Contemporary Issues Module (CIM)

Bitcoin Price Prediction Project

AYA, GWENAELLE

I- Introduction

Bitcoin is one of the most famous crypto currencies, since it got the highest valuation during the last decades since its creation in 2009. Unlike a normal currency, it is generated by the blockchain process. It allows the storage and transfer of the information safely without any institution to regulate it.
We can think, from their different approaches to data, that Blockchain and Data Sciences are purely independent disciplines. While the Blockchain is currently in full emergence, especially with the global craze around cryptocurrencies, data science is an already well-established technology.
However, these two innovations, which are making it possible to revolutionize the world of work and the relationship between humans and technology, are not that far apart. Statisticians and data scientists provide statistics on the cryptocurrency market, and develop predictive analysis algorithms to forecast market trends and developments.
We will see it in more detail in this project using a dataset from Kaggle composed of seven variables and 1556 observations.

- **Date**: from 2013 to 2017
- **Open:** Means the price at which a stock started trading when the opening bell rang.
- **High:** Means the highest price in a given period of time.
- **Low:** Means the lowest price in a given period of time.
- **Close:** Refers to the price of an individual stock when the stock exchange closed shop for the day.
- **Volume:** Is the amount of shares bought/sold of a stock in a given period of time. It is important because the more volume the more people agree with the price of the stock.
- **Market Cap:** Or market capitalization—refers to the total value of all a company's shares of stock.

II- Methodology

1. Dealing with the missing data

While going through the data, we notice a lack of information within the volume variable. We can check for missing data during the whole period and see the variables side by side to get a better perspective on the activity of each station. Using the library **missingno**, we get a fast and easy-to-understand visualization of when the data is present in each variable.

2. Time Series modeling of Bitcoin prices

In this section we will be comparing the variables Open, High, Low and Close along with each other so that we can understand the behavior of each variable and if there's a correlation between them. The main focus will be mainly on the Close variable. This can be explained by the fact that it is a stock's closing price that determines how a share performs during the day. The close price is considered the reference point for any time frame. It's the price traders agreed on after all the action throughout the day. When researching historical stock price data, financial institutions, regulators, and individual investors use the closing price as the standard measure of the stock's value as of a specific date. For example, a stock's close on December 31, 2019, was the closing price for not only that day, but also that week, month, quarter, and year. We mainly use **forecast**, **tidyverse, statsmodels, matplotlib** and **sklearn.metrics** libraries from Python to perform our analysis. We will be using graphs that are relevant and significant to our study which are Candlestick, Boxplots, line charts and a Heatmap.

3. Forecasting

In this section, we use a notebook from Kaggle, to forecast the highest price of bitcoin. Here, we are presenting the building steps of our model. Thus we are going to recall the definition of a **time series and its characteristics**. Then, we will present the **augmented Dickey- Fuller Test** and see how it works. To finish, we will present the **ARIMA** model used and test it .

A **time series** is a set of numerical values taken at equally spaced intervals over the time. Mathematically, a time series model is described as the sum of three components which are the trend, the seasonality and the zero-mean error. The **trend** is the overall direction of the series. Besides, we can sometimes observe repeated patterns at regular intervals of time. This is what we call a **seasonal** phenomenon. We again use **matplotlib** to visualize the price evolution into the given data set. Since the seasonality and the trend are respectively varying the mean and the variance over time, they define non stationary series which are to forecast.
Therefore, before predicting the bitcoin's price, we had to look for the stationarity of the time series. To do that we choose to perform the **augmented Dickey- Fuller Test**. It is a **unit root test**, where the null hypothesis is the presence of a unit root in the time series. The unit root is also called **random walk with drift** because it is describing the unpredictable irregularities. It is mathematically defined by the

following equation: $Y_t = \alpha Y_{t-1} + \beta X_e + \varepsilon$, where Y_t is the value of the time series at t and X_e is an exogenous variable.If the null hypothesis is accepted the time series is not stationary and we can't apply ARIMA.
Hence, we use the function **adfuller** from the **statsmodels.tsa.stattools** library to perform this test. The augmented Dickey Fuller test is an expanded version of the Dickey Fuller Test for higher order . Let $Y_t = \alpha Y_{t-1} + \beta X_e + \phi_1 \Delta Y_{t-1} + \ldots \phi_p \Delta Y_{t-p} + \varepsilon_t$, the null hypothesis is that α is equal to 1. The null hypothesis is rejected when the p-value is less than the significance level. The algorithm is computing the difference between $Y_t = \alpha Y_{t-1} + \beta X_e + \sum_{i=1}^{p} \phi_i Y_{t-i} + \varepsilon_t$ and Y_t, which is equal to the unit root equation, and calculate the Dickey coefficient ($\widehat{\alpha} / std(\widehat{\alpha})$) and find the critical value into the Dickey Fuller distribution. If the coefficient is less the critical value we reject the null hypothesis. We saw that it is necessary to make the series stationary before forecasting it. To do it, we have to apply the logarithmic function thanks to the **log** function from **Numpy** to remove the variance variation by removing the seasonality. Besides, we differentiate the series by subtracting Y_t and Y_{t-1} . We use the **shift** method from python.

The ARIMA model is using the value of the past to predict the future value. It is built with to model which are **autoregressive** model (AR) defined by:
$Y_t = \alpha + \sum_{i=1}^{p} \beta_i Y_{t-i} + \varepsilon_t$, where α is the intercept, and β_i is the coefficient of the lag i.
and by the **moving average** model (MA) defined by:
$Y_t = \alpha + \sum_{i=1}^{q} \phi_i \varepsilon_{t-i} + \varepsilon_t$ where $\phi_i \varepsilon_{t-i}$ are the error of the autoregressive models at each lag. Then, since in the ARIMA model the predicted values Y_t are defined by:
$Y_t = \alpha + \sum_{i=1}^{q} \phi_i \varepsilon_{t-i} + \sum_{i=1}^{q} \phi_i \varepsilon_{t-i} + \varepsilon_t$. Therefore, we need to find p, q parameters which are the orders of the AR and MA model respectively , so we use the **autocorrelation** function and the **partial autocorrelation** function from **statsmodels.graphics.tsaplots** library to study the lags, and find the parameters of the model to make the best forecast. Then, we use the **summary** on the model to see its parameters significance and the model's accuracy. The accuracy of the model can be evaluated by the BIC and AIC values. The **Akaike Criterion** or **AC** is defined by the following formula: $AIC = 2k - 2ln(L)$. Where k is the number of observations estimated by the model and L is the **likelihood function**'s maximum. Briefly, we can say the likelihood function measures the acceptability of the model parameter given the data set. Besides, the **BIC** or **Bayesian Information Criterion** is defined by : $BIC = -2ln(\bar{L}) + kln(N)$ where N is equal to the number of observations and $\bar{L}$is the maximum of the maximized likelihood function. The best model will be the model with the least BIC or the AIC. Moreover, to validate the model, we use the **Ljung Box test** on the model residuals to check if they are independently distributed. Besides, we use the acf and pacf function on the residuals to see if they are looking like a white

noise time series. To finish, we remove the differentiation and the logarithm, to obtain the predicted values.

III- Results

1. Missing data

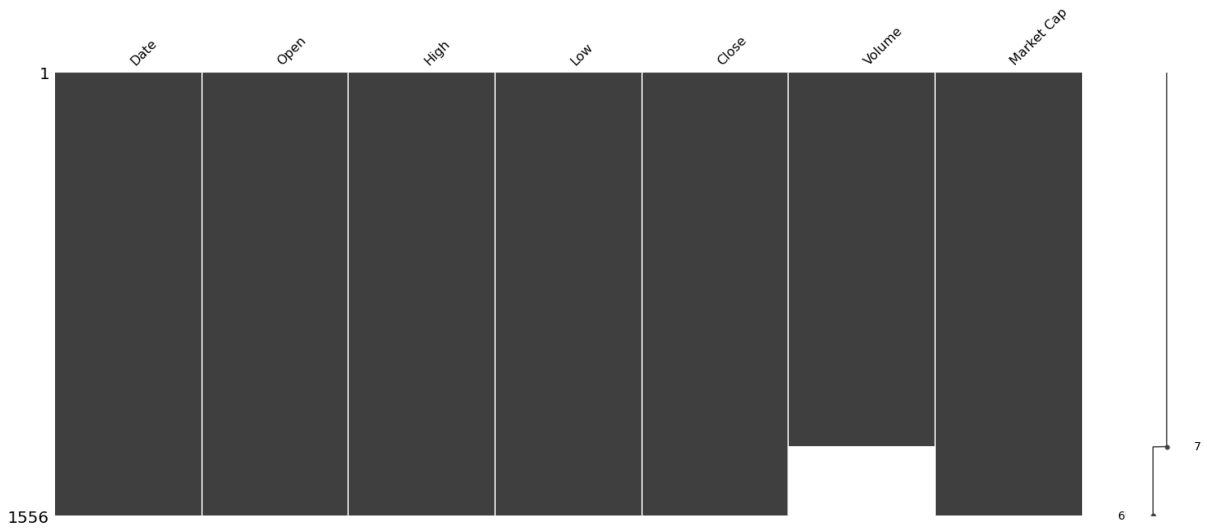


Figure 1: Missing data in each variable of our dataset during the period 2013-2017

In Figure 1, missing values are represented by gaps. The columns are the dataset variables mainly Date, Open, High, Low, Close, Volume and Market Clap. We can denote that almost all the variables are not missing values. The dataset is pretty clean except for Volume.

1. Time Series modeling of Bitcoin prices

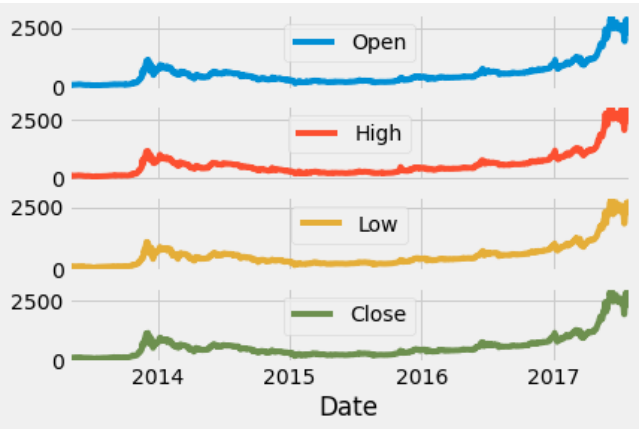


Figure 2: variation of Close values during the period 2013-2017

Line charts are used to represent the relation between Date and the 4 variables Close, High, Low, Open on a different axis. Note that the values have a positive trend overall, but there are ups and downs over the course. There is a significant rise starting from 2017 where it reached the pick.

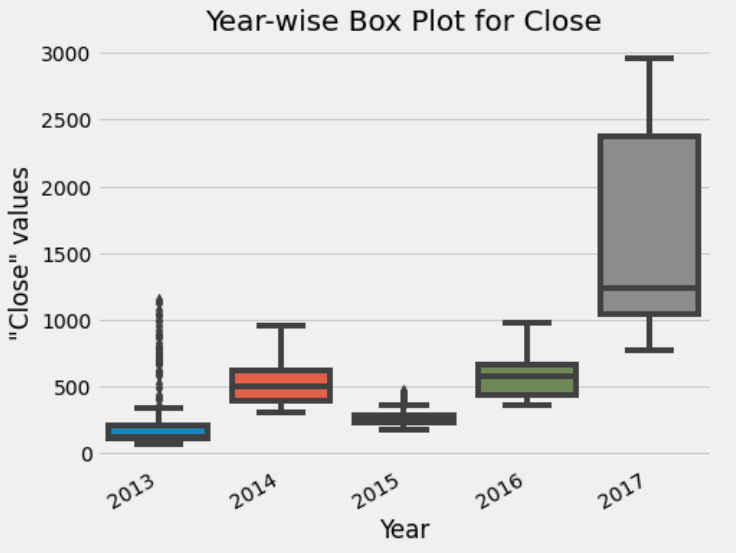


Figure 3: Close value ranges (Year-wise Box Plot).

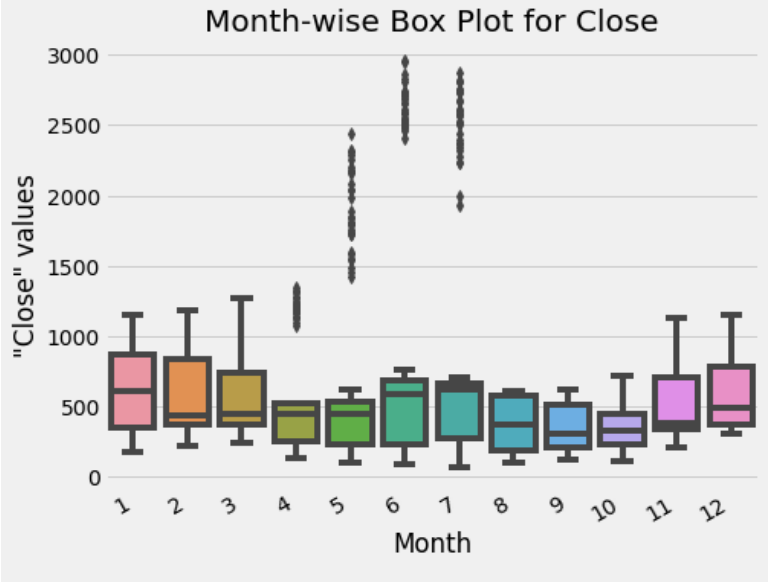


Figure 4: Close value ranges (Month-wise Box Plot).

From figure 3 and 4, we are able to obtain an intuition of the ‘Close’ value ranges of each year as well as each month. For figure 3, the 5 boxplots do *not* overlap with one another, then there *is* a difference each year. The medians are not similar either and the boxes are of different sizes. Short boxes mean their data points consistently hover around the center values. Taller boxes imply more variable data. We can observe that

the value range is much higher for 2017 compared to the remaining years. For figure 4, there are overlapping boxes. We can observe that the value range is slightly higher in Jan and Feb, compared to other months. And there are outliers, dotted outside the whiskers for Apri, May, June and July.

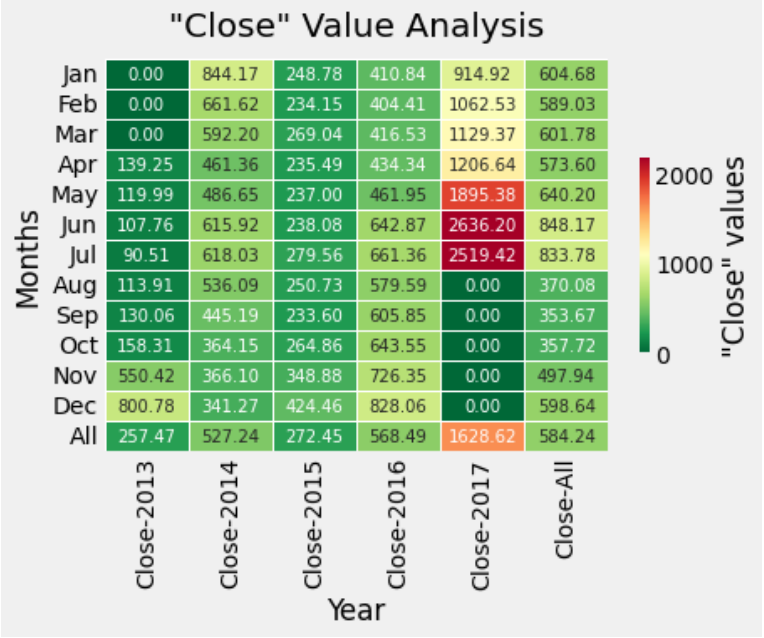


Figure 5: Variation of Close across Years as well as Months.

We can interpret the trend of the Close values across the years sampled over 12 months, variation of values across different years. In figure 5, The color of the tiles is representative of the variation of Close across Years as well as Months, differentiated using a Colormap. The colormap provides a visually appealing way to identify monthly patterns or anomalies over the years. The highest value is represented by the red color which is 2636.20 and the lowest 0 and this can be explained by the fact that there were no recorded values during the first months of 2013 and the last months of 2017.

Figure 6: Daily price movements of securities for August 2017

A candlestick chart is a type of financial chart that displays the price movements of securities over time.

Each candlestick represents the price movement of the security on a particular day. The color of the candlestick tells us whether the price closed higher (green) or lower (red) than the previous month.

Green candles show rising prices, so the open price is at the bottom of the body and the close price is at the top.

Red candles show falling prices. The opening price is therefore at the top of the body and the closing price is at the bottom.

For 2017-07-06 we have a Doji candlestick that has no body because the open and close prices are the same. This can generally be interpreted as a sign of indecision in the market, and is a possible indicator of an upcoming price reversal.

For 2017-07-29 we have an umbrella that has a particularly long lower wick. A red umbrella is also known as a hammer. When you see a hammer, it often means that the asset is the subject of some serious buying action - and the price may be on the rise soon.

If the body occupies most of the candle, with very short wicks (or no wicks visible) on either side, this may indicate a strong bullish (on a green candle) like 201707-18 or strongly bearish (on a red candle) sentiment like 2017-07-16.

2. Forecasting with ARIMA:

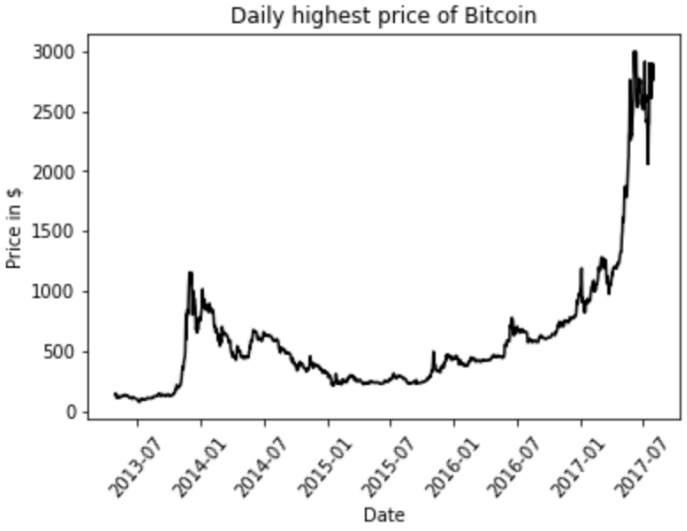


Figure 7: Daily highest price of the bitcoin

In Figure 7, it represents the higher price of bitcoin each day. It is easy to see that the overall trend is increasing, even if there are some local drops. Besides, it is hard to identify any seasonal patterns just by looking at this graph. So, it is easy to see that it is not stationary. Therefore, we performed the logarithmic, hence we obtained Figure 8.a. Then, we computed the differentiation method in order to transform the series, as we can see in Figure 8.b below. By using the augmented Dickey Fuller test, we find that in both cases the series after transformation are stationary.

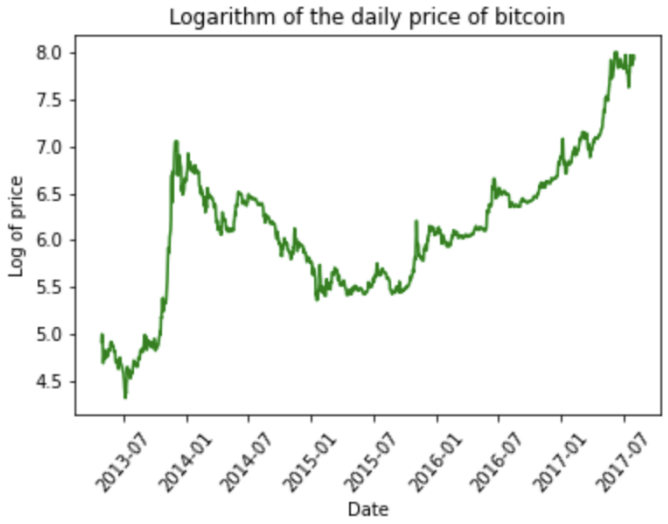


Figure 8.a Logarithm of the price

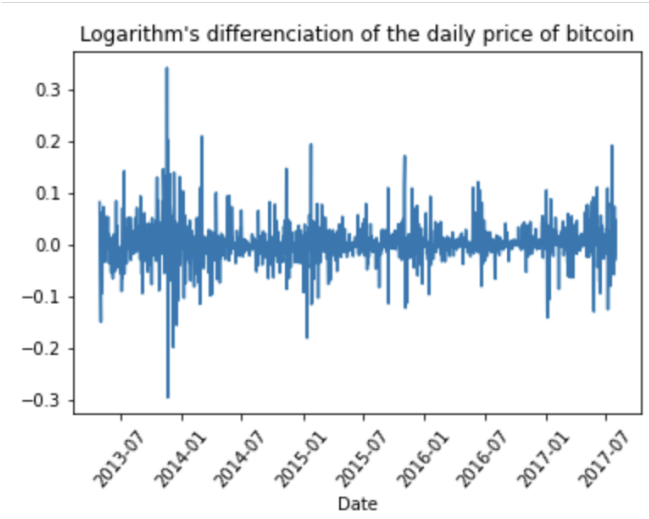


Figure 8.b Differentiation of the logarithm price.

The figures below show the partial autocorrelation function and the autocorrelation function of the time series represented in Figure 8.a. In Figure 9.a, we can see that values are tailing off after lag 2. This means the order of the autocorrelation function order p is equal to 2. Besides, in Figure 9.b, we can denote that function is tailing off after the third lag. Then, the moving average order is equal to 3. Moreover in both plots we can see that there are some seasonal components. It implies a mixed ARIMA model called SARIMA which is more sophisticated. We will not go further in the analysis of these components in this report.

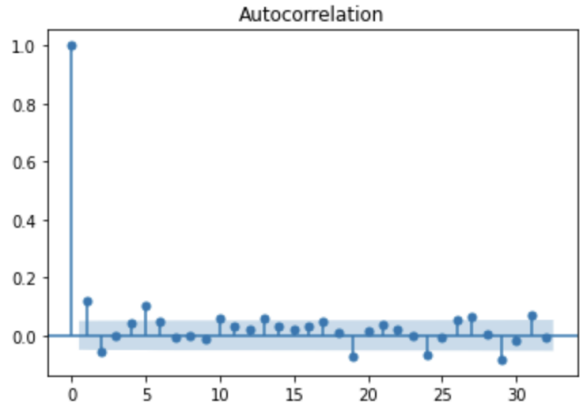


Figure 9.a: Autocorrelation function on the time series

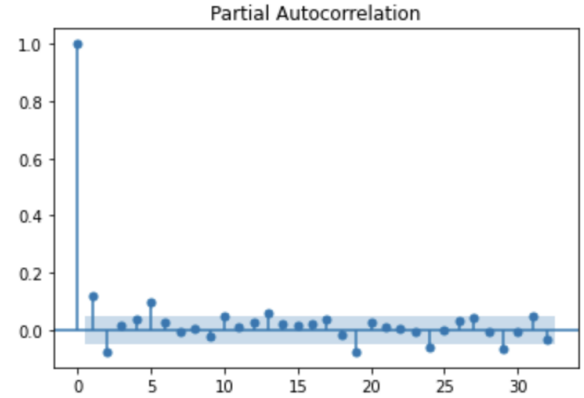


Figure 9.b: Partial autocorrelation function on the time series

After choosing the parameter, we use the model to predict the values. Then, when we remove the change made at the beginning of the analysis to make the time series stationary. We obtain Figure 10.

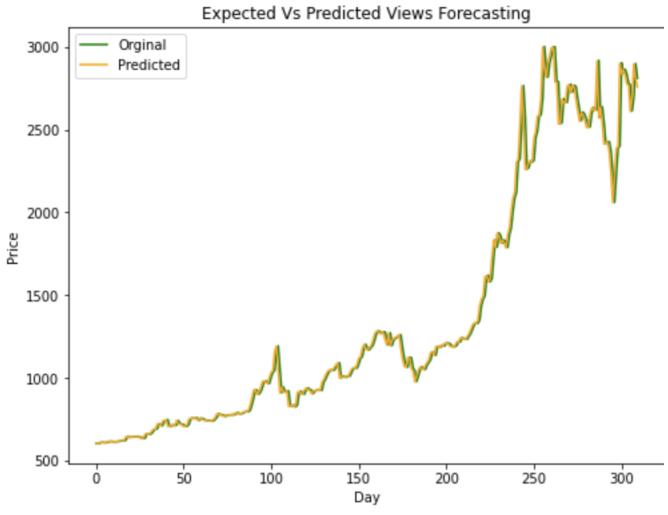


Figure 10: The predicted price and the real prices

With Figure 10, we cannot see properly the difference between the two curves. Therefore, by looking at Figure 11, we can see the daily percentage of error. The values are between 0 and 17%. But we still do not know if our model is the most accurate.

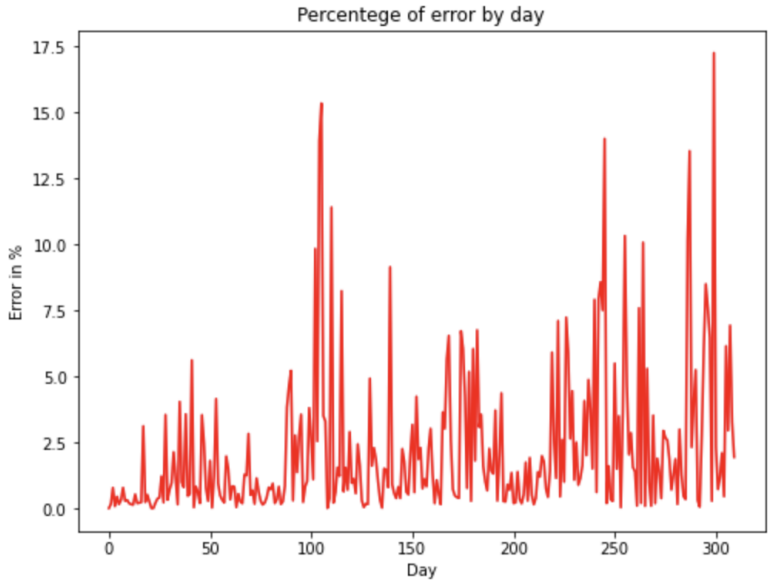


Figure 11: Errors' percentage of the time series

By looking at the Ljung Box test , we see that at 21 lags the p-value is less than 0.05. So, we can reject the null hypothesis that the residuals are independently distributed. Therefore, the residuals cannot be modeled as a white noise, we can validate this hypothesis by looking at Figure 12.

IV- Discussion

1. Time Series modeling of Bitcoin prices

From the previous graphs we can conclude that Bitcoin has been witnessing a rise in the behavior, people are investing more than the previous years and shares and trades are more important.the value range is much higher for 2017 compared to the remaining years. There are ups and downs over the course but the values have a positive trend overall.
Although we notice from the Candlestick chart that the number of green and red boxes are equal, we can't conclude anything unless we take a closer look since a long wick at the bottom of a candle, for example, can mean traders are buying an asset as prices fall, which can be a good indicator that the asset is moving higher. However, a long wick on the top of a candle could suggest that traders are looking to take profits, signaling a large potential selloff in the near future. So It's important to note that while one-candle signals can be an

important clue, an accurate reading of the market requires understanding the bigger picture.

2. Forecasting

From the prediction graph above , we saw that it is possible to get an approximation of the bitcoin prices, by using an ARMA(2,0, 3) model. In figure 10, the results look accurate. Nevertheless, after performing the Ljung Box test on the residuals we saw that they were not white noise, so that the chosen model is not the best one.

V- Conclusion

These emerging projects make it possible to bridge the gap between two technologies that will continue to revolutionize our lives in the years to come, going even further day by day. This combination helps ensure more resources, security, reliability, and speed for all users.

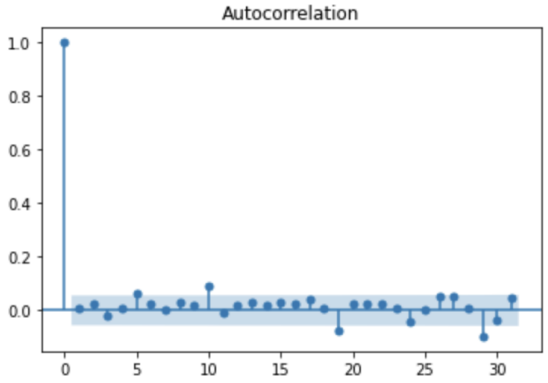
VI- Appendix

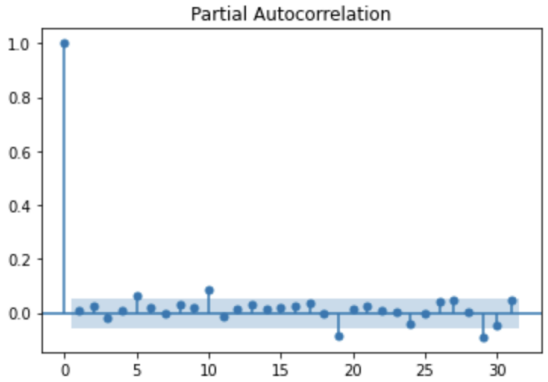
Arima model Result:

```
                              ARMA Model Results
==============================================================================
Dep. Variable:                      y   No. Observations:                 1244
Model:                     ARMA(2, 3)   Log Likelihood                2301.056
Method:                       css-mle   S.D. of innovations              0.038
Date:                Thu, 16 Dec 2021   AIC                          -4588.112
Time:                        18:08:00   BIC                          -4552.230
Sample:                             0   HQIC                         -4574.619

==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0012      0.001      0.978      0.328      -0.001       0.004
ar.L1.y        1.0441      0.149      7.022      0.000       0.753       1.336
ar.L2.y       -0.6769      0.115     -5.873      0.000      -0.903      -0.451
ma.L1.y       -0.9218      0.148     -6.210      0.000      -1.213      -0.631
ma.L2.y        0.4687      0.119      3.942      0.000       0.236       0.702
ma.L3.y        0.1669      0.029      5.744      0.000       0.110       0.224
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            0.7713           -0.9394j            1.2155           -0.1406
AR.2            0.7713           +0.9394j            1.2155            0.1406
MA.1            0.7864           -0.8655j            1.1694           -0.1326
MA.2            0.7864           +0.8655j            1.1694            0.1326
MA.3           -4.3802           -0.0000j            4.3802           -0.5000
-----------------------------------------------------------------------------
```

ACF and PACF on the residuals to visualize the Ljung Box test:


Autocorrelation
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25
30


Partial Autocorrelation
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25
30