of historical stock prices are available for the public which are suitable for machine learning to generate such financial models.

The equity forecast [1] by Nikola Milosevic predict whether some company's value will be 10% higher over a period of one year using machine learning. The paper has motivated me to write my own model. Unfortunately, I dont think I'll have access to all the company information that he used as his features. I'll just use whatever information will be easily accessible to me.

Using the machine learning techniques and algorithms I have learned from Machine Learning Engineer Nanodegree, I would be generating a stock price predictor model. It would help predict when to buy or sell a certain stock base on its current performance. Using that model, I could generate a portfolio optimization that automates trading and see how it would perform.

1.3 Problem Statement

The stock market is very volatile and unpredictable. Although there are plenty of information available per stock, such as earnings, market cap, profit/expense ratio, news, etc, it is still hard to predict where the stock price will go. If it was easy, then every one could have made millions from stocks already.

For this project, I would only be using the daily trading data over a certain date range as training set. The data would contain metrics such as opening price (Open), highest day price (High), lowest day price (Low), closing price (Close), and volume (Volume). The model would then predict whether to buy or sell that certain stock, on that specific day.

1.4 Datasets and Inputs

As mentioned aboved, I'll be using the Open, High, Low, Close, Volume dataset per stock. I'll perform a few pre-processing techniques before it can be fed to the model. Multiple moving averages (2, 3, 5, 10, 30, 45, 60 days) will be calculated and normalized. Multiple stocks will also be used, such as: AAPL, GOOG, YHOO, T, IMAX, IBM, NFLX, SIRI, S, PLUG, C, BAC, P, NOK,

XONE, SSYS, TSLA, AMZN, SDRL, DDD, DBO, SRPT, SPWR, SCTY, FB, URRE, NQ, TWTR, F, BAH, MZDAY, FSYS, BIDU, KORS, HLF, ORCL, MBLY. The range of the dates will be about 7 years, from Jan 2010 - Jul 2017. Any rows with NA will be dropped, including holidays, weekends, and the first 6 months. Since the 60 day average will be NA for these times. Each stocks will have about 1785 rows of data. It'll have less data if the stock wasn't available from Jan 2010. The next day closing price would be use as the target. This will be used to predict whether the stock is going up or down, base on the current trend. Since stock prices vary a lot, I'll be normalizing them in ratio with the previous closing price, 2-day, 3-day, 5-day, 10-day, 30-day, 45-day, or 6-day average prices. I'll try multiple target options as well. Such as:

- Below 0.98 (Sell), Above 0.98 (Buy)
- Below 1 (Sell), Above 1 (Buy)
- Below 0.99 (Sell), Above 1.01 (Buy), In between (Neutral)
- Below 0.98 (Sell), Above 1.02 (Buy), In between (Neutral)
- Below 0.97 (Sell), Above 1.03 (Buy), In between (Neutral)

All of these would be ran, and will determine which combination has the best performance.

1.5 Solution Statement

A model would be determined using all the information described from the dataset/input section above. Multiple supervised learning algorithms such as DecisionTreeClassifier, GaussianNB, SVC, AdaBoostClassifier, will be evaluated in order to determine the best one. The model could then be optimized using GridSearchCV or something similar.

1.6 Benchmark Model

I plan to compare the results with just buying and holding a stock for a certain amount of period as a baseline. For this project, we will always hold a stock for a year as a benchmark. The portfolio optimizer model would be executed within the same time frame, 1 year. Basically, the benchmark model will buy and hold SHOP, BA, SD, FCEL, HEMP, TPLM, CHK, OLED, HON, LMT, CMG, MA for a year. The portfolio optimizer model will buy/sell multiple times for a year with the same stocks. I will compare the dollar amount of the two models at the end, to determine if the porfolio optimizer model is better than the benchmark model.

1.7 Evaluation Metrics

Calculating the accuracy of the model on the test set, I would be able to see how well the model is doing in predicting whether the price will go up or down the next day. Will use the sklearn accuracy score along with F-score metrics, but this wont be enough to see how well our model is doing though. I would still generate a porfolio and its performance. Compare that with the benchmark model, of holding the stock for a year. The dollar amount at the end of the year will be compared against each other, to determine if it was better or worst.

1.8 Project Design

First step is to gather all the stock data mentioned above. I would perform preprocessing to gather more information, such as moving averages. I would then normalize all the data base on multiple categories. Normalize it base on previous closing price, 2-day, 3-day, 5-day, 10-day, 30-day, 45-day,

or 60-day moving averages. I would fit these data to multiple supervised learning algorithms and see which one performs best. Determine which normalization performs the best too. Possibly perform some optimization using GridSearchCV, to gain that extra edge. Base on preliminary test I performed, it seems that the AdaBoost classifier performs the best. The DecisionTree classifier comes in second, but it also takes the longest.

After determining a predictor model. I could use that to automate a portfolio, to buy or sell, within a given time frame. I would pick different stocks than was used before, just to make sure it didnt just overfit the training data. The porfolio would then have a value at the end and I would compare it to the portfolio that just bought and held the stock for that same time frame. Basically against the benchmark model. The automated portfolio could be buying/selling multiple times within a given time frame. Thus the commision fee will have to be considered. The taxes should also be considered. The benchmark model might have a long term gain tax incurred while the automated porfolio might have multiple short term tax incurred. All these items need to be taken into account to have a better evaluation and to determine whether its really better to just hold long term or whether to execute higher frequency trading with the stock price predictor model.

1.9 References

[1] Milosevic, Nikola. "Equity forecast: Predicting long term stock price movement using machine learning." PDF file. https://arxiv.org/ftp/arxiv/papers/1603/1603.00751.pdf