

**University of Texas Arlington**

**IE 5300 DATA MINING PROJECT**

**BITCOIN PRICE PREDICTION USING FORECASTING TECHNIQUES AND DEEP LEARNING**

**Submitted to**

**Dr. Shouyi Wang**

**By**

Arvindhan Rameshbabu 1001138603

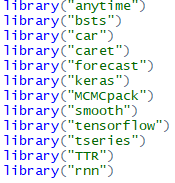
Samraj Muralikrishnan 1001428211

**Introduction:**

Bitcoin is one of the revolutionary cryptocurrency in the market today. Even though it was silent all along in the marketing world, it caught the eye of the entire world in the past few weeks by defying every forecasting algorithm ever made for it. Today, as of Dec 12 2017, it stands at a market value of $17,287.6. This brings us to the question on whether investing on bitcoin worth it and is profitable at the end of the day. Bitcoin came into existence as to a form of token which was awarded for solving mathematics puzzles and Cryptography. This was until 2013, when renowned payment processor Coinbase reported selling over $1 million worth of bitcoins. Today, most of the online merchants have started recognizing it as a form of currency. Since then it has become kind of a stock market where people invest to generate profits.

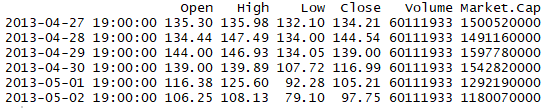
Forecasting the value of bitcoin as forecasting the flow of stock market. Over the years many algorithms have been developed for forecasting stock markets, but very few have focused on bitcoin price prediction. In this data mining project, we have focused on predicting the price of Bitcoin prices from Nov 8-Nov17 based on historical price data by studying trends in the form of both seasonal and general trends, existing forecasting methods, Bayesian statistics using Markov Chains and Recurring Neural Networks. The forecasted prices would help you understand the risk factor involved in investing in the bitcoins and the possibility of having a profit or loss in it. With that said, it is to be recalled that forecasts models are most probably wrong and hence should always include a prediction error with it.

The entire project was done using R programming on the R-Studio interface. The dataset for the project was taken from Kaggle and can be found [here](https://www.kaggle.com/sudalairajkumar/cryptocurrencypricehistory/data). The following are the libraries required for running the program:

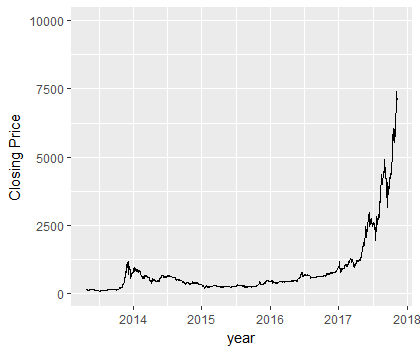


**Importing Data and Data manipulation:**

The first step of the project is to import the data set from Kaggle. We have used two data sets for the project. One is the training data set ranging from April 28,2013 to November 7,2017 and a test data set from November 8,2017 to November 17,2017. The forecasting models are created using the training data set which are used to predict the testing data set. The Open, High, Low and Close columns in the data set signify the Opening, highest, lowest and Closing price of Bitcoin against the USD on that particular day. The volume refers to the volume of bitcoin transferred in the market on a particular day and Market Cap refers to the total amount of bitcoin on circulation which is to be capped at 21 million.



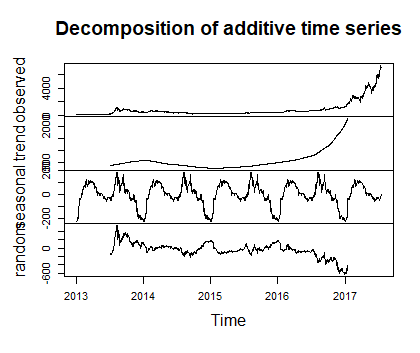
We can see from our data set that the volume column of the data set has missing values. Ignoring the volume column from the data set would result in lot of information and hence would give us a poor model fit in models it is supposed to be used at. The missing values can be filled by understanding the pattern in the data. The volume of bitcoin transferred corresponds to the volume of bitcoin transferred on a particular day. Using this data, we can use the difference between the highest and lowest price on a day to fill the missing values of volume using the rest of the information in the data set.



