Gold Price Prediction

Project Report

Project By: -

Ashfaque Jamal

Beauty Atara

Content

1. Abstract ---------- 1

2. Introduction ---------- 2

3. Design Details ---------- 3 - 28

   3.1 About Data

   3.2 Exploratory Data Analysis (EDA)

   3.3 Data cleaning & Pre-processing

   3.4 Model Creation & hyperparameter tuning

   3.5 Model Evaluation

   3.6 Finalizing Model

   3.7 Deployment

4. Conclusion ---------- 29

1. Abstract

Prediction of future rates is one of the challenging applications of modern time series forecasting and very important for the success of many businesses and financial institutions. This project focuses on time series analysis using different Forecasting Algorithms to forecast the gold prices.

2. Introduction

Gold has a special place among metals. It is the oldest metal exploited by man, it plays an important role in world economics, it is highly prized. It was the ultimate goal of alchemists, and it is stored the vaults of banks. Gold has been used in gliding, to make funeral masks, and for many other uses.

Gold are extensively traded commodities, and there are several macroeconomic factors which affect the prices, thus causing the prices of these commodities to be extremely volatile. Certain factors are also interdependent, which makes it extremely difficult to estimate the extent to which an individual factor could affect the price of a commodity.

Moreover, the relationship between the factors and the prices could also vary over time. This makes the prediction of gold prices a very complex and challenging problem.

Historically, gold had been used as a form of currency in various parts of the world including USA. In present times, precious metals like gold are held with central banks of all countries to guarantee re-payment of foreign debts, and also to control inflation which results in reflecting the financial strength of the country. Also, it is one of the most traded commodities considering its stability and potential as an investment tool.

Forecasting rise and fall in the daily gold rates, can help investors to decide when to buy (or sell) the commodity.

There are several forecasting techniques available for making future predictions. These techniques are developed mainly based on different assumptions, mathematical foundations and specific model parameters. However, for better result, it is important to find the appropriate technique for a given forecasting task. In our project, we have tried different forecasting methods to predict the gold prices for 30 days from, 22nd Dec 2021 to 20th Jan 2022 using the data of around 6 years, from 1st Jan 2016 to 21st Dec 2021.

3. Design Details

3.1 About Data:

Data for this study is collected from **January 1th 2016** to **December 21st 2021** from various sources. The data has 2182 rows in total and **2** columns in total. The columns are Date and Price. The datatype for Date column is **object** and that of Price column is **float64.** The data collected here is on daily basis.

3.2 Exploratory Data Analysis (EDA):

3.2.1 Data Transformation:

The datatype of the Date column is converted to date-time index since it is a time series problem and the data should be in that format only. Then the date column is converted into the index of the data.

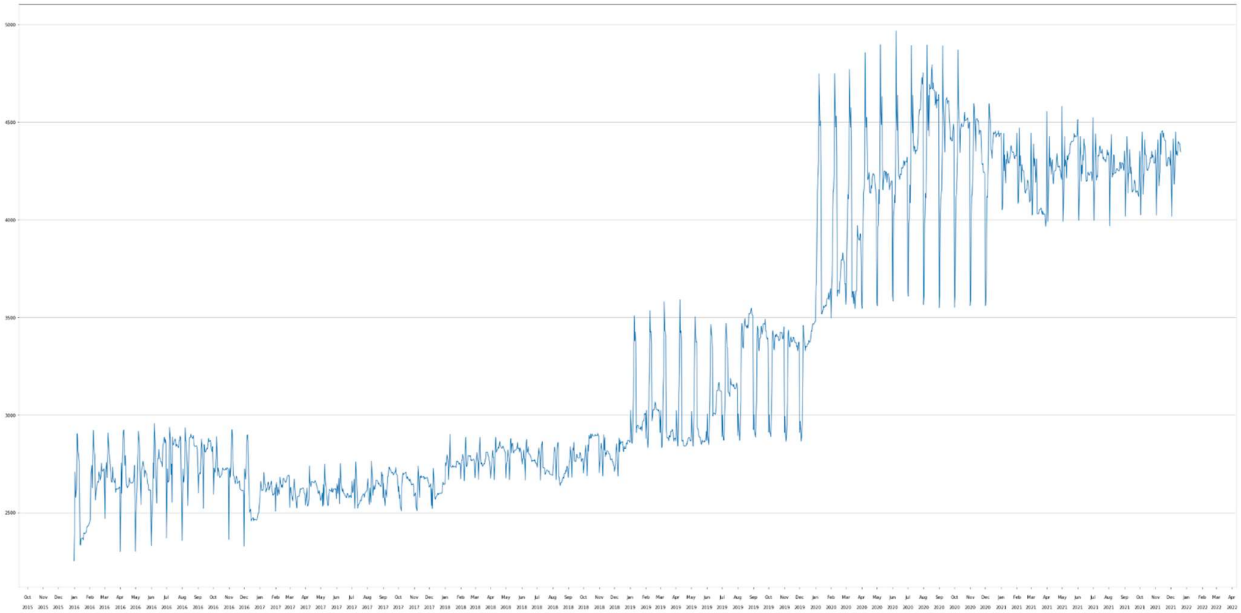
3.2.2 Description of Data:

In this data, the average gold price is Rs. 3284.45 per gram, the maximum gold price is Rs. 4966.30 per gram, and the minimum gold price is Rs. 2252.60 per gram.

3.2.3 Plots:

a) Line Plot:

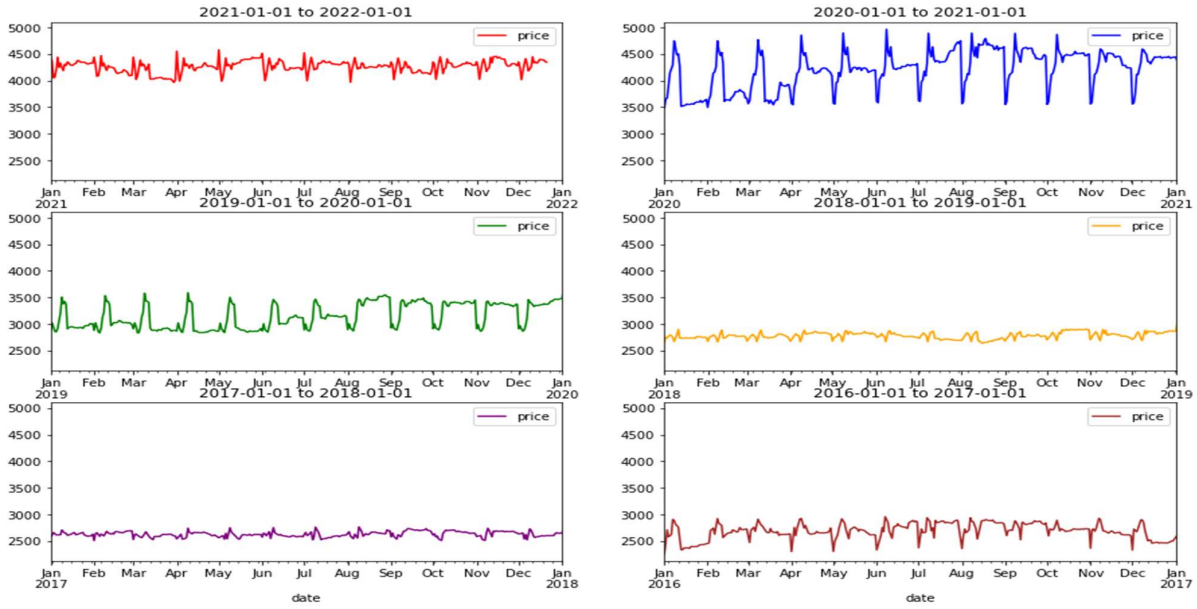
The following is the image of a line plot. From the line plot we can conclude that the trend of gold prices is increasing over the years.



