**Bitcoin Price Analysis and Forecasting**

**Prof. Yazan Roumani | Forecasting | Group** 2

**Team Members**

Ashwini Nelagondanahalli Virupakshappa

Prachi Dalvi

**Contents:**

[**Introduction**](#_heading=h.k6gu4i7jov8t) **2**

[**Problem**](#_heading=h.t7ccujvf4nv9) **3**

[**Data**](#_heading=h.ttoq0tr21hkc) **3**

[**Data Cleaning/Validation**](#_heading=h.unq0m6w2dqzg) **3**

[**Exploratory Data Analysis**](#_heading=h.1a9rfvjqgzrt) **4**

[**Model Selection**](#_heading=h.2veev8sdqhqe) **5**

[● Holt’s Exponential Smoothing](#_heading=h.9vvhblq4jntc) 5

[● DampedTrend Exponential Smoothing](#_heading=h.lef3ldr0k92z) 6

[● Simple Regression](#_heading=h.rwxchla36g54) 8

[● ARIMA](#_heading=h.scnc7u86xfxx) 10

[**Conclusion:**](#_heading=h.li38dbma6emt) **15**

[**SAS CODE**](#_heading=h.8iycula251mz) **17**

# **Introduction**

Bitcoin is a decentralized cryptocurrency, which is a type of digital asset that serves as the foundation for blockchain-based peer-to-peer financial transactions. Bitcoin was invented in 2008 and its initial value was around $1. Bitcoins price started soaring from 2017 and have been high since then.

# **Problem**

As bitcoin is one of the top-performing currencies, its prediction holds great promise, providing motivation for further research in the field. The value of Bitcoin fluctuates in the same way the value of any other stock does. Many algorithms are used to forecast stock market prices using stock market data. However, the factors influencing Bitcoin differ. As a result, forecasting the value of Bitcoin is necessary in order to make sound investment decisions. Unlike the stock market, the price of Bitcoin is not affected by business events or intervening government authorities. As a result, we believe it is necessary to use appropriate forecasting models to forecast the price of Bitcoin.

The goal of this project is to determine the best forecasting method that will predict the Bitcoin Closing price and to forecast the monthly average of Bitcoin closing price for the next 6 months

# **Data**

The dataset we are using was downloaded from Kaggle site available at:

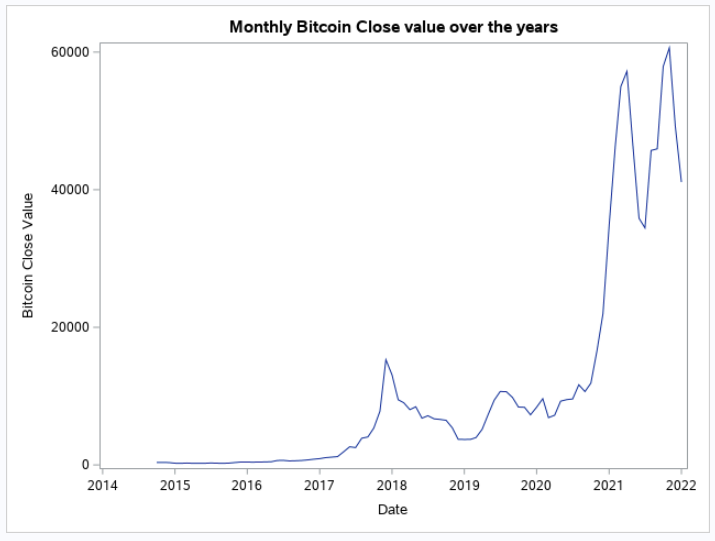
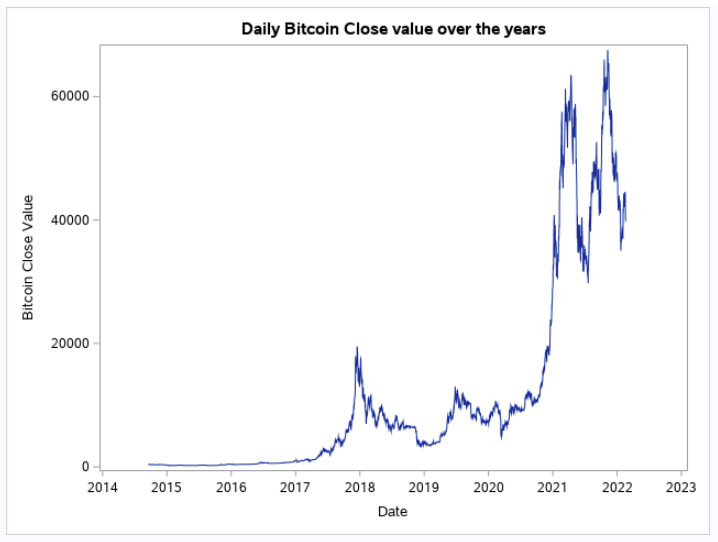
[https://www.kaggle.com/datasetsept-17-2014-august-24-2021s/meetnagadia/bitcoin-stock-data-](https://www.kaggle.com/datasets/meetnagadia/bitcoin-stock-data-sept-17-2014-august-24-2021)

The dataset has the daily Opening, Closing, high's and low's of the bitcoin price from September 2014 to February 2022. The variable of interest for this project is the close price. We selected Closing price over opening price, because firstly, there isn't much difference between the closing and opening price which is less than $1 and having a good forecast overview of closing price can help to be prepared for what is coming in future.

