Ethereum Gas Cost Prediction Using Time Series Analysis

ABSTRACT

Since the blockchain technologies are absolute secure but hectic as well takes costly hardware producing wasteful energies. To cater with this problem Ethereum Blockchain came up with idea to minimize the consumption of energy by writing the efficient smart contract, i.e., the amount of code and the way it is written will cost likewise gas cost which programmer has to pay. To measure it on Ethereum Block chain the motive of the project Was to predict the gas cost using time series analysis. The TSA model we implemented in our project based on Autoregression Integrated Moving Average (ARIMA). We were successful in training upon the given dataset based on Dates and the gas cost.

INTRODUCTION

The number of state and global variables taking part upon the main chain the higher the storage and processing power they will be acquiring for the execution of the respective smart contract. But to understand in terms of blockchain the programmer of the smart contract and the one who runs it on the node might be different persons. So, the person who is offering his hardware and electricity is paying for the directly dependent upon the quality of the smart contract. To get the reward as well as to minimize the consumption of energy Ethereum has proposed the term GAS. GAS is nothing but the TAX which is inversely proportional to the efficiency of code (Smart contract).

So, we came up with the idea of predicting the future consumption of GAS by using the historic data.

Proposed Methodology

The prediction of future based on the previous data using probability and statistical tools termed as *Time Series Analysis*. This is a bit trickier and different from traditional machine learning because we are given with one parameter which itself is the X and Y.

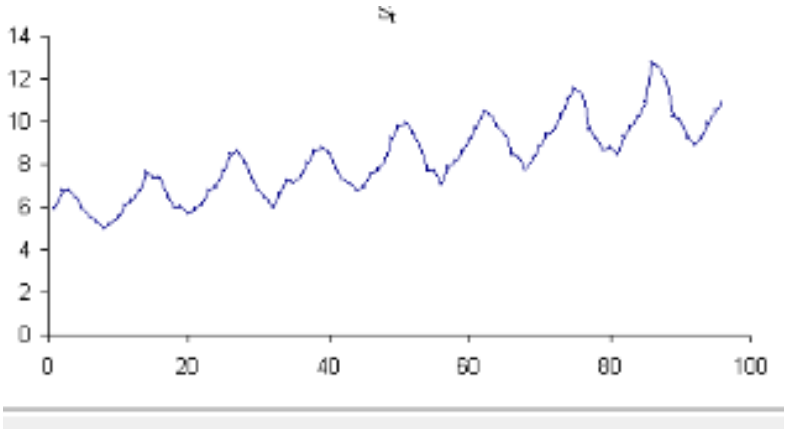
The model of TSA we are using is ARIMA model.

- What is Arima?

    1. AR" reflects the evolving variable of interest is regressed on its own prior values,
    2. "MA" infers that the regression error is the linear combination of error terms values happened at various stages of time priorly, and
    3. "I" shows the data values are replaced by the difference between their values and the previous values.

- Observations before performing ARIMA
  Time series shows some behavioral properties such as Trend, Seasonality
  And unexpected events.
    1. Trend is a pattern in data that shows the movement of a series to relatively higher or lower values over a long period of time. It is observed when there is an increasing or decreasing slope in the time series.

    2. Seasonality is a characteristic of a time series in which the data experiences regular and predictable changes that recur over the calendar year. For Example, the inflation rate in the month of Ramadan
       Is relatively higher.



3. Unexpected events mean some dynamic changes occur in history which cannot be captured and predicted. These events are rare and random but do play the critical role in model training. For Example,
   The occurrence of pandemic and its aftereffects on the parameters considering that events-based series.

These three events are observed, and the respective actions are taken before making the model on ARIMA.

To train the model we must make time series stationery.

What is the stationary time series?

Since to predict the future and train the model over the dataset. For Training the model, series must be stationary. It has constant mean and variance over the time. These two statistics define the series in mathematical terms. time series has above mentioned components Trend, seasonality, and unexpected events. These events needed to be handled because they make time series nonstationary.

Mathematically:

We have a series:

T1	T2	T3	T4	T5
10	20	45	56	88

If this series is stationary, then the mean across at any two given points must be approximately the same.

