BONUS PROJECT

BITCOIN PRICE PREDICTION

ABHISHEK RAMESH GADEKAR

B20EE089

INTRODUCTION

This report is about predicting bitcoin price using time series forecasting. Time series forecasting is quite different from other machine learning models because -

1. It is time-dependent. So, the basic assumption of a linear regression model that the observations are independent doesn't hold in this case.

2. Along with an increasing or decreasing trend, most time series have some form of seasonality trends, i.e. variations specific to a particular time frame

In this article time series models like AR(Auto-Regressive model), MA (Moving Average model), and ARIMA (Autoregressive Integrated Moving Average model) are used for forecasting the price of bitcoin.

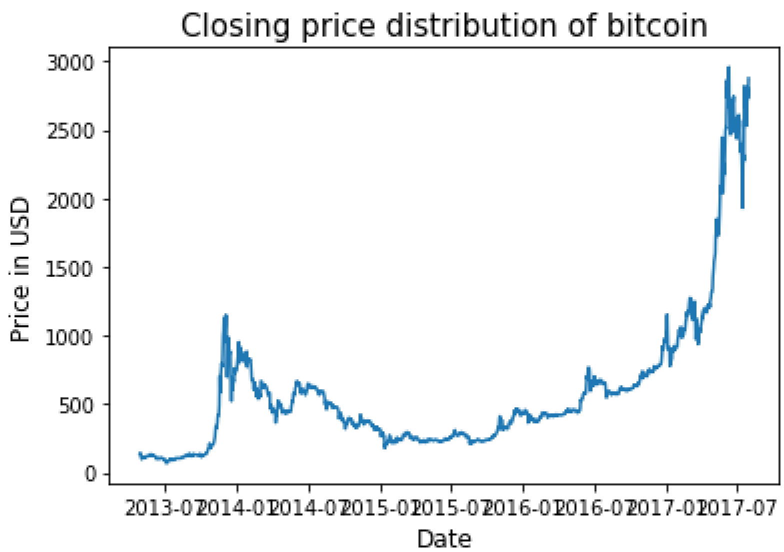
ABOUT DATASET:

The dataset contains the date, open, high, low, close, volume, and market cap values of bitcoin from April 2013 to August 2017.

	Date	Open	High	Low	Close	Volume	Market Cap
0	2017-07-31	2763.24	2889.62	2720.61	2875.34	860,575,000	45,535,800,000
1	2017-07-30	2724.39	2758.53	2644.85	2757.18	705,943,000	44,890,700,000
2	2017-07-29	2807.02	2808.76	2692.80	2726.45	803,746,000	46,246,700,000
3	2017-07-28	2679.73	2897.45	2679.73	2809.01	1,380,100,000	44,144,400,000
4	2017-07-27	2538.71	2693.32	2529.34	2671.78	789,104,000	41,816,500,000

DATA VISUALIZATION AND PRE-PROCESSING:

The closing price of the bitcoin is plotted with respect to the date.



TESTING THE STATIONARY:

Augmented Dicky Fuller(ADF) Test:

The Augmented Dicky Fuller test is a type of statistical test called a unit root test. The intuition behind a unit root test is that it determines how strongly a time series is defined by a trend. There are no. unit root tests and ADF is one of the most widely used

1. Null Hypothesis (H0): Null hypothesis of the test is that the time series can be represented by a unit root that is not stationary.

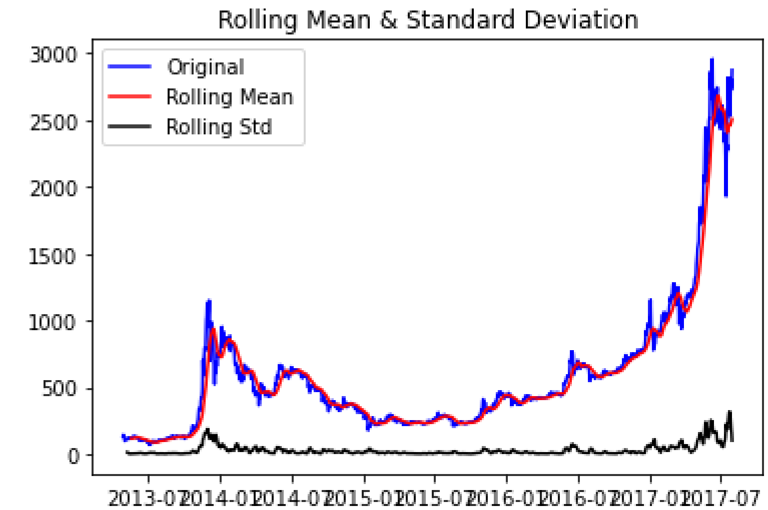
2. **Alternative Hypothesis (H1):** Alternative Hypothesis of the test is that the time series is stationary.