# **Data Cleaning/Validation**

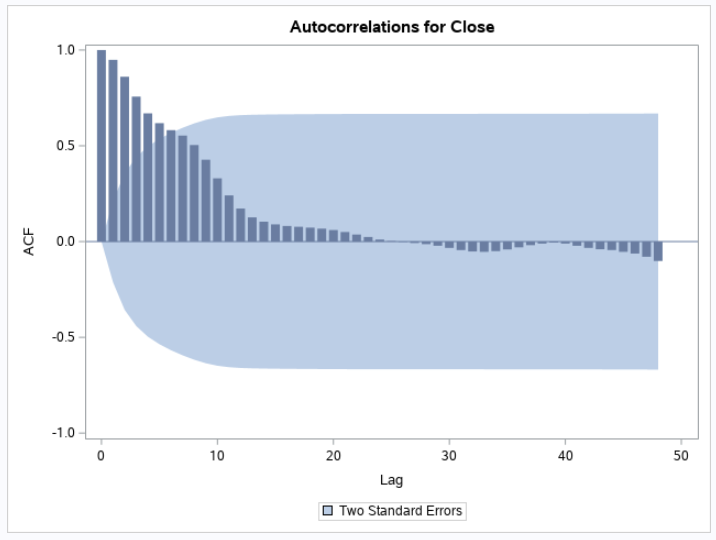
The data available at the Kaggle site was pretty consistent without null and missing values, so there was no need for cleaning the dataset. However, the time period for the data was daily. After reviewing the data, we converted the data to monthly by averaging the Close price to remove irregularities and noise components from the data and to get a better and clear understanding of the data.

# **Exploratory Data Analysis**



From the Daily and monthly timeseries plot, we can see that the irregularities from the daily data is smoothened in the monthly graph and we can clearly see the upward/positive trend in the closing price of the Bitcoin. Initial value of Bitcoin was $1. In 2017, its price started increasing and since then it is high. The huge spike that we can clearly see from the monthly timeseries plot during the first half of 2021 was due to the pandemic. Then, slowly decreased and again there is one more spike which was due to the launch of bitcoin ETF in the United States.

**ACF Plot**



The trend component is confirmed from the ACF plot of the closing price of the Bitcoin as the values are slowly decreasing and not hitting zero even till the 10th lag. There is no seasonality in the closing price or cyclical component that can be confirmed from the monthly timeseries plot.

# **Model Selection**

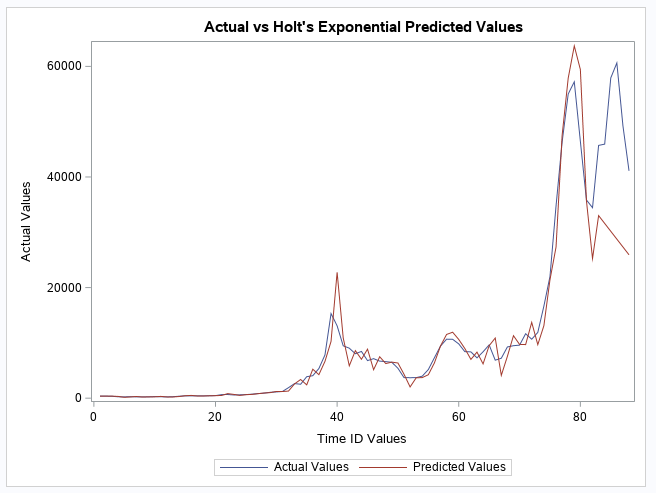
From the timeseries and ACF Plot, we strongly see the trend components and no seasonality or cyclical components. Considering the presence of trend, we have the following options to try and determine a good fitting model out of them:

* Holt’s Exponential Smoothing
* DampedTrend Exponential Smoothing
* Simple Regression
* ARIMA

We tried the above models to understand, how they predict the closing price and the factors they consider when predicting it, for eg. Holt’s give weightage to the trend component and damped trend to trend as well as to damping factor etc.

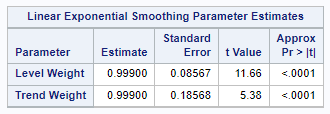
## Holt’s Exponential Smoothing

Holt’s Exponential Smoothing is best used for data that displays some linear trend but no seasonal component, which is consistent with this dataset. The 6 month forecast for average bitcoin closing cost resulted in the below forecast plot.



Between March 2020 and April 2021, the average bitcoin closing cost increased from $6,871 to $57,206, respectively. According to Time Magazine, “when the coronavirus pandemic shut down the economy and stirred fears of inflationary pressures on the U.S. dollar, bitcoin’s price started to accelerate in its upward climb” (DeMatteo, 2022). This upward climb, as shown in the above forecast plot, will impact forecasted values.

When reviewing the parameter estimates for Holt’s, the Level Weight and Trend Weight both had values of 0.99900. For Level Weight, this indicates most of the data’s weight was applied to the most recent observations. For Trend Weight, this indicates that the slope is drastically changing.



To determine the model fit and accuracy of Holt’s, the RMSE, MAPE, AIC and BIC were reviewed (see table below). Based on the below statistics, there is a concern with overfitting. Based on the RMSE and MAPE only, the model fit resulted in the better model (where model fit RMSE = 2683.19 and MAPE = 16.05 and model accuracy RMSE = 21,840.36 and MAPE = 40.14). When reviewing AIC and BIC values, the values for model fit were higher than the values for model accuracy (where model fit AIC = 1298.74 and BIC = 1303.55 and model accuracy AIC and BIC = 119.90).

