COMP 541 - Data Mining

---

# Predicting Cryptocurrency Prices


*California State University Northridge*

*COMP 541 - Data Mining (Fall 2018)*

*Group 4*

**Sachin Bhalekar**

**Sumit Jawale**

**Moises Alvarez**

# Table of Contents

# 1. Introduction

## 1.1. Objective

Digital forms of money are emerging as an alternative for customary cash all over the world. Cryptocurrencies are accessible to buy in a wide range of spots, making it open to everybody, and with retailers accepting different digital forms of money it could be an indication that cash; as we probably are aware of, is going to experience a noteworthy change. We propose a prototype to predict the direction and changes in prices of cryptocurrency based on the historical trends in the prices.

## 1.2. Background Information

A cryptocurrency (or crypto currency) is a digital asset designed to work as a medium of exchange that uses strong cryptography to secure financial transactions, control the creation of additional units, and verify the transfer of assets.

Cryptocurrencies are a kind of digital currency, virtual currency or alternative currency. Cryptocurrencies use decentralized control as opposed to centralized electronic money and central banking systems. The decentralized control of each cryptocurrency works through distributed ledger technology, typically a blockchain, that serves as a public financial transaction database.

Cryptocurrencies itself have huge benefits such as eliminating chargeback fraud, lower transaction costs, identity theft, decentralization, immediate settlement and so on. Predicting the direction and changes in cryptocurrency will help customers to plan and gain most out of their future cryptocurrencies. It will help them to achieve huge profit with less amount of resources such as time and strategy planning.

Cryptocurrency is quickly becoming an important part of our economy. It is the ability to predict the direction and changes in prices that can help to gain profitability for a multibillion-dollar industry. We plan to test our model by splitting the data into 80:20 ration where we will use 80% of the data for training the data and the remaining 20% will be used to compare the predicted prices with the actual prices. The accuracy of the model can be tested by calculating the Root Mean Square Error (RMSE) where lesser the value of RMSE means better the prediction model.

# 2. Overall description

## 2.1. Design principle

### 2.1.1. Dataset

The prototype will use the dataset containing all the historical daily prices for all cryptocurrencies as listed on **CoinMarketCap**. Every record in the dataset contains information for an opening price for the day, high price for the day, low price for the day and closing price for the day.

Along with the cryptocurrency data, we will use the **Google Search Trends** for the about the same period of dataset collected of cryptocurrency.

The data will be gathered from open data repository provided by CoinMarketCap. The dataset provided by CoinMarketCap is open source and legal to use for data mining purpose.

The prototype will use the dataset containing all the historical daily prices for all cryptocurrencies as listed on CoinMarketCap. The real time data for cryptocurrencies is monitored and for each day the value for following attributes is recorded: Date, Symbol, Open, High, Low, Close, Volume, Market Cap

- The dataset is semi-structured and fetched in JSON format.
- The dataset is a sequence data with historical record of the cryptocurrency in a time-series.
- The attribute types for the dataset is as follows:
    - **Date** - Interval : Date of the record
    - **Symbol** - Nominal : Identifier of the cryptocurrency index

- **Open** - Ratio : It is the price of the stock at the beginning of the trading day (it need not be the closing price of the previous trading day)
- **High** - Ratio: It is the highest price of the stock on that trading day
- **Low** - Ratio: It is the lowest price of the stock on that trading day
- **Close** - Ratio: It is the price of the stock at closing time
- **Volume** - Ratio: Also known as market capitalization, it is obtained by multiplying the circulating supply of coins by the current coin price. It is one way to rank the relative size of a cryptocurrency.
- **Market Cap** - Ratio: Itis a measure of how much of a given financial asset has been traded in a given period of time and even though so simple, it can be a powerful indicator for trading.

### 2.1.2. Data Mining Technologies

The prototype will be built using **Python Jupyter Notebook**. We plan to compare models like Simple **Linear Regression**, Artificial **Neural Network**, **ARIMA** (Auto Regressive Integrated Moving Average) and **SARIMA** (Seasonal - Auto Regressive Integrated Moving Average).

The predictions might be incorrect from these models. In such a case we will shift to a different model or technique and iterate the whole process with new models.

We plan to test our model by splitting the data into 80:20 ration where we will use 80% of the data for training the data and the remaining 20% will be used to compare the predicted prices with the actual prices.

The results might vary immensely, so we might change the ratio such that the training data size will be increased.

The models used in this paper for prediction uses the past trends from time series data to predict future values. However the actual prices for cryptocurrencies may depend on a wide range of other external factors such as the GTrends data i.e. considered by our model.

**a) Simple Linear Regression**

Simple linear regression is a statistical method that allows us to summarize and study relationships between two continuous (quantitative) variables:

(1) One variable, denoted ***x***, is regarded as the **predictor**, **explanatory**, or **independent** variable in our case google trends data.

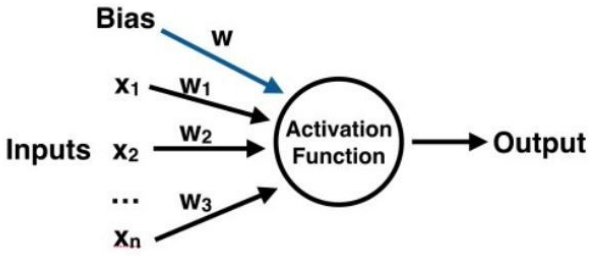
(2) The other variable, denoted ***y***, is regarded as the **response**, **outcome**, or **dependent** variable in our case bitcoin price data

**b) Neural Network**

Neural Networks are a machine learning framework that attempts to mimic the learning pattern of natural biological neural networks. Biological neural networks have interconnected neurons with dendrites that receive inputs, then based on these inputs they produce an output signal through an axon to another neuron. The process of creating a neural network begins with the most basic form, a single perceptron.

Neural networks are a set of algorithms, modeled loosely after the human brain, that are designed to recognize patterns. They interpret sensory data through a kind of machine perception, labeling or clustering raw input. The patterns they recognize are

numerical, contained in vectors, into which all real-world data, be it images, sound, text or time series, must be translated.




**c) ARIMA (Auto Regressive Integrated Moving Average):**

An ARIMA model is a class of statistical models for analyzing and forecasting time series data.

It explicitly caters to a suite of standard structures in time series data, and as such provides a simple yet powerful method for making skillful time series forecasts.

ARIMA is an acronym that stands for AutoRegressive Integrated Moving Average. It is a generalization of the simpler AutoRegressive Moving Average and adds the notion of integration.

This acronym is descriptive, capturing the key aspects of the model itself. Briefly, they are:

**AR:** Autoregression. A model that uses the dependent relationship between an observation and some number of lagged observations.

**I:** Integrated. The use of differencing of raw observations (e.g. subtracting an observation from an observation at the previous time step) in order to make the time series stationary.

**MA:** Moving Average. A model that uses the dependency between an observation and a residual error from a moving average model applied to lagged observations.

Each of these components are explicitly specified in the model as a parameter. A standard notation is used of ARIMA(p,d,q) where the parameters are substituted with integer values to quickly indicate the specific ARIMA model being used.

The parameters of the ARIMA model are defined as follows:

**p:** The number of lag observations included in the model, also called the lag order.

**d:** The number of times that the raw observations are differenced, also called the degree of differencing.

**q:** The size of the moving average window, also called the order of moving average.

A linear regression model is constructed including the specified number and type of terms, and the data is prepared by a degree of differencing in order to make it stationary, i.e. to remove trend and seasonal structures that negatively affect the regression model.

A value of 0 can be used for a parameter, which indicates to not use that element of the model. This way, the ARIMA model can be configured to perform the function of an ARMA model, and even a simple AR, I, or MA model.

## 2.2. Implementation issues and solutions

### 2.2.1. Missing Values

- Missing Bitcoin Data:
    - The Data from Google trends spans a time period from Jan-01-2013 to Sep-28-2018.
    - The Bitcoin data spans across a time period from Apr-28-2013 to Jun-02-2018.

    - The data from Google trends from Jan-01-2013 to Apr-27-2013 and from Jun-03-208 to Sep-28-2018 will be ignored while constructing the model.
- Missing values for Volume in Bitcoin Data: The values missing were fetched from Coin Market Cap and integrated with the data.

### 2.2.2. Entity ID Problem

- The google trends data for term “bitcoin” contains the search volume for each day from Jan-01-2013 to Sep-28-2018.
- Bitcoin price data contains bitcoin prices recorded per minute from Apr-28-2013 to Jun-02-2018.
- Bitcoin data per minute is used to create records per day by analysing the prices for entire day and recording the opening, high, low and closing prices for a day.
- Google trends data is integrated with the Bitcoin price data using record date as entity identifier.

### 2.2.3. Redundancy

In the Google Trends data the Search percentage field and the Search Volume field are redundant and provide same information.

```
Pearson Corelation between Search Percentage and Search Volume
(0.9999999999998386, 0.0)
```

Correlation coefficient (Pearson's correlation coefficient)

$$r_{A,B} = \frac{\sum (A - \overline{A})(B - \overline{B})}{(n-1)\sigma_A \sigma_B} = \frac{\sum (AB) - n\overline{A}\,\overline{B}}{(n-1)\sigma_A \sigma_B}$$

where n is the number of tuples, $\overline{A}$ and $\overline{B}$ are the respective means of *A* and *B*, $\sigma_A$ and $\sigma_B$ are the respective standard deviation of *A* and *B*, and $\Sigma(AB)$ is the sum of the *AB* cross-product.

If $r_{A,B} > 0$, A and B are positively correlated (A's values increase as B's). The higher, the stronger correlation.

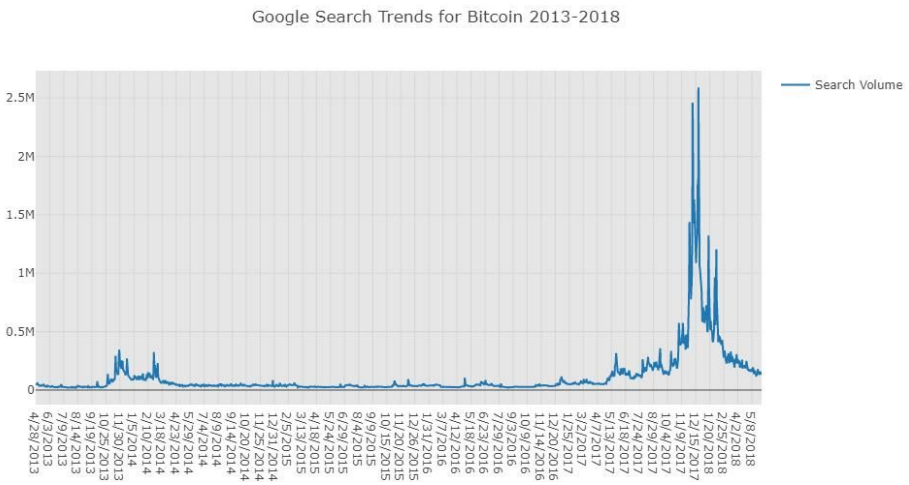
$r_{A,B} = 0$: uncorrelated; $r_{A,B} < 0$: negatively correlated

### 2.2.4. Attribute Transformation

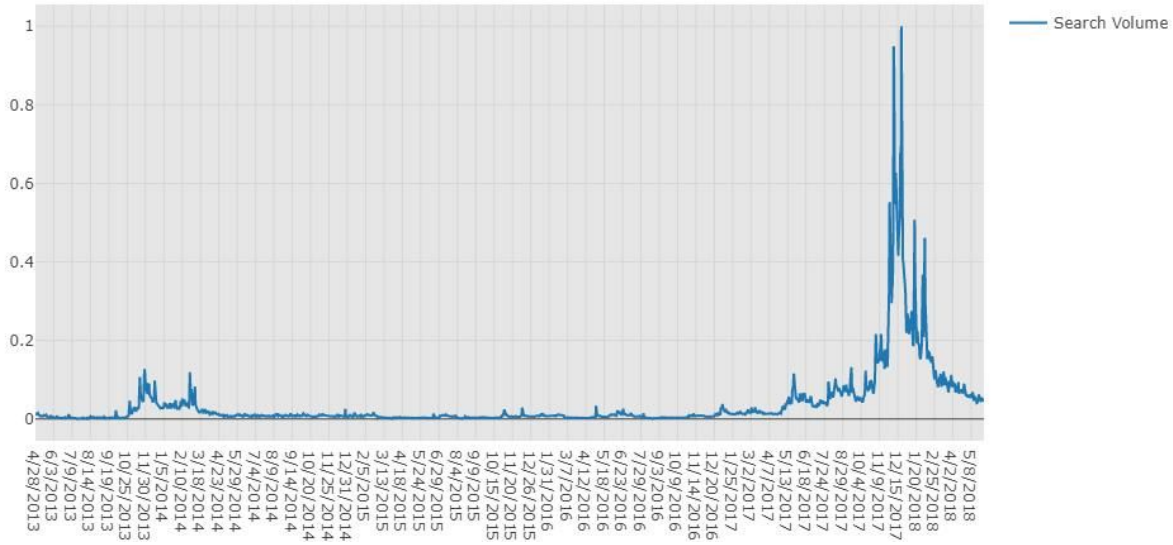
- Converted Linux timestamp to date-format.
- Eg.: 1538505925 == 10/02/2018

### 2.2.5. Normalization

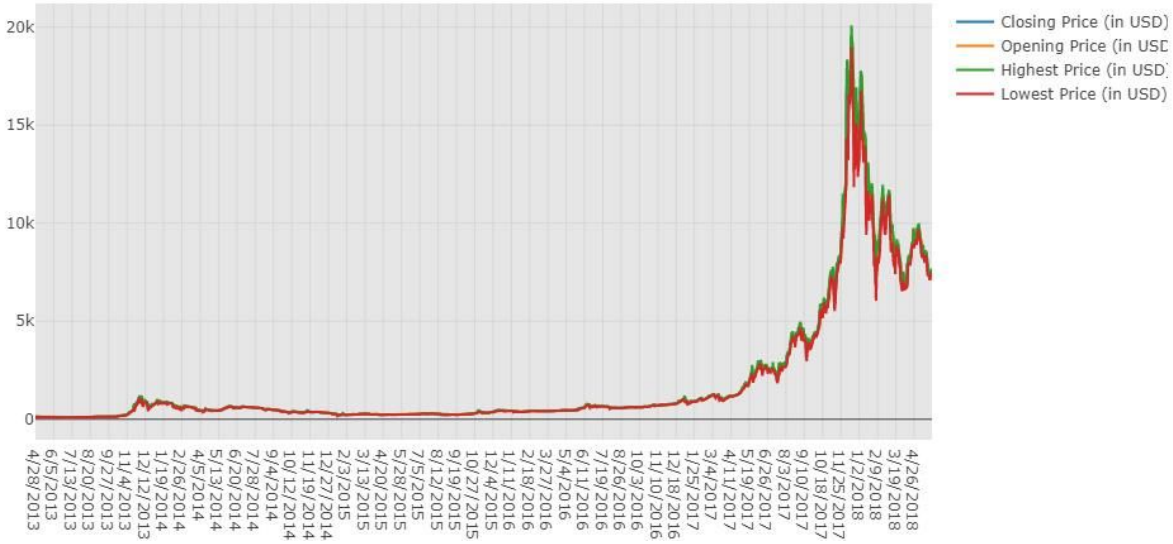
- The Bitcoin price is normalized to values lying in range 0 to 1.
- The Google search volume is normalized to values lying in range 0 to 1.

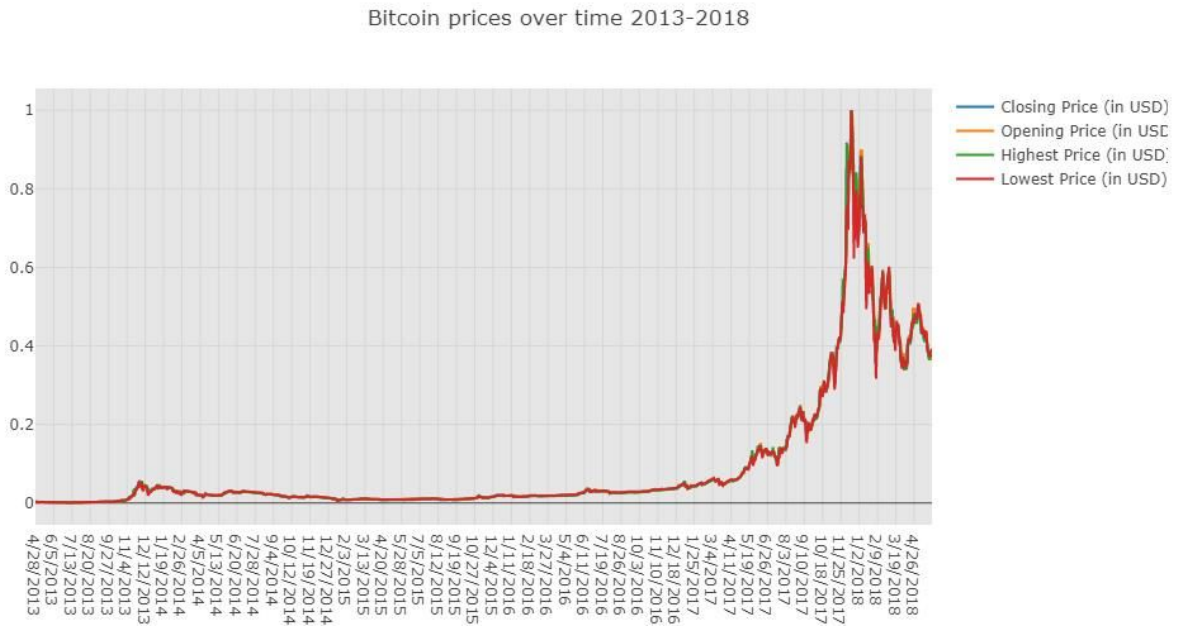


Google Search Trends for Bitcoin 2013-2018




Bitcoin prices over time 2013-2018






### 2.2.6. Data Cube Aggregation

- The time series data collected per minute is reduced to data per day by recording the open, high, low and close price for the day.
- After this reduction, the data is more reduced to only Bitcoin cryptocurrency among all the cryptocurrencies recorded.

### 2.2.7. Attribute Selection

- The Bitcoin volume and Search percentage are correlated so BTC volume will not be considered for BTC price prediction.
- Market Capitalization is not relevant to the problem we are trying to address so this column will be removed from the dataset.

## 2.3. Analysis of results

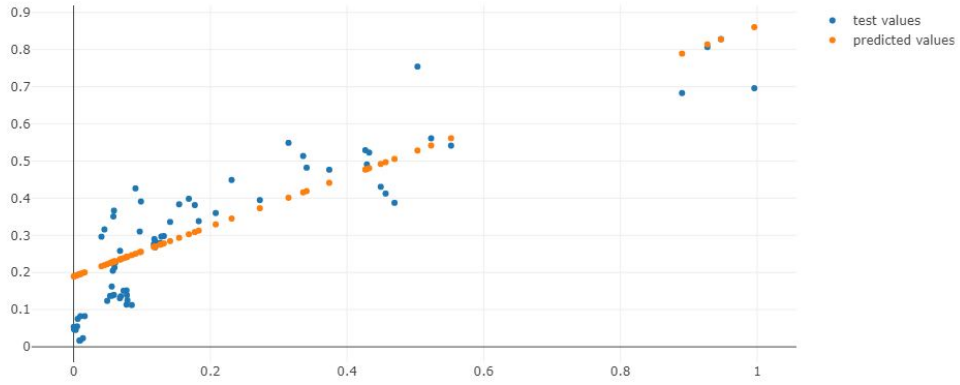
**A. Simple Linear Regression:**

- R2 value:

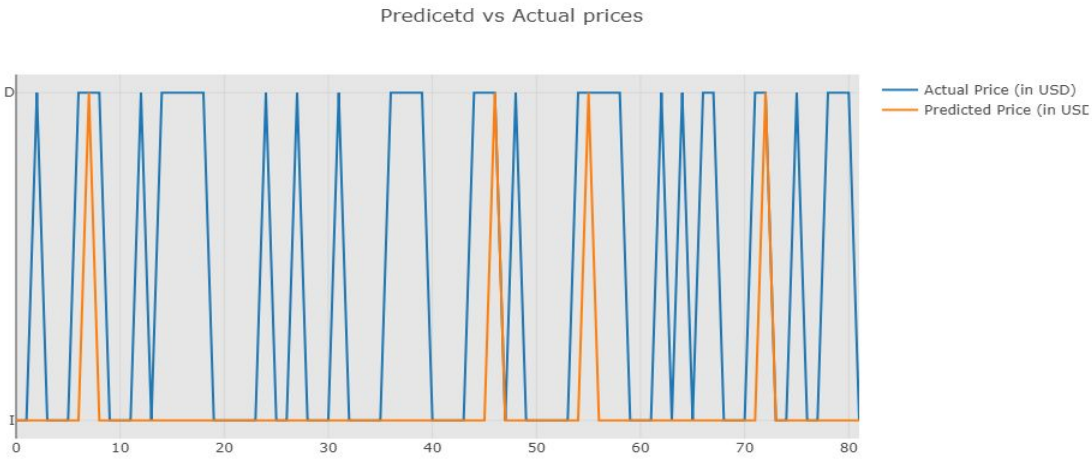
```
from sklearn.metrics import r2_score
r2_score(Y_test, Y_pred)
```

```
0.7532882980794404
```

- Scatter Plot:

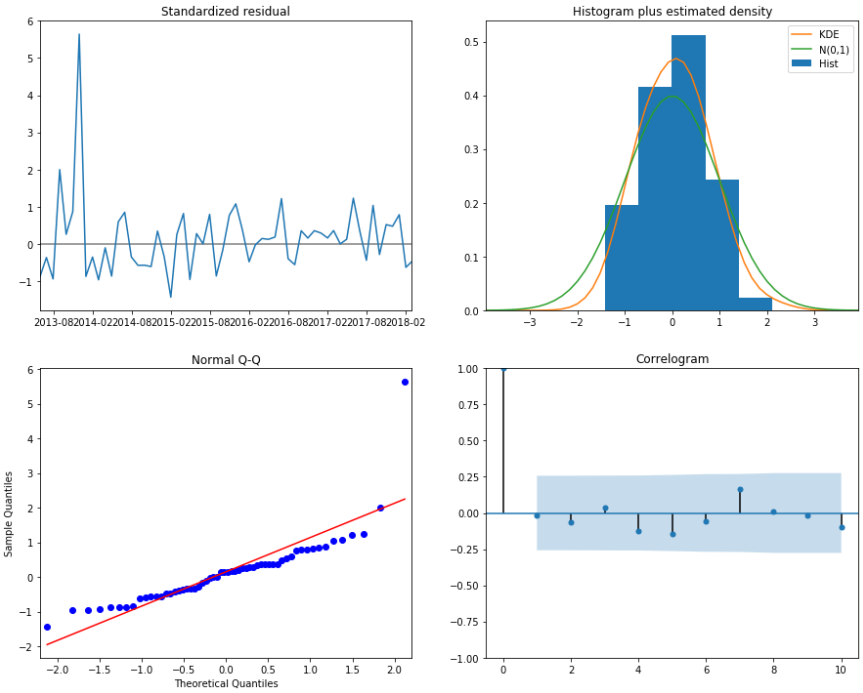



## B. Artificial Neural Network:

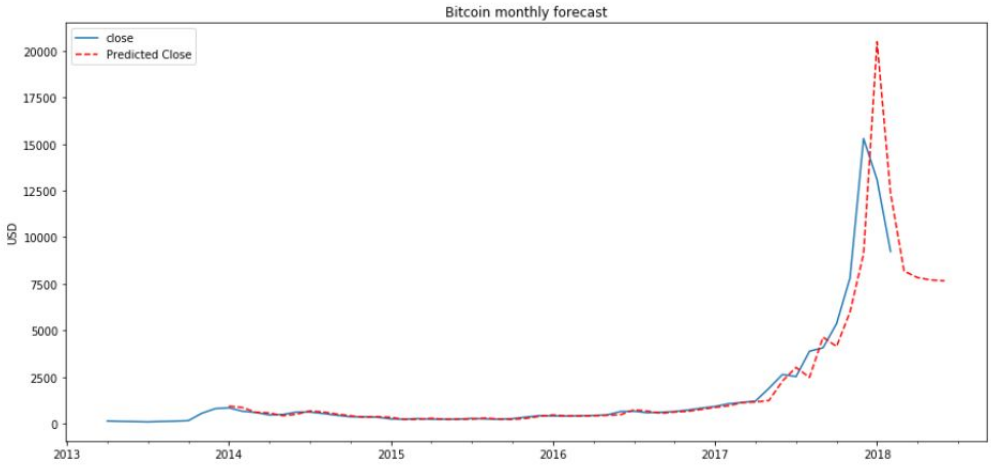



| | precision | recall | f1-score | support |
|---|---|---|---|---|
| D | 1.00 | 0.11 | 0.20 | 36 |
| I | 0.59 | 1.00 | 0.74 | 46 |
| avg / total | 0.77 | 0.61 | 0.50 | 82 |

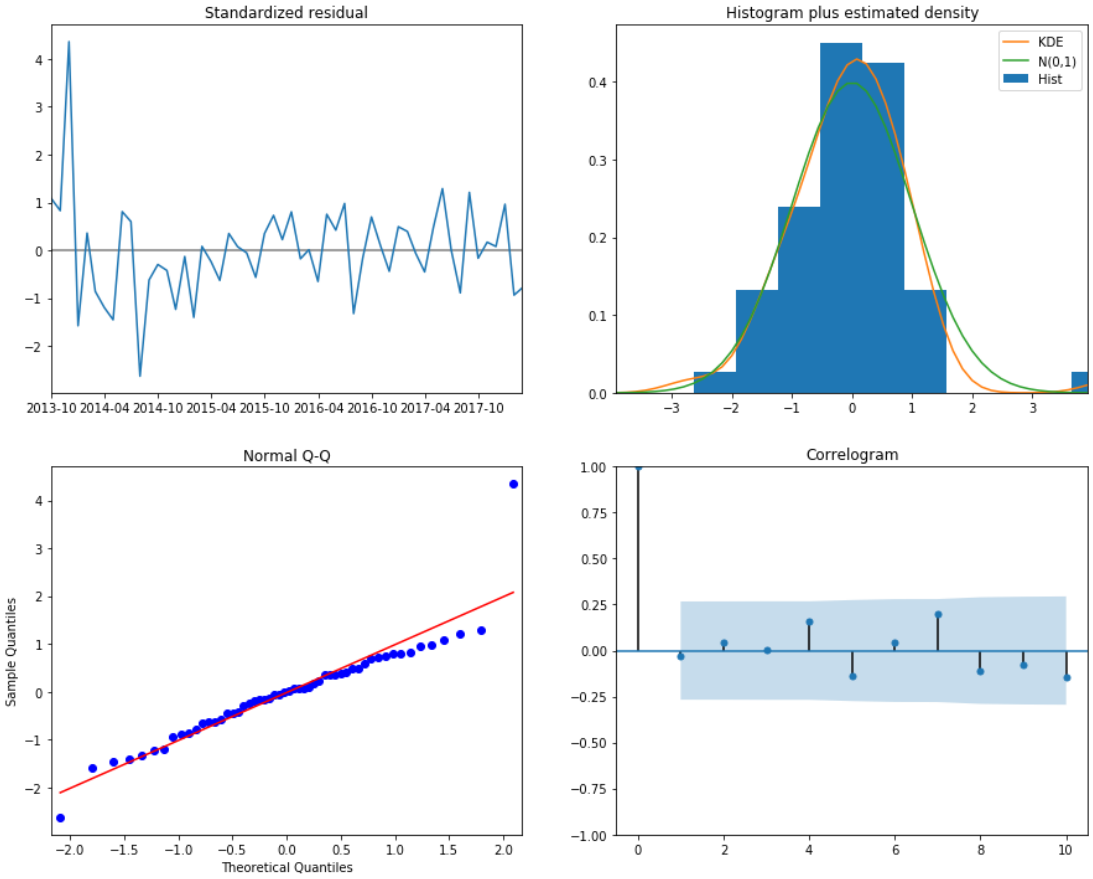
## C. ARIMA:




- Our primary concern is to ensure that the residuals of our model are uncorrelated and normally distributed with zero-mean.
- In the histogram (top right), the KDE line should follow the N(0,1) line (normal distribution with mean 0, standard deviation 1) closely.
- In the Q-Q-plot the ordered distribution of residuals (blue dots) should follow the linear trend of the samples taken from a standard normal distribution with N(0, 1).
- The standardized residual plot doesn't display any obvious seasonality.
- This is confirmed by the autocorrelation plot, which shows that the time series residuals have low correlation with lagged versions of itself.
- **Result:**




## D. SARIMA:




- The four plots analyze the residual after applying the chosen parameters. There is no long- or short-term trend remaining in the

autocorrelation factor(ACF). However, the histogram plot (upper right) shows that the residual is not perfectly normally distributed: it has a long right tail.

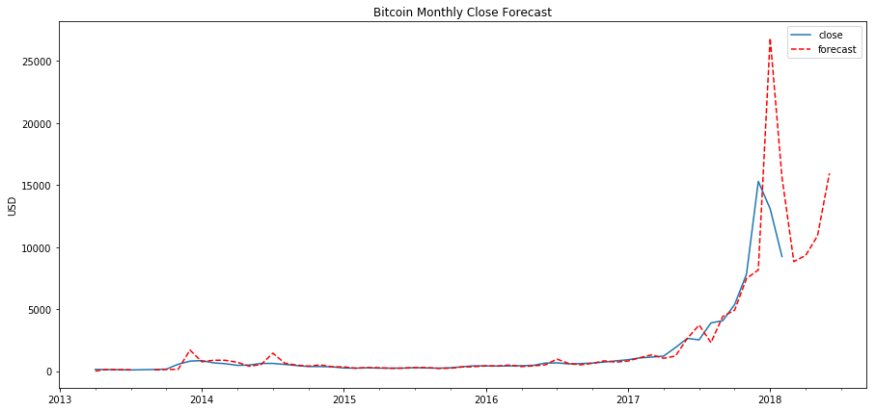
- Q-Q Normal Plot, the graph is between the actual distribution of residual quantiles and a perfectly normal distribution residuals. If the graph is perfectly overlapping on the diagonal, the residual is normally distributed.
- The correlations are very low (the y axis goes from +.1 to -.1) and don't seem to have a pattern.The gray areas are confidence bands (e.g. tell you whether the correlation is significant).
- A simple indicator of how accurate our forecast is is the root mean square error (RMSE).
- The RMSE represents the square root of the second sample moment of the differences between predicted values and observed values or the quadratic mean of these differences.
- Lower the value, the better the predictions for that model.

```
In [28]:
y_forecasted = btc_month2.forecast
y_truth = btc_month2['2015-01-01':'2017-01-01'].close

# Compute the root mean square error
rmse = np.sqrt(((y_forecasted - y_truth) ** 2).mean())
print('Mean Squared Error: {}'.format(round(rmse, 2)))

Mean Squared Error: 85.18
```

- **Result:**



# 3. Summary

- Result:
    - For Simple Linear Regression, the R2 value was 0.75, which is decent and shows that the results or predictions is accurate.
    - For Neural Network, the decrease were predicted with 100% accuracy and the increases were predicted with 51% accuracy, which is total of 75% accuracy.
    - The RMSE value for ARIMA and SARIMA was 85.18, which was the best out of all tried models.

- Learning Experiences:
    - This project helped us explore various data ming models.
    - We got a very clear understanding of data cleaning and preprocessing techniques.
    - We were able to compare four different models which helped us learn a lot about data mining and its implementation.

# 4. References


1) https://en.wikipedia.org/wiki/Linear_regression
2) https://en.wikipedia.org/wiki/Artificial_neural_network
3) https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/
4) https://machinelearningmastery.com/sarima-for-time-series-forecasting-in-python/
5) https://www.springboard.com/blog/data-mining-python-tutorial/
6) https://www.analyticsvidhya.com/blog/2018/05/starters-guide-jupyter-notebook/

# 5. Appendix

- https://github.com/sachinbhalekar/COMP541-CryptoCurrencyPrediction