Rickord, Jake JJ
Applied Data Science
Project 2 Milestone 2

Business Problem

Every individual young and old has likely wanted to retire early, and enjoy the rest of our lives worry-free. With the advent of penny-stock similar cryptocurrencies, we've seen an uptick of young persons in particular play their own hand at what are now traditionally seen as a scam tactic: buying dollars worth of cryptos valued at 16ths of a penny in the hopes that they hit the jackpot akin to the bitcoin or ethereum jumps we've in the past decade. With the demand for this market being high, the supply has exploded with a million different Shibcoins, Dogecoins, and so many others that jump and fall based simply on tweets or other random noise they create. But how random is that shift really? Is there a possibility that, in reality, we can leverage historical crypto data to predict these shifts and provide better predictions with which to invest our hard-earned dollars?

Background/History

If you asked 100 people in 2010 what cryptocurrency or Bitcoin was, it'd be an incredibly low margin, if anyone, that would know what you're talking about. Now, a little over a decade later, it is common to talk about in boardrooms, lunch hours, and over dinner. The "crypto craze" has really taken off as companies, criminals, and even countries have started to adopt cryptocurrencies as a common circumnavigation to conventional cash. In 2021, an Argentinian lawmaker put forth legislation that aims to allow workers to opt into receiving their payments in cryptocurrency as opposed to pesos.

All this interest in the future usages of cryptocurrency has led to a two-fold explosion: crypto coin makers aiming to come up with the safest and most popular coin that could be adopted going forward, and a swathe of investors young and old looking to get in on the action before it's officially adopted. To add fuel to the fire, recent memories of early investors in Bitcoin (represented as BTC on exchanges) hitting the jackpot and becoming millionaires practically overnight, has lent itself to younguns and older folks alike looking to join in the earnings.

Rickord, Jake JJ
Applied Data Science
Project 2 Milestone 2

Figure 1: Bitcoin Price Explosion

The problem in this, however, is that, thanks to the former consequence of more cryptos on the market, it is increasingly difficult to make educated guesses to which will fall and which will rise (as seen most recently with Dogecoin, which was generated around a meme, but quickly gathered a cult following including that of Tesla CEO Elon Musk). As the trepidation for some continues, and the ever-present goal of aiming to attain the golden ticket of living a life financially-worry free continues, we aim our sights into leveraging historical crypto data into predicting their cryptographically complex futures.

Data Explanation

The primary dataset we will use will be provided from the Kaggle user Sanskar Hasija, who generated it from the www.investing.com website, which can be found below:

https://www.kaggle.com/odins0n/top-50-cryptocurrency-historical-prices

The dataset itself, as explained in the title there, is the historical pricing data from the top 50 cryptocurrencies currently available on the market by cap size. The variables available to us for modeling for each crypto are open, high, low, close prices as well as the volume of trading for each day since inception of each coin up until August of 2021. Each cryptocurrency had its own csv file, however we chose to operate off of the “All_combined.csv” file which contains all the data from the individual csvs labeled with the coin they belong to.

Once we had the dataset imported, our first step was to check for missing values and in general, inspect data integrity. As expected, we identified 50 unique cryptocurrency labels, one for each coin type. We checked the data type of each column and were pleased to find that each column of data was formatted as a float with

expectation of the labels and the Date column. Similarly, we were also excited to see that the data did not hold any null values, and thus no data fills were required.

From here, we made some small modifications to the naming conventions of the dataset for purposes of clarity, such as renaming the “Price” column to “Close” to better understand what the price shown there represents, and then reordered that column header to come after Open, High, and Low values.

All the above checks and reorganization being completed, we moved onto visualizing and modeling our data.

Methods

Once data preparation was complete, we took the opportunity to visually analyze the different distributions of coin prices over time. To do so, we graphed each distribution via the matplotlib method, the results of which are shown below:

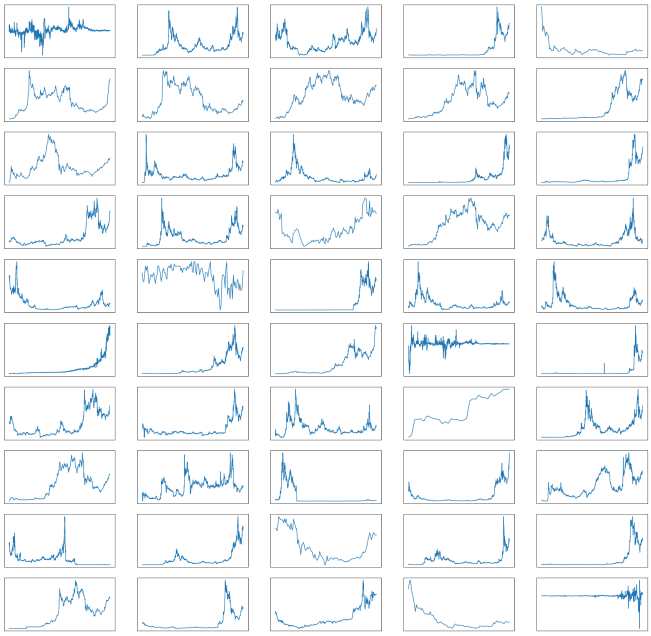

Figure 2: All Cryptocurrency Distributions Graphed

While each individual chart is not necessarily of interest on its own, it is notable that quite a few of these distributions seem to have values peaking towards the most recent data (left-skewed). This lines up with our earlier discussion of the massive increase in popularity in recent years. There are, however, a few oddballs hanging around in these depictions, which may serve as a way of showing if our models are overfitting.

All that being said, now that we have our dataset all cleaned up, and we’ve taken a look at these distributions, we needed to dive into how we planned on predicting these

crazy values. To start, we knew we needed to have our test data to compare against in order to attain any measure of accuracy. As such, our next logical step was to divide these up. Once we began this process, we actually realized we needed to take a step back and present an idea on how we were going to leverage our regression modeling by asking questions about the inputs for it. Did we want to simply use the bulk of the data here and simply see how the mass of crypto data can predict future data? Or is there a way we could iterate over each crypto and see how accurate a predictor each is on the rest of the cryptocurrencies?

After careful debate, we decided, why not go for the best of both worlds? As such, we decided on selecting an n-gram-akin technique in which we'd start by using linear regression built on each crypto to predict all the others, and compare each accuracy to one-another. Then, we'd move on to pairing the first crypto to the currency following it alphabetically, i.e. Avro to Bitcoin, then Bitcoin to Ethereum as an example and so on through all 50. This would proceed all the way through a model built using the entire dataset of all 50 cryptos.

Having this decided, we then needed to ensure we kept the appropriate labels within our test/train splits. Once we carried this down, we could move onto building out that modeling process. We established a single variable, b_a for best accuracy which we measured via r_squared values. Iterating over each of the n-grams as previously discussed, we combined the data for n-cryptocurrencies, and then created a multiple linear regression model off of it, and drew up our r-squared value for the model.

Analysis

One additional complication came to actually analyzing how that R-squared should be accounted for. Generally speaking, we apply that via our predicted r-squared based on the X-test set vs the actual output seen in the y-test set. However, as we had multiple different crypto currencies involved in the creation of the x-test, our train-test split had carved out an equal percentage of each cryptocurrency. This provided a particularly nasty challenge best explained via example:

> If our model is 95% accurate for cryptocurrencies that have data since 2015 (thus the testing set is large), and is only 10% accurate for a cryptocurrency started in 2019 (thus much smaller in test size), should those r-squared values be weighted equally?

From our critical thinking, we came to the conclusion that it did not make sense to weigh them equally. Thus we adjusted the r-squared value for these elements as a measure of its size in the combined test set. Once each model was tested against all cryptocurrencies, the r-squared values for each were then summed. If this total was greater than that of the previous best model, it would take the new top spot as the best

overall, and its r-squared number stored along with the cryptocurrencies used to develop it.

From our results, we can conclude that the best multiple linear regression model for predicting cryptocurrencies would be built from the use of historical data from Dai, Litecoin, Tezos, Polygon, and Shiba Inu. This conglomerate makes up our best conglomerated r-squared of 0.907. From here we took this model and once more synthesized the results of the modeling procedure against the test data, this time to examine the root-mean squared error value for the model against the testing data which provided values as seen in Figure 3.

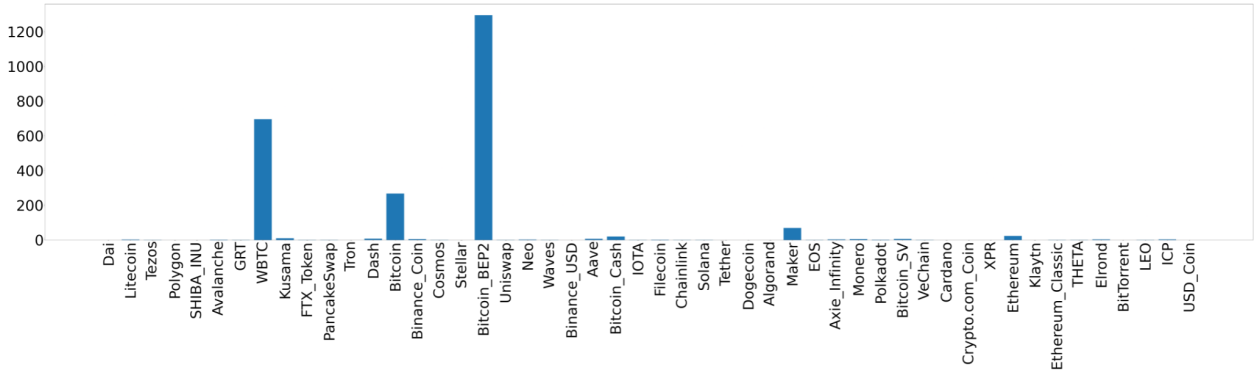

Figure 3: Root Mean Squared Error for each Crypto Coin

These results indicate that our model was pretty good at hitting a majority of coins with low variance, with the exception of bitcoin, which it struggled with in all areas (bitcoin, wrapped bitcoin, bitcoin bep2, bitcoin cash).