**Figure 1 Bitcoin Historical Prices**

**Forecasting Models:**

Before creating the forecasting models, the date element is removed from the training data set and the Closing price of the bitcoin is converted to a time series. We need to study the trends and seasonality involved within the closing price of Bitcoin. Based on the seasonal patterns, we can decide on which forecasting model to use.

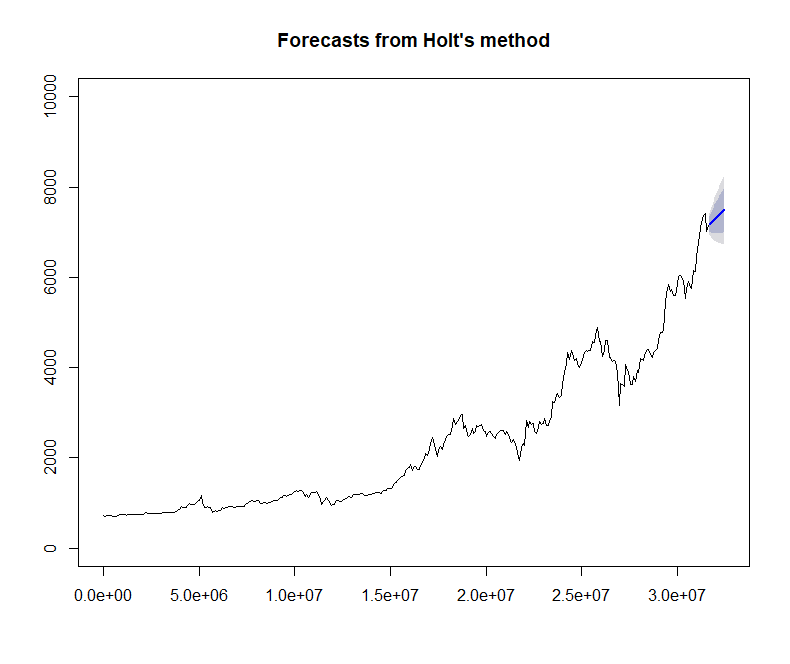


**Figure 2 Seasonal and Trend Parameters**

It can be seen that there is a uniform seasonal variation in the closing price of bitcoins over the years and the trend has been almost constant till the end of 2016. Hence the forecasting is done based on the data from Nov 7, 2016 for standard algorithms and the entire data set for Multilinear and Bayesian regression. The following are the forecasting algorithms used in predicting the price of bitcoins and each of it is reported with its prediction plots and accuracy:

**Holt’s Forecasting model:**

Holt’s forecasting or Double exponential smoothing method uses exponential smoothing based on the given input values of alpha and beta. Based on this data and the trend, we have chosen alpha as 0.2 and beta as 0.1057 for this model. Initially the model is constructed using Holt’s algorithm and this model is used to predict the forecasted price for bitcoins for the next 10 days. The dark blue line in the graph below displays the forecasted values and the light blue band displays the confidence intervals.



**Figure 3 Forecasts from Holt's method**

Accuracy of forecast:



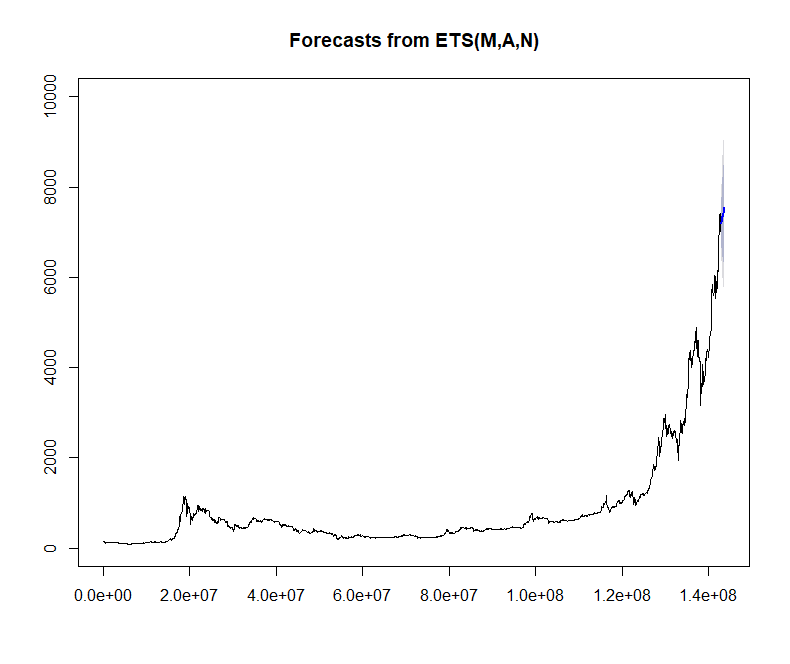
The mean accurate prediction error comes out to be 8.94% which is a pretty good fit as long as it is below 10%.

