Machine Learning Engineer Nanodegree

Capstone Proposal

Youness Assassi
February 10, 2018

Proposal

Domain Background

Using machine learning to solve equity trading problems is something that I am highly interested in. According to JPMorgan[1], computer generated trades account for almost 90% of all trading volume as of February 2018. Where does this leave the small investor who does not have access to the same resources as large hedge funds and investment banks do? Resources such as computers capable of high frequency trading and expert statisticians that use quantitative analysis to beat the market.

After the market crash of 2008, I decided to get an MBA with focus on finance in order to understand why the markets fluctuate the way they do. I believed that it was important for me to gain more knowledge about the financial market so I do not make the same mistake that others made when they suffered major losses during the recession. My takeaway at the time was to not try to play the market as no one beats it in the long run. Now, with the advent of machine learning, I am not so sure that this theory holds anymore. The goal of this research is to find out if the market can be beat using some of the new theories in conjunction with machine learning and automation. This will be a breakthrough for someone like me as I may be able to invest a small amount of money and let an automated system trade on its own with the goal of maximizing profits with lower risk.

There is ample amount of research being done in this field, but much of it is not published for the obvious reason that the sponsors of the research would like to keep the information to themselves. Luckily, Udacity with the help of professor Tucker Balch of Georgia Tech has published a free course online called Machine Learning for

[1] https://www.cnbc.com/2017/06/13/death-of-the-human-investor-just-10-percent-of-trading-is-regular-stock-picking-jpmorgan-estimates.html

Trading.[2] In this course, the professor explains how hedge funds use different machine learning methods to devise strategies that can beat the market. Another research paper with the same name was published by Gorden Ritter[3], a professor at NYU where he shows how machine learning, specifically reinforcement learning or Q learning can be applied in long term portfolio management problems.

Problem Statement

Given only the freely available historical stock market information such as adjusted stock price and volume, can a stock portfolio be actively managed for a period of time that will beat the returns of the S&P 500? The goal of this project is to build an application that can generate an optimized portfolio of stocks given a certain amount of money, then adjust the portfolio as the market changes. The portfolio returns will have to beat the market returns within a defined time period of no more than 1 day in the future through a simulation using real market data. This is regression type of a problem, where I will use the S&P500 statistics as input and individual stock price as output. Some of the input variables I will be trying out will be the sharpe ration, bolinger bands and volatility.

Datasets and Inputs

I will be using the S&P 500 historical stock prices[4] that includes the high, low, open, close and adjusted close of each stock. Out of the 505 stocks, my program will generate a portfolio of about 10 stocks. This portfolio will be optimized for its high Sharpe ratio value with a percentage allocation for each of the stocks. The idea is to then analyze these stocks by generating a number of statistics that can be used as input variables or features. Some of these statistics include the Bolinger Bands, momentum and volatility, while the future stock price will be the label or output variable. The model will be trained using a number of days in the past (possibly 1 to 2 weeks) with a target price of 1 day in the future. I will be using TimeSeriesSplit from Sklearn to split the data between training and testing. This will ensure that my model is not able to peak into the future.

[2] https://www.udacity.com/course/machine-learning-for-trading--ud501

[3] Gordon Ritter https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3015609

[4] https://github.com/younessassassi/machine-learning-udacity-capstone/tree/master/data/stock_dfs

The goal is to eventually be able to train the model with the most up-to-date information on stock performance so that it can be able to generate recommendations around necessary adjustments to the portfolio in order to maximize its return.

According to Tucker Balch[5], technical analysis can be used to build a short term strategy (hours to a handful of days) that is able to beat the market which I plan to use here. The opposing strategy called fundamental analysis would be more appropriate for a long term strategy (months to years).

Solution Statement

The plan is to start with a portfolio of the entire S&P 500, then based on the historical performance of the stocks that make up the S&P 500, generate a portfolio of about 10 stocks that is optimized using the Sharpe ratio. This would ensure that we have a portfolio with high return and low volatility. The next step is to generate other statistics for each of these stocks, such as the Bolinger Bands, momentum and volatility that can be used as features for our learning algorithm. The label of course will be the stock price 1 day in the future.

The plan is to try different algorithms to generate the model, including KNN and Random Forests. Once the model is trained with the features from each of the stocks, I will use the model to predict the portfolio price 1 day into the future and then compare it with the actual price. The total portfolio value will then be compared to the value of S&P index.

Benchmark Model

The benchmark model that I will be using is the performance of the S&P 500 during the same period used for the generated portfolio. Since models built based on technical analysis do not perform well in the long term, the comparison will be limited to 1 day in the future.

Evaluation Metrics

As mentioned earlier, each predicted portfolio will be compared to the returns of the S&P 500 in the same period. The formula used will be: (P1 – P0)/P0 where P1 represents the total value at the end of the test period and P0 represents the total value at the

[5] https://www.cc.gatech.edu/~tucker/

beginning of the testing period. I will also use the Root Mean Squared Error of Predictions as a metric to evaluate the performance of each model.

Project Design

The following are the steps I plan to follow to complete this project:

1- Obtain the full list of stocks currently making up the S&P 500 through page scraping of a wiki page.
2- Use this list obtained in step 1 to query google or yahoo finance for historical stock performance up to the year 2000 or so.
3- Apply data processing to clean up the data. This includes removing stock price information when the S&P was not trading, filling empty cells using the fill forward and backward methodology and dropping tickers if they were not trading during the specified date range.
4- Generate a portfolio of about 10 stocks that is optimized using the Sharpe ratio. This will ensure that we are accounting for volatility when trying to maximize the return. It also allows to focus on a smaller number of stocks for analysis.
5- Calculate useful statistics such as volatility, momentum and Bolinger bands.
6- Use TimeSeriesSplit from Sklearn to split the data between training and testing.
7- Train various machine supervised learning regression models including KNN and Random Forests using the generated statistics as features and future stock prices as labels.
8- Compare different model's Root Mean Squared Error and fine tune the models to reduce the error by changing training data date range, changing the number of features, changing the number of k neighbors used for KNN and so on.
9- Compare the portfolio results to that of the S&P 500.