Cole Sprouse

**Cryptocurrency Forecasting using ARIMA and ARMA-GARCH Models**

**Introduction:**

I used the “quantmod” package in R that is designed for traders so they can easily import up to date financial data. I’m using the yahoo finance as the source for the cryptocurrency data. The columns in each cryptocurrency data frame are open\_price (price of cryptocurrency at the start of the day), highest\_price (highest price in the given day), lowest\_price (lowest\_price in the given day), closing\_price (price of cryptocurrency at the end of the day), volume (USD amount traded in the given day), adjusted (adjusted closing price). For my analysis, I am going to use adjusted closing price to forecast Bitcoin and Ethereum within the dates November 29, 2020, and November 29, 2021.

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

Above are the time series plots of Bitcoin (left) and Ethereum (right). Since November 29, 2020, Ethereum has roughly increased 636.4%, while Bitcoin has increased roughly 218%. These are very high percent changes within a year, and I wanted to test whether I can predict the volatility and forecast the closing price for Bitcoin and Ethereum while also comparing the two to see if Bitcoin is easier to predict or vice versa.

**Statistical Analysis**

From the ACF and PACF plots for Bitcoin and Ethereum we can see that there is no seasonal trend and that these time series plots are random walks.

Chart

Description automatically generatedChart

Description automatically generated

**Bitcoin:**

Before we move onto to further analysis, I need to make sure that the autocorrelation and data is stationary. To do this I will first take the log of the data. This will help stabilize the variance of the series.

Chart

Description automatically generated

The time series and ACF plot still are not stationary. So, I need to take the first difference of the log of bitcoin data. This is the same as the log daily returns. This will in turn take help stabilize the mean of the time series.

**A picture containing chart

Description automatically generated**

A picture containing text, device

Description automatically generated

**Ethereum:**

We are also going to take the log difference for Ethereum. When we take the log difference, we see a significant lag at lag 10 for both ACF and PACF plot; however since it’s just one significant lag we will disregard it. As you can see though, there is a high cluster of volatility right between May and June.

Timeline

Description automatically generated

To forecast the data, I am going to using an ARIMA model first. ARIMA models have a linear structure, so it is widely used in model forecasting. They also have three parameters/orders p, d, and q. ARIMA(p,d,q) can be written as:

Xt = α + φ1Xt-1 +…+ φpXt-p+Wt + θ1Wt-1+…+ θqWt-q

Where p and q are the orders of the model and Wt is a white noise time series following a Normal distribution (N,sigma^2).

To reiterate, ARIMA models is a method for linear data and does not work well for forecasting when there is new information. This is the best model for linearity; however it does not have a role when the data is nonlinear. Financial returns data is not linear and very volatile so a model that will help us forecast nonlinear data will be GARCH model. We know this model will work better because of the residual plot has clusters of volatility.

Chart

Description automatically generated

GARCH models are used frequently for financial time series to estimate the volatility of returns. A series of returns (rr) follows a GARCH(1,1) model if it can be written as

Rt = μt + σt εt

μt = φ0 + φ1rt-1 ,

σ2t = α0 + α1 r2t-1 + β1 σ2t-1

assumed that, α1+ β1 <1

**ARIMA Model (Bitcoin):**

Given that the ACF and PACF plot do not have any significant lags (shown on page 2), we choose the orders p = 0 and q = 0 and d = 1 because we differenced the data to transform to become stationary. So our ARIMA model is ARIMA(0,1,0).

Graphical user interface

Description automatically generated with low confidence

This graph above is residual diagnostics to ensure the normality and making sure that the data is stationary before I forecast the data. The Ljung-Box statistic p values are essential for determining whether a series is white noise and given that all of the p values are above 0.05 confirms that the data is stationary because we fail to reject the null hypothesis that the data is not stationary.

**ARIMA(0,1,0) Model Forecast (Bitcoin):**

Chart, line chart

Description automatically generated

This graph shows 20 days of predicted values on the logarithm bitcoin price. The grey around these points is the confidence interval, as you can see the more days we forecast the higher variance and wider the confidence interval is because we have less information.

**ARMA(0,0) – GARCH(1,1) Fitting (Bitcoin):**

Using the garchFit() function in R we use orders p=0 and q = 0 for ARMA and GARCH(1,1). Also, this function runs the Ljung-Box Test and we obtain p-values greater than 0.05 which is great.