Conclusion

Examining our modeling data, we were able to build multiple successive n-gram models to attempt to calibrate our linear regression technique onto predicting a wide variety of cryptocurrencies. If our model were to be perfect, we'd be making millions. Unfortunately, based on the root mean squared error data, it doesn't seem to hit some of the biggest players in the industry. One possible explanation may be regarding the relative size of most bitcoin and bitcoin offshoots are higher than that of many of the other coins. As such, an effort to scale values down for those such as bitcoin and ethereum may create a more inclusive and applicable modeling here.

Assumptions

As akin to most projects that aim to predict the future, the biggest assumption is always that historical data is representative of future activity. Beyond this, our models were built using the top 50 cryptocurrencies by size as of August 2021. It is quite possible that leveraging smaller cryptos, or subbing others out as the market values ebb

and flow, may provide additional lenses to increase the accuracy of our model. Additionally, we leveraged r-squared to identify the model that “best fit” the data. Use of other measures such as mean absolute error, mean squared error, root mean squared error, or adjusted r-squared values to identify the best performing model may also increase predictive accuracy. Finally, our choice to weight the underlying r-squared values based on size within the overall test set may be better implemented in an alternative way.

Limitations

Some limitations of the analysis include the lack of recent data coming after August of 2021, lack of computing power necessary to increase the number of crypto to all available on the market (which may or may not impact performance), as well as limiting our dataset strictly to the pricing and volume details involved with the dataset. As we’ve seen in some of these coins, something as odd as Twitter data can have a large impact on pricing. One change in analysis simply using n-grams as opposed to trying out all combinations of crypto training data additionally could be used for full breadth of linear regression results.

Challenges

Challenges encountered within the application of this modeling technique almost solely surrounded identifying a solid method of implementation for keeping the labels intact and additionally about how to apply weights appropriately to address the issues regarding each individual coin’s representative portion of the testing data sample.

Future Uses / Additional Applications

Future uses of this modeling would be to use similar techniques across subsets of the traditional stock market elements such as stocks, etfs, and/or mutual funds (which operate more awkwardly with implementation. As this idea was built off of those traditional stock regression modelings save the n-gramming, it would likely apply once honed in.

Recommendations

Moving into fixes for this model in order to bring it to a successful implementation, more data should be ingested as cryptos continue to burgeon. Additionally, scaling methods should be implemented in order to better include those ebbs and flows in larger coins like BTC.

Rickord, Jake JJ
Applied Data Science
Project 2 Milestone 2

Implementation Plan

In order to implement this model according to the recommendations, the steps needed to be taken include:

- Ingest additional historical data
- If necessary, add additional crypto currencies
- Leverage regression model to predict future crypto price and,
- Buy/Sell a coin in question based on predicted trend
- Profit.

Ethical Assessment

One particular ethical question would be based around how these are implemented. If this model's accuracy were to improve significantly, it would need to be available to the public to make better informed decisions, otherwise private industries could use this to their advantage identifying where each are likely to go and betting for/against them (akin to Gamestop situation before Reddit got ahold of it).

Questions:

1. How would the inclusion of additional cryptocurrencies affect the prediction rate of the model?
2. What made you select r-squared to be the most appropriate measure of model accuracy?
3. Which independent variables were most important in determining the closing price?
4. Would other regression models be applicable to this situation?
5. With regard to weightings, what other ways could the weights be selected?
6. We see you used the sum total of the r-squared values as your measure for accuracy as opposed to mean or median. Why is that?
7. If you could go back and start this analysis over, what would you do differently?
8. Can this model be used to predict crypto ETFs as well?
9. Besides more data, are there any other ways you can think of to improve the predictive accuracy of your model?
10. Which do you think will be the next bitcoin based on your analysis?

Rickord, Jake JJ
Applied Data Science
Project 2 Milestone 2

Appendix

1. I. (2021, April 29). *How To Do Stock Market Forecasting Using Linear Regression In Python*. Imurgence. https://www.imurgence.com/home/blog/how-to-do-stock-market-forecasting-using-linear-regression-in-python
2. *TOP 50 Cryptocurrencies Historical Prices*. (2021, September 11). Kaggle. https://www.kaggle.com/odins0n/top-50-cryptocurrency-historical-prices?select=All_combined.csv
3. Vadapalli, P. (2021, December 31). *Stock Market Prediction Using Machine Learning [Step-by-Step Implementation]*. upGrad Blog. https://www.upgrad.com/blog/stock-market-prediction-using-machine-learning/#Step_1_-_Importing_the_Libraries
4. Wilson, A. (2021, December 23). *Stock Prediction Using Linear Regression - Analytics Vidhya*. Medium. https://medium.com/analytics-vidhya/stock-prediction-using-linear-regression-cd1d8351f536
5. www.aionlinecourse.com. (2019, March 14). *4 Best Metrics for Evaluating Regression Model Performance*. https://www.aionlinecourse.com/tutorial/machine-learning/evaluating-regression-models-performance
6. Yadav, H. (2022, January 6). *Multiple Linear Regression Implementation in Python*. Medium. https://medium.com/machine-learning-with-python/multiple-linear-regression-implementation-in-python-2de9b303fc0c