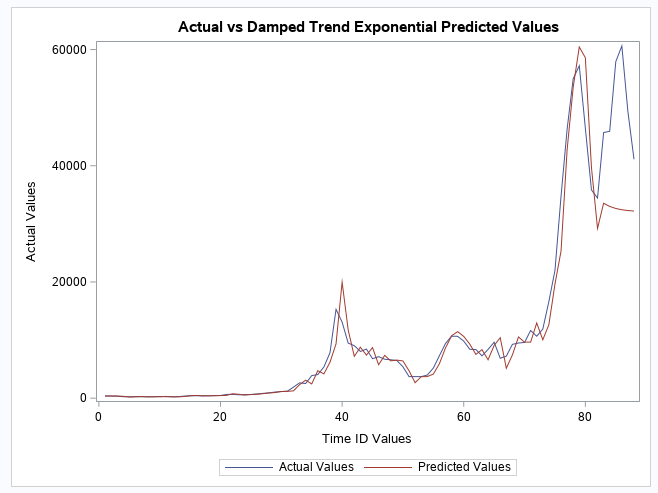
|  |  |  |
| --- | --- | --- |
| **Holt’s Exponential Smoothing** | | |
|  | **Model Fit** | **Model Accuracy** |
| RMSE | 2683.19 | 21,840.36 |
| MAPE | 16.05 | 40.14 |
| AIC | 1298.74 | 119.9 |
| BIC | 1303.55 | 119.9 |

*Note: AIC and BIC values were included to understand how the resulting forecast impacted additional statistics for model fit and accuracy. Since AIC and BIC are analyzed across similar models, these amounts are excluded from the overall conclusion of this report.*

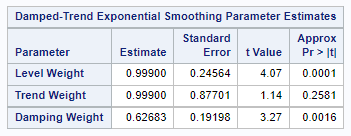
It is possible overfitting in the model was due to the significant increase that occurred from 2020 to 2021 for the average closing cost.

## DampedTrend Exponential Smoothing

Similar to Holt’s, Damped Trend Exponential Smoothing is used for data that displays some linear trend but no seasonality. The main difference is Damped incorporates a parameter that curves (or dampens) the flat trend line generated by Holt’s Exponential Smoothing. The 6 month forecast for average bitcoin closing cost resulted in the below forecast plot.



When reviewing the parameter estimates for Damped Trend, the Level Weight and Trend Weight both had values of 0.99900. Similar to Holt’s, the Level Weight, this indicates most of the data’s weight was applied to the most recent observations and the Trend Weight, this indicates that the slope is drastically changing. The Damping Weight indicates the amount of damping applied to the forecast, where the lower value the more damping applied and the higher the value the less damping applied. For this model, Damping Weight had a value of 0.62683 indicating more damping was applied.



The RMSE, MAPE, AIC and BIC were reviewed to determine the model fit and accuracy of Damped Trend (see table below). Based on the below statistics, there is a concern with overfitting (similar to Holt’s). Based on the RMSE and MAPE only, the model fit resulted in the better model (where model fit RMSE = 2420.83 and MAPE = 14.59 and model accuracy RMSE = 18,777.33 and MAPE = 33.50). When reviewing AIC and BIC values, the values for model fit were higher than the values for model accuracy (where model fit AIC = 1283.87 and BIC = 1291.09 and model accuracy AIC and BIC = 118.08).

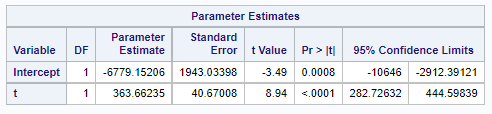
|  |  |  |
| --- | --- | --- |
| **Damped Trend Exponential Smoothing** | | |
|  | **Model Fit** | **Model Accuracy** |
| RMSE | 2420.83 | 18,777.33 |
| MAPE | 14.59 | 33.5 |
| AIC | 1283.87 | 118.08 |
| BIC | 1291.09 | 118.08 |

*Note: AIC and BIC values were included to understand how the resulting forecast impacted additional statistics for model fit and accuracy. Since AIC and BIC are analyzed across similar models, these amounts are excluded from the overall conclusion of this report.*

Consistent to the results seen in the Holt’s, overfitting in the model was due to the significant increase that occurred from 2020 to 2021 for the average closing cost.

## Simple Regression

Simple Regression is a linear regression model with a single explanatory variable, i.e, it concerns two-dimensional sample points with one independent variable and one dependent variable. For performing the simple regression, we created a time index variable t, and used it as an independent variable to check for the relation to the closing price of Bitcoin.

Equation: 

y = - 6779.15 + 363.66(x)

Where y = Closing price of Bitcoin and x = Time period

Intercept – At time = 0 the closing price of Bitcoin is - 6779.15

Slope - with each month the bitcoin closing price is increasing by $363.66

**Model Evaluations**

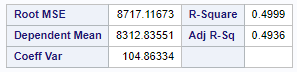
* Is the model logical?

Model is logical as the positive slope makes sense, closing price of Bitcoin has increased over time.

* Is the Slope Statistically significant?

Yes, both the p-values are less than alpha (0.5) and 0 is not a possible value in the lower and upper 95% confidence interval limits.

* What percent of variation in the dependent variable is explained by variation in the independent variable?

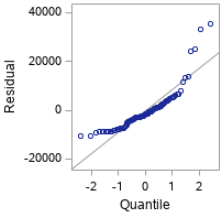
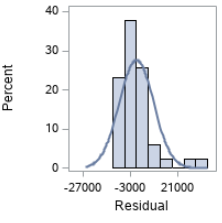


The coefficient of determination (R-sq) – 49%, which shows that the model is a moderate fit to the data.

**Model Assumption:**

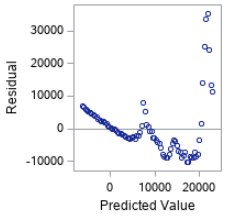
The model is linear

* Normality assumption-



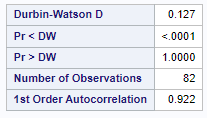
The Residual plot shows normally distributed values and most of the points in the Quantile plot are either on the line or close to the line, which confirms the normality assumption.

* Equal variance assumption:

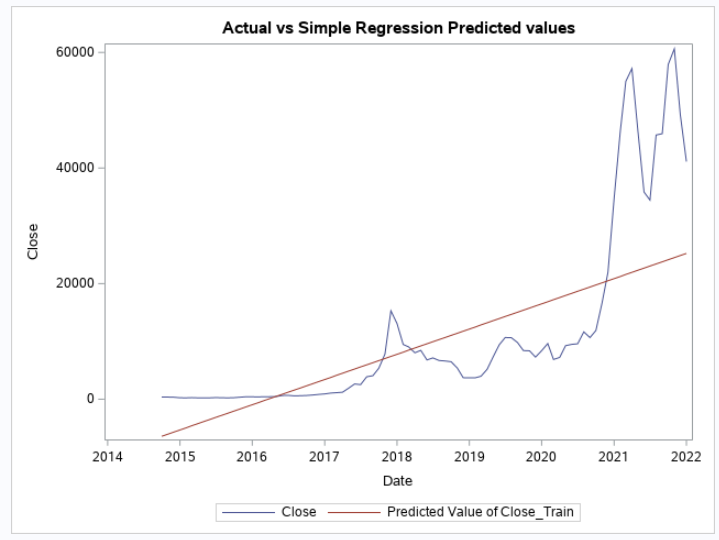
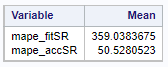


The predicted value Scatter Plot does not show a clear funnel-in or fan-out pattern but there is a slant line which seems like a pattern and hence the equal variance assumption is not true..

* Independence assumption:



As the Pr<DW is less than alpha (0.05) there is a positive autocorrelation. which is a violation of the independence assumption

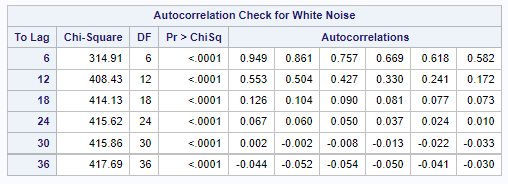


Even though 2 out of 3 assumptions did not meet, we continued to calculate the MAPE for the model fit and model accuracy and to plot the forecasted value to analyze how well or worst the model fits the data. Model fit of MAPE 360 and model accuracy of MAPE 50 indicates that it’s not a good fit to the data and we are discarding the simple regression from any further considerations against our models.

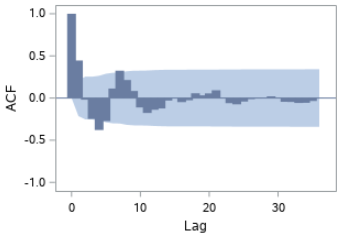
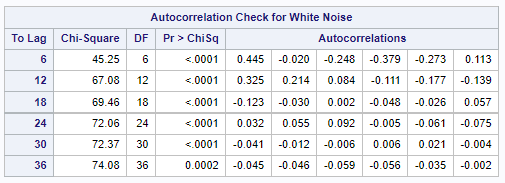
## ARIMA

ARIMA(Auto Regressive Integrated Moving Average) is a combination of 2 models AR(Auto Regressive) & MA(Moving Average). It has 3 hyperparameters - p (auto regressive lags),d(order of differentiation),q(moving avg.) which respectively come from the AR, I & MA components. The AR part is correlation between previous & current time periods. To smooth out the noise, the MA part is used. The **I** part binds together the AR & MA parts.  
In order to find the values of p and q, we will be using the ACF(Autocorrelation Function) & PACF(Partial Autocorrelation Function) plots. Before going any further, the first step is to check for the white noise in the data and for Stationarity.

**White noise and Stationarity** – From the Autocorrelations check for White Noise table, we see that all pr>Chisq values are less than alpha, so the data is not white noise, and we can move forward with our further analysis for our data using the ARIMA model.



Throughout this document, from the monthly ACF plot(shown earlier) we have seen that our data is not stationary as it has a trend and the value of the lags are slowly decreasing to zero. To move further with the ARIMA model we need to make the data stationary by either differencing the time series or by taking logs of the original time series. In this project, we will be using the most frequently used method of differencing the time series.



Once we have applied differencing, we can see from the ACF plot above that the differenced variable no longer has the trend component and the data still is not white noise. This step has prepped us to move forward to evaluate the possible model for our ARIMA analysis.

**Evaluating the model –**

As interpreting whether the pattern is present in ACF and PACF is confusing from the ACF and PACF plots of the differenced variable, we started by analyzing the possible first AR, MA and then the ARMA by assuming the pattern is present in the ACF and PACF. We came with the below 4 possible ARIMA models.

ARIMA(p,d,q)

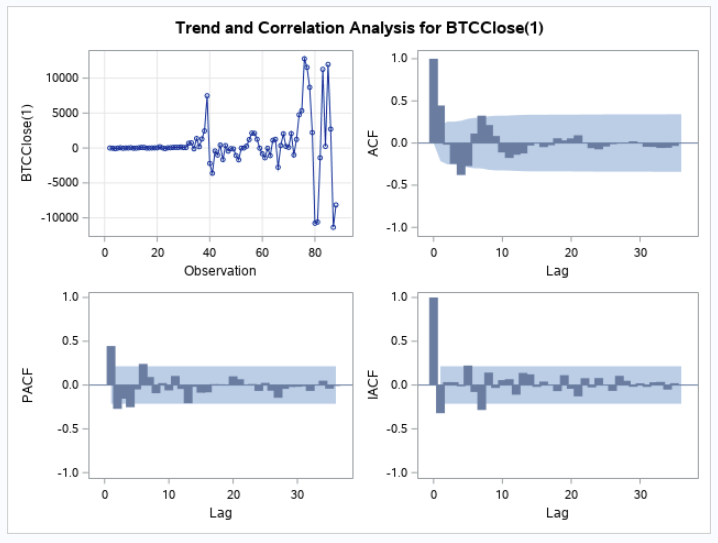
Where p = order of nonseasonal AR terms (found from the PACF Plot)

d = order of nonseasonal differencing

q = order of nonseasonal MA terms (found from the ACF Plot)

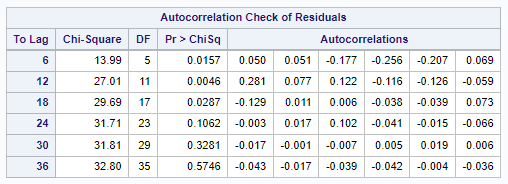
* ARIMA(0,1,1)
* ARIMA(2,1,0)
* ARIMA(2,1,1)
* ARIMA(3,1,0)

We built the ARIMA models using the above options to determine which model residuals have white noise, lower AIC, BIC and that passes parsimony principle too.



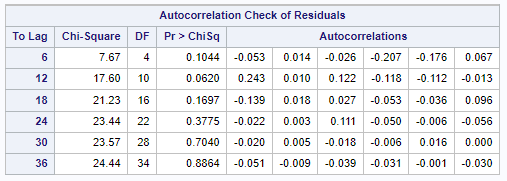
**Diagnostic check for an adequate model (are estimated residuals white noise) –**

/\*ARIMA(0,1,1)\*/



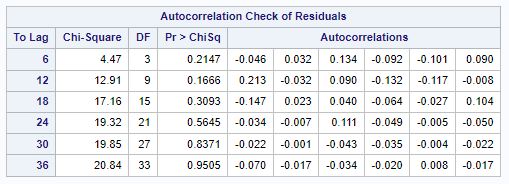
For model ARIMA(0,1,1) the p-values of residuals are less than alpha(0.05) for the first 18 lag which shows that the residuals are not white noise using ARIMA(0,1,1) model.

/\*ARIMA(2,1,0)\*/



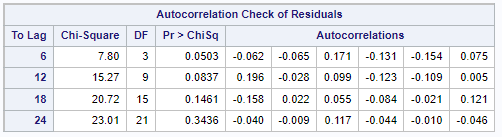
As p-values of all the residuals are greater than alpha which means the residuals are white noise using model ARIMA(2,1,0) and the model is appropriate too.

/\*ARIMA(2,1,1)\*/



For model ARIMA(2,1,1) as well, all p-values of the residuals are greater than alpha which shows that the residuals are white noise using the model ARIMA(2,1,0) and the model is appropriate too.

/\*ARIMA(3,1,0)\*/



For model ARIMA(3,1,0), the p-values are greater than alpha and as mentioned earlier, this model is appropriate too.

Checking the AIC and BIC of all the ARIMA models:

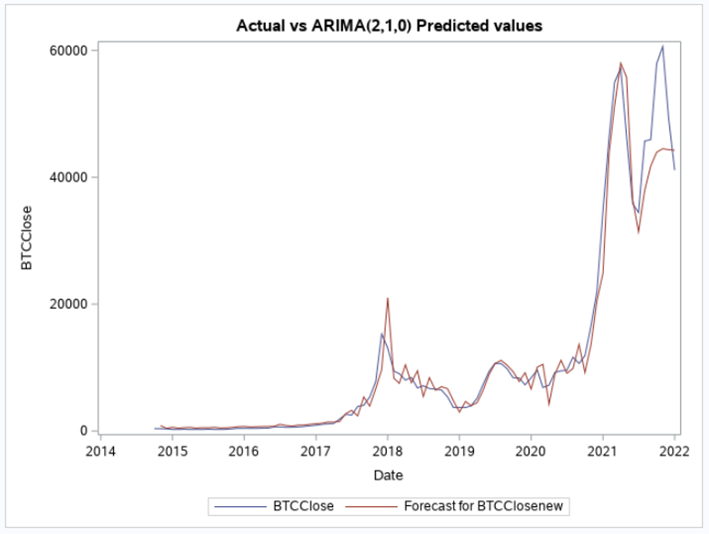
|  |  |  |
| --- | --- | --- |
| **ARIMA Model** | **AIC** | **BIC** |
| ARIMA(0,1,1) | 1661 | 1666 |
| **ARIMA(2,1,0)** | **1659** | **1667** |
| ARIMA(2,1,1) | 1656 | 1666 |
| ARIMA(3,1,0) | 1658 | 1668 |

We will discard the first model as the residuals for it were not white noise and from the next 3, as the AIC and BIC for all the other models are approximately similar, as per the Parsimony principle (Keep It Simple) we select the second model ARIMA(2,1,0) for forecasting the Closing price of Bitcoin.

**Forecast with the chosen model –**

|  |  |  |
| --- | --- | --- |
| **Model** | **MAPE -Model Fit** | **MAPE -Model Accuracy** |
| ARIMA (2,1,0) | 36.77 | 15.73 |

Using model ARIMA(2,1,0), we first divided the data in train and test set to analyze how well the ARIMA model fits both the sets compared to our earlier models (Holt’s and Damped Trend). We considered the last 6 months data as a test set, because the same period is considered in Holt’s and Damped trend to calculate the error metrics and to keep the comparison unbiased.

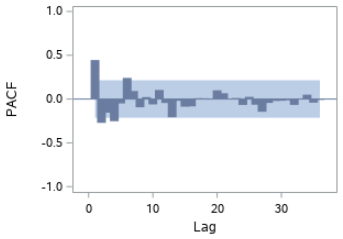
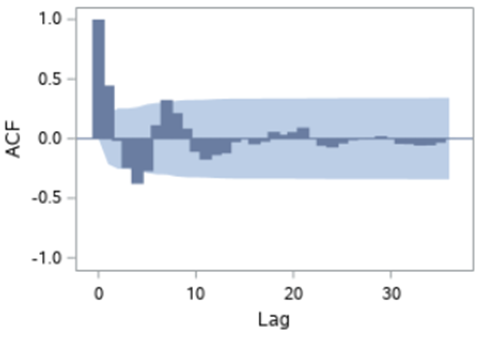


From the Actual Vs ARIMA predicted values plot and from model fit 36.77% we see that the ARIMA model is fitting the data good, but the lower model accuracy of 15.73% confirms that the model predicts the test data better and the ARIMA model is a good fit for the data.

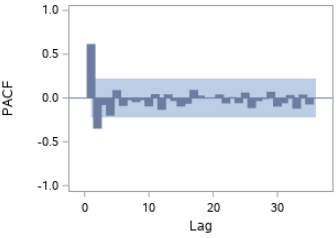
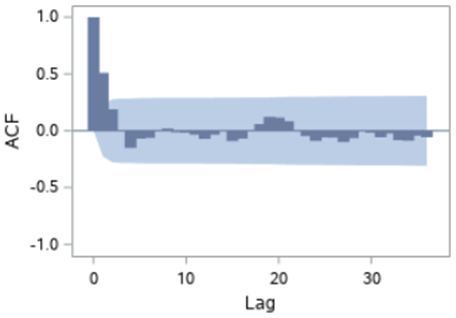
**Check for Overfitting –**

Before we compare all our potential models and forecast the closing price of the Bitcoin using the best model, we need to check if the ARIMA model selected is overfitting the data. To check for overfitting, we have used the same train set divided data and of the first 82 months to evaluate the ARIMA(2,1,0) model for overfitting

ACF and PACF of whole data



ACF and PACF of train data – 82 months.



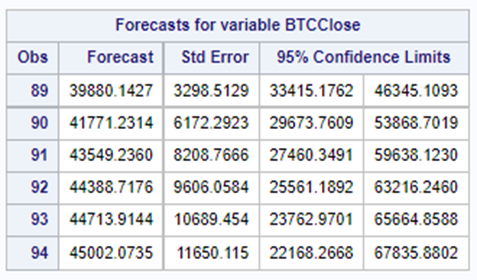
From the ACF and PACF plot of the train set, we see that there is not much change in the values of ACF and PACF spikes and patterns, which confirms that the model is not overfitting the data. Also, the model fit 36.77% and a lower model accuracy of 15.73 % confirm that the ARIMA model is not overfitting the actual data.

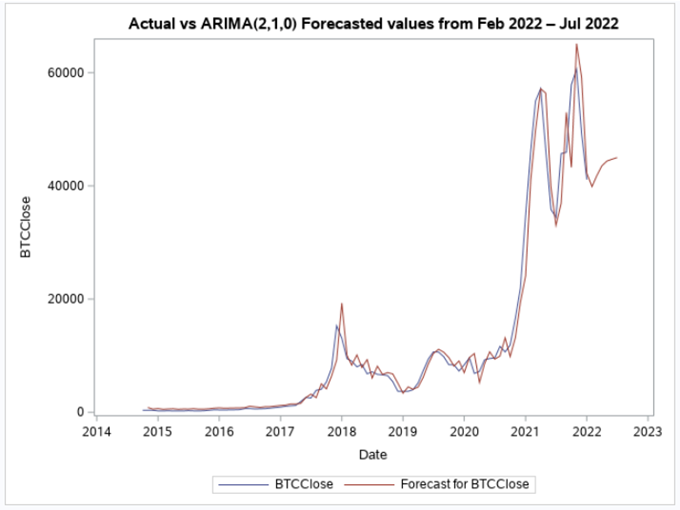
# **Conclusion:**

|  |  |  |
| --- | --- | --- |
| **Model** | **MAPE FIT** | **MAPE ACCURACY** |
| Holt's Exponential Smoothing | 16.05 | 40.14 |
| Damped Trend Exponential Smoothing | 14.59 | 33.5 |
| ARIMA(2,1,0) | 36.77 | 15.73 |

Now that we have reviewed and analyzed all the 3 potential models, from the MAPE fit and MAPE accuracy details of all the 3 models, we see that the Damped trend has the lowest MAPE for model fit of all the 3 models and it fits the train set better, However, ARIMA model has the lowest MAPE accuracy of all 3. As our goal is to find a model which will forecast better values, we will continue with the ARIMA model to forecast our closing price of Bitcoin.

Using ARIMA(2,1,0), below are the forecasted values for the next 6 months i.e. from Feb 2022 – July 2022.





Forecasting changes in the stock market will always consider an element of unpredictability. Fluctuations in bitcoin closing price do not follow seasonal or cyclical components because adjustments to the stock market are triggered by buyers and sellers impacting the supply and demand of investments. As investing in bitcoins increases in popularity, this report provides a good starting point to forecast future closing prices. Additional consideration should be given to account for the large fluctuations in price to generate accurate forecasts. Also, the analysis and forecasting done were completely based on timeseries historical data and as the stock market there are also additional external factors that affect the pricing of Bitcoin, getting understanding and performing analysis on them will help to improve the Bitcoin price forecasting.

References

DeMatteo, M. (2022, October 31). *Bitcoin price history 2009-2022 | nextadvisor with Time*. Time. https://time.com/nextadvisor/investing/cryptocurrency/bitcoin-price-history/

Keating, B., & Wilson, J. H. (2019). *Forecasting & predictive analytics: With ForecastX* (7th ed.). McGraw-Hill Education.

Nagadia, M. (2022, February 19). *Bitcoin price dataset*. Kaggle. https://www.kaggle.com/datasets/meetnagadia/bitcoin-stock-data-sept-17-2014-august-24-2021

# **/\*SAS CODE\*/**

proc import out=BTC datafile="/home/u61150141/sasuser.v94/Project/BTC-USD.csv"

dbms=csv replace;

run;

/\*Timeseries plot of the Actual data\*/

proc sgplot data=BTC;

/\*series x=date y=Open;\*/

series x=date y=Close;

title "Daily Bitcoin Close value over the years";

xaxis label="Date";

yaxis label="Bitcoin Close Value";

run;

/\* ACF plot of Actual data\*/

proc timeseries data=BTC plots=acf out=\_null\_;

var Close;

corr acf/nlag=48;

run;

/\*Converting to Month\*/

proc expand data=BTC out=BTC\_Month from=Day to=Month;

convert Close/observed=average;

id date;

run;

/\*Timeseries plot of Monthly data\*/

proc sgplot data=BTC\_Month;

series x=date y=Close;

title "Monthly Bitcoin Close value over the years";

xaxis label="Date";

yaxis label="Bitcoin Close Value";

run;

/\*ACF plot of Monthly data\*/

proc timeseries data=BTC\_Month plots=acf out=\_null\_;

var close;

corr acf/nlag=48;

run;

/\*================================================================================================================================\*/

/\*Holts Exponential Smoothing\*/

Proc esm data=BTC\_Month print=all outfor=BTCHOut lead=6 back=6 out=\_null\_ plot=forecasts;

/\*Print option to all the results and values print will append data, lead option to specify the number of periods to forecast, out will create a new column \*/

Forecast Close/model=Linear;

run;

proc sgplot data=BTCHOut;

series x=\_timeid\_ y=Actual;

series x=\_timeid\_ y=Predict;

title "Actual vs Holts Exponential Predicted values";

run;

/\*================================================================================================================================\*/

/\*DampedTrend Exponential Smoothing\*/

Proc esm data=BTC\_Month print=all outfor=BTCDOut lead=6 back=6 out=\_null\_ plot=forecasts;

/\*Print option to all the results and values print will append data, lead option to specify the number of periods to forecast, out will create a new column \*/

Forecast Close/model=damptrend;

run;

proc sgplot data=BTCDOut;

series x=\_timeid\_ y=Actual;

series x=\_timeid\_ y=Predict;

title "Actual vs DampedTrend Exponential Predicted values";

run;

/\*================================================================================================================================\*/

/\*Simple Regression - Worst Fit but though of trying\*/

/\*Building a regression model\*/

data BTC\_Month;

set BTC\_Month;

t=\_n\_;

Close\_Train=Close;

if t>82 then Close\_Train=.;

tsq=t\*t;

run;

Proc REG Data=BTC\_Month;

model Close\_Train=t/clb dwprob;

output out=BTCSROut1 predicted=BTCSRPred residual=BTCSRres; /\*p= is same as predicted create an o/p database it will have original x and y + predicted values and residuals columns\*/

run;

data BTCSROut1;

set BTCSROut1;

mape\_fitSR=(abs(BTCSRres)/Close\_Train)\*100;

if t>82 then mape\_accSR=(abs(BTCSRPred-Close)/Close)\*100;

run;

PROC Means data=BTCSROut1 mean;

var mape\_fitSR mape\_accSR;

run;

proc sgplot data=BTCSROut1;

series x=date y=Close;

series x=date y=BTCSRPred;

title "Actual vs Simple Regression Predicted values";

run;

/\*Checking for Non-Linearity\*/

Proc REG Data=BTC\_Month;

model Close\_Train=tsq/clb;

output out=BTCSROut2 predicted=BTCSRPred residual=BTCSRres; /\*p= is same as predicted create an o/p database it will have original x and y + predicted values and residuals columns\*/

run;

data BTCSROut2;

set BTCSROut2;

mape\_fitSR=(abs(BTCSRres)/Close\_Train)\*100;

if t>82 then mape\_accSR=(abs(BTCSRPred-Close)/Close)\*100;

run;

PROC Means data=BTCSROut2 mean;

var mape\_fitSR mape\_accSR;

run;

proc sgplot data=BTCSROut2;

series x=date y=Close;

series x=date y=BTCSRPred;

run;

proc sgplot data=BTCSROut2;

series x=date y=Close;

series x=date y=BTCSRPred;

title "Actual vs Non-Linear T-sq Predicted values";

run;

/\*================================================================================================================================\*/

/\*ARIMA \*/

/\*Changing Close to BTCClose to avoid Keyword close confusion\*/

data BTC\_AR;

set BTC\_Month;

rename Close=BTCClose;

run;

/\*We know the data has trend so using ARIMA to check for whiteNoise\*/

/\*Check if the data has white noise\*/

Proc ARIMA data=BTC\_AR;

identify var=BTCClose nlag=36 whitenoise=ignoremiss; /\*ignoremiss is for ljung box test\*/

run;

/\*Pr > ChiSq is less than alpha, data is not white noise. we reject the null hypothesis \*/

/\*Trend in data from ACF plot so we apply first order differencing\*/

Proc ARIMA data=BTC\_AR;

identify var=BTCClose(1) nlag=36 whitenoise=ignoremiss;

run;

/\*After differencing Data is stationary now from the ACF plot.

Possible models are ARIMA(0,1,1) ARIMA(2,1,0) ARIMA(2,1,1) and ARIMA(3,1,0) are the options\*/

/\*ARIMA(0,1,1)\*/

Proc ARIMA data=BTC\_AR;

identify var=BTCClose(1) nlag=36 whitenoise=ignoremiss;

estimate q=1 whitenoise=ignoremiss;

run;

/\*ARIMA(2,1,0)\*/

Proc ARIMA data=BTC\_AR;

identify var=BTCClose(1) nlag=36 whitenoise=ignoremiss;

estimate p=2 whitenoise=ignoremiss;

run;

/\*White noise in the Data\*/

/\*ARIMA(2,1,1)\*/

Proc ARIMA data=BTC\_AR;

identify var=BTCClose(1) nlag=36 whitenoise=ignoremiss;

estimate p=2 q=1 whitenoise=ignoremiss;

run;

/\*ARIMA(3,1,0)\*/

Proc ARIMA data=BTC\_AR;

identify var=BTCClose(1) nlag=24 whitenoise=ignoremiss;

estimate p=3 whitenoise=ignoremiss;

run;

/\*Based on parsimony principe and the AIC and BIC Using Model 2 for forecasting \*/

/\*ARIMA(2,1,0)\*/

/\*Dividing the data in Train and Test set to evaluate how well the model fits to test set before forecasting\*/

data BTC\_AR;

set BTC\_AR;

t=\_n\_;

BTCClosenew=BTCClose;

if t>82 then BTCClosenew=.;

run;

Proc ARIMA data=BTC\_AR;

identify var=BTCClosenew(1) nlag=36 whitenoise=ignoremiss;

estimate p=2 whitenoise=ignoremiss;

forecast lead=6 id=date interval=month out=BTCout1; /\*Forecasting test data 6 months \*/

run;

data BTC\_AR\_combined;

merge BTCout1 BTC\_AR;

mape\_fit\_AR=(abs(Residual)/BTCClosenew)\*100;

if t>82 then mape\_acc\_AR=(abs(BTCClose-Forecast)/BTCClose)\*100;

run;

PROC Means data=BTC\_AR\_combined mean;

var mape\_fit\_AR mape\_acc\_AR;

run;

proc sgplot data=BTC\_AR\_combined;

series x=date y=BTCClose;

series x=date y=forecast;

title "Actual vs ARIMA(2,1,0) Predicted values";

run;

/\* Checking for overfitting of ARIMA model \*/

proc arima data=BTC\_AR;

identify var=BTCClosenew(1) nlag=36 whitenoise=ignoremiss; /\*Considered first 82 months(Train set)\*/

run;

proc arima data=BTC\_AR;

identify var=BTCClose(1) nlag=36 whitenoise=ignoremiss; /\*Considered the entire data\*/

run;

/\*Forecasting the next 6 months using the final model\*/

Proc ARIMA data=BTC\_AR;

identify var=BTCClose(1) nlag=36 whitenoise=ignoremiss;

estimate p=2 whitenoise=ignoremiss;

forecast lead=6 id=date interval=month out=BTCForecastout; /\*Forecasting next 6 months \*/

run;

proc sgplot data=BTCForecastout;

series x=date y=BTCClose;

series x=date y=forecast;

title "Actual vs ARIMA(2,1,0) Forecasted values from Feb 2022 – Jul 2022";

run;