How to check for stationary Time series

- Augmented Dickey Fuller Test

  There is a statistical Test known as Augmented Dickey Fuller Test. It builds the null hypothesis based on the Z area and value. It computes the mean and variance of series and checks the precision on 5%. If the p-value returned is less than 5% then null hypothesis is rejected otherwise the null hypothesis is accepted and smoothen of the series is next consideration.

$$z = \frac{(x' - \mu)}{(\sigma - \sqrt{n})}$$

- Rolling Statistics

  In this test we calculate the mean and variance upon the window of dataset
  In the above time series shown from T1 to T5
  If window size is 2 then the mean of every two values is calculated and plotted.

- Lagged Time series

  A lag is the fixed amount of passing time: On set of observation in a time series is plotted (lagged) against a second, later set of data. The Kth lag is the period that happened k time before the time I.

  Some time we are given a time series that is incomplete or tends to be incomplete on any point in time series which is considered is to be the corelative of previous points

Date	Values	Lag1	Lag2
D1	1	-	-
D2	2	1	-
D3	3	2	1
D4	4	3	2
D5	5	4	3
		5	4

			5

Auto regression models predict this and return the number of lags. Auto Regression is the first component of time series analysis.

- P, D, Q

  ARIMA is the integration of Auto Regression (AR) and Moving Average (MA).

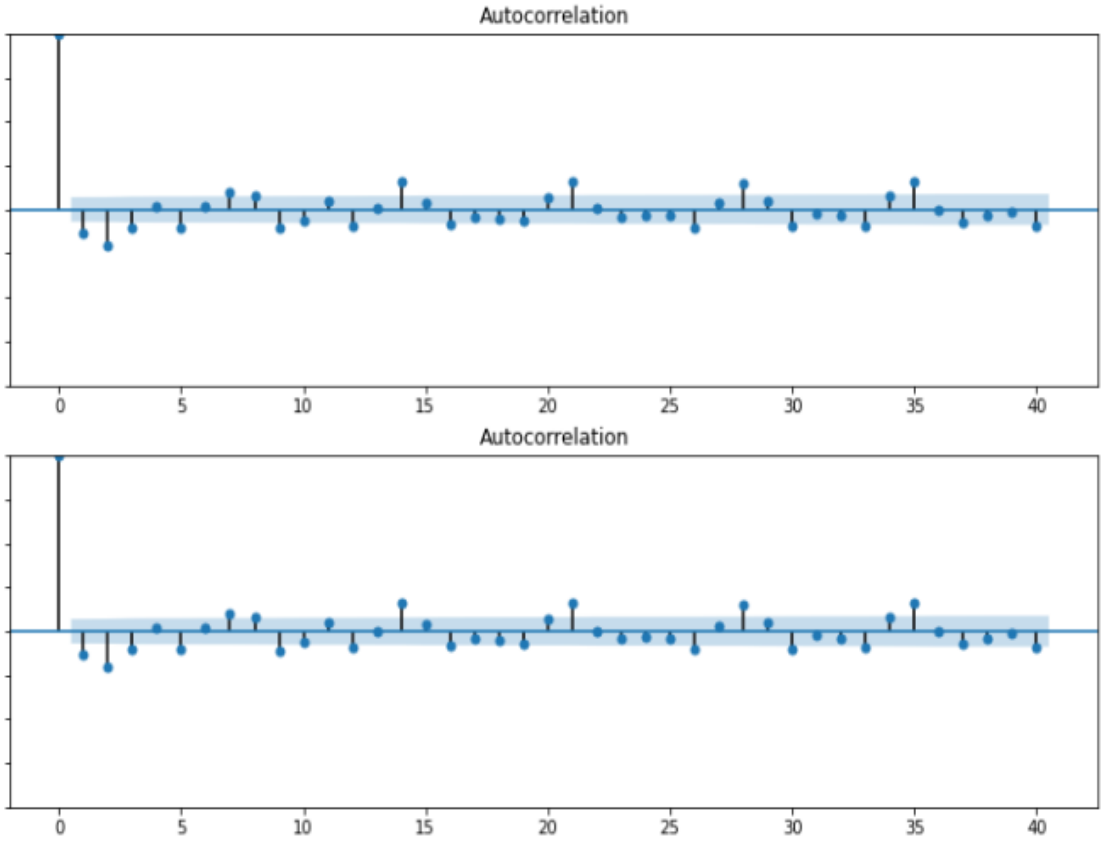
  The AR component in ARIMA is represented by p value.

- P

  Auto regression is performed by auto regressive models to predict the p values for AR component of ARIMA.