**Exponential Triple Smoothing:**

Exponential triple smoothing is an algorithm which estimates the initial states and smoothing parameters of a forecast by optimizing the likelihood function to find the local minimum and is restricted within a parametric space to make sure data is forecastable. The dark line in the forecasting graph points towards the forecasted values and the light blue band refers to the confidence interval.

Forecasting accuracy:





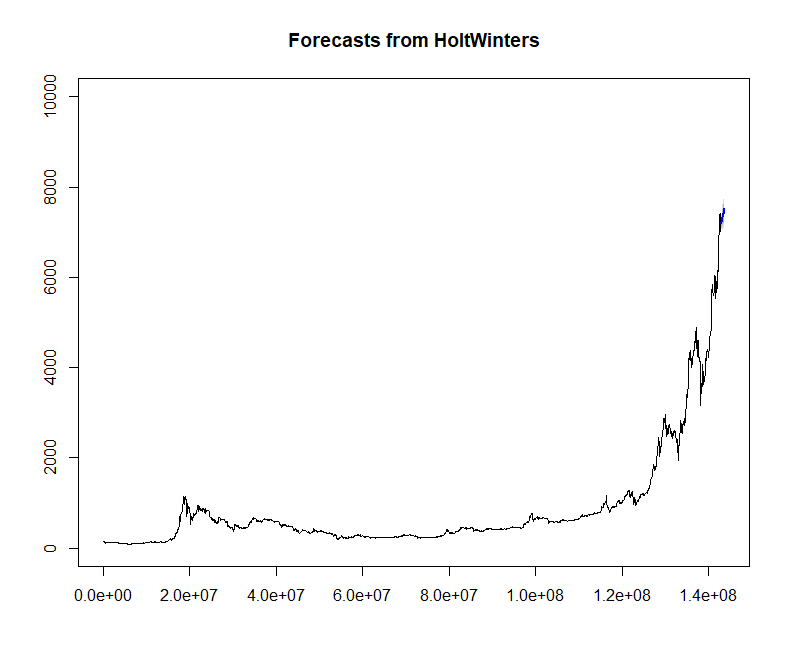
**Figure 4 Forecasts from ETS**

The forecasted model has a mean accuracy prediction error of 9.022% which is still a very good prediction model but is not as accurate as the Holt’s forecasting model.

**Holt Winter’s forecasting method:**

Holt winter’s algorithm is a self-learning algorithm which uses heuristic values for the initial states and the interprets the smoothing parameters for the model by optimizing the mean square error. This is also a form of triple exponential forecasting with the only difference that it ignores the parametric space while fitting the model. The smoothing parameters for this method is initialized to null since it is a self-taught algorithm and the model automatically selects the best smoothing parameters during fitting. The model is then used to forecast values for the next 10 days which is compared with the test data set to check for accuracy. The dark blue line in the forecast graph signifies the direction and value of forecasted values.





**Figure 5 Forecasts from Holt-Winters**

The model gives a prediction accuracy error of 8.98% which is still lower that Holt’s forecasting model.

**ARIMA forecasting:**

Autoregressive Integrated Moving Average forecasting is a method extensively used for time series forecasting. It divides the time series data into models based on the number of time lags in the autoregressive model, the differencing and the order of the moving average model. R provides functionality for two kinds of ARIMA models. One is the *auto.arima* function which automatically fits the data to the best possible order and other one is ARIMA forecasting in which the order needs to be determined based on the ACF and PACF plots. Both the models are created for forecasting and are compared for accuracy.