Interpretation of p-value:

1. p-value > **0.05:** Accepts the Null Hypothesis (H0), the data has a unit root and is non-stationary.

2. p-value < = **0.05:** Rejects the Null Hypothesis (H0), the data is stationary.

Statistics of the data:



STATISTICS:

ADF Statistics: 2.535589

p-value: 0.999060

The graph is non-stationery

Critical values:

1%: -3.435

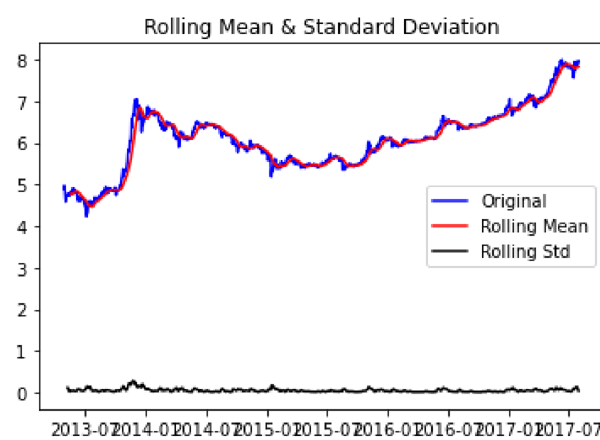
5%: -2.863

10%: -2.568

Since the p-value is greater than 0.05 the time series is nonstationary. We now process the data and use transformations to make the series stationary.

Log Transformation:

Log transformation is used to unscrew highly skewed data. Thus helping in the forecasting process.



STATISTICS:

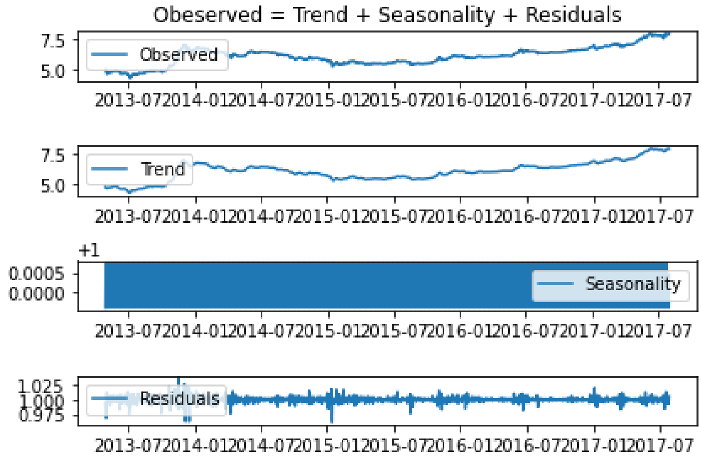
ADF Stastistic: -0.790465
p-value: 0.821907
The graph is non stationery
Critical values:
1%: -3.435
5%: -2.863
10%: -2.568

Since the p-value is still greater than 0.05, we need to apply some more transformations. The next transformation we are going to apply is differencing.