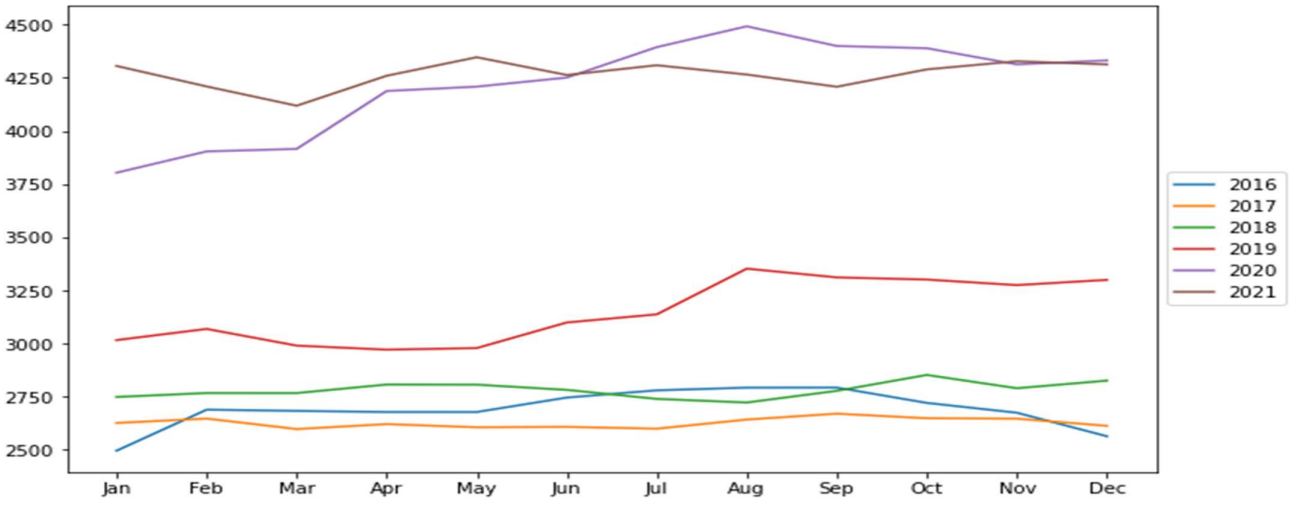
Also, the seasonality is not visible clearly from the graph, since the fluctuations in the prices are not constant over the years. We can say that our data is following a cyclic behaviour.

b) Yearly Line Plot:

Yearly line plots were generated in order to get a clear understanding of the seasonality of the data, and also the price ranges for each year. As we can see, the highest fluctuation in gold prices is observed in the year 2020 and the lowest fluctuation is observed in the year 2017.



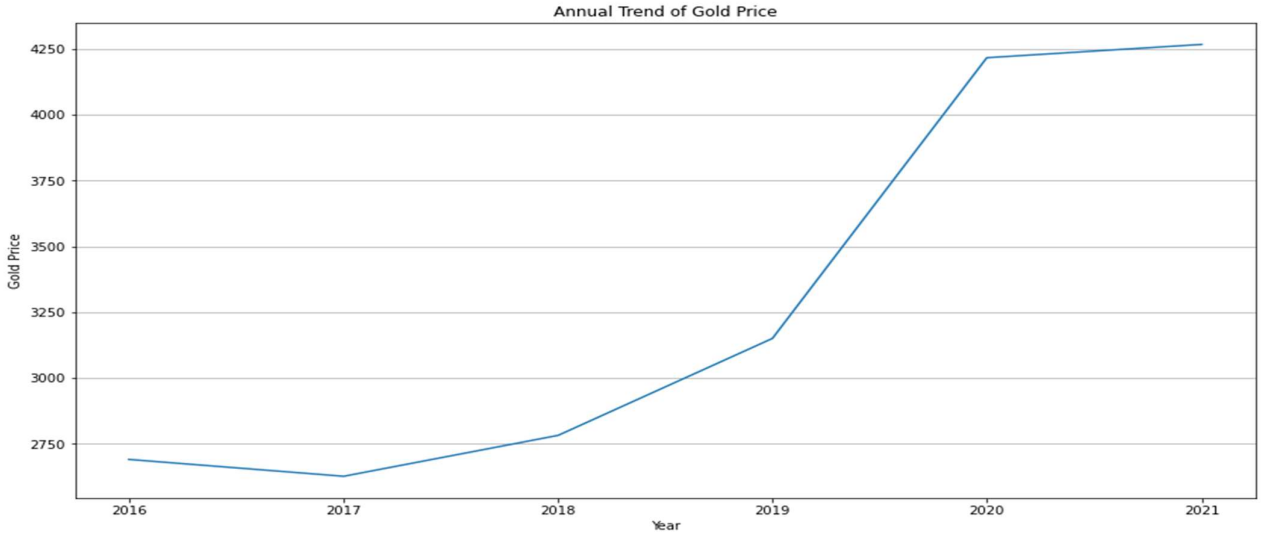
c) Monthly Line Plot:



Below shown graph is a monthly line plot. Each line represents a different year. From the graph we can observe that the gold prices went lowest in the year 2016 and the highest in the year 2020.

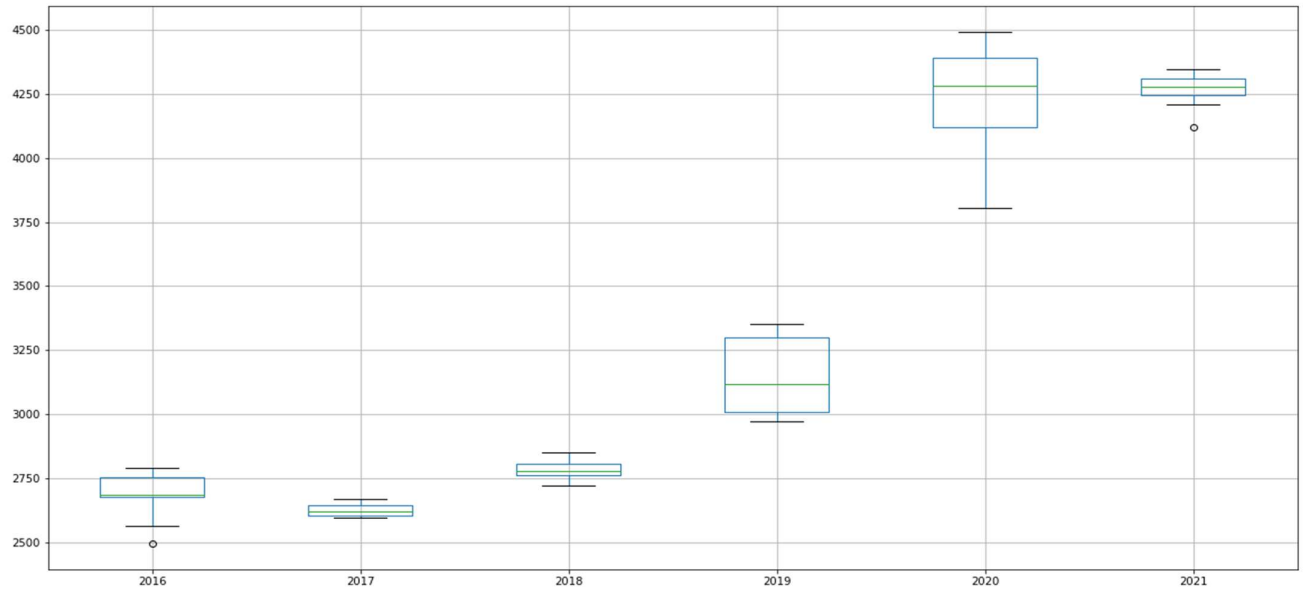
d) Annual Trend:

The following graphs is also a line plot of the time series, and from this graph we can get a clear picture of the trend in the gold prices. From 2016 to 2017 the gold prices decreased, and after 2017 it gradually increased. In the year 2019 there is a sharp increase in the prices, and this may be due to the pandemic situation. After the year 2019 the prices are stable but increasing.



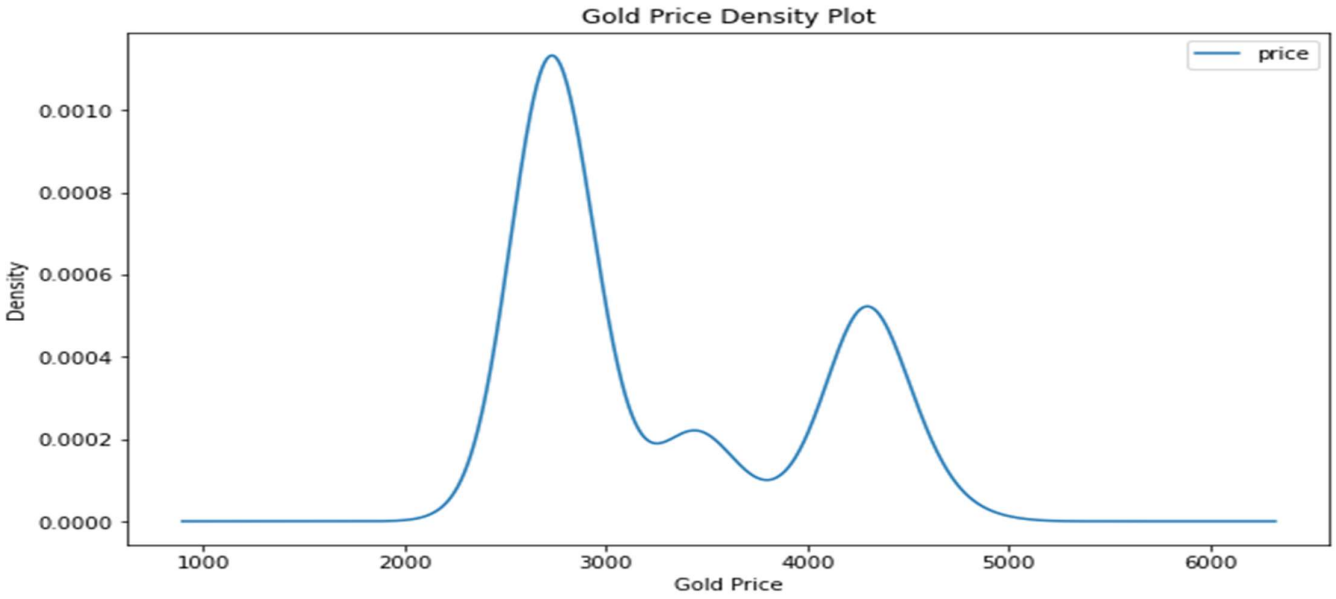
e) Box Plot:

The following graph shows the box plot for each year. Box plots help us in detecting any outliers present in the data. In our data, we have a few outliers in the year 2016 and 2021.



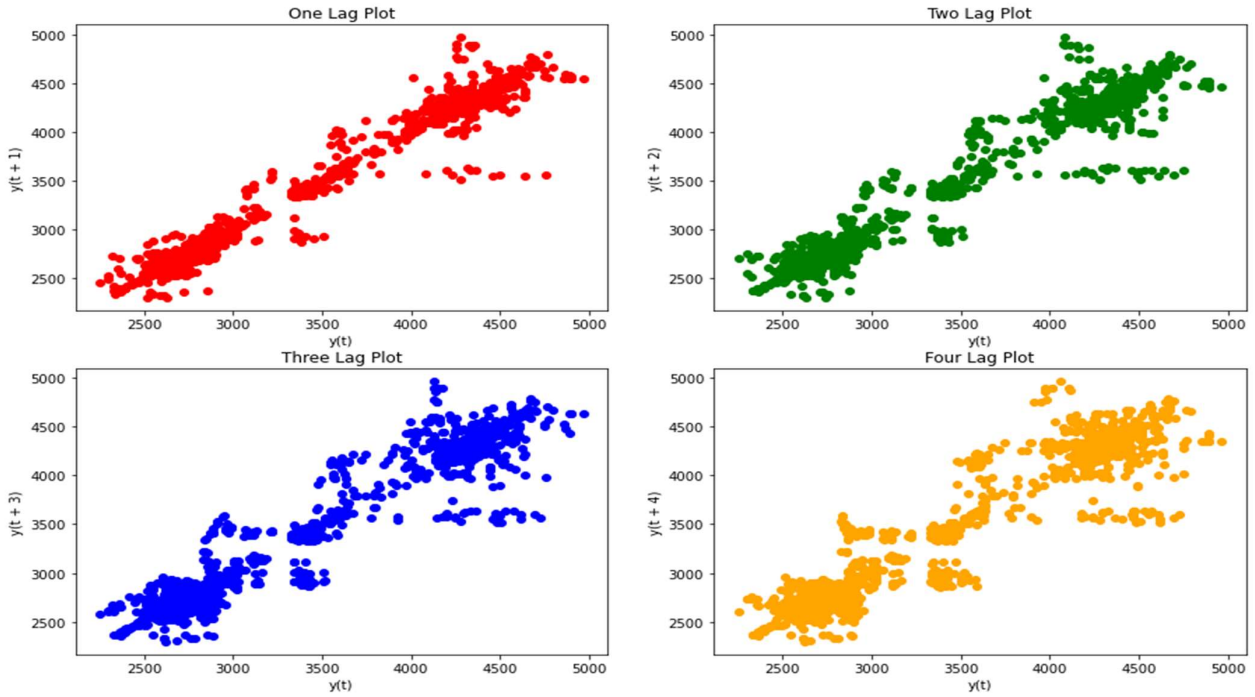
f) Density Plot:

Density plot help us in identifying whether the data is normally distributed or not. From the graph, we can say that our data is not normally distributed.



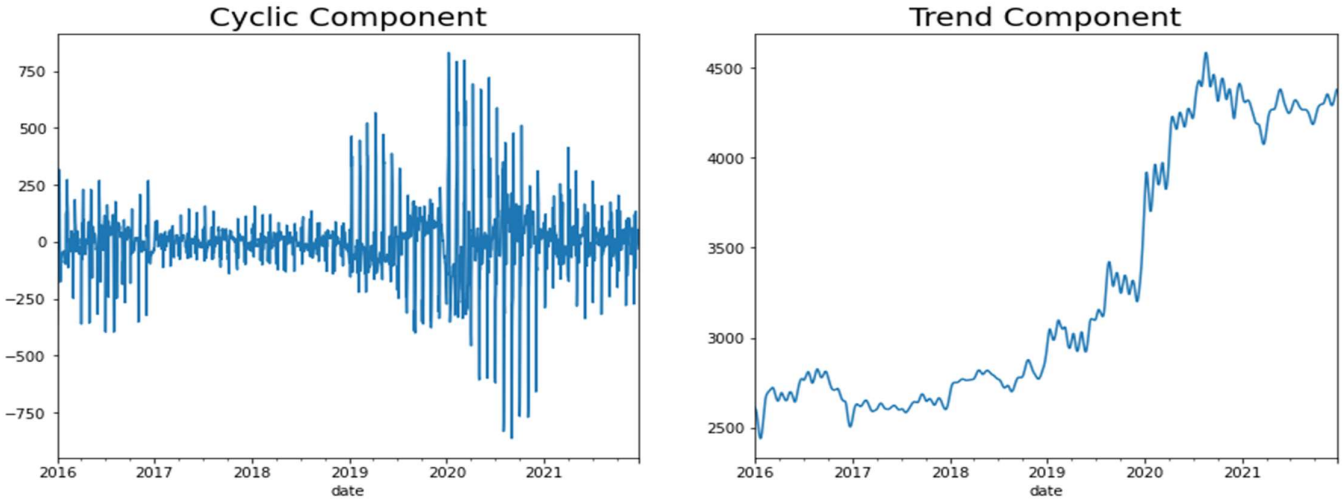
g) Lag Plot:

The following images shows the Lag plots of past 4 lags. The plots are not random and shows a positive corelation. This means that the present gold prices depend on the past gold prices and the relation is positive.



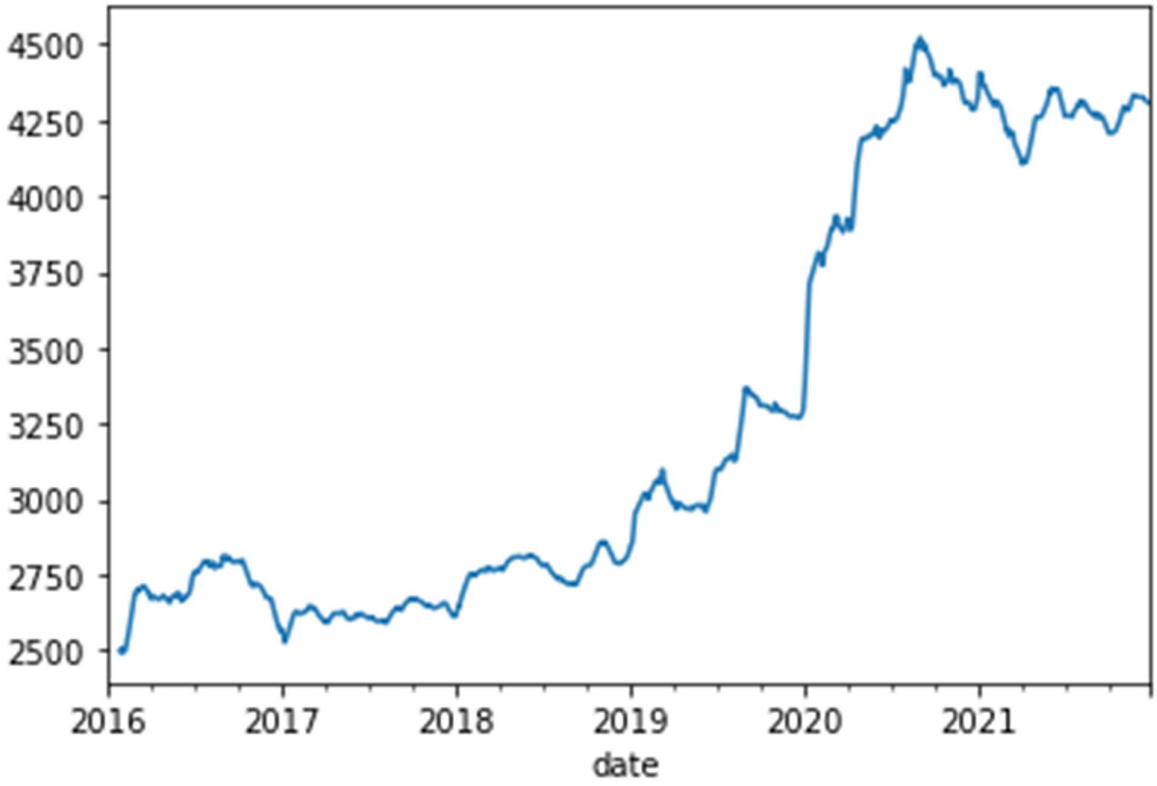
h) Cyclic and Trend Component:

Using hp_filter library, we separated the Trend and Cyclic component of the time series.



3.2.4 Moving Average:

The following plot shows the Moving Average of the time series with a window size 30. By doing moving average, we get a smooth plot from which we can get a clear understanding of the trend of the time series.



3.2.5 Decomposition of Time Series:

a) Components of Time Series:

- Trend: Change in direction over a period of time
- Seasonality: Seasonality is about periodic behaviour, spikes or drops caused by different factors, for example: Naturally occurring events, like weather fluctuations, Business or administrative procedures, like start or end of a fiscal year, Social and cultural behaviour, like holidays or religious observances, Calendar events, like the number of Mondays per month or holidays shifting year to year
- Residual: irregular fluctuations that we cannot predict using trend or seasonality.
- Cyclic behaviour: Another important thing to consider is the cyclic behaviour. It happens when the rise and fall pattern in the series does not happen in fixed calendar-based intervals. We should not

confuse 'cyclic' effect with 'seasonal' effect. If the patterns are not of fixed calendar-based frequencies, then it is cyclic. Because, unlike the seasonality, cyclic effects are typically influenced by the business and other socio-economic factors

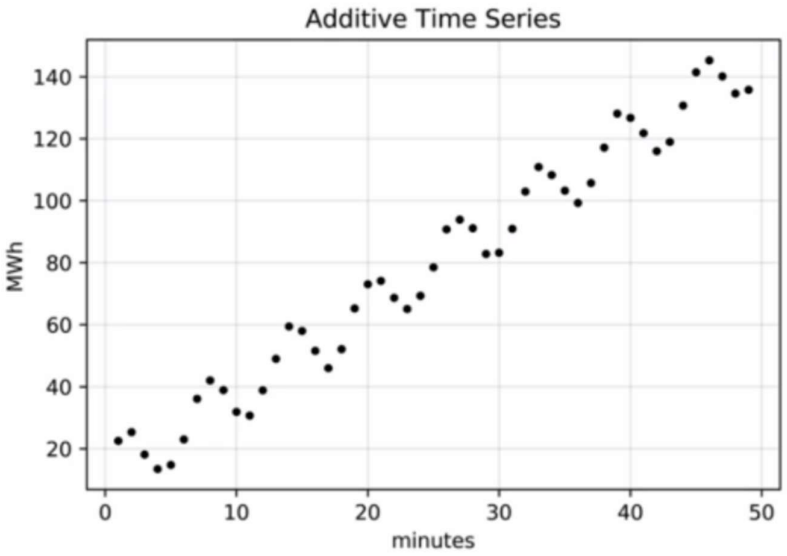
b) Decomposition Models:

Additive model:

- The additive model assumes the observed time series is the sum of components:

  Observation = trend + seasonality

- Additive models are used when the magnitude of seasonal and residual values is independent of the trend.



Multiplicative Model:

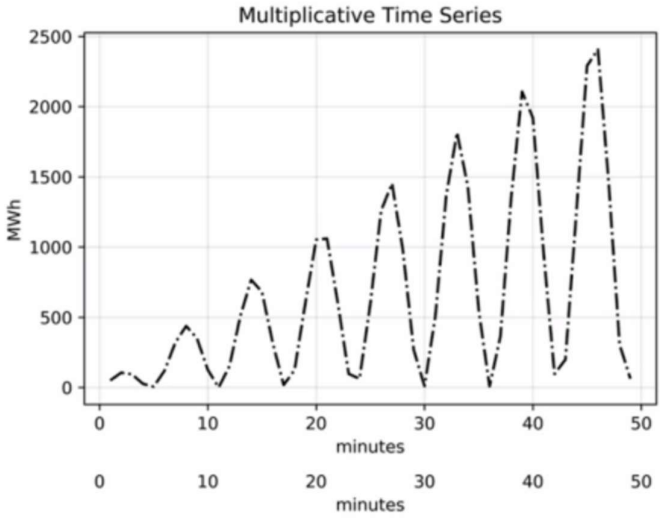
- The multiplicative model assumes the observed time series is a product of its components:

Observation = trend * seasonality * residual

- We can transform the multiplicative model to an additive model by applying a log transformation:

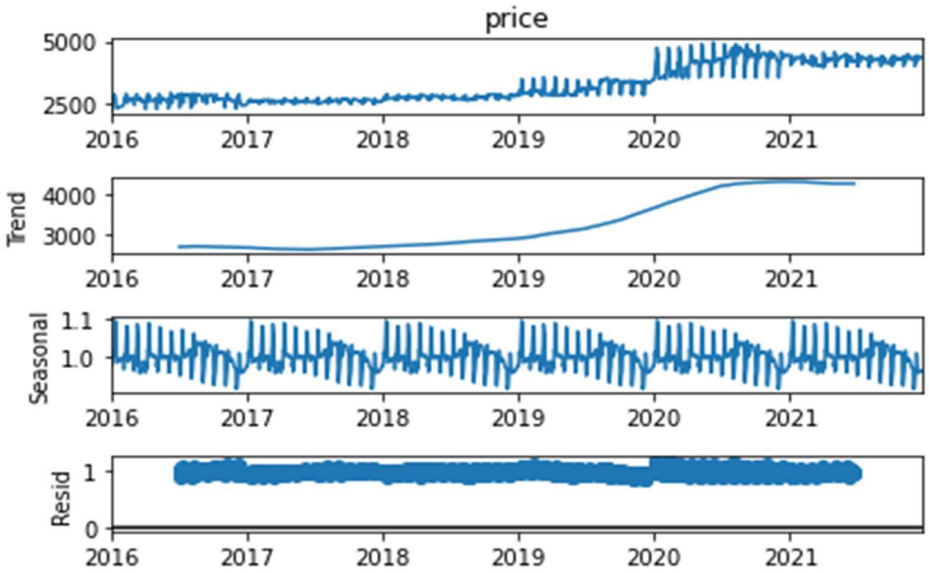
  Log (time * seasonality * residual) = log (Time) + log (seasonality) + log (residual)

- These are used if the magnitudes of seasonal and residual values fluctuate with the trend.



c)Decomposition Time Series:

The following plot shows the decomposition of time series into its components. We have done multiplicative decomposition, because in our data the magnitudes of seasonal and residual values fluctuate with the trend.



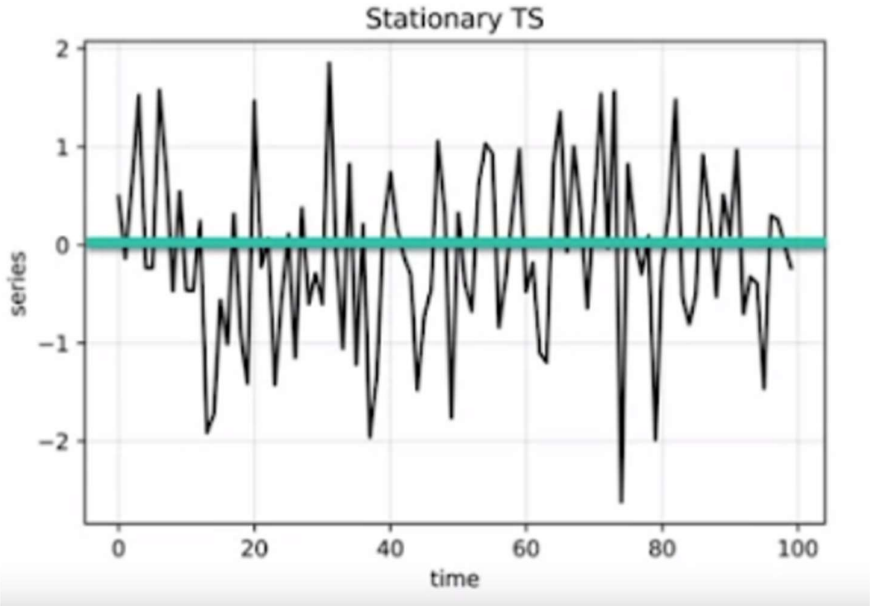
3.2.6 Stationarity Of Time Series:

a) What is Stationarity Time Series?

A stationary time series is a series whose properties remain constant over time. These properties are variance, mean, and covariance. Stationary time series do not have trends and repetitive cycles/seasonality. Most time-series is non-stationary, especially stock prices and other financial data. To make a time series stationary, we apply the differencing technique.

Constant Mean:

