DATA MINING AND PREDICTIVE MODELING FOR CRYPTO-CURRENCIES

[Final Report]

James Beedle
University of Colorado
Boulder, CO USA
James.Beedle@Colorado.edu

Lauren Deans
University of Colorado
Boulder, CO USA
Lauren.Deans@colorado.edu

Inderpal Dhillon
University of Colorado
Boulder, CO USA
indh2640@colorado.edu

Theo Shin
University of Colorado
Boulder, CO USA
Ted.Shin@colorado.edu

ABSTRACT

In this paper, we describe a project conducted on five different cryptocurrencies (Bitcoin, Bitcoin Cash, Ethereum, Litecoin, and Ripple) where we performed multiple types of analysis to search for trends in our data in hopes of creating a predictive model for the different cryptocurrencies. The main questions we sought to answer include how a $1 increase in Bitcoin influences the other cryptocurrencies in our dataset, as well as how to best set up data to produce the most effective and accurate predictive models which will best be able to predict future trends. We utilized various data mining and machine learning techniques including multilinear regression analysis, k-nearest neighbor, decision trees, random forest model, and a Dickey-Fuller statistical test. These data analysis and machine learning techniques allowed us to answer our question of how a $1 increase in Bitcoin influences the other cryptocurrencies with the results including:

A $1 increase in Bitcoin will result in the following price value fluctuations:

BCH = - $0.0162

ETH = - $0.0182

LTC = + $0.0082

XRP = - 2.394e-05

After conducting the other statistical analysis methods (k-nearest neighbor, decision trees, and feature selection of moving averages and relative strength index), we concluded what the best features were to include within each model to yield the most accurate results. For the moving averages and relative strength index, we found that MA15 and MA200 along with RSI200 were the most optimal choices for features for the predictive models. We also found that using the K-nearest neighbor method, the optimal number of clusters was 34 given our dataset. We also concluded that using decision trees, a max-depth of 3 yielded the most optimal results to utilize with our predictive models.

KEYWORDS

Data Mining, Cryptocurrency, Machine Learning, Linear Regression, Predictive Modeling, Python, Bitcoin, Ethereum, Litecoin, Ripple

ACM Reference format:

James Beedle, Lauren Deans, Inderpal Dhillon and Theo Shin. 2019. Insert Data Mining and Predictive Modeling for Crypto-Currencies: Proposal Paper. *University of Colorado, Boulder, CO, USA, 4 pages.*

1. PROBLEM STATEMENT

With over one million data points structured to varying bit-currencies, our objective is to discover correlations involved with forecasted predictions and prevailing trends. We have looked deep into the data and have discovered interesting insights. Through data cleansing, our group focused on aggregating crypto data through prior years and following the removal of incomplete data, we handled the outliers and noisy data. Our preliminary pre-processing efforts led to a complete integration containing the additional datasets from alternate crypto currencies, in order to simplify our analyses and form prediction models.

Questions and deeper meanings our group sought to answer include which currency provides the best average returns for investments, which trends are consistent amongst various groups/countries, any unique correlations which can potentially be linked to our predictions, and which method can yield the most accurate forecast. Similarly, another question we sought to answer is whether a Bitcoin dollar increase necessarily lead to increases across the basket of currency indices? Our hypothesis was that the alternate cryptocurrencies may even have a negative relationship compared to Bitcoin.

Once correlations were discovered, we were able to become closer to understanding the reason why these correlations exist and determine how these trends came about. We were also able to determine which formulated prediction models can be the most accurate in the future and seeing how correlations are tied to our data analysis.

2. LITERATURE SURVEY (PREVIOUS WORK)

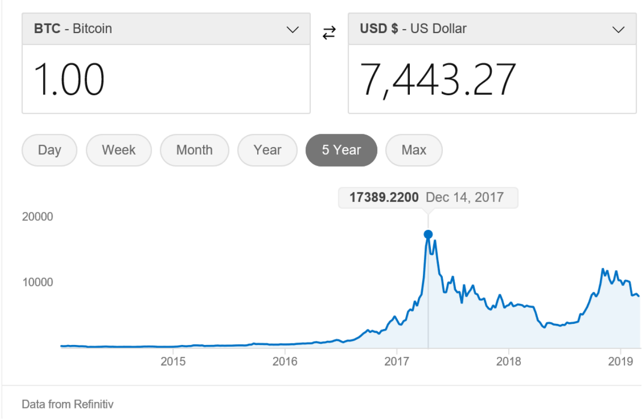
Bitcoin came onto the scene as the first decentralized cryptocurrency and was first released as open-source software in 2009 (created by alias Satoshi Nakamoto [1]) and since, there have been over 4,000 alternative variants of cryptocurrency in the market. Bitcoin has grown tremendously since then with prices less than 15 USD in 2012 to its current market value at 8122 USD (as of Oct 24, 2019). The historical high for Bitcoin occurred on Dec 14, 2017 at a value near $18,000 USD [2].

Coincidentally, data mining tools and methods have improved within the same timeframe which has encouraged many individuals to attempt prediction models and figure out when the next jump in price could take place.

Cryptocurrencies have become a popular form of transaction amongst merchants and investors for multiple reasons. To understand why more and more people are using cryptocurrencies for market exchange, let's look at the advantages:

- Not needing a middle man (no financial institutions and no transaction fees)
- Anonymous transactions and omission of personal details
- Facilitating International transactions (Bitcoin is not tied to any country's regulations)
- Investment opportunity (If one had invested $100 in Bitcoin in its early stages the investment would be worth north of $25 million).

The uncertainty and volatile nature underlying Bitcoin has created curiosity and even concern, as no one really knows what will become of Bitcoin. Looking at the pattern for the past 2 years, the constant fluctuation in the price of Bitcoin is a challenge for prediction. Despite its unevenness, Bitcoin's value still remains relatively high.



BTC - Bitcoin ⇄ USD $ - US Dollar

1.00 | 7,443.27

Data from Refinitiv

Figure 1: Five-year trend of Bitcoin value [2].

What our group would like to do is replicate and even continue the predictive analysis for cryptocurrencies using various data mining techniques. An applicable machine-learning model we are initially implementing are Time Series models to examine past patterns and trends to anticipate future fluctuations.

3. DATASET

The dataset includes a collection of five different cryptocurrencies along with their transactions. The datasets start on 8/15/18 and end on 9/22/18 inclusively, for a total of 39 days of acquired data with a total of 1,314,830 data points. The dataset includes the following four attributes:

- Trade price in USD
- Transaction date and time
- Coin valued amount per transaction
- Type of Coin

We found our dataset from https://www.kaggle.com/albala/ticks-bitcoin-ethereumlitecoin-ripple [4], though the site states it was originally aggregated from 'Coinbase.' The dataset is downloaded and accessible using our personal computers, as well as stored online in our GitHub repository. In CSV format, the dataset is 50.7MB in size. Since our datasets are static in nature, our analysis should be accurate so long as we mitigated any intrusive trends (using Dickey-Fuller) and preserved our relative data points following cleaning and pre-processing.

4. TOOLS

Python offers extensive access to built-in libraries and modules which will be utilized in our project that are especially useful for financial analysis and linear regression models. Specifically, we be utilizing modules such as seaborn through matplotlib, sklearn, pandas, numpy, nltk, scipy, and scipy.stats. We created a pandas dataframe containing cleaned and organized data which was then be converted to a numpy array, and then further our prediction model using k-neartest neighbor, decision trees, and multilinear regression models. Most of this analysis will be conducted within a Jupyter notebook. Microsoft Excel's currency converter will be utilized when necessary to relate each cryptocurrency to one another and relative to a USD. We will be conducting analysis using statistical methods such as Naïve Bayes classification models, as well as the machine learning task of supervised learning.

5. MAIN TECHNIQUES APPLIED

In order to begin the data collection and analysis, we first pre-processed the dataset. Using historical bitcoin data acquired from https://www.kaggle.com/albala/ticks-bitcoin-ethereumlitecoin-ripple, we began by formatting the CSV data into a Pandas DataFrame to facilitate exploratory data analysis (EDA) adding plots and calculations. We aimed to back the accuracy of our findings using statistics. Plots were developed in order to visualize price movements in our DataFrame between the various cryptocurrencies over a set amount of time. This involved adding a numerical tag to categorize and describe the cryptocurrency type:

1. Bitcoin cash
2. Bitcoin
3. Ethereum
4. Litecoin
5. Ripple

Since each alternate cryptocurrency is stored in separate CSV files, we combined all into a single list totaling 1,314,830 rows of data points. This invariably led to cleaning the data (null/missing, redundancies, and removal of noisy data). To account for redundancies, our plan was to remove repeatedly occurring data that did not aide in our analysis and predictions. For missing data, we had to determine values to drop or replace with the most applicable statistical measure. By choosing to replace missing data with the mean or other method, we would minimize any negative impact towards our conclusions and insights.

After dropping the categories 'order number,' 'buy order number,' 'sell order number,' and 'type' columns, the data was rearranged and grouped by correlations (chronological, closing costs) to increase efficiency in our analysis and prediction models and to make clearer sense of the data.

Once we were able to rearrange the data and begin working with the data, we performed exploratory data analysis (EDA) and we noted that the data would be much more efficient to work with if transposed into a matrix. This was conducted by generating a pivot table which put each unique value ("price") in a specified column ("type of coin"). Utilizing this pivot table function allowed us to convert back to the original data frame after conducting the analysis. We also casted the timestamp from an object type to a datetime64-bit type which allowed for facilitating time series analysis.

To account for missing data values, we utilized multiple different methods. First, we transformed the data matrix to convert all zero values to NaN, which indicated no trade occurring for that particular currency at that particular time. Then, we utilized a back-fill method to propagate the next values backward when performing the multilinear regression analysis, and a front-fill method when performing the remaining statistical analyses which propagated the previous value forward. Ultimately, the multilinear regression analysis also additionally performed the same methods using a front-fill propagation and there was not a significant difference between the two propagation methods. The data matrix for the multilinear regression analysis was also conducted by truncating the seconds down to minutes, and the results were compared to check for any correlations between the data. There was very little deviation between both methods (seconds vs. minutes) which showed us that utilizing minutes would not cause a change in correlation between the currencies.

One thing to note is that the value of bitcoin was significantly and consistently higher than any of the other coins. This can be displayed in Figure 2, which resulted in the need to normalize the data to better compare the correlations and work with the data.

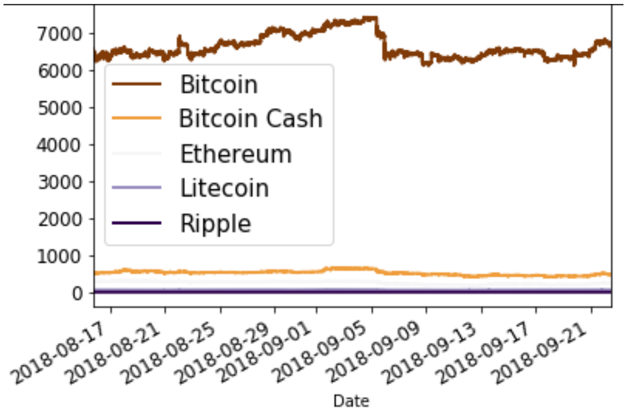


Figure 2: Time series of all cryptocurrencies

To account for the wide range between the cryptocurrency values (bitcoin often over 6500 vs. Ripple with values less than 1), we normalized the dataset. This allowed direct comparisons between the cryptocurrencies and better allowed us to visualize trends within a time series plot. Figure 3 below displays the normalized cryptocurrencies plotted on a time series graph with the time on the x-axis and the normalized value on the y-axis.

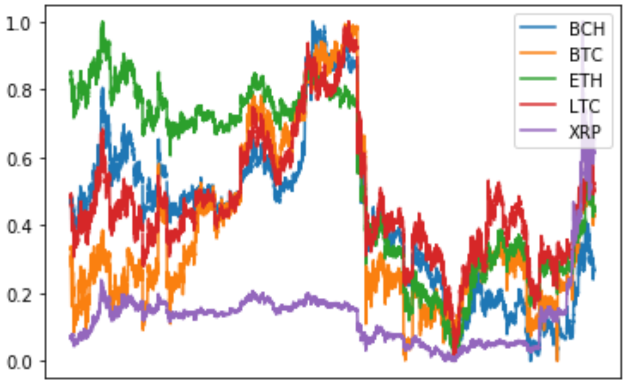


Figure 3: Time series of normalized cryptocurrencies

Our project will entail observing the historical data associated with Bitcoin along with the four other alternatives, which should provide interesting correlations or distinctions between the datasets.

Throughout our analysis, we were able to keep track of which correlations were prevailing and which form of cryptocurrency is directly influenced by others. If Bitcoin increases by a significant measure, does that necessarily mean Ethereum or Litecoin necessarily increase? And by how much?

To begin answering these questions, we first performed multilinear regression analysis on our data. This allowed for exploration of correlation between the different cryptocurrencies but stops short of allowing any predictive outcomes. The multilinear regression was utilized to search for correlation between the data, with other models implemented for predictive measures extending into the future. To visualize these potential correlations, a pair plot was generated and can be seen in Figure 4 below. Examining the pair plot, it appears there are correlations between many of the coins individually. One outlier is the XRP(Ripple) coin, which can be seen on the far right and far bottom plots. Some of the XRP regression lines are nearly flat. This coin seems to be moving differently compared to the other coins. On the other hand, LTC(Litecoin) has very high correlation with both BTC(Bitcoin) and BCH(Bitcoin Cash).

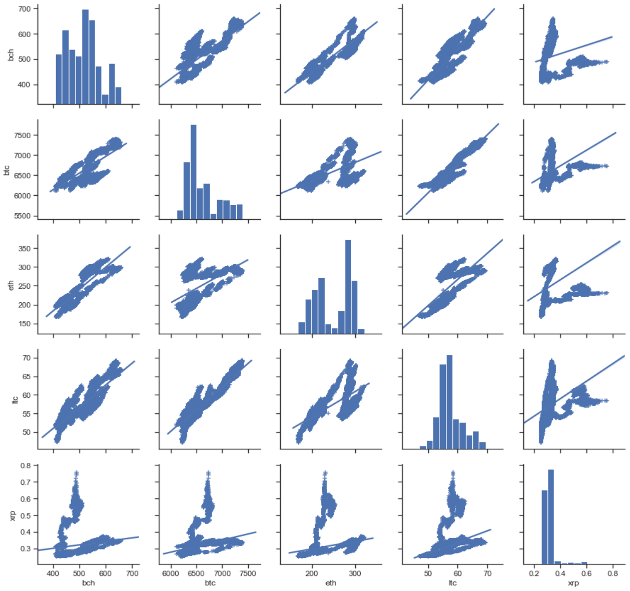


Figure 4: Pair-plot generated using multilinear regression analysis

Using OLS Least squares corroborates much of what had been predicted visually from the pair plot. In order to get a full grasp of how these coins relate to each other, MLN/OLS was performed on each coin. Here we are trying to predict the continuous value of one coin using the value of all other coins. We were looking for R^2 values close to 1, which would signify a high correlation for that particular coin. Figure 5 below shows the R^2 values for each coin, given the other coins.

BCH given [BTC,ETH,LTC, XRP]	$R^2 = .889$
BTC given [BCH,ETH,LTC,XRP]	$R^2 = .877$
ETH given [BTC,BCH,LTC,XRP]	$R^2 = .775$
LTC given [BTC,BCH,ETH,XRP]	$R^2 = .927$
XRP given [BTC,BCH,LTC,ETH]	$R^2 = .268$

Figure 5: R^2 values of each coin, relative to the other coins

These results agree with the visual inspection of the data from the pair plot. What is surprising is that even though XRP shows a lower correlation when predicted using all other coins, removing XRP from the regressions above give lower R^2 values. The LTC R^2 value is exceptionally high in this dataset. Suggesting that given other coin data, we would be able to predict the LTC value with reasonable accuracy.

We then took our original dataset with 'NaN' values and used the linear replacement method using the interpolate method(linear) that identifies all missing values and fills the values based on a linear path. Using the interpolate method, we were also able to revert to the backfilled dataset by changing the parameters within the code.

Before doing any further evaluation to check for any relationships between the coins we then ensured the reference coin, Bitcoin (BTC), fell into a normal distribution using a histogram of daily price change percent (Figure 6).

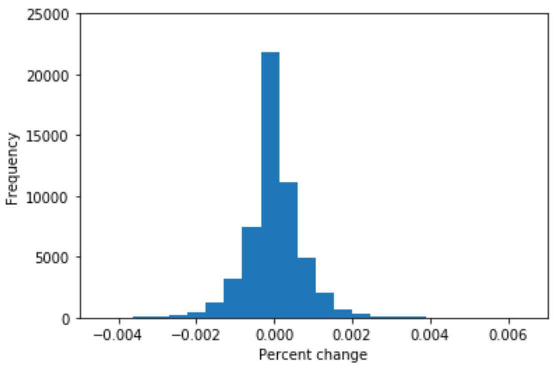
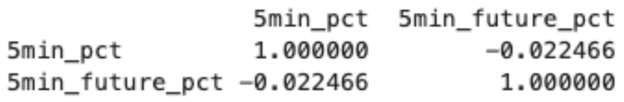


Figure 6: Histogram of BTC daily percentage price changes

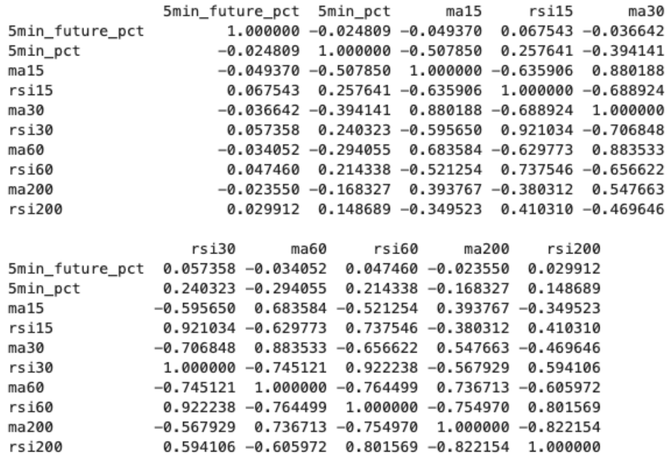
Above we can see a nearly normal distribution. We then used Pearson's correlation coefficient to detect any linear relationships. We checked the correlations between current price changes to see if previous price changes can predict future ones. In Figure 7 below, based on the outputted correlation matrix between 5-minute percentage changes (current and future), we can discern a slightly negative correlation (-0.0225) to the change in the last 5 minutes, an example of mean reversion (stock prices bounce around as opposed to following an upward trend).



```
                     5min_pct  5min_future_pct
5min_pct             1.000000        -0.022466
5min_future_pct -0.022466             1.000000
```

Figure 7: Correlation matrix between 5-minute percentage changes (current and future)

Using the 5-minute time period showed almost no correlation in the matrix and scatter plot, so we then performed the same calculations using 15, 30, 60, and 200-minute intervals. We used moving averages (ma) as well as relative strength index (rsi) indicators. The correlation matrix can be found below (Figure 8).



```
                 5min_future_pct  5min_pct      ma15     rsi15      ma30
5min_future_pct         1.000000 -0.024809 -0.049370  0.067543 -0.036642
5min_pct               -0.024809  1.000000 -0.507850  0.257641 -0.394141
ma15                   -0.049370 -0.507850  1.000000 -0.635906  0.880188
rsi15                   0.067543  0.257641 -0.635906  1.000000 -0.688924
ma30                   -0.036642 -0.394141  0.880188 -0.688924  1.000000
rsi30                   0.057358  0.240323 -0.595650  0.921034 -0.706848
ma60                   -0.034052 -0.294055  0.683584 -0.629773  0.883533
rsi60                   0.047460  0.214338 -0.521254  0.737546 -0.656622
ma200                  -0.023550 -0.168327  0.393767 -0.380312  0.547663
rsi200                  0.029912  0.148689 -0.349523  0.410310 -0.469646

                    rsi30      ma60     rsi60     ma200    rsi200
5min_future_pct  0.057358 -0.034052  0.047460 -0.023550  0.029912
5min_pct         0.240323 -0.294055  0.214338 -0.168327  0.148689
ma15            -0.595650  0.683584 -0.521254  0.393767 -0.349523
rsi15            0.921034 -0.629773  0.737546 -0.380312  0.410310
ma30            -0.706848  0.883533 -0.656622  0.547663 -0.469646
rsi30            1.000000 -0.745121  0.922238 -0.567929  0.594106
ma60            -0.745121  1.000000 -0.764499  0.736713 -0.605972
rsi60            0.922238 -0.764499  1.000000 -0.754970  0.801569
ma200           -0.567929  0.736713 -0.754970  1.000000 -0.822154
rsi200           0.594106 -0.605972  0.801569 -0.822154  1.000000
```

Figure 8: Correlation matrix for moving averages(ma) and relative strength index(rsi) for time intervals of 15,30,60, and 200 minutes

We can see some high correlations for relative strength index (RSI) rsi60 and rsi30, moving average (ma) ma60 and ma200, rsi15 and rsi30, and strong negative correlation between ma200 and rsi200, ma60 and rsi30. Compared to the target ['5min_future_pct'], rsi15 has the highest correlation at 0.066734**.**

We then set up a training dataset training set that is 85% of the number of samples to be used for preparing some prediction models and compare various performances between models discussed in our class such as decisions trees and k-nearest neighbors. We used linear model and least squares fit to determine p-values, searching for features with p <= 0.05 which are typically considered significantly different from 0. We then used predictions from our model for train and test sets and plotted them to compare the outcomes (Figure 9).

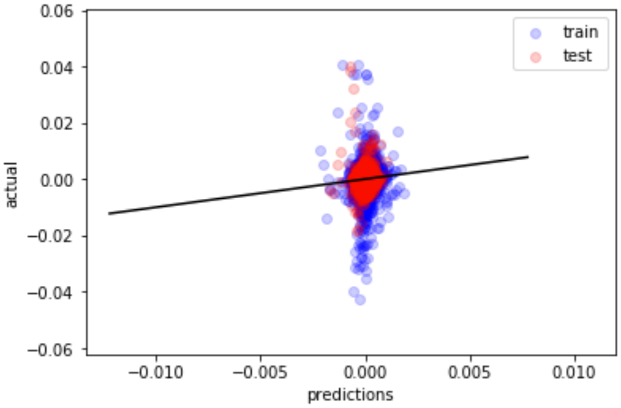


Figure 9: Scatterplot using the training and testing dataset and perfect prediction line.

We also utilized a non-linear machine learning model of decision trees. Decision tree split our data into groups based on the features. We used this to determine which max depth can provide the best prediction without overfitting. We fit the model to the training features and targets, and then checked the score on the train and test datasets. This showed a near perfect fit of our training data (0.98698) but not with our testing data (-1.22532). Testing multiple max depths of 3,5, and 10, we were able to see the best fit is a max depth of 3 at a score of -0.02125.

We also briefly looked at a random forest model. We used random sample of training data points to test our results. We fit our models on the oldest data and evaluated on the newest data. To accomplish this, we used sklearn's ParameterGrid to create combinations of hyperparameters to search. We used the best scores to fit a random forest model and generated a scatter plot with train/test actual vs predictions. We obtained the feature importances from our random forest model, which can be seen plotted below (Figure 10) in a bar chart from greatest to least importance. We can see moving average 200 (ma200) and 15 minutes along with relative strength index 200 (rsi200) minutes as the most contributing features to the predictions. We focused on these features for even stronger predictions.

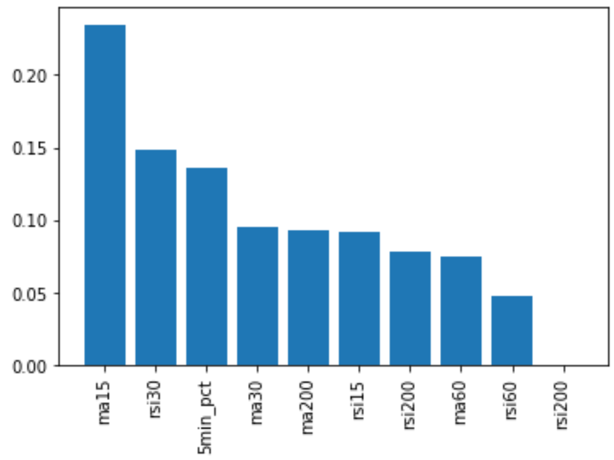


Figure 10: Bar chart of features in order of importance for random forest model

We then implemented the k-nearest neighbor method. As discussed in lecture, KNN takes the k-nearest points to a new point and averages their target values to obtain a prediction. We scaled the data, and then fit the KNN model to the training data. From the model we were able to see which number of neighbors yielded the highest n-score, which resulted in an optimal number of 34. The scatterplot of the KNN model can be found below in Figure 11 where actual vs. predicted values were plotted on the x and y axes respectively**.**

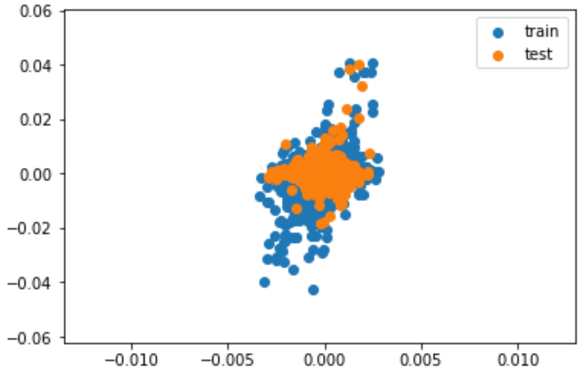


Figure 11: Scatterplot of k-nearest neighbor model with actual vs. predicted values for the train and test datasets.

As previously mentioned, prediction forecasting for cryptocurrencies has been developed and studied in the past. Many individuals seek to predict not only cryptocurrencies, but the stock market, housing prices, and other investment vehicles to discover financial opportunities. Some of the past work have aspects of technical work beyond the scope of our project (TensorFlow, Keras, Deep Learning), however, it is interesting to glean into one of many possibilities using data science practices [6].

We then examined the time period (denoted by green dashlines in Figure 12) when Bitcoin prices drop into a slight valley. Prior to the drop, Bitcoin was very steadily increasing at a positive rate. It was, however, difficult to discern any sort of trends between the other currencies due to the wide range in currency values and Bitcoin being significantly higher. To better analyze the trends in each currency, a sublot was generated for each particular currency (Figure 13).

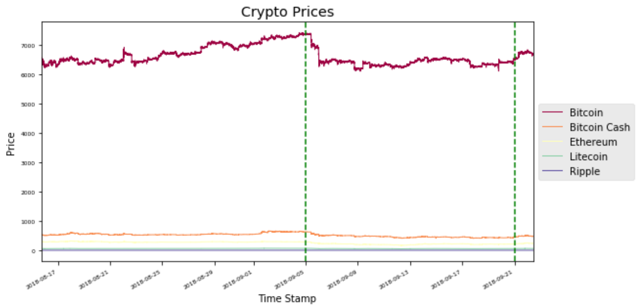


Figure 12: Time series of the different cryptocurrencies with the green dashed line indicating significant area of interest for our analysis

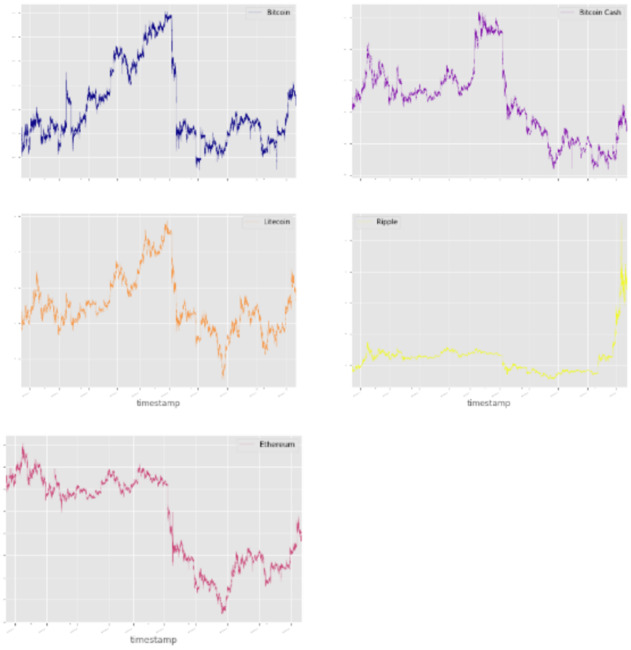


Figure 13: Time series of each cryptocurrency within its own subplot

We can observe that for most of the coins, around 09-04-2018 (~middle of x-axis) the prices all similarly dropped. From this we can note that it appears there is at least a direct negative correlation as all currencies were affected by a negative drop in value. Next we wanted to determine if and how the coins are correlated with one another. Here, we computed the correlation coefficients using Pearson and Spearman methods, depending on whether the relationships are thought to be linear or not.

By using the Spearman method, though not necessarily ordinal values, we ranked each coin and correlation coefficients, which can then be used to summarize the monotonic (entirely nonincreasing or nondecreasing) strength and direction of the relationships. Later, we used the Pearson correlation method to determine the linear strength and direction between the variables. Comparing the Pearson graph below (Figure 15) with the produced Spearman Graph (Figure 14), we can denote anytime correlation values for (S)pearman > (P)earson (Litecoin/Ethereum; Bitcoin Cash/Ripple), we have a correlation that is monotonic but not linear. With Pearson correlation higher, we can discern the linear correlation is larger than rank. This may be due to influential observations in the distribution tails having a large influence relative to ranked values. If linearity holds in our dataset, we can associate Pearson correlation to be of stronger measure.

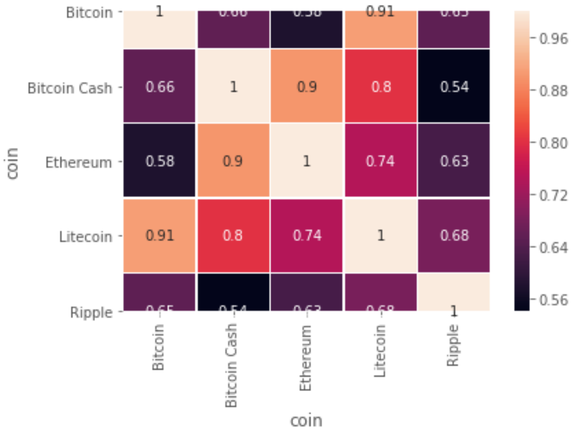


Figure 14: Heatmap of correlation matrix using Spearman coefficient method

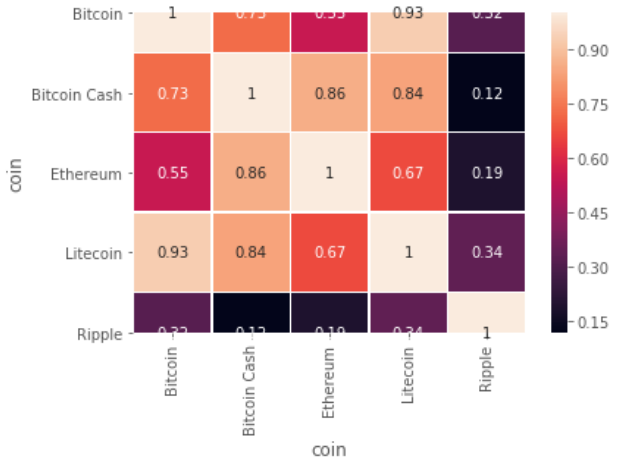


Figure 15: Heatmap of correlation matrix using Pearson coefficient method

You can see from the heatmap correlation matrices above using both the Spearman and the Pearson coefficients show high correlation between Bitcoin and Litecoin.

Next, let's introduce some Time series decomposition as it can be a great way to reveal the time series structure. We obtained the seasonal decomposition and visualized the components which can be seen below (Figure 16).

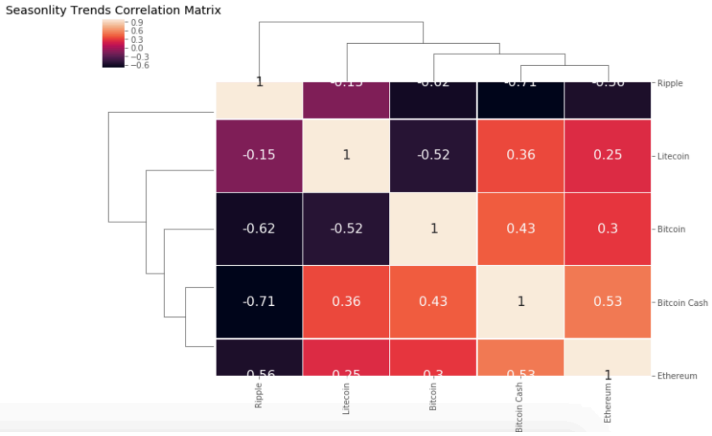


Figure 16: Seasonality trends matrix for all cryptocurrencies

From the seasonality trends matrix above, we can say Ripple is negatively correlated with Bitcoin Cash (-.71), Ripple has some positive correlation with Ethereum (0.56), and there does not seem to be much correlation between Bitcoin with Litecoin (-0.15). * Due to the 1-year timeframe of our dataset, the seasonality trends would certainly make for stronger analysis on a larger timeframe.

Taking a closer look at just the Bitcoin trend, let's revisit and highlight the steep drop occurring at 2018-09-04 which is displayed below (Figure 17).

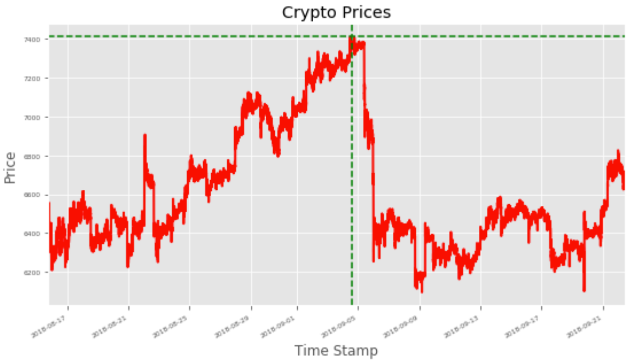


Figure 17: Time series graph of Bitcoin, with the green dashed line indicating a steep drop on 2018-09-04

Looking at the summary statistics and the graph above, we found the max value for Bitcoin was 7410.80 occurring on 2018-09-04, before the price dropped a low of 6094.47. Also, we calculated the 5% and 95% values as 6250.62 and 7237.37 for Bitcoin. We then examined a box plot for Bitcoin and zoomed in on whiskers and outlying data (Figure 18).

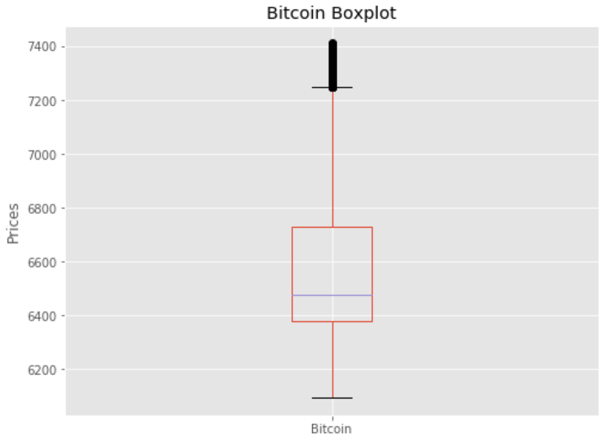


Figure 18: Boxplot of Bitcoin to examine whiskers and outliers

We also tested autocorrelation for Bitcoin, which yielded autocorrelation values close to 0, so we can conclude values between consecutive observations are not correlated with one another (Figure 19).

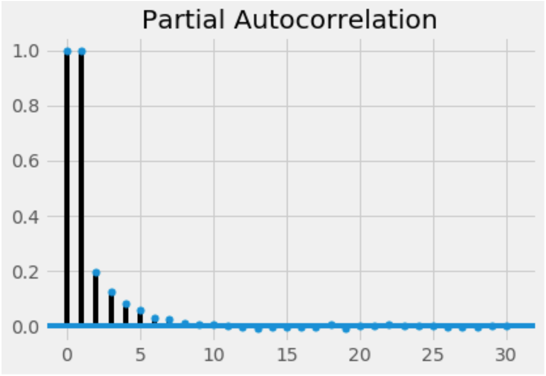


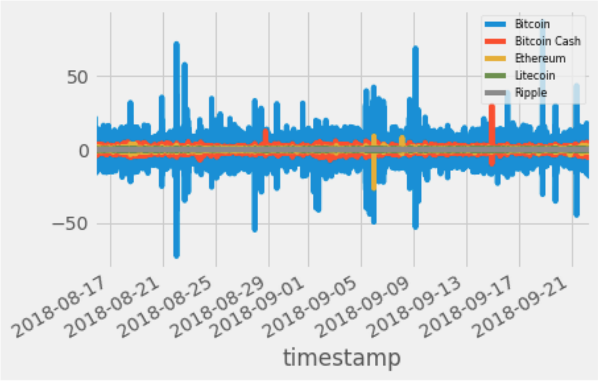
Figure 19: Testing autocorrelation for Bitcoin, which yielded results mostly close to zero signifying a lack of correlation

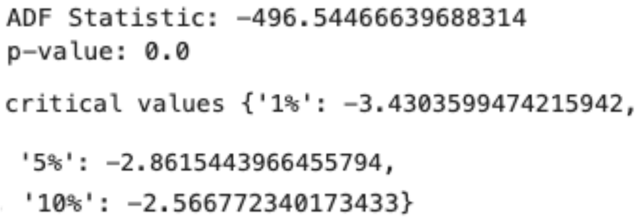
What if we wanted to be able to predict the future price of the coins based on our data and past trends? To do this, we first needed to split our data into train-test splits so we could the quality of our model fit.

A statistical implementation we feel is important to our analysis is the Dickey-Fuller statistical test, which implements a stationarity and thus removing possible intrusiveness to our dataset and allowing for better performing prediction models. With the Dickey-Fuller test, we aimed at P-Values less than the critical 5% and attempted at lowering P-Values as close to zero as possible for a high measure of accuracy [3]. To test, the Null Hypothesis (H_0) can be represented as the time series being not stationary and alternatively, our H_1 hypothesis will state the time series is stationary. As already mentioned, we can reject the Null Hypothesis if P-Values are determined to be less than or equal to 0.05 and conclude the data is stationary [8].

In order to statistically test whether the null hypothesis in our time series data is non-stationary due to trend, we can implement the augmented Dicky-Fuller test (this can be imported in Python as adfuller). If we can decide the data is non-stationary, then we will have to transform it into a stationary set prior to making our predictions. This can often be done by transforming the data by taking the difference, log, square root, or proportional change. We also looked to find the simplest yet effective implementation. This time, we switched over to the Litecoin column of the dataset to determine if it's stationary (p-value < 0.05

significance). If we wanted a p-value of 0.05 or below, the test statistic (ADF Statistic) needs to be below the 5% (or -2.8615 below) critical value of the test statistic. The results can be found below in Figure 20.





```
ADF Statistic: -496.54466639688314
p-value: 0.0

critical values {'1%': -3.4303599474215942,

 '5%': -2.8615443966455794,
 '10%': -2.566772340173433}
```

Figure 20: ADF results, p-value, and critical values of Dickey-Fuller test

6. KEY RESULTS

Throughout our investigative analysis, we were able to form multiple conclusions relating to our dataset relating to our proposed work. One question we sought to answer was how much a $1 increase in bitcoin would affect the other cryptocurrencies. Using the coefficient measure from the MLR summary helps us answer this question. Given a 1$ increase in bitcoin, the other cryptocurrencies reactions can be seen in Figure 21. BCH, ETH, and XRP on average lose value when bitcoin goes up. Bitcoin cash (BCH) was created during a "split" from bitcoin. In this sense, the market cap is a zero-sum game where loss in confidence in one coin can lead to a gain with the other.

BCH = - $0.0162
ETH = - $0.0182
LTC = + $0.0082
XRP = - 2.394e-05

Figure 21: Bitcoin's influence on other coins as Bitcoin increases $1

Another interesting cryptocurrency to examine influence of is Litecoin(LTC) which has the highest correlation in our regression summaries to Bitcoin. We see that a one dollar increase in BTC, BCH, XRP all lead to increases with LTC. It seems that with the exception of ETH, increases in other coins are good for LTC. This can be seen in Figure 22.

BTC = + 0.082
BCH = + 0.0278
ETH = -.0048
XRP = + 5.4566

Figure 22: Litecoin's influence on other coins as Litecoin increases $1

After conducting the other statistical analysis methods (k-nearest neighbor, decision trees, and feature selection of moving averages and relative strength index), we concluded what the best features were to include within each model to yield the most accurate results. For the moving averages and relative strength index, we found that MA15 and MA200 along with RSI200 were the most optimal choices for features for the predictive models. We also found that using the K-nearest neighbor method, the optimal number of clusters was 34 given our dataset. We also concluded that using decision trees, a max-depth of 3 yielded the most optimal results to utilize with our predictive models.

7. APPLICATIONS

Looking at this at a high level, this model will attempt to tell us if the current market price of a coin is below what it is really worth in comparison to the other coins because of the fairly tight correlation of price movements, especially with LTC. If we assume what we have observed in the past is true for the future, we can trade based on this. In the event that we are correct we will make a ton of money, but if we are incorrect, we could amplify our losses.

Consider these fabricated trades as an example:

2018-08-15 18:55:02 BTC traded at 6519.39	LTC traded at 57.82
2018-08-15 18:55:03 BTC traded at 6519.60	
2018-08-15 18:55:05 BTC traded at 6520.39	

The price of BTC went up by a dollar in a short span of 4 seconds. The last LTC trade executed was at the first second of the minute at 57.82. According to our algorithm, the value of LTC at 18:55:05 given the rise in BTC is 57.8282. Our goal now is to buy as many LTC coins in the open market for under 57.8282 as possible. During the inverse event when the price is dropping, our algorithm will attempt to sell coins it deems as overvalued. If our model turns out to be accurate, and we are fast enough. Over the long run, we will be able to keep these small differences between our predicted value of a coin and its market value as profit.

REFERENCES

[1] Wikipedia. 2019. Wikipedia: The Free Encyclopedia. Retrieved from http://en.wikipedia.org/wiki/Bitcoin.

[2] Refinitiv. 2019. Retrieved from http://Refinitiv.com

[3] Bhavesh Bhatt. 2019. Adf-test-stationarity-python. (October 2019). Retrieved October 24, 2019 from https://github.com/bhattbhavesh91/adf-test-stationarity-python/commits/master/augmented-dickey-fuller-test-python.ipynb.

[4] DataJuicers. 2018. DataJuicers cryptodatabase: bitcoin, ethereum,litecoin, ripple. (September 24, 2018). Retrieved on October 24, 2019 from http://www.kaggle.com/albala/ticks-bitcoin-ethereumlitecoin-ripple.

[5] Wikipedia. 2019. Wikipedia: The Free Encyclopedia. Retrieved from http://en.wikipedia.org/wiki/Cryptocurrency.

[6] Cryptocurrency Price Prediction Using LSTMs | TensorFlow. 2019. Retrieved from https://towardsdatascience.com/cryptocurrency-price-prediction-using-lstms-tensorflow-for-hackers-part-iii-264fcdbccd3f.

[7] Machine Learning Mastery. 2019. Retrieved from https://machinelearningmaster.com/arime-for-time-series-forecasting-with-python/.

[8] Stationarity and Differencing. 2019. Retrieved from https://people.duke.edu/~ma/411diff.htm.