Machine Learning Engineer Nanodegree

Capstone Proposal
Michal Kozinski

Domain Background

Equity Option traders try to predict future volatility of stock and implided option's volatility (vol) based on past data. They use a variety of signals to chose which options seem to provide the most edge and place their bets in the market. One of the most important part of the puzzle is predicting future realized volatility of stocks. Traders, hedge funds and option market makers (MM) use a variety of tools, ranging from looking at simple stock charts and looking at past volatility data and trades high touch (aka "manually" trades) to using machine learning/statistical models and trading low touch (traders only oversees the "flow" of transactions but not what is bought and sold on individual basis). I am personally interested in the topic as I have been option trader for over a year and I am continously very interested in the topic.

Problem Statement

The problem I want to try to solve is to predict next term realized volatility based on past period of time. I am especially not assigning units of time as there are different approaches to the topic concerning what is relevant unit of time when trying to predict future realized volatility. In general, the most recent data provide more insight into next few days, however when we want to predict long term volatilty we should also look at long term past data. The measure of success of our model here should be how far our model predicts future vol vs the actual realized vol. One solution could be looking at past realized volatility as from practice it is the best predictor of the future realized vol.

Dataset and Inputs

The dataset that I will be working with has been obtained through Kaggle. It contains Spanish stock data from 2009 to 2019. It contains daily returns for 27 spanish stocks, and for each day data has typical 4 prices: open, close, high and low. This dataset is appropriate as it contains closing day prices, the most important predictor of future vol.

Solution Statement

The solution to the problem will be predicting future vol within reasonable accuracy. Option traders first try to predict future vol, and if they see that the market is mispricing vol, they try to buy low and sell high (implied vol). The model should take as an input past realized vol data (in other words past daily movements) and predict future average vol. For option traders it is important that their prediction of reazlized future vol is as close to the actual as they will be able to extract that mispricing by process called delta hedging (too complicated explanation but basically if option implied volatility is 20 and trader predicts accurately it will 30 and buys it, he can "earn" that 10 vol points in $, more can be read in Sheldon Natenberg).

Benchmark Model

Usually such models are implemented as proprietary tools by hedge funds and prop shops therefore there is very limited data. Furthermore, in most cases the models use much more data than is available to me here, thereofre I will just use "naive" model where I will use "lookback" realized vol as prediction for the look forward period (so if I am using past 20 days data to predict next 5 days realized vol I will just assume next 5 days realized vol is equal to past 20 days realized vol).

Evaluation Metrics

Evaluation metric that can be used is RMSE (root mean squared error) and MAD (mean absolute deviation). We should also look at the median and how the distribition looks like, and it should be expected that it has much bigger tails than normal distribution which is typical when working with stock data.

Project Design

The model that I want to implement will take as an input realized vol data + movement of the stock and will try to predict future realized vol. The important thing to note is that for option traders it is actually not relevant if stock is going up or down, as because of the delta hedging it is not relevant to their capability of extracting mispricing from the market. However, the model should have this information as maybe big moves down are more likely to precede higher realized vol than big moves up.

The data will need first have to be transformed into log returns. Then, I will have to create our dependent variable (what our model will be trying predicting) by summing up the relevant time period and then calculating what would be the normalized yearly volatility (the standard unit of time). Thus, we have a realized volatility for some period of time that will be our Ys values. Then, I will create a vector of realized daily volatility in similar fashion, and add back at the end whether the the move on the day was up or down. In such a way we have our y variable a realized vol over period of time (which always must be positive) and then we will have a vector of realized daily vols (adjusted to be in the same units as our y value) with additional information wether moves was up or down (so in cause the calculated daily vol was 40, and move was down it would be equal to -40.)

I will create such Ys and vector of Xs for each possible date where data is available (as if our look past is = 10, predict future = 5) we need at least 15 points of data for the model. I will combine all data for all stocks, get rid of all Y-Xs pairs where there is any NaN (first, as some of them will have missing value as realized vol can only be calculated if there is enough points) and then I am sure there will be some missing data as every stock data I have worked with before was missing something. Then, I have two versions of dividing my data, by stock and random. I can either combine all stocks, shuffle and divide it into 3 sets of train, validation and test, or divide or divide it by stock (as there are in total 27 stocks and for example I can choose 20 for train, 3 for validation and 4 for testing).

When this is done, I will feed my data into models. I am thinking about using AWS's built in XGBoost algorithm using linear regression package, as well as linear with squared error. I will also look at how well Amazon's Deep AR will work in this situation. LAstly, I will look at custom made PyTorch model with full fully connected Neural Network and look at which one of them have the best results on validation data, and then I will do hyperparameter tuning on the best model using automatic hyperparameter tuning function of AWS.

Sources: Bartolome, Alvaro. “Spanish Stocks Historical Data from 2000 to 2019.” Kaggle, www.kaggle.com/alvarob96/spanish-stocks-historical-data.
Natenberg, Sheldon. Option Volatility and Pricing Strategies: Advanced Trading Strategies and Techniques. McGraw-Hill, 1994.
“RealVol Daily Formula (Realized Volatility Formulas).” Calculating Realized Volatility, www.realvol.com/VolFormula.htm.

In [ ]: