Machine Learning Engineer Nanodegree

Capstone Proposal

Amrit Prasad
December 12th, 2017

Motivation

I have always been intrigued by the price movements of the stock market and have tried my hand at investing. My initial foray into finance 5 years ago was during an internship, where I developed a model to predict the price for BSE-Sensex and its constituents based on Artur Wojtak's Fourier Transform model[1]. Hopefully I've learnt enough during the course of this Nanodegree to better my results using Machine Learning.

Proposal

I wish to use my machine learning knowledge to predict whether a stock/index will close higher or lower than its open price. This knowledge can then be exploited to generate profits by taking positions accordingly in the asset or its derivatives. To achieve this, I decided to model the most commonly known benchmark index, i.e. S&P 500, and downloaded 5 years' worth of data from Yahoo Finance[2].

Domain Background

Financial trading is at its core the buying/selling of financial assets in the hopes of making a profit (speculating/investing depending on intent) or limiting downside (hedging). These financial instruments can be anything under the sun, from cash (FX) to company ownership (equity) to debt (credit/rates) to cattle (commodities). It has seen massive changes in method from open outcry to electronic trading.

In the recent past, algorithmic trading has become the rage, which is essentially removing or automating away human involvement. It can be as simplistic as automating the data download via website crawling to the creation of very complex models that use advanced math and finance domain knowledge that trade without human intervention. The current project aims to develop an aspect of one such model by trying to utilise Machine Learning techniques in the hopes of developing a financial trading model that would beat the returns of a selected benchmark on an out-of-sample data set.

Problem Statement

The ML part of this project is to predict whether S&P 500 will close higher or lower than its open. Since there are two possible outcomes, this can be tackled as a classification problem, and I aim to use a Decision Tree, SVM and Ensemble Leaners (Random Forests and Gradient Boosting), and choose the model with the highest cross validation score. A label 1 should imply close above open the next day, while 0 should imply close below open. In case of predicted label 1, go long futures, while go short in the other case. For unchanged index values, the labelling should be according to the average of the previous five trading days. This prediction can then be used by any amateur trader to profit by taking appropriate positions in SPY ETF/SPX Index Futures/derivatives on the former two.

Datasets and Inputs

For this classification problem, I have tried to limit myself to numerical raw data like open/close price, volume and market expectation of 30-day volatility (VIX). The following raw data fields are going to be used in this problem-

1) Date: Date on which the data was recorded
2) Open_SPX: Open value of SPX Index
3) High_SPX: High value of SPX Index
4) Low_SPX: Low value of SPX Index
5) Close_SPX: Close value of SPX Index
6) Volume_SPX: Volume of SPX Index constituents traded
7) Open_VIX: Open value of SPX VIX
8) High_VIX: High value of SPX VIX
9) Low_VIX: Low value of SPX VIX
10) Close_VIX: Close value of SPX VIX

The above dataset was downloaded from Yahoo Finance[2] for the last 5 years of trading days from 6th Dec 2012 to 6th Dec 2017. Each data field has a value corresponding to the Date of trade. If any of the fields is missing/invalid, then it should be filled by the average of the 5 previous trading day values. Since the numerical ranges aren't standardised for these data fields, they need to be used for engineering new standardised features, that will be the input for training the model. These new features are mentioned in the Project Design section.

As mentioned in the Problem Statement above, close > open → label = 1, close < open → label = 0, and close = open → label = average of last 5 labels. If last 5 labels aren't available, then discard the data rows.

Calculation of VIX

VIX is calculated as 100 times the square root of the annually scaled expected 30-day variance of the daily returns of the S&P 500 Index.

$$VIX = 100\sqrt{var}$$

$$var = \frac{365}{30} \times Expected\ 30 - day\ variance$$

The expected 30-day variance is calculated using a portfolio of out-of-money puts and calls where the reference strike is the maximum value smaller than forward price. The forward price is in turn estimated from the following formula-

$$F = Strike + e^{RT}(Call\ Price - Put\ Price)$$

Where,
Strike: Options' strike where the difference between calls and puts is minimum
R: Risk Free rate of return
T: Maturity of options

If options maturing in exactly 30 days aren't available, then a weighted average of the closest maturing (between 23 to 37 days) options' variance is used for estimating the 30-day variance.

Solution Statement

The solution of the problem would be to predict accurately whether the SPX 500 on day x, will close above or below the open. Since only data prior to x will be used to predict this, the information could be used to take the relevant position on the open of day x.

Benchmark Model

The benchmark model to be beaten would be the passive investing strategy in the most popular benchmark index of the stock/index's origin country, which advises buying and forgetting about the investment. This is the benchmark which 80% of fund managers are unable to beat year on year[3]. For our case, S&P 500 would be the benchmark index. We can invest in SPY ETFs/long dated forward contracts for simulating the benchmark in real life. For the purposes of this project, I'll be using the SPX Index values as the theoretical benchmark, which should correspond very closely with the real life version.

Evaluation Metrics

There are a couple of evaluation metrics to consider here. The first one would be the F1 score, and the other could be the cumulative PnL of the strategy implementation

assuming that the SPX Index can be bought and sold at the index values, which is not too bad an assumption (SPY ETF/Emini Futures do a pretty decent job of capturing the daily moves), and comparing both of these against the benchmark of buy and hold.

Project Design

The initial work would focus on getting the benchmark setup, and its evaluation metrics calculated. Post that, comes the part where we need to try and extract useful, standardised information out of the dataset that could be used for predicting whether SPX closes above or below the open. A very important thing to keep in mind while working with this data is to ensure that look-ahead bias is avoided. Hence, if we are trying to predict something on trading day x, then all data should correspond to trading days x-1 and before. A few useful features that could help us in our prediction for trading day x are-

1) Trailing_1d_Return: Close(x-1)/Close(x-2) – 1 of SPX Index
2) Trailing_1d_Max_Move: High(x-1)/Low(x-1) – 1 of SPX Index
3) Trailing_2d_Return: Close(x-1)/Close(x-3) – 1 of SPX Index
4) Trailing_3d_Return: Close(x-1)/Close(x-4) – 1 of SPX Index
5) Trailing_4d_Return: Close(x-1)/Close(x-5) – 1 of SPX Index
6) Trailing_5d_Return: Close(x-1)/Close(x-6) – 1 of SPX Index
7) Trailing_10d_Return: Close(x-1)/Close(x-11) – 1 of SPX Index
8) Trailing_22d_Return: Close(x-1)/Close(x-23) – 1 of SPX Index
9) Trailing_63d_Return: Close(x-1)/Close(x-64) – 1 of SPX Index
10) Trailing_252d_Return: Close(x-1)/Close(x-253) – 1 of SPX Index
11) Trailing_1d_Return_VIX: Close(x-1)/Close(x-2) – 1 of SPX VIX
12) Trailing_1d_Max_Move_VIX: High(x-1)/Low(x-1) – 1 of SPX VIX
13) Trailing_5d_Return_VIX: Close(x-1)/Close(x-6) – 1 of SPX VIX
14) is_MA5_above_MA20: 1 if average of Close(x-1) to Close(x-6) > average of Close(x-1) to Close(x-21); else 0 of SPX Index
15) is_Trailing_1d_Vol_above_MA10_Vol: 1 if Vol(x-1) > average of Vol(x-1) to Vol(x-11); else 0 of SPX Index
16) Trailing_1d_Gap_Return: Open(x-1)/Close(x-2) – 1 of SPX Index

Close(y) above refers to the Close price on trading day y.

Once the above features have been calculated for however many days they can be, the next step would involve cleaning up the data by dropping all NaN values, which will creep in if the date ranges are out of scope. Post the removal of NaN, 4 years' worth of features should be left (the 1-year return eats up that year). After the data cleaning comes the part where we split the data set into training, cross-validation and test sets. The remaining 4 years can then be split into 60%-20%-20%. A key thing to note here is that the data should be split *sequentially* rather than *randomly*, to avoid look-ahead bias.

The model should be trained by using Decision Trees, SVM, Random Forests and Gradient Boosting. Initially for these models, I would train them on the training dataset while using the default values, and then look to fine tune the hyper parameters on the cross-validation set using GridSearch. The best tuned model out of them should be chosen on the basis of the best F1 score on the cross-validation set. This best model should then be compared against the benchmark on the test dataset using the evaluation metrics of F1 score and cumulative PnL. If both of these are better, then the ML strategy is ready to be implemented in the live markets. Maybe test it out for a month or two by paper trading, and then take it for a spin.

For implementing the strategy, we can utilise the predicted labels 1 and 0 by interpreting the former as the signal to go long, and the latter as the signal to go short. A major assumption of no transaction costs will be taken, and the portfolio would be closed out daily, i.e. if the predicted label for tomorrow is 1, then buy at open and sell at close, and vice-versa.

Bibliography

[1] https://web.wpi.edu/Pubs/E-project/Available/E-project-022808-142909/unrestricted/FulllQPReport7.pdf
[2] https://finance.yahoo.com/quote/%5EVIX/history?period1=1354818600&period2=1512585000&interval=1d&filter=history&frequency=1d
[3] http://us.spindices.com/documents/research/research-fleeting-alpha-evidence-from-the-spiva-and-persistence-scorecards.pdf
[4] https://www.ig.com/uk/learn-to-trade/what-is-financial-trading
[5] https://www.investopedia.com/terms/a/algorithmictrading.asp
[6] http://cfe.cboe.com/cfe-education/cboe-volatility-index-vx-futures/vix-primer/cboe-futures-exchange-nbsp-nbsp-education
[7] https://www.cboe.com/micro/vix/vixwhite.pdf