**Ethereum Cryptocurrency Price Prediction**

**Data Preparation:**

It was initially my intent to query granular minute by minute ethereum blockchain data from Google bigquery. Due to running against bandwidth and hardware limitations I shifted to daily. This has reduced my data from over 2 million rows down to 1600. Using the results from delivery 2, I identified features that correlate highly with Ethereum price and merged them all into the same dataframe based upon a datetime index. These features are: number of addresses, number of unique addresses, number of transactions, transaction fee, Google trends, price open, price close, price high, price low, and exchange volume. In depth EDA of these features can be seen in delivery 2.

**Feature engineering:**

I used financial theory to develop 3 new sets of features. First, I added in moving averages. As prices can change quickly, I am concerned that this can throw off my model. Moving averages smooth overs this variance. Second, I calculated the relative strength index (RSI). RSI is a heavily utilized metric in the investment industry. It is a measure for how overbought or oversold an asset is. Please see appendix for the formula. I believe RSI will prove to be a powerful feature because it is so commonly used by investors. I calculated a third technical metric, Moving Average Convergence Divergence (MACD). MACD is a “trend-following momentum indicator that shows the relationship between two moving averages of a security’s price. The MACD is calculated by subtracting the 26-period Exponential Moving Average (EMA) from the 12-period EMA. The result of that calculation is the MACD line. A nine-day EMA of the MACD called the "signal line," is then plotted on top of the MACD line, which can function as a trigger for buy and sell signals. Traders should buy the security when the MACD crosses above its signal line and sell - or short - the security when the MACD crosses below the signal line.”

With the addition of the above 3 features my dataset is finalized, I am ready to select a machine learning model.

**Model Selection:**

When conducting a literature review, I found a similar project called “Predicting short-term Bitcoin price fluctuations from buy and sell orders”[1]. This paper constructed numerous models, but found the Extreme Gradient Tree gave the best results. I therefore researched the xgboost python library (extreme gradient boosting). Gradient boosting uses a collection of ensemble methods to make a prediction while using boosting to decrease computational resources. I found that this is applicable to my dataset which is a time series regression problem and extremely resource hungry.

**Time Series Prediction:**

A time series prediction is one where the order of events is important. For cryptocurrency pricing this is crucial. This adds a challenge as Time Series data cannot be simply is shuffled randomly or ordering would be lost.[3] Therefore instead of using the randomized test/train split included in scikitlearn , I created distinct time windows to separate my test, train, and validation sets (fig 2). Cross validating my data set is going to be a challenge to tackle during delivery 4.

**Preliminary Results:**

The preliminary run of model produced an underfitted prediction (fig 3). However, it looks like I have a strong starting point for my model. As I go into delivery 4, I need to work on tweaking the model parameters as well as my test/train time windows.

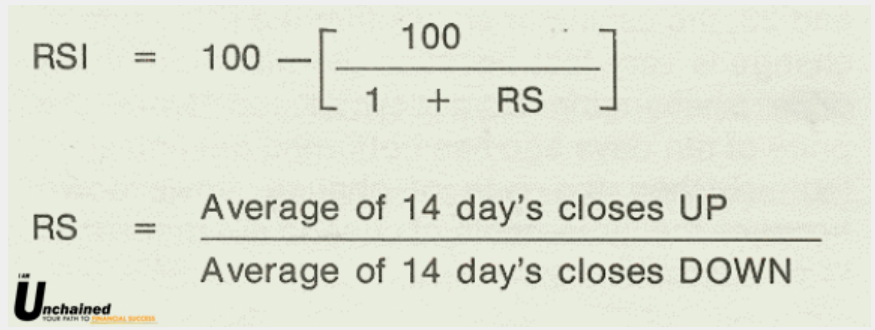
For more details on the methods I employed, please view the python notebook in delivery 3 of my github page.

**References:**

1. <https://www.investopedia.com/terms/r/rsi.asp>
2. <https://www.investopedia.com/terms/m/macd.asp>
3. <http://francescopochetti.com/pythonic-cross-validation-time-series-pandas-scikit-learn/>
4. <https://plotly.com/python>
5. <https://www.researchgate.net/publication/323141771_Predicting_short-term_Bitcoin_price_fluctuations_from_buy_and_sell_orders?enrichId=rgreq-c5c3e2551eac9650e731da8639be436f-XXX&enrichSource=Y292ZXJQYWdlOzMyMzE0MTc3MTtBUzo1OTgzMTQwMTM3MDgyODhAMTUxOTY2MDU4NTI4OQ%3D%3D&el=1_x_3&_esc=publicationCoverPdf>
6. <https://www.kaggle.com/johanvandenheuvel/lstm-model-of-stockdata>

**Appendix:**

Fig (1)



Fig(2)



Fig(3)

