Udacity ML Engineer Nanodegree Capstone Project

Predicting Bitcoin Movement Using MLP

Wah Chi Shing, Anthony

Background

The recent surge in Bitcoin price and historical high at $24,000 has again shed light on this legendary cryptocurrency. The sky-rocketting volatility of this coin has driven investors and trades to build extensive predictive models to look for any profitable opportunity from it.

Figure 1: Bitcoin Price since 2020 July

Unlike traditional assets where intrinsic value of the commodity/stock can be taken as a refernece price for trading, Bitcoin itself is not backed by any physical asset. It is solely subject to investors sentiment/expection on its price movement and thus it has a highly volatile nature. Classical financial theories and models learnt in classroom will no longer work here (e.g. CAPM, WACC). What we have for analysis (freely and publicly available data) will only be the historical price. Therefore, we are trying to take advantage of the power of Machine/Deep learning, to find out the hidden and untrivial patterns of the price movement from its historical data.

Problems and Targets

There are different ways to predict an asset price, such as predicting the actual price, expected forward % change, volatility, price range ... etc. Due to the fact that number of investors in Bitcoin and the pool size (USDT used for trading BTC) change over time, the expected actual price may lose its reference value if either of them changes. Therefore, we will target on the expected forward % change of Bitcoin over a fixed time interval in this model (e.g. forward 1-minute % change).

Then here comes the core part of this model: features to be used for predicting forward % change. This looks very much like a time-series problem and it seems trivial to use RNN to solve this problem. However, we don't want the price data to be the only

input for the problem (or else it will be too simple that anyone with a RNN model can do it). We are trying to include some other proprietary information computed on our own analysis (e.g. moving average price, average volume traded ... etc). Therefore, we will need to:

1) Pick some target time intervals for constructing forward % change to predict, and;
2) Model a pool of features so that we can train a model based on the feature-expected-change relationship, and lastly;
3) Select a target algorithm for predicting the forward % change

Note, we will try to first predict the NUMERICAL value of the forward % change directly using the model. If the result isn't good enough, we may try grouping the forward % change into different bins and do a binary/multi classification.

Data Source

Bitcoin are traded in a lot of different crytocurrency exchanges, so its price/volume data may vary between different venues (but the difference shouldn't be too large, see here for more). Here we will use historical Bitcoin price data from Binance as it is one of the largest crytocurrency exchanges in the globe. We will get the Bitcoin 1-minute open-high-low-close-volume data from its publicly available API.

date_time	open	high	low	close	volume
2020-11-19 00:00:00	17873.89	17880.12	17828.51	17870.0	235.474692
2020-11-19 00:01:00	17870.0	17939.13	17870.0	17936.43	149.612749
2020-11-19 00:02:00	17936.44	17960.0	17927.63	17941.68	201.725584
2020-11-19 00:03:00	17941.68	17978.78	17937.82	17978.78	154.594181
2020-11-19 00:04:00	17975.7	17982.76	17958.0	17969.09	163.60663
2020-11-19 00:05:00	17969.08	17985.58	17953.53	17958.5	125.26772
2020-11-19 00:06:00	17958.5	17965.13	17925.01	17963.58	113.729178
2020-11-19 00:07:00	17963.59	18000.0	17962.32	17988.96	209.569132

Figure 2: Binance data sample

Hypothesis

First of all, we will prepare a pool of features/indicators that may be taken as an input to our model. After some analysis and screening based on their correlation, we may come up with a final size of ~ 5-10 features.

Then, we will try to test on forward % change over different time intervals (e.g. 1min, 5min, 10min). We expect that not every time interval will show an extraordinary result or predict the forward % change very accurately, but at least the model should be able to tell the direction of movement in some of the time intervals.

Benchmark Models

There has been different types of machine learning/deep learning approach towards predicting Bitcoin price movements. For example, this article described how a RNN/LSTM model can be applied to predicting Bitcoin prices in the next 10 days. As mentioned above, we are trying to predict the forward % change of Bitcoin in a fixed time interval but not the exact price. So there is a bit difference here, but still, it can be a good benchmark model for our comparison.

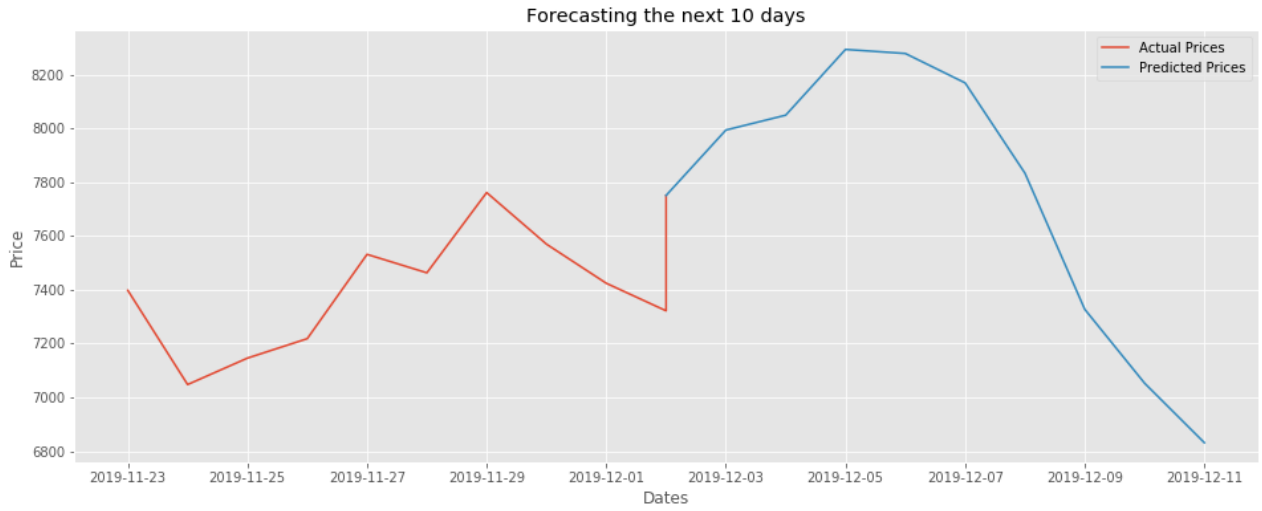

Figure 3: Forecast Visualization of Benchmark Model (Source)

Evaluation Metrics

After we train the model, we will try to plot the following to first visualize our result.

1) Predicted forward % change VS Actual forward % change
2) Predicted expected Bitcoin price VS Actual Bitcoin price

where expected Bitcoin price is computed by predicted forward % change $\times$ current price

Then we may compute the following statistics to evaluate our predictions (on test data):

1) Error Variance: $\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}$
2) 95% Confidence Interval (Assuming Normal distribution of forward return): $\hat{y} \pm 2 \times \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$

Solution Outline

The way to develop a solution will be outlined as the following:

1) Prepare dataset by calling historical data from Binance;
2) Prepare a pool of all features to be selected for training the model;
3) Analyze correlation/significance of each features and pick a few of them;
4) Pick a/some target time interval(s) for constructing the forward return to be predicted;
5) Train a trial model and analyze the prediction accuracy;
6) Change the number of features/apply PCA/change the target time interval and go back to step 5);
7) Choose the target model/parameters to be used and test on the validation set