**Output:**

**Statistic P-value**

Ljung-Box Test R Q(10) 8.54869 0.5754012

Ljung-Box Test R Q(15) 14.6528 0.4767037

Ljung-Box Test R Q(20) 21.30279 0.379507

Ljung-Box Test R^2 Q(10) 9.80234 0.4580026

Ljung-Box Test R^2 Q(15) 12.3709 0.6507652

Ljung-Box Test R^2 Q(20) 17.54214 0.6175416

**ARMA(0,0) – GARCH(1,1) Forecasting (Bitcoin):**

meanForecast meanError standardDeviation

1 0.003515319 0.03805905 0.03805905

2 0.003515319 0.03823608 0.03823608

3 0.003515319 0.03840571 0.03840571

4 0.003515319 0.03856827 0.03856827

5 0.003515319 0.03872411 0.03872411

6 0.003515319 0.03887351 0.03887351

7 0.003515319 0.03901678 0.03901678

8 0.003515319 0.03915419 0.03915419

9 0.003515319 0.03928600 0.03928600

10 0.003515319 0.03941245 0.03941245

This table shows the prediction interval for each step ahead and shows that the volatility increases as steps ahead increase.

**ARIMA Model Fitting (Ethereum):**

At lag 10 we see a significant lag for both the pacf and acf plot for Ethereum, however we will disregard this and use the auto.arima() function to get the orders because of this. The function suggests an AR(1) so p =1, q = 0 and d = 1: ARIMA(1,1,0). The residual diagnostics look good, and the data is stationary and normal.

A picture containing graphical user interface

Description automatically generated

**ARIMA(1,1,0) Model Forecast (Ethereum):**

Chart, scatter chart

Description automatically generated

Similarly, to bitcoin graph, this graph shows 20 days of predicted values on the Ethereum price. The grey around these points is the confidence interval, as you can see the more days we forecast the higher variance and wider the confidence interval is because we have less information.

**ARMA(1,0) – GARCH(1,1) Fitting (Ethereum):**

Using the garchFit() function in R we use orders p=1 and q = 0 for ARMA and GARCH(1,1). Also, this function runs the Ljung-Box Test and we obtain p-values greater than 0.05 which confirms that it is stationary.

**Output:**

**Statistic P-value**

Ljung-Box Test R Q(10) 15.60884 0.1113907

Ljung-Box Test R Q(15) 20.01109 0.1715076

Ljung-Box Test R Q(20) 25.81083 0.1721508

Ljung-Box Test R^2 Q(10) 2.503908 0.9908187

Ljung-Box Test R^2 Q(15) 3.999581 0.9977387

Ljung-Box Test R^2 Q(20) 15.18628 0.765651

**ARMA(0,1) – GARCH(1,1) Forecasting (Ethereum):**

meanForecast meanError standardDeviation

1 0.004811632 0.04786682 0.04786682

2 0.006995159 0.04851198 0.04838389

3 0.006834452 0.04899436 0.04886409

4 0.006846280 0.04944210 0.04931043

5 0.006845409 0.04985857 0.04972560

6 0.006845473 0.05024621 0.05011203

7 0.006845469 0.05060725 0.05047195

8 0.006845469 0.05094369 0.05080735

9 0.006845469 0.05125738 0.05112006

10 0.006845469 0.05154999 0.05141176

This table shows the prediction interval for Ethereum price each step ahead and shows that the volatility increases as steps ahead increase.

**So which model is best and which coin is easier to predict?**

The GARCH Model is better than the ARIMA given that cryptocurrency data is not linear and there are clusters of volatility. The GARCH model uses new information and analyzes the series with up-to-date information. It then uses this information to forecast future values. The ARIMA model only predicts future value linearly and does not incorporate volatility in the data.

Overall, Ethereum is more volatile, because the standard deviation (volatility) increases more as more days are forecasted compared to bitcoin. Essentially Bitcoin is easier to predict because of this. I believe that these models are not the best to predict price action of cryptocurrency longer than couple days; however, it can be useful for next day forecasting when the volatility is low. The cryptocurrency market, compared to other financial markets, is extremely volatile and I believe the reasoning for a lot of these clusters of volatility are in these so called “bull markets” that happen every so often when cryptocurrency is heavily talked about in the news and more innovative decentralized ideas come out for example, decentralized finance, decentralized gaming and NFTs. We can see from the forecasting of the GARCH model that Ethereum has higher volatility, and this makes sense because of the percentage change in the past year compared to bitcoin. Ethereum has increased, respectively, 3 times more than bitcoin has this year. Some reasonings for this could be because of the applications you can build on Ethereum network while bitcoin is only used for a store of value.