International Institute of
Information Technology, Bangalore

Milestone 4 - Clustering and Association
rules
DS 707 Data Analytics

Cryptocurrency Analysis and Blockchain Understanding

Akanksha Dwivedi - MT2016006
Hitesha Mukherjee - MS2016007
Nayna Jain - MS2017003
Tarini Chandrashekhar - MT2016144

Instructors :
Prof. Ramanathan Chandrashekhar
Prof. Uttam Kumar

November 26, 2017

Contents

1 Background

As with stock market, cryptocurrency is a growing investment area for the daily traders and investors. Cryptocurrency is very new and still stabilizing because of which it is very volatile in nature and might get affected by different factors. In such an environment, it is very difficult to predict the price movement of cryptocurrencies. Since the trend and investment point of view is similar to stock market, the established techniques used in stock market prediction, are considered for cryptocurrency.

2 Trial & Error on Classification Model

To improve the accuracy of the classification models built, the following criteria were used:

2.1 Application of Dimensionality Reduction

According to the Wikipedia, dimensionality reduction or dimension reduction is the process of reducing the number of random variables under consideration by obtaining a set of principal variables. The main linear technique for dimensionality reduction, principal component analysis, performs a linear mapping of the data to a lower-dimensional space in such a way that the variance of the data in the low-dimensional representation is maximized. In practice, the covariance (and sometimes the correlation) matrix of the data is constructed and the eigen vectors on this matrix are computed. The eigen vectors that correspond to the largest eigenvalues (the principal components) can now be used to reconstruct a large fraction of the variance of the original data. Moreover, the first few eigen vectors can often be interpreted in terms of the large-scale physical behavior of the system[citation needed][why?]. The original space (with dimension of the number of points) has been reduced (with data loss, but hopefully retaining the most important variance) to the space spanned by a few eigenvectors. We used 'prcomp' function to compute the principal components which uses Singular Value Decomposition technique. The number of principal components were selected from the scree plot and elbow curve on the respective datasets, which clearly highlights the important principal components. After choosing the principal components, the data was again projected back(with reduced dimension) to be used for classification models. We could achieve only a maximum of 58% accuracy after applying PCA.

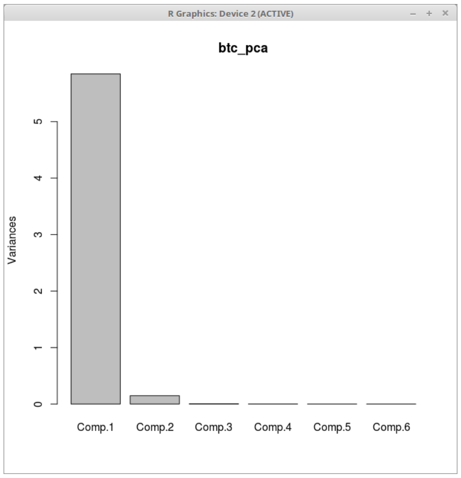

Figure 1: Scree plot of principal components of bitcoin

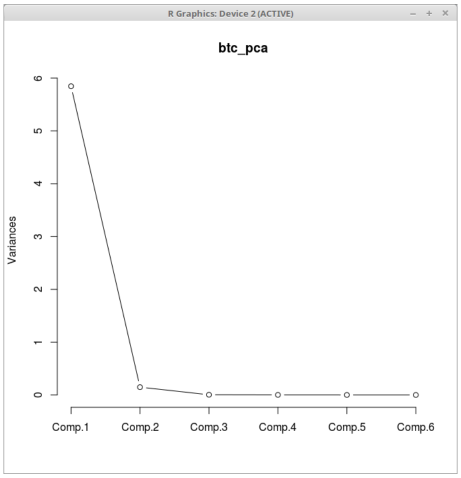

Figure 2: Elbow Curve on bitcoin dataset

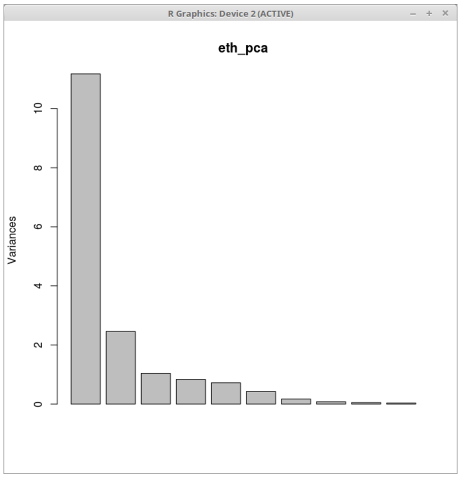

Figure 3: Scree plot of principal components of ethereum

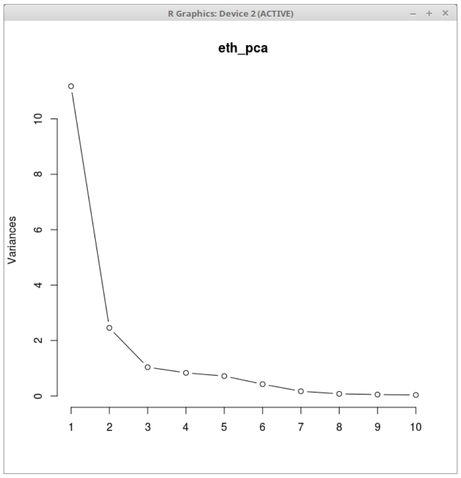

Figure 4: Elbow Curve on ethereum dataset

2.2 K-Fold Loss Cross Validation Strategy

Cross-validation is a technique to evaluate predictive/classiification models by partitioning the original sample into a training set to train the model, and a test set to evaluate it. In k-fold cross-validation, the original sample is randomly partitioned into k equal size subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k-1 subsamples are used as training data. The cross-validation process is then repeated k times (the folds), with each of the k subsamples used exactly once as the validation data. The k results from the folds can then be averaged (or otherwise combined) to produce a single estimation. The advantage of this method is that all observations are used for both training and validation, and each observation is used for validation exactly once. For classification problems, one typically uses stratified k-fold cross-validation, in which the folds are selected so that each fold contains roughly the same proportions of class labels. In our case, we have use Leave-One-Out cross validation strategy, in which only one datapoint is left as testing point which the remaining are used for training. This process is repeated for 10 times i.e. in k folds. The plot below shows the different combinations of hyper-parameters used and the respective accuracy achieved with model. A maximum accuracy of 54% was achieved with this strategy. From the above trial and error strategies, we inferred that this low accuracy of the classfication models might be due to insufficient number of datapoints and the granularity of datapoints available.

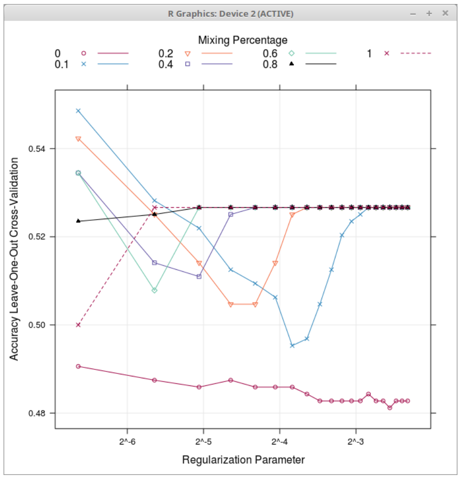

Figure 5: Plot of Leave One Out Cross Validation Accuracy with regularization parameters

3 Clustering Techniques

3.1 K-Means Clustering

[1][2] refers to the papers which shows how clustering is used in the process of stock price prediction. We have applied similar techniques from [1] for the cryptocurrency forecasting. [1] discusses three steps for forecasting:

- Normalizing the data.
- K-means clustering of the normalized data to identify the outliers.
- ARIMA model for forecasting on clustered data.

This is done to see how clustering is applied to remove the outliers from the clusters. It is a centroid based partitioning technique that uses the centroid of a cluster, C$_i$ to represent the cluster. Conceptually, the centroid of a cluster is its center point. This algorithm requires to specify the number of clusters (k) beforehand. This method is not guaranteed to converge to the global optimum and often terminates to a local optimum. The algorithm on Close/High attributes of the daily prices. Out of cluster values from 3, 4 and 5, we found that cluster values of 4 and 5 give similar values. Hence, we attempt to identify the outliers based on these attributes, with 4 clusters.

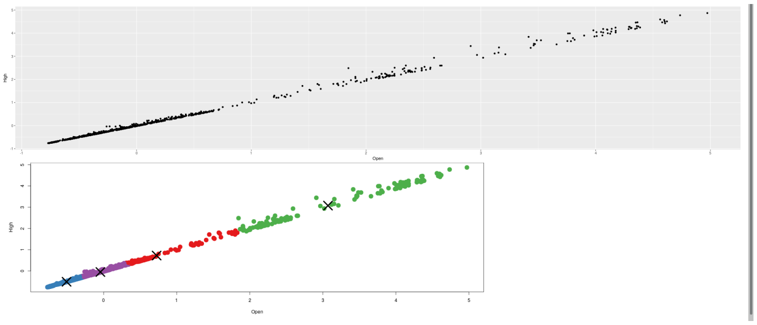

Figure 6: Before and After K Means Clustering

3.2 Hierarchical Clustering

Hierarchical clustering algorithm called CHAMELEON that measures the similarity of two clusters based on a dynamic model. In the clustering process, two clusters are merged only if the inter-connectivity and closeness (proximity) between two clusters are comparable to the internal inter-connectivity of the clusters and closeness of items within the clusters. We have considered the Bitcoin Data Set which has 24 features or attributes in it. We have extracted 16 important features and build a subset of the data. This dataset has been clustered using Chameleon. From the below figure we see that we have six clusters of different size, shape, and orientation, as well as random noise points and special artifacts such as streaks running across clusters.In this case CHAMELEON finds eleven clusters, out of which six of them correspond to the genuine clusters in the data set, and the rest contain outliers.

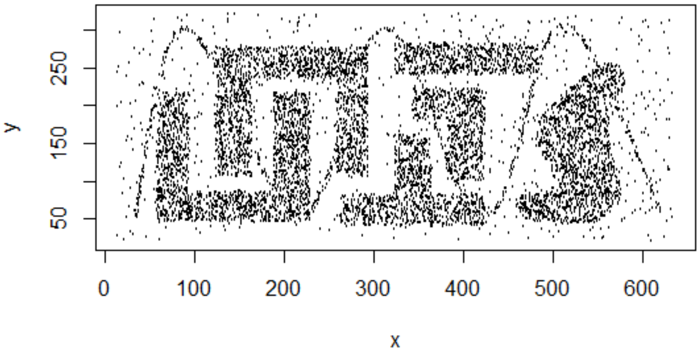

Figure 7: Chameleon Clustering Bitcoin

The next data set is ethereum Dataset with 12 attributes. From Figure 8 we see it has eight clusters of different shape, size, and orientation, some of which are inside the space enclosed by other clusters. Moreover, it also contains random noise and special artifacts, such as a collection of points forming vertical streaks. In this case CHAMELEON also finds eleven clusters, out of which nine of them correspond to the genuine clusters, and the rest contain outlier points.

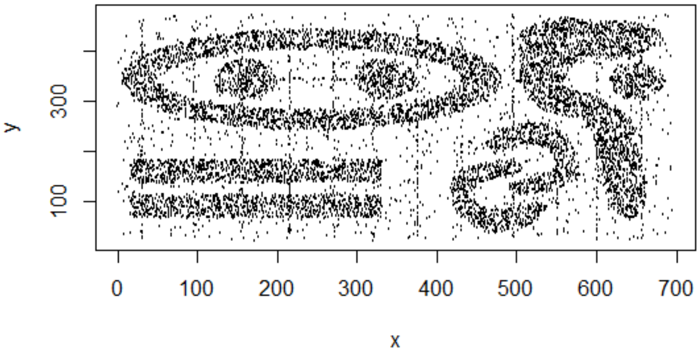

Figure 8: Chameleon Clustering Etherium

3.3 Partition along medioids clustering(PAM) : CLARANS

CLARANS (Clustering Large Applications based uponRANdomized Search) was proposed to improve the quality and the scalability of CLARA. It combines sampling techniques with PAM.It does not confine itself to any sample at a given time. It draws a sample with some randomness in each step of the search.
Advantages

- Experiments show that CLARANS is more effective than both PAM and CLARA.
- Handles outliers
  Disadvantages
- The computational complexity of CLARANS is O(n^2), where n is the number of objects.
- The clustering quality depends on the sampling method

We take the Bitcoin Dataset into consideration. It has 16 attributes. We have considered 2 attributes(Market Price Label and Market Cap) for CLARANS Clutering. From Figure 9, we observe that the Market Cap and Market Price varies almost in the same way for negative and positive values. It means that they are highly correlated. There is a similarity in their pattern, hence the clusters are based on similarity measure. We have used Euclidean Distance Parameter for Computation.

The next dataset we considered is the BitCoin Price Dataset which has 7 attributes. From below figure 10, we have considered 2 attributes(Volume and Market Cap) for CLARANS Clutering. We observe a certain randomness about

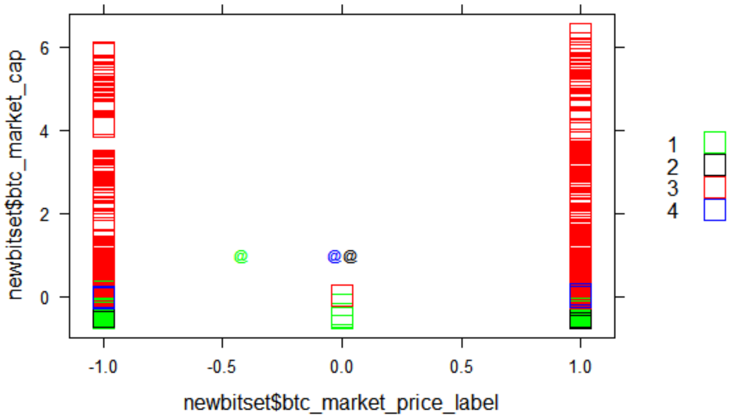

Figure 9: Clarans Clustering for Market Price and Market Cap

the clusters. No similarity pattern is observed. The volume and Market Cap vary randomly.

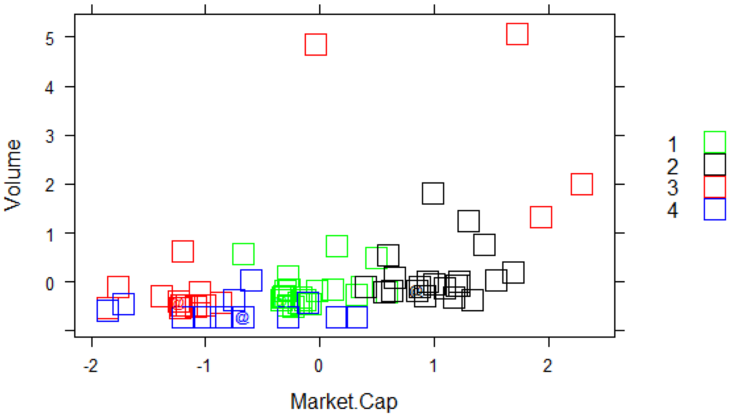

Figure 10: Clarans Clustering for Market Cap and Volume

4 Density-Based Clustering : DBSCAN

Density-based clustering is a technique that allows to partition data into groups with similar characteristics (clusters) but does not require specifying the number of those groups in advance. In density-based clustering, clusters are defined as dense regions of data points separated by low-density regions. Density is measured by the number of data points within some radius.

Advantages of density-based clustering:
As mentioned above, it does not require a predefined number of clusters,clusters can be of any shape, including non-spherical ones,the technique is able to identify noise data (outliers).

Disadvantages:
Density-based clustering fails if there are no density drops between clusters, it is also sensitive to parameters that define density (radius and the minimum number of points); proper parameter setting may require domain knowledge.

This algorithm works on a parametric approach. The two parameters involved in this algorithm are: e - (eps) is the radius of our neighborhoods around a data point p.Using these two parameters, DBSCAN categories the data points into three categories:

- minPts (K)- It is the minimum number of data points we want in a neighborhood to define a cluster.
- Core Points: A data point p is a core point if Nbhd(p,eps) [eps-neighborhood of p] contains at least minPts; $|Nbhd(p, eps)| \geq$ minPts. Border Points: A data point *q is a border point if Nbhd(q, eps) contains less than minPts data points, but q is reachable from some core point p.
- Outlier: A data point o is an outlier if it is neither a core point nor a border point. Essentially, this is the "other" class.

The idea is to calculate, the average of the distances of every point to its k nearest neighbors. The value of k will be specified by the user and corresponds to MinPts.We have used K-Nearest Neighbour Algorithm to determine the K value (minPts) to be used as a parameter for the DBSCAN Algorithm. The minimum value obtained for K was 5.

The value of epsilon is calculated in the following manner. The function kNNdistplot() [in dbscan package] can be used to draw the k-distance plot. k-distances are plotted in an ascending order. The aim is to determine the “knee”, which corresponds to the optimal eps parameter.

A knee corresponds to a threshold where a sharp change occurs along the k-distance curve. From the below figure, It can be seen that the optimal eps value is around a distance of 0.15.

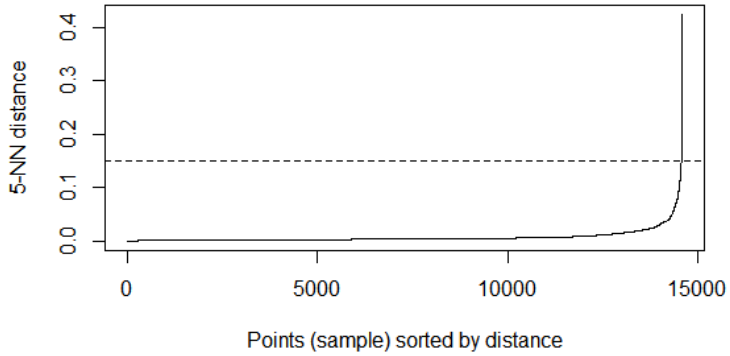

Figure 11: K-Distance Curve for determining Eps Value

We take the Bitcoin Dataset into consideration. It has 16 attributes. We have considered 2 attributes(Total Bitcoins and Market Cap) for DBSCAN Clustering.

From the figure 12, the variation of Total Bitcoins and Market Cap is constant for negative values, but as the value increases, we observe a sharp spike after zero for total BitCoins with respect to the Market Cap. The blue Cluster has maximum density, it is densely connected and has maximum density reachable value. We observe there are two outliers in the dataset.

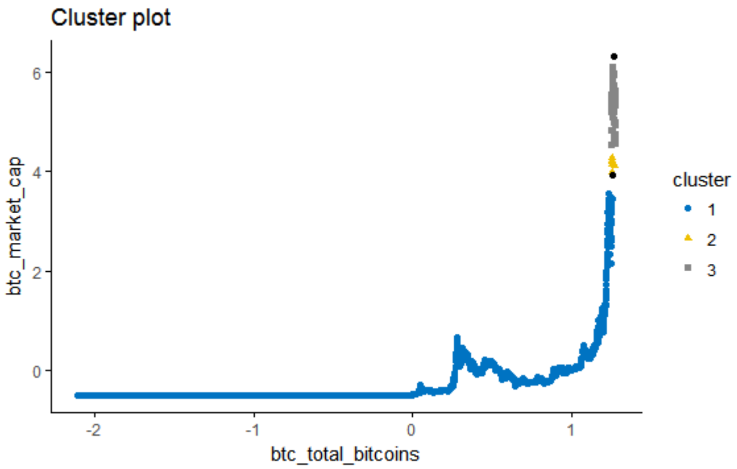

Figure 12: DBSCAN Clustering for Total Coins and Market Cap

We have considered 2 attributes Total Bitcoins and Trade Volume for DBSCAN Clustering. From figure 13, we observe that there is sharp change in the value of Total Bitcoins and Trade Volume. Also there is an overlap of clusters(Blue Cluster and Gray Cluster).The entire Blue Cluster and some parts of the Gray Cluster can be considered as core points and the remaining parts of gray cluster are the Border Points. There are 2 points away from the clusters one is yellow point and the other single gray point, which can be termed as outliers.

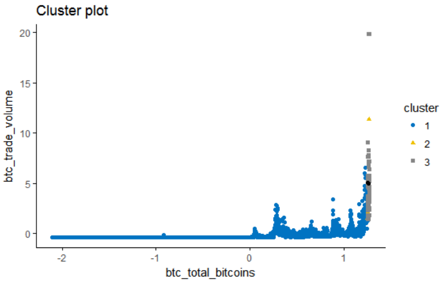

Figure 13: DBSCAN Clustering for Total Coins and Trade Volume

We have considered 2 attributes Market Cap and Trade Volume for DBSCAN Clustering. From Figure 14, we observe that the first cluster is very dense. The second and third clusters are very sparse. We observe that as the Market Cap increases the clusters become less denser, After the market cap and trade volume crosses the value of 5 in the x and y axis we observe very few points lie on that region. We can therefore conclude that both Market Cap and Trade Volume are directly proportional to each other. We also observe 2 outliers in the dataset.

5 Association Rules Mining

Association rule learning is a rule-based machine learning method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using some measures of interestingness It is often used by grocery stores, retailers, and anyone with a large transactional databases.Association rules use the R arules library. The arulesViz add additional features for graphing and plotting the rules.

We have considered the Bitcoin Data Set which has 24 features or attributes in it. We have extracted 16 important features and build a subset of the data.First we have converted our dataset to transactions which is necessary step for rule creation. The rules can then be created using the Apriori function on the transaction dataset. After running the Apriori Function 860931 rules were created and the following statistics were obtained; Confidence of 0.8 and Minimum Support count of 292. The figure 15 illustrates it.

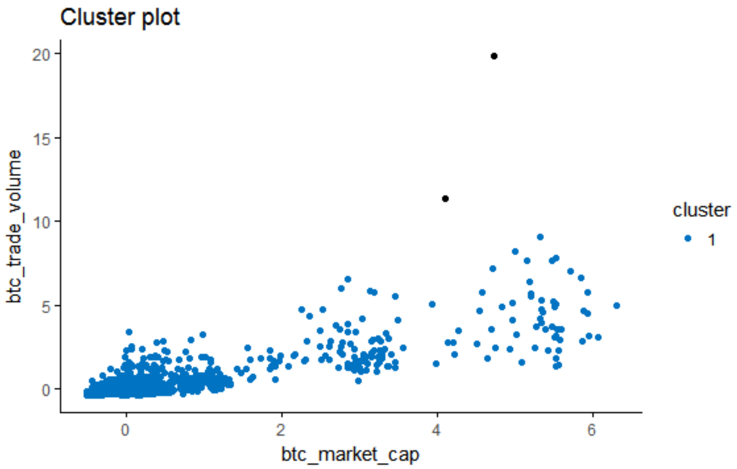

Figure 14: DBSCAN Clustering for Market Cap and Trade Volume

We can now create the subset of rules to visualise the set of rules. Every rule is composed by two different sets of items, also known as itemsets, X and Y, where X is called antecedent or left-hand-side (LHS) and Y consequent or right-hand-side (RHS).From figure 17 we get these graphs we can see the two parts to an association rule: the antecedent (IF) and the consequent (THEN). These patterns are found by determining frequent patterns in the data and these are identified by the support and confidence. The support indicates how frequently the items appear in the dataset. The confidence indicates the number of times the IF/THEN statement on the data are true.These IF/THEN statements can be visualized by the following graph.

We can then subset the rules to the top 30 most important rules and then inspect the smaller set of rules individually to determine where there are meaningful associations. We plot the graph as in the above figure for top 30 rules and inspect them to find meaningful correlation between different attributes based on LHS and RHS. In practical applications, a rule needs a support of several hundred transactions before it can be considered statistically significant and datasets often contain thousands or millions of transactions.

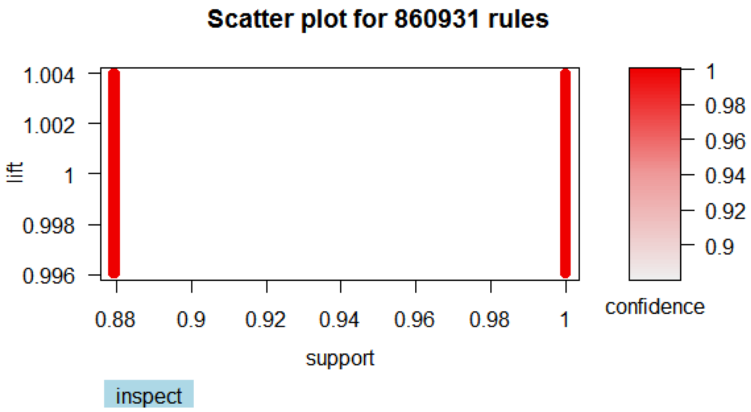

Figure 15: Association Rule Mining Scatter Plot

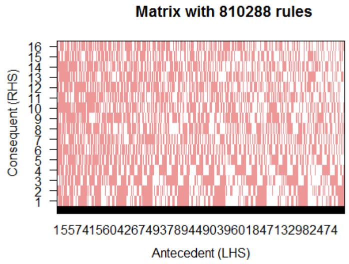

Figure 16: Antecedent Consequent for the Subrules Created

6 Time Series Analysis

6.1 Introduction

Time Series data is the collection of observations which are collected over a period of time, generally over fixed intervals. The data can be divided into univariate or multi-variable based on number of attributes. Time series data analysis is applied for the purpose of:

- Understanding what happened in the past (Trend).
- Predicting/Forecasting the future.

6.2 Stationary Series

One of the basic requirement for time series modeling is to first check whether the data is stationary or non-stationary. The data is said to be stationary if it

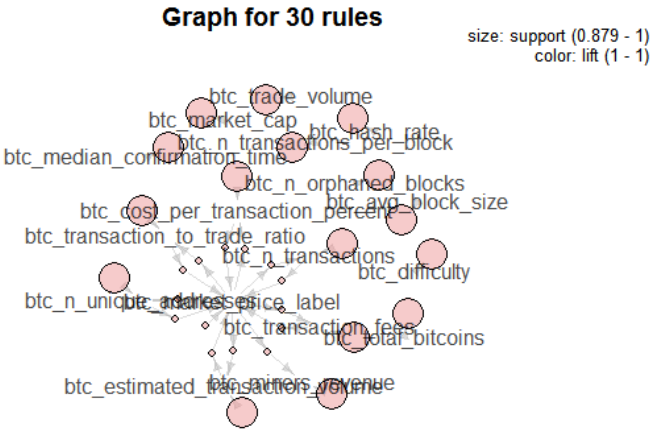

Figure 17: Graph for 30 Rules

satisfies the following property:

- The mean of the series should not be a function of time but should be a constant.
- The variance of the series should not be a function of time.
- The co-variance of the ith term and (i + m)th term should not be a function of time.

So, for the time series modeling, if the data is not stationary, it has to be first converted into stationary data. To verify whether data is stationary or not, we can use Dickey Fuller Test of Stationarity. For the Dickey-Fuller Test, the p value has to be less than 0.05 or 5. If it satisfies the above condition, it is considered as stationary series, else it is a non-stationary series. To convert non-stationary into stationary, differencing should be applied before proceeding.

6.3 Time Series Components

There are four components of the Time Series:

- Trend Component: Movement of data over a period of time like whether increasing, decreasing or remaining unchanged are defined as Trend.
- Seasonal Component: Fluctuations observed during specific seasons or chunks of time e.g. during 4th quarter of very year, stocks tends to show better.
- Cycle Component: Some type of data tends to repeat itself over a longer period of time, thereby exhibiting some cycles. It can be combination of trend and seasonal data.

- Irregular Component: Randomness or noise in the data.

The time series modelling can be represented as additive or multiplicative model. Additive Model:

$$Y(t) = T(t) + S(t) + C(t) + R(t) \tag{1}$$

Multiplicative Model:

$$Y(t) = T(t) * S(t) * C(t) * R(t) \tag{2}$$

where, T - Trend, S - Seasonal, C - Cycle, R - Random

6.4 Identify p, d and q for ARIMA/Multivariate-ARIMA models

The two key concepts which helps in identifying the p and q values used in AR(p) and MA(q), are ACF and PACF

6.4.1 Autocorrelation Function(ACF)

It is a correlation function which rather than showing correlation between two different variables, shows the correlation between different values of same variable i.e X_i and X_{i+k}.
ACF is used to identify the q value for **MA** component of ARIMA/Multivariate-ARIMA. To calculate the MA term of the model, the lag at which the ACF cuts is considered. It displays the sharp cut-off at the negative correlation.

6.4.2 Partial Autocorrelation Function(PACF)

It is also a correlation function, but it gives the partial correlation of time series with its own lagged values, controlling for the values of the time series at all shorter lags. It contrasts with the autocorrelation function, which does not control the other lags.
PACF is used to identify the p value for the AR component of ARIMA/Multivariate-ARIMA. The lag at which the PACF cuts off is the indicated number of AR terms. It displays a sharp cutoff at the positive autocorrelation.

6.5 Finalizing values for p, d and q

- p - This is calculated by finding the lag k via PACF function
- d - This is the differencing order used while making the data stationary.
- q - This is calculated by finding the lag k via ACF function.

6.6 Generating ARIMA model

After identifying the values of p, d and q, we can create the ARIMA/Multivariate-ARIMA models. The ARIMA model can then be used to forecast or predict the future values.

ARIMA model can be validated by examining ACF and PACF for residual models. After fitting the correct model, it can then be used to do forecast and prediction.
We can also verify the forecast and prediction, by reserving a portion of our dataset and applying the prediction on top of it, thereby comparing the forecasted and actual observed values.

7 Application of ARIMA/Multivariate-ARIMA models

7.1 ARIMA Model

Dataset used: bitcoin_price.csv
Attribute used for modelling - Close Price
ARIMA model - univariate analysis as trading prices are often analyzed based on how they had been behaving over the period of time.
Data has been cleaned before applying ARIMA/Multivariate-ARIMA model.

7.1.1 Exploring Raw Data

Plot for visualizing the raw data as obtained is shown in Figure 18.

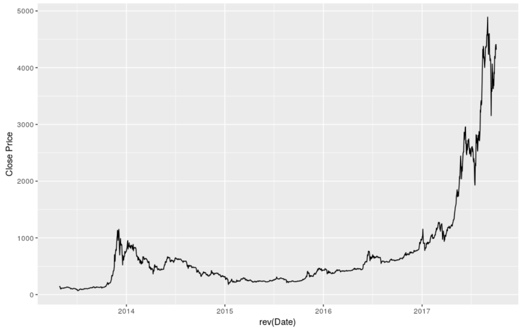

Figure 18: Original Raw Close Price Data

With the K-Means clustering it was identified, that there existed some outliers. So, with tsclean function of R, data has been cleaned for outliers to avoid any skewing.
Figure 19 shows clean data.

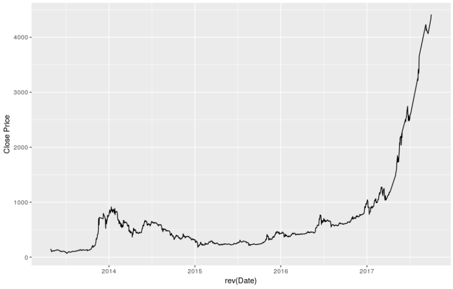

Figure 19: Cleaned Close Price Data

As part of exploration, we also calculated the moving average data. However, this moving average was to smoothen the data rather than actual modelling. Further analysis is done on dataset with this moving average value. Figure 20. shows the smoothened data.

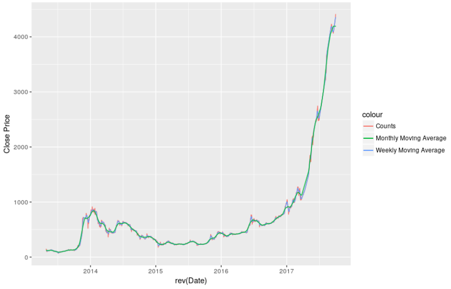

Figure 20: Moving Average Data

7.1.2 Trend Components

Figure 21 shows the trend, seasonal and residual components for the dataset.

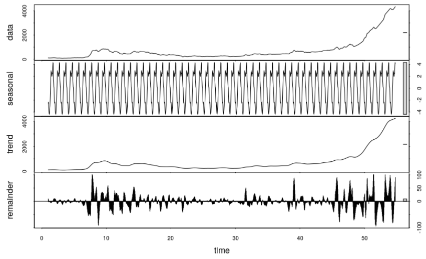

Figure 21: Time Series Components

Since there did exist some seasonal component, we subtracted it as shown in Figure 22.

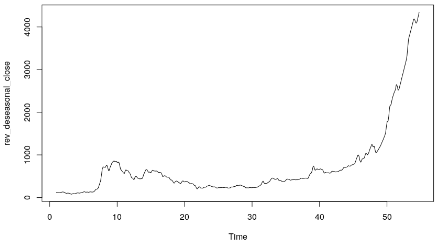

Figure 22: Subtracting Seasonal Components

Further analysis is done on data after subtracting seasonal components.

7.1.3 Stationary Series

Next step identifies if the data is stationary or not, by applying Dickey Fuller Test and verifying with Plot.

Differencing 0 Without any differencing, the result of Dickery Fuller Test, showed following value

p-value = 0.99, Alternative Hypothesis: stationary

Since p-value > 0.05, it implies that there is a need of differencing. It can also be seen from Figure (Cleaned Close Price Data) This is applied without differencing so it is same as Figure 3.

Differencing 1 Differencing value as 1, the result of Dickery Fuller Test, showed following value

p-value = 0.01, Alternative Hypothesis: stationary Since p-value < 0.05, it implies that data is now stationary, also seen from Figure 23 for differenced Close Price Data.

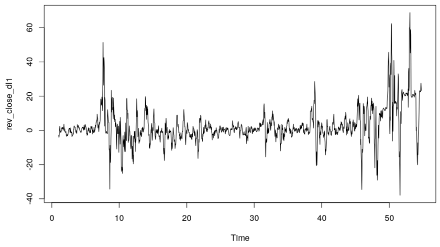

Figure 23: Close Price after 1 order differencing

Differencing 2 Differencing value as 2, the result of Dickery Fuller Test, showed following value

p-value = 0.01, Alternative Hypothesis: stationary Since p-value < 0.05, it implies that data is now stationary, also seen from Figure 24 for differenced Close Price Data.

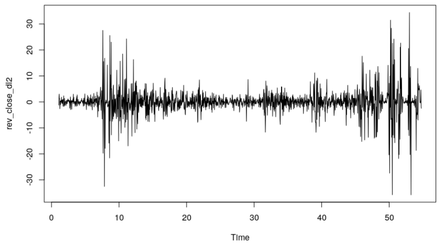

Figure 24: Close Price after 2 order differencing

Since both with d = 1 and d = 2, we get p < 0.01, we can choose any of these for further analysis. However, if we see the plots we see that the graph with d = 2 is more around 0 value rather than d = 1. And so we decide to take d = 2 for further analysis.

7.1.4 Identify p, d and q for ARIMA models

ACF and PACF for Differencing 0

Figure 25 and 26 shows ACF and PACF plots for data without any differencing.

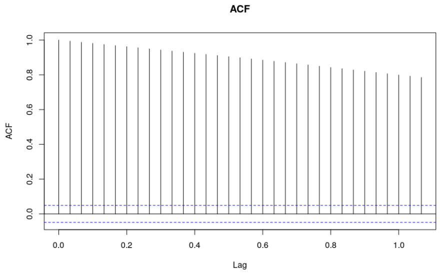

Figure 25: ACF with 0 order differencing

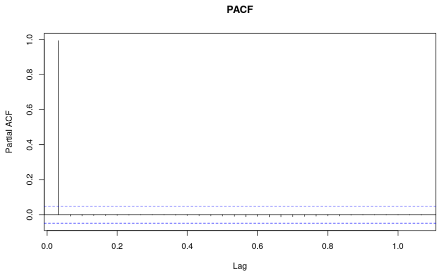

Figure 26: PACF with 0 order differencing

ACF and PACF for Differencing 1

Figure 27 and 28 shows ACF and PACF plots for data with differencing order 1.

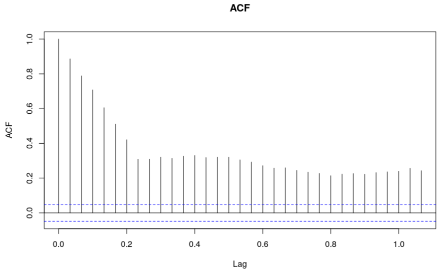

Figure 27: ACF with 1 order differencing

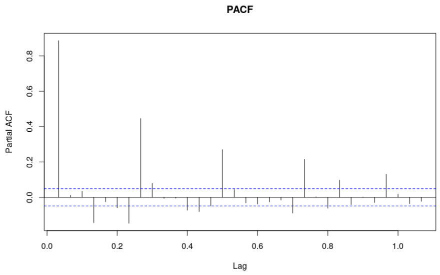

Figure 28: PACF with 1 order differencing

ACF and PACF for Differencing 2

Figure 29 and Figure 30 shows ACF and PACF for data with differencing order 2.

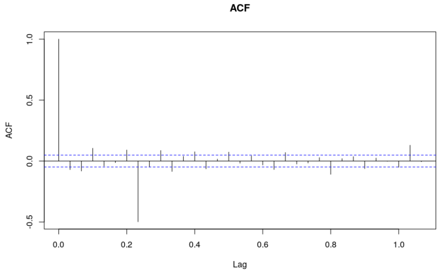

Figure 29: ACF with 2 order differencing

We can see ACF and PACF plots for differencing order 1, 2, and 3 as above. And we see that ACF plot shows the lags in the graph with d = 2. So, this again matches with our previous conclusion when we considered the stationary data for d = 2. With this we take p = 3 from ACF and q = 7 from PACF as that is where we see real positive and negative correlation sharp cut-off respectively.

Generating ARIMA model With p = 3, d = 2, and q = 7, we have following ARIMA model generated as shown in Figure 31.

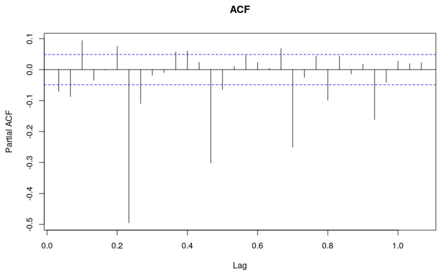

Figure 30: PACF with 2 order differencing

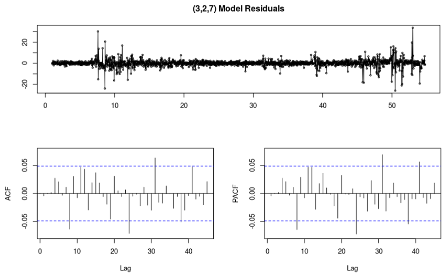

Figure 31: ARIMA with 3,2,7

7.1.5 Forecasting

After generating the above model, we did the forecasting for next 25 elements.

Figure 32 shows the

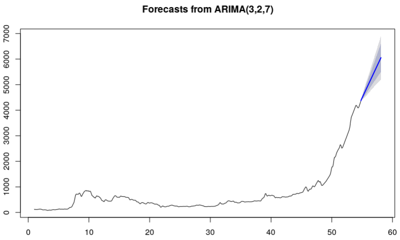

Figure 32: ARIMA with 3,2,7

To confirm the model, we did the forecast for a part of existing data and see how it compares with original data as shown in Figure 33.

Last we tried future trend prediction which matches with our original trend as shown in Figure 34.

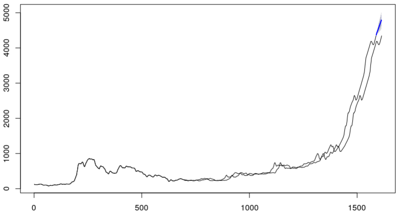

Figure 33: Forecast and comparision with original data

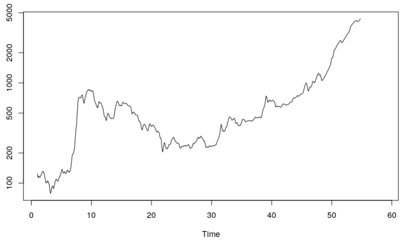

Figure 34: Future Trend Prediction

7.2 Multivariate-ARIMA Model

Multivariate-ARIMA model is applied on time-series data containing multiple attributes as opposed to univarate time series in ARIMA model. The marima model has been applied to **ethereum price** dataset. As explained above, non stationary dataset is made stationary to perform time series analysis. Different differencing techniques were applied and plotted. Difference(of order 1) of log of data and difference(of order 1) of square root of data seemed to show the constant variance in data. Hence the data was reduced into one of the former and was converted into a time-series object. For defining the marima model, ACF/PACF plots were used to check the lag values for AR and MA components of the model. The Date attribute was removed from the defined model for further time series analysis and one of the attributes (i.e. Close, Open, High, Low and Market Capitalization) were chosen as regression variables. A penalty is defined in the interval [0, 2] for stepwise model reduction i.e. defining the level of significance to consider AR and MA parameters in estimating the accurate model. As stated above, some data points were reserved for forecasting through the estimated model. Two models were created i.e. one with the normalized and non-stationary dataset without the differencing and other on the differenced/stabilized data.

The plot below shows the trend of bitcoin dataset features (such as Open, Close, High, Low, etc) with time. Non Linearity of attributes can be seen from the first and second plots. The non stationarity of the attributes can still be seen after doing the first order differencing. However, after applying difference of either log or square root transformating, the data becomes stationary, as visible in the third plot.

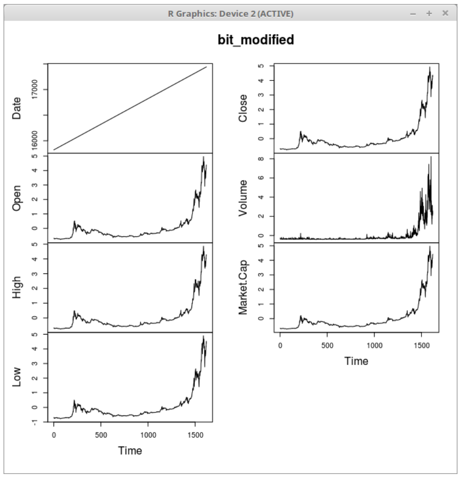

Figure 35: Original trends of bitcoin attributes

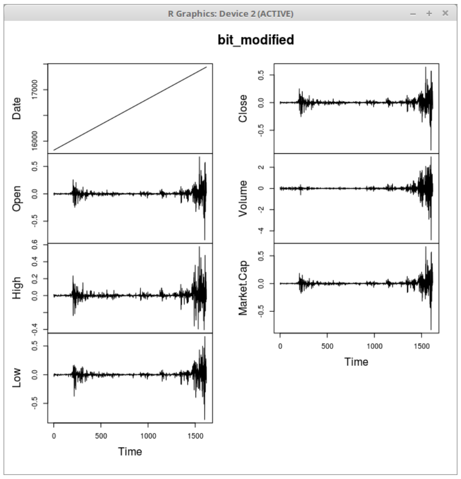

Figure 36: Trends of bitcoin attributes after first order differencing

attributes.png

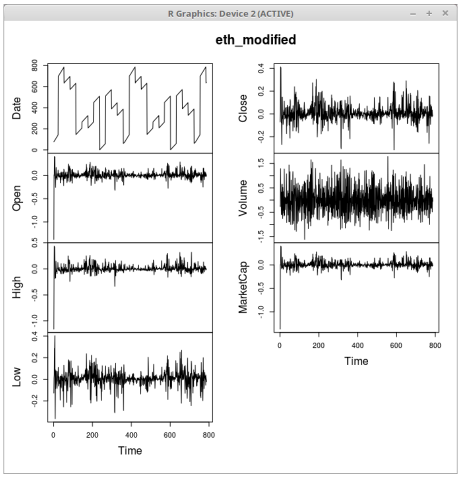

Figure 37: Trends of stationary bitcoin attributes

The ACF plot of residuals is caclulated bolow to get approximate values of p and q values for respective AR and MA components of MARIMA model for accurate estimation.

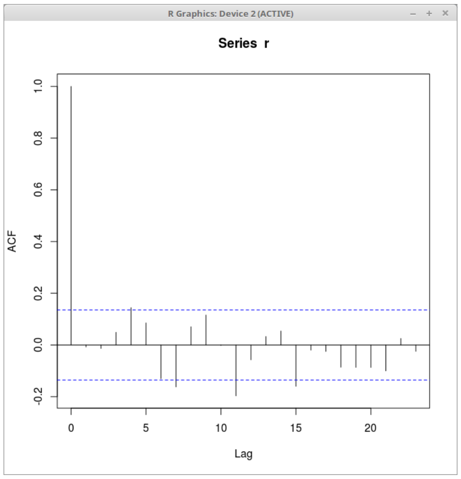

Figure 38: ACF Plot for residual on bitcoin data

The plots below shows the forecasting done for Close and Market Capitalization attributes, on the non stationary as well as stationary data from the estimated MARIMA model. The black line shows the ground truth with the green line showing the results of forecasting.

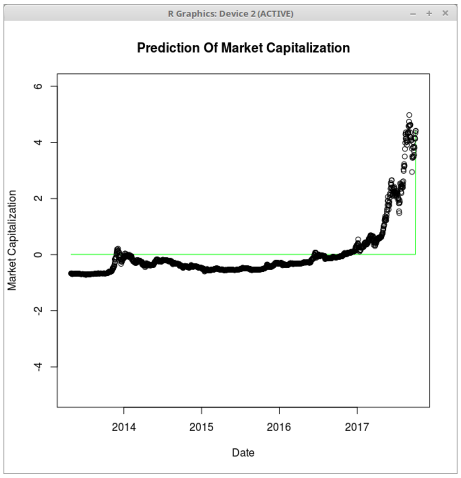

Figure 39: Forecasting of Market Capitalization

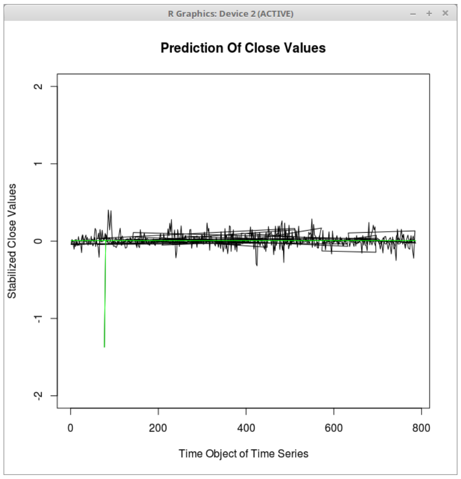

Figure 40: Forecasting of Close Attribute

References

[1] STOCK MARKET TRENDS USING CLUSTER ANALYSIS AND ARIMA MODEL by Joyti Badge, Published in stock Market Trends Asian-African Journal using of Economics Cluster Analysis and Econometrics, and ARIMA Model Vol. 13, No. 2, 2013: 303-308

[2] ANALYSIS OF NIFTY FIFTY STOCKS BASED ON K-MEANS CLUSTERING TECHNIQUE FOR STOCK MARKET PREDICTION Dr. T.Chitra kalarani* and S.Indrakala**

[3] Book named "Data Mining Concepts and Techniques", Jiawei Han, Micheline Kamber, Jian Pei

[4] https://www.datascience.com/blog/introduction-to-forecasting-with-arima-in-r-learn-data-science-tutorials

[5] https://www.analyticsvidhya.com/blog/2015/12/complete-tutorial-time-series-modeling/

[6] https://www.slideshare.net/21_venkat/arima-26196965

[7] https://people.duke.edu/ rnau/411arim3.htm

[8] https://www.openml.org/a/estimation-procedures/1

[9] https://en.wikipedia.org/wiki/Dimensionality_reduction