  P value is the lagged value i.e., the number of lags that correlates with the time series.

  ACF and PACF tests are performed.



The ACF function perform the regression and predicts the best fitting P value.

- Q

  Q component is the moving average of fixed numbers of items in the time series which move through the series by dropping the top items of the previous averaged group and adding the next in each successive average.

  1. Exponential Smoothing

This analyzes the time series and gives the values percentage to the next one.

$$y = \alpha \times x_t + (1 - \alpha) y_{t-1}$$

Here yt is the current value and alpha lies in between 0 and 1 and assigns the weightage to the previous one.

This smoothing basically used for simple linear time series

2. Double Exponential Smoothing

   It also keeps track of trend components along with the linearity in the time series.

$$y = \alpha \times x_t + (1 - \alpha)(y_{t-1} + b_{t-1})$$

   Last term is the trend component in time series.

$$\beta_t = \beta(y_t - y_{t-1}) + (1 - \beta) b_{t-1}$$

   This is at any given time what was the prediction at time t and previous of that.

3. Triple exponential smoothing

$$y = \alpha \frac{x_t}{c_t - L} + (1 - \alpha)(y + b_{t-1})$$

$$c_t = \gamma \frac{x_t}{y_t} + (1 - \alpha) L_t - L$$

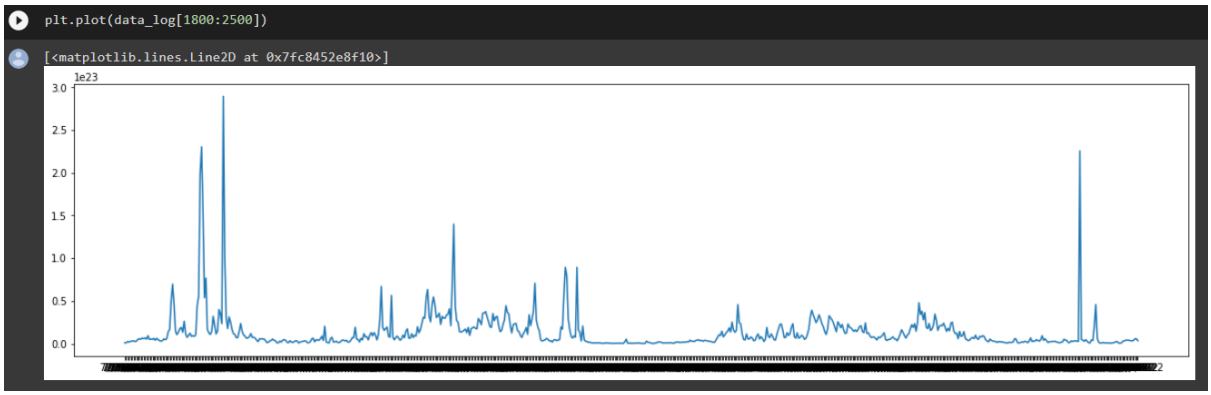
   This keeps track of all three components of time series.

Dataset Details

Dataset included gas cost and the capturing dates.

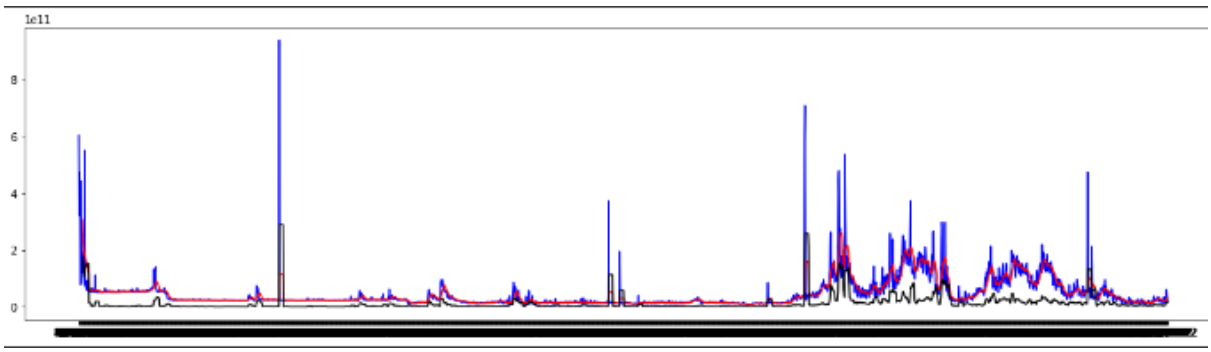
Date	cost
2015-08-07	6.050000e+11
2015-08-08	3.230000e+11
2015-08-09	4.750000e+11
2015-08-10	4.220000e+11
2015-08-11	7.783882e+10

Experiments and Results



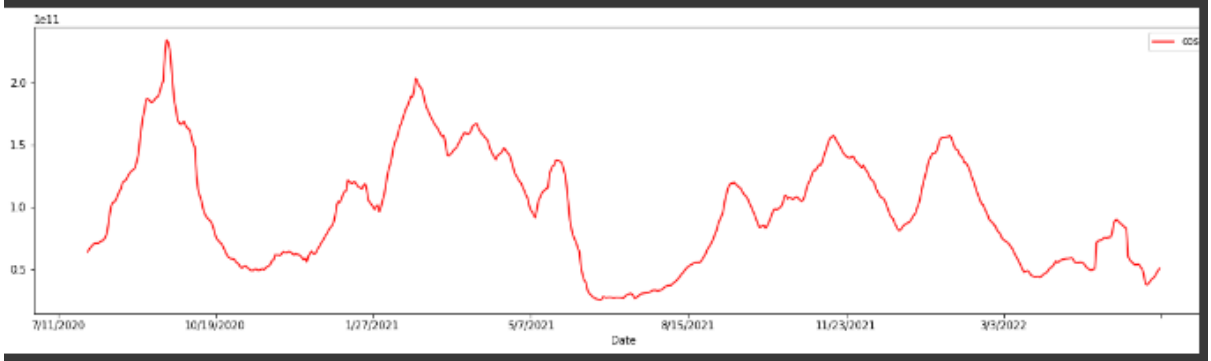
This dataset does not show the exact behavior to train for ARIMA model as it is. It has irregular behavior sometimes it shows the trend component for a period and trends fades out for the next period. At some points it shows some seasonality figure as well. And mostly it is Linear in behavior.

Square and log transformations were performed so the mean would be approximately constant over the period. It shows some results:

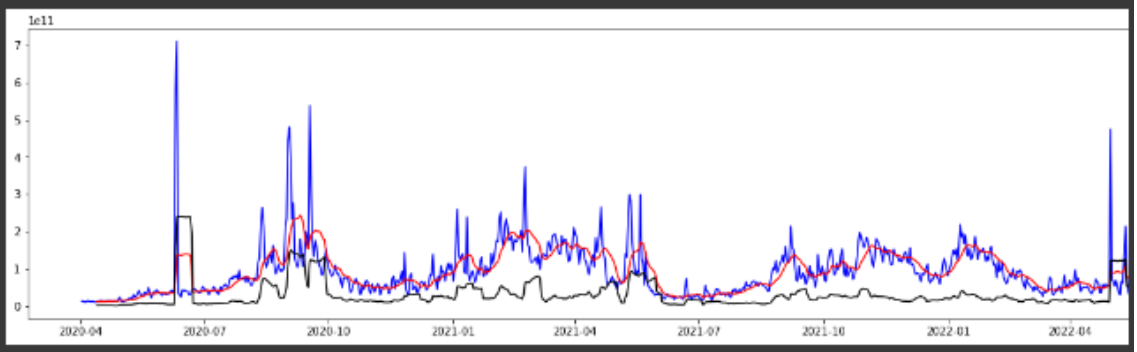


Since the above figure does not show some stationarity. Then Rolling Mean was performed.

```
rollingmean.plot(color = 'red')
plt.show()
```

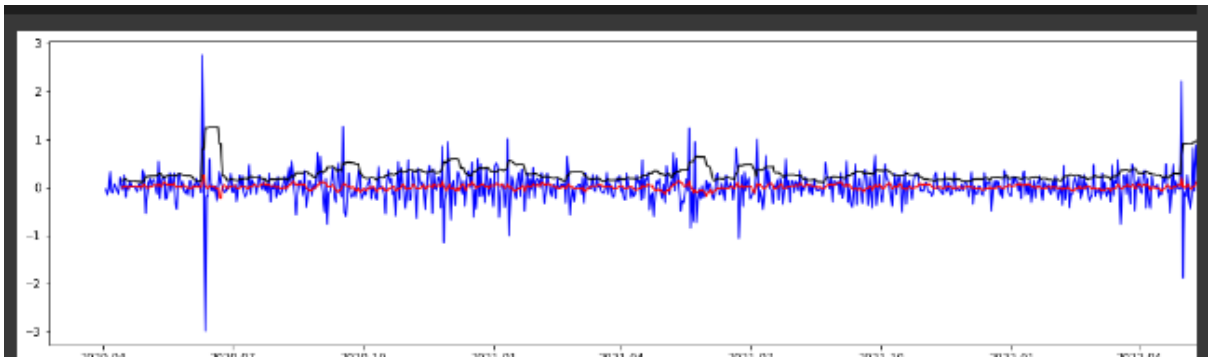



```
org_plot = plt.plot(data,color = 'blue',label = 'Data Frame')
mean = plt.plot(rolling_mean,color = 'red',label = 'Data Frame')
std_deviation = plt.plot(rolling_dev,color = 'black',label = 'Data Frame')
```

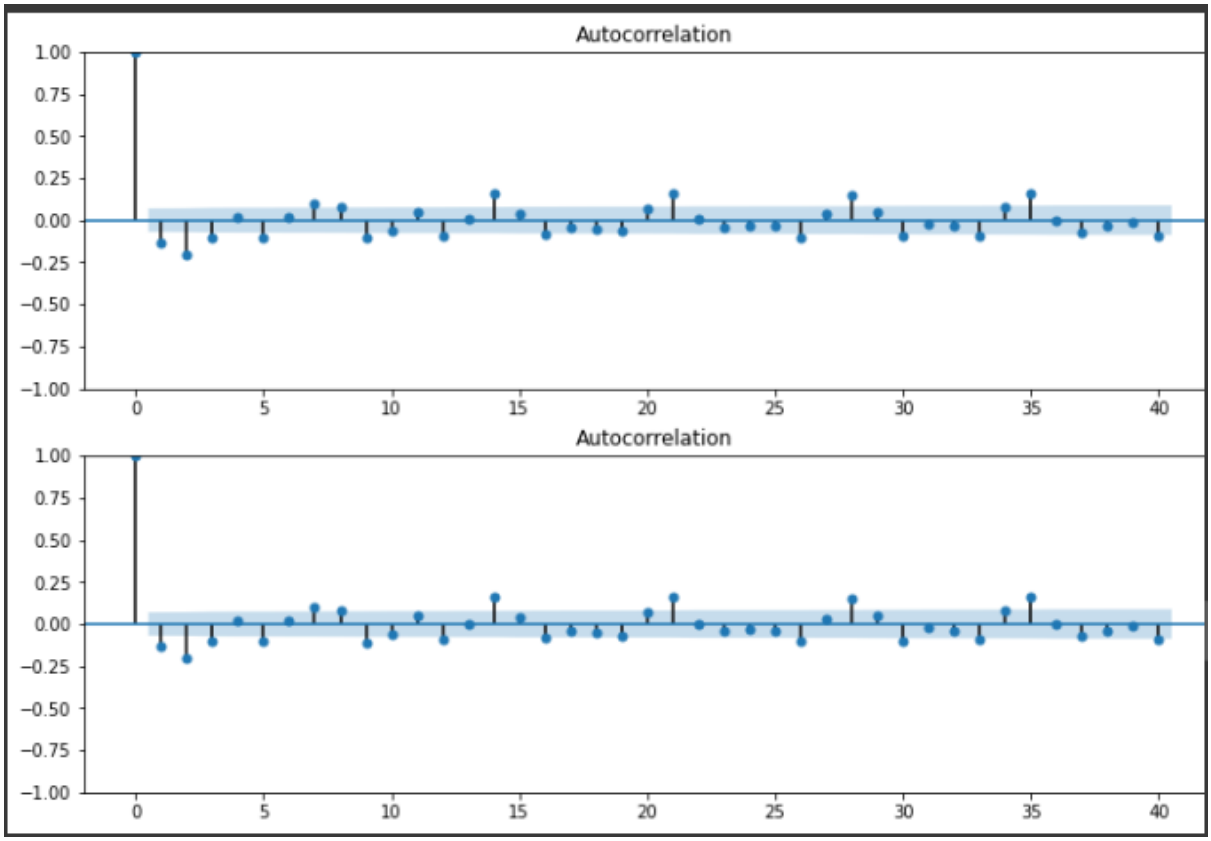


As we can see relative consistency in mean and variance.