Removing Trend and Seasonality with Decomposition

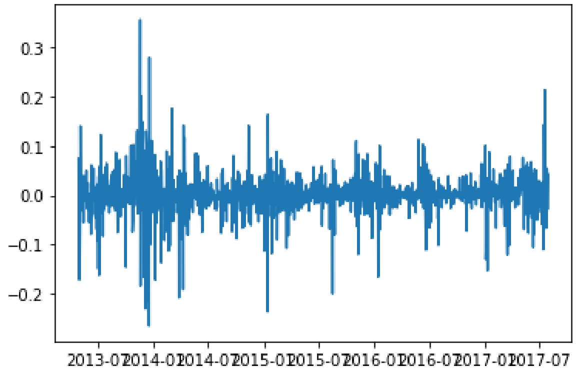
Time series decomposition is a mathematical procedure that transforms a time series into multiple different time series. The original time series is often split into 3 component series:

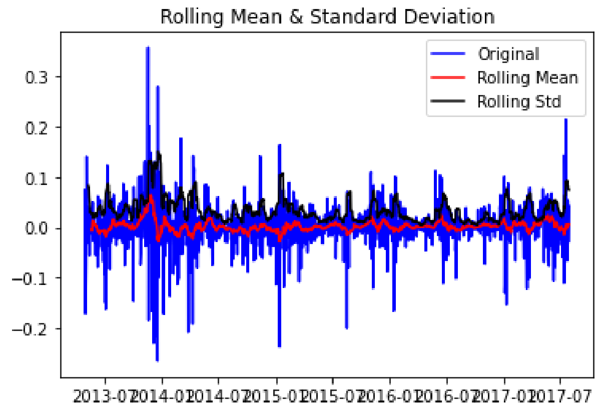
- Seasonal: Patterns that repeat over a fixed period of time. For example, a website might receive more visits during weekends; this would produce data with seasonality of 7 days.
- Trend: The underlying trend of the metrics. A website increasing in popularity should show a general trend that goes up.



Removing Trend and Seasonality with Differencing:

In this case of differencing to make the time series stationary the current value is subtracted from the previous values. Due to this, the mean is stabilized and hence the chances of stationarity of time series are increased.





STASTICIS:

ADF Stastistic:-7.285034

p-value: 0.000000

The graph is stationery

Critical values:

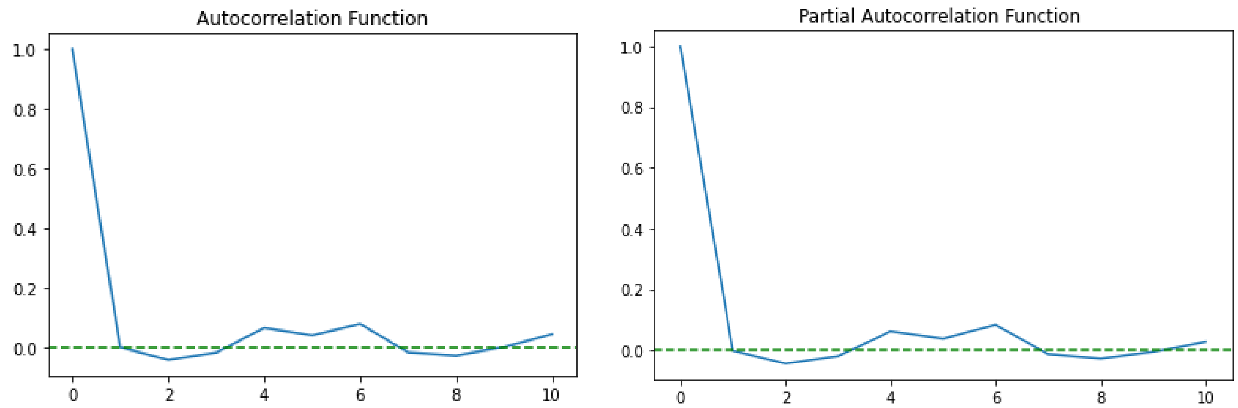
1%: -3.435

5%: -2.863

10%: -2.568

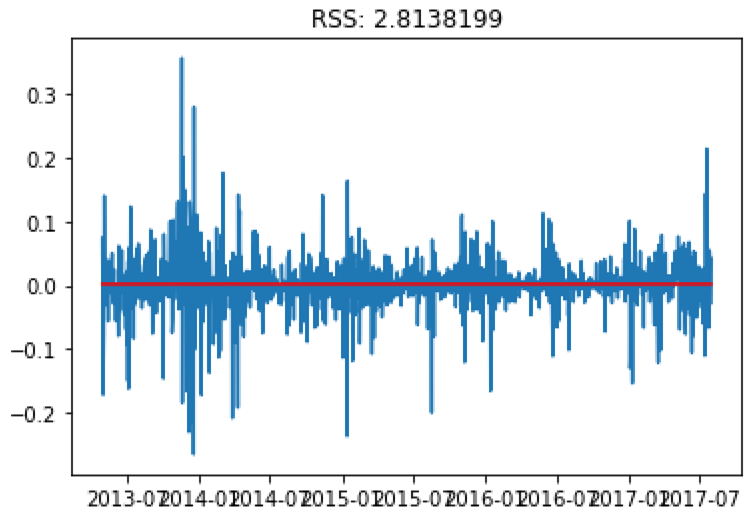
Since the p-value is less than 0.05, the time series is now stationary, and therefore we can apply time series, forecasting models.

Plotting the auto co-relation and partial co-relation functions:



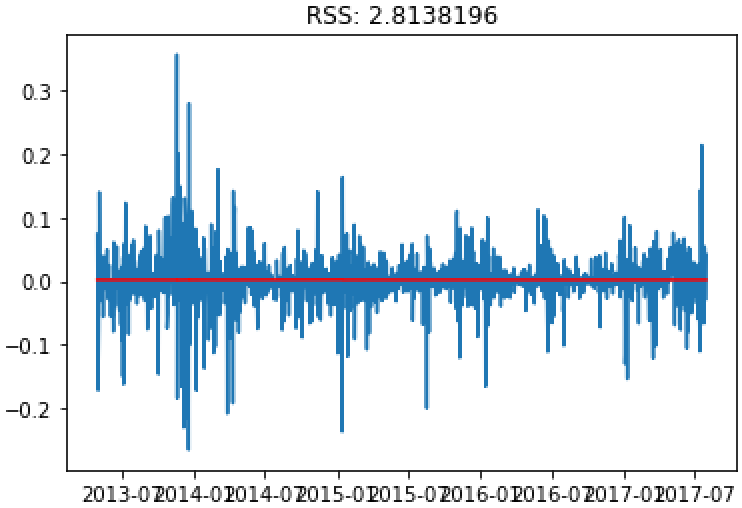
Auto-Regressive Model:

An autoregressive model is a time series forecasting model where current values are dependent on past values.



Moving Average Model

In the moving average model, the series is dependent on past error terms.

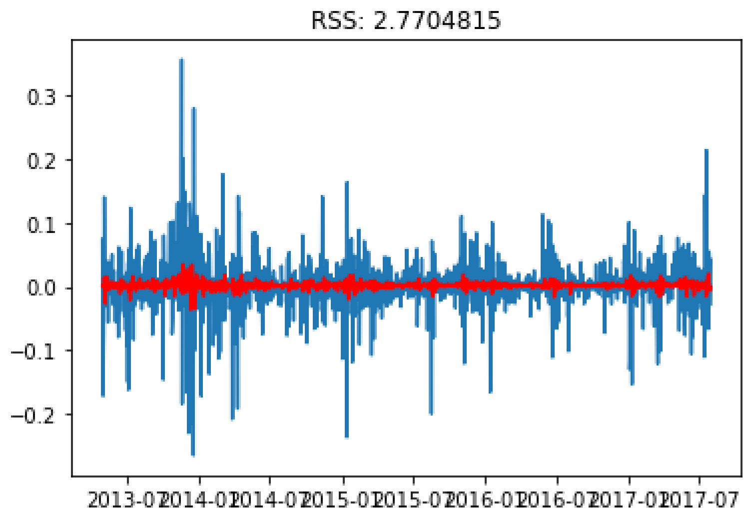


Summary of the Moving Average model is given below:

```
                             ARIMA Model Results
==============================================================================
Dep. Variable:                D.Close   No. Observations:                 1555
Model:                 ARIMA(0, 1, 1)   Log Likelihood                2703.220
Method:                       css-mle   S.D. of innovations              0.043
Date:                Sun, 01 May 2022   AIC                          -5400.441
Time:                        17:51:20   BIC                          -5384.393
Sample:                    04-29-2013   HQIC                         -5394.473
                         - 07-31-2017
=================================================================================
                    coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------
const             0.0020      0.001      1.829      0.068      -0.000       0.004
ma.L1.D.Close    -0.0012      0.027     -0.045      0.964      -0.053       0.051
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
MA.1          833.4465           +0.0000j          833.4465            0.0000
-----------------------------------------------------------------------------
```

Auto-Regressive Integrated Moving Average Model(ARIMA):

It is a combination of both AR and MA models. It makes the time series stationary by itself through the process of differencing. Therefore differencing need not be done explicitly for the ARIMA model.



ARIMA MODEL RESULTS:

```
from statsmodels.tsa.arima_model import ARIMA
from pandas import DataFrame
print(results_ARIMA.summary())
```

```
                             ARIMA Model Results
==============================================================================
Dep. Variable:                D.Close   No. Observations:                 1555
Model:                 ARIMA(8, 1, 0)   Log Likelihood                2715.504
Method:                       css-mle   S.D. of innovations              0.042
Date:                Sun, 01 May 2022   AIC                          -5411.009
Time:                        17:51:26   BIC                          -5357.516
Sample:                    04-29-2013   HQIC                         -5391.117
                         - 07-31-2017
=================================================================================
                    coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------
const             0.0020      0.001      1.662      0.097      -0.000       0.004
ar.L1.D.Close    -0.0056      0.025     -0.221      0.826      -0.055       0.044
ar.L2.D.Close    -0.0432      0.025     -1.702      0.089      -0.093       0.007
ar.L3.D.Close    -0.0144      0.025     -0.566      0.572      -0.064       0.035
ar.L4.D.Close     0.0689      0.025      2.704      0.007       0.019       0.119
ar.L5.D.Close     0.0395      0.026      1.549      0.122      -0.010       0.089
ar.L6.D.Close     0.0845      0.026      3.299      0.001       0.034       0.135
ar.L7.D.Close    -0.0129      0.026     -0.502      0.616      -0.063       0.037
ar.L8.D.Close    -0.0273      0.026     -1.062      0.288      -0.078       0.023
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1            0.5050           -1.2650j            1.3620           -0.1896
AR.2            0.5050           +1.2650j            1.3620            0.1896
AR.3            1.6805           -0.2518j            1.6993           -0.0237
AR.4            1.6805           +0.2518j            1.6993            0.0237
AR.5           -0.6481           -1.3097j            1.4613           -0.3231
AR.6           -0.6481           +1.3097j            1.4613            0.3231
AR.7           -1.7738           -0.2415j            1.7902           -0.4785
AR.8           -1.7738           +0.2415j            1.7902            0.4785
-----------------------------------------------------------------------------
```

Thus we see that the RSS (Residual Sum of Squares) error is minimum for the ARIMA model. Therefore ARIMA model is the best among the three models because of the use of dependence on both lagged values and error terms. Therefore it is further used to calculate the mean square error. Here in the below code snippet, the dataset is divided into train and test.

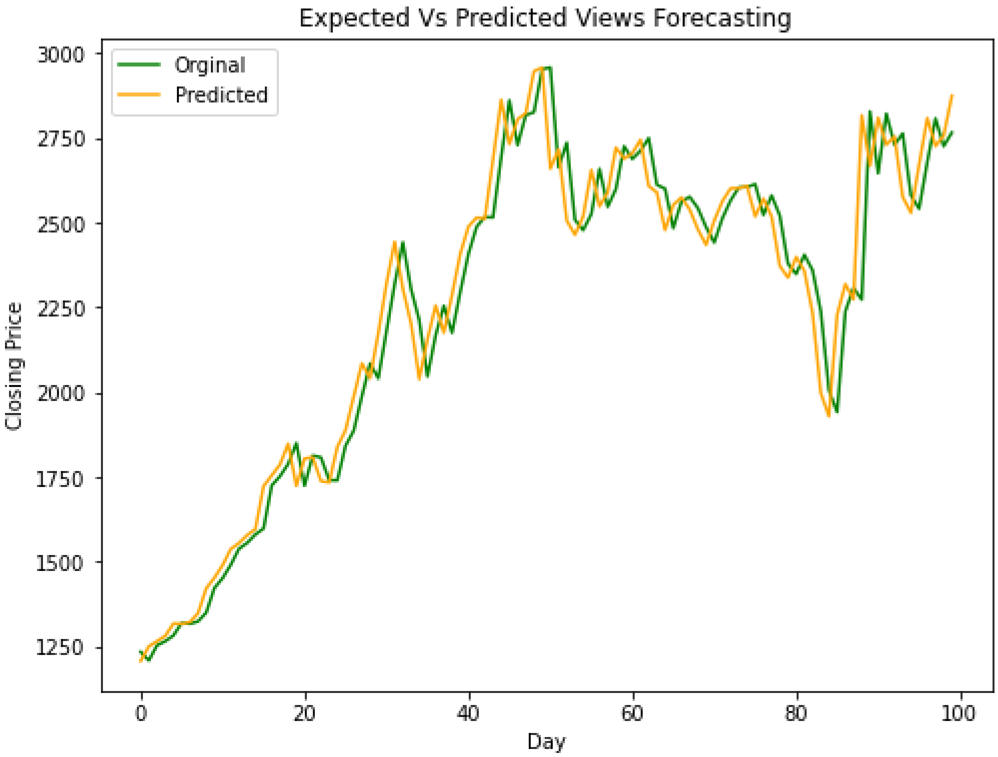
For every value in the test, we apply an ARIMA model and then the error is calculated and then after iterating over all values in the test set the mean error between predicted and expected value is calculated.

```
predicted = 2613.835793,   expected = 2518.660000,   error = 3.778827  %
predicted = 2523.203679,   expected = 2571.340000,   error = 1.872033  %
predicted = 2580.654924,   expected = 2518.440000,   error = 2.470375  %
predicted = 2521.053568,   expected = 2372.560000,   error = 6.258791  %
predicted = 2379.066832,   expected = 2337.790000,   error = 1.765635  %
predicted = 2348.468564,   expected = 2398.840000,   error = 2.099825  %
predicted = 2405.299996,   expected = 2357.900000,   error = 2.010263  %
predicted = 2359.650922,   expected = 2233.340000,   error = 5.655696  %
predicted = 2239.002236,   expected = 1998.860000,   error = 12.013960  %
predicted = 2006.206510,   expected = 1929.820000,   error = 3.958219  %
predicted = 1942.244793,   expected = 2228.410000,   error = 12.841677  %
predicted = 2238.149971,   expected = 2318.880000,   error = 3.481423  %
predicted = 2307.325761,   expected = 2273.430000,   error = 1.490953  %
predicted = 2272.890193,   expected = 2817.600000,   error = 19.332404  %
predicted = 2829.051256,   expected = 2667.760000,   error = 6.045943  %
predicted = 2646.110464,   expected = 2810.120000,   error = 5.836389  %
predicted = 2822.356841,   expected = 2730.400000,   error = 3.367889  %
predicted = 2730.087040,   expected = 2754.860000,   error = 0.899246  %
predicted = 2763.766195,   expected = 2576.480000,   error = 7.269072  %
predicted = 2580.946825,   expected = 2529.450000,   error = 2.035890  %
predicted = 2541.493498,   expected = 2671.780000,   error = 4.876393  %
predicted = 2679.029939,   expected = 2809.010000,   error = 4.627255  %
predicted = 2808.092214,   expected = 2726.450000,   error = 2.994451  %
predicted = 2726.150578,   expected = 2757.180000,   error = 1.125404  %
predicted = 2766.298165,   expected = 2875.340000,   error = 3.792311  %

 Means Error in Predicting Test Case Articles : 3.593134  %
```

Note: Mean Error in Predicting Test Case Articles: 3.593133 %

FINAL PREDICTION AND VISUALISING:



Thus we have seen that we were able to use different transformations and models to predict the closing price of bitcoin with a mean error of 3.59 %., which is shown graphically above.