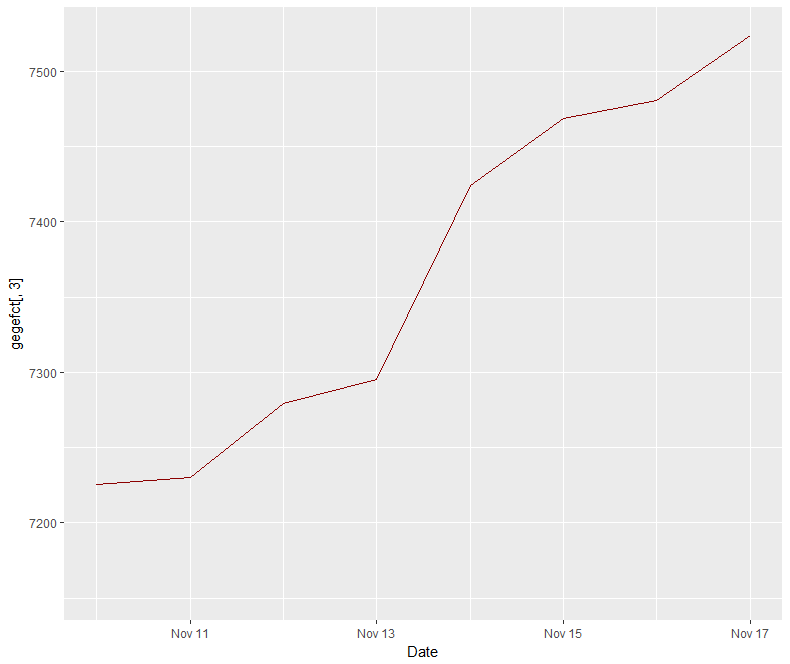
Accuracy for ARIMA based on ACF and PACF plots- order(4,2,11):



Accuracy for auto.arima fit – order(0,2,1) :



It can be clearly seen that the ARIMA model we obtained using ACF and PACF plots provide high prediction accuracy compared to the auto.arima model. This is infact the best forecasting model we have obtained so far. The following are the ARIMA forecast plots for the model with order(4,2,11) :



**Figure 6 ARIMA forecasting for order(4,2,11)**

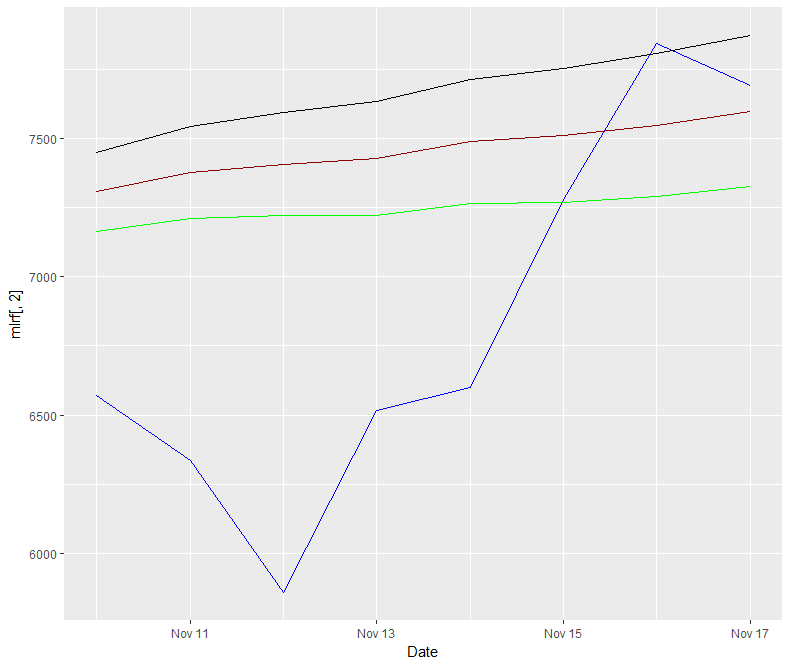
**Forecasting using Multilinear Regression:**

This is a forecasting method we implemented based on our understanding of the data and how it is varying over time. We initially used the entire data set to form a multilinear regression equation by choosing the closing price as the response variable and the rest of the parameters as regressors. Then the regressors were forecasted for the next 10 days using the ARIMA forecasting model(best model so far) and these values were built on the regression model to predict the Closing prices of bitcoins for the next 10 days.

The following is the prediction accuracy of the model:



The following is the forecast graph using the MLR model. The green and the black line signify the confidence intervals, the dark red line signifies the forecasted values and blue line the actual values. The model gives a prediction accuracy error of 9.46% which is still pretty good given the amount of information it contains.

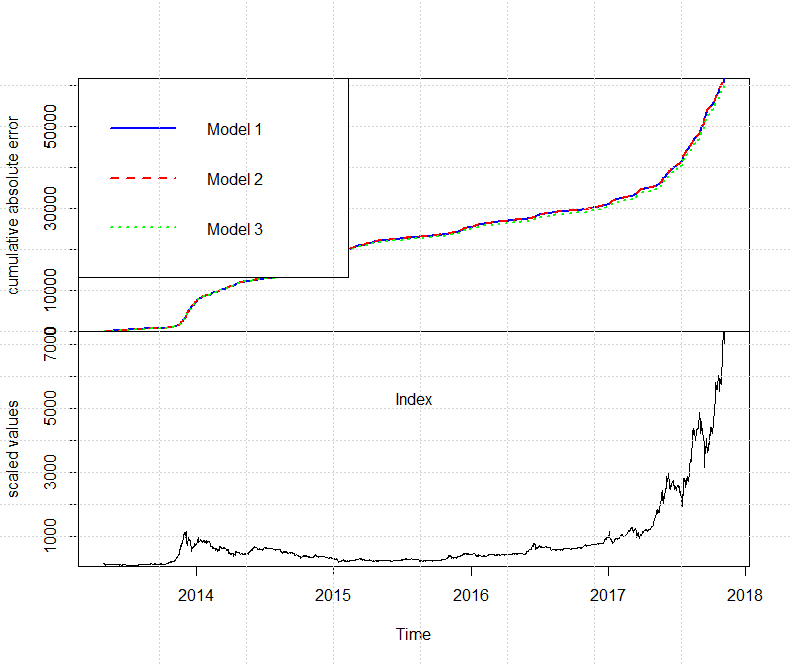


**Figure 7 Forecast plot using MLR model**

**Forecasting using Bayesian Regression:**

Bayesian linear regression is one of the ways to deal with linear regression. Here, the statistical investigation is attempted inside the setting of Bayesian derivation. The earlier conviction about the parameters is joined with the information's probability work as per Bayes hypothesis to yield the back conviction about the parameters and. This data model is that an information vector x of length m duplicates a coefficient matrix A to create a yield vector y of length d, with Gaussian noise included. This is a conditional model for y just: the appropriation of x isn't required. As we should see, conditional models make a small reference in Bayesian inference.

We have created 3 difference Bayesian models-1) Only using the times series of Closing price of bitcoins 2) Simple Bayesian regression using only a single model. 3) 10 different Bayesian models normalized to one.



**Figure 8 Bayesian statistics models**

The following is the prediction accuracy of the Bayesian model:



It can clearly be seen that Bayesian regression gives the best forecasting model for bitcoin price prediction. It has a very low Mean accuracy prediction error of 7.82 which is way lower compared to other models. Our model is run only for 10 iterations or Markov chains and increasing this number would further improve the model accuracy.

**Long Short Term Memory – RNN:**

LSTM is one of the simplest neural network algorithms which works as a building block in creating bigger neural networks. The way this algorithm works is that the model is trained over 300 times via recurrent nets to make sure the error is as minimal as possible. This is the latest advent in the concept of forecasting. Though we were able to write the code for this Deep learning algorithm, we were not able to execute it due to the given time and machine constraints to show it in the report, but this will be included as part of our presentation.

**Conclusion:**

From the comparison of the forecasting techniques it can be seen that Bayesian forecasting technique using Markov chains gives us the best fit for Bitcoin price prediction. As a matter of fact, this technique has been used by Harvard professionals for over the past 2 years to study ad determine patterns in the bitcoin data. Even though the price is really volatile now due to the Bitcoin bubble, this algorithm might be a good algorithm to use for its price prediction in the future. There is further scope of improvement in this project by studying the seasonal patterns of bitcoins which can be done using more resources and time.

**References:**

* <https://rpubs.com/zkajdan/279967>
* <https://cran.r-project.org/web/packages/available_packages_by_name.html>
* <https://www.investopedia.com/articles/07/mean_reversion_martingale.asp>