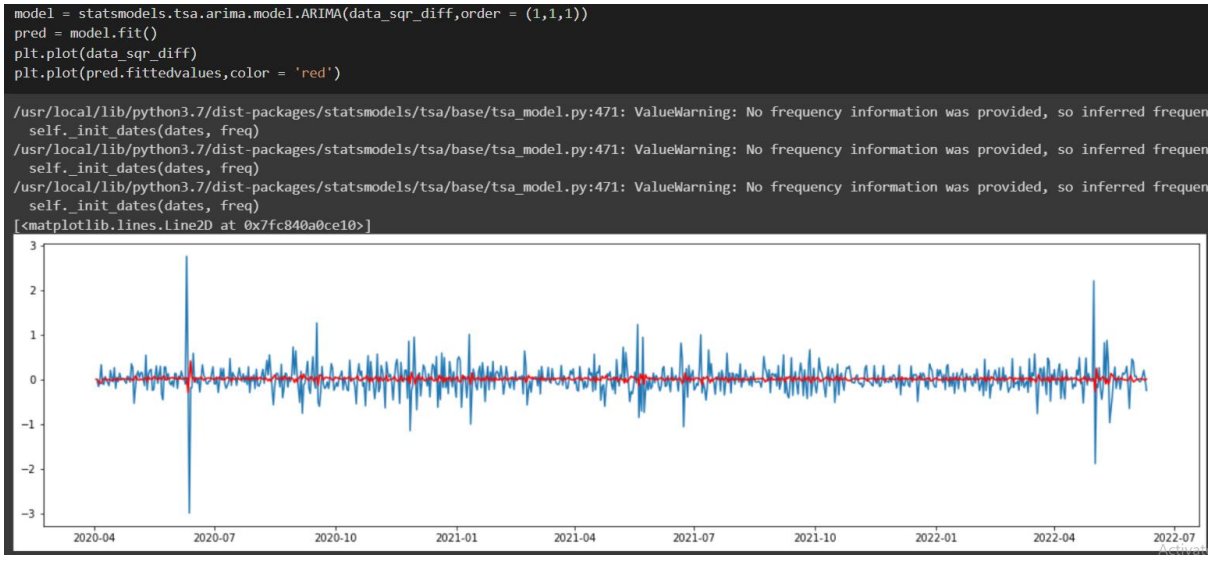
But for square transformation it shows some opposite results:



Then ACF and PACF is calculated:



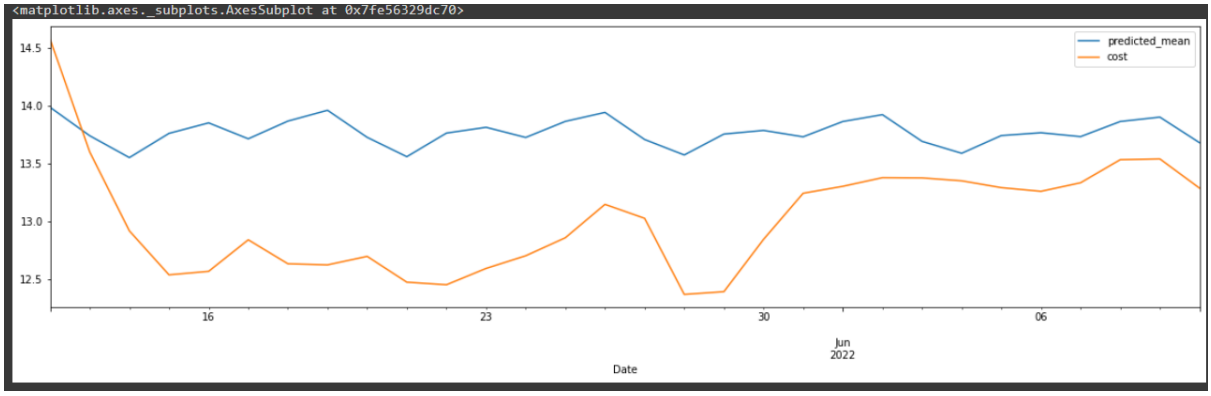
Then model is fitted which shows well fitted behavior:



Order (1,1,1) are used on the basis.

Since our model best fits upon the log transformation and shows acceptable result.

And accuracy of 89% RMSE approximately.



Now the future forecast of 30 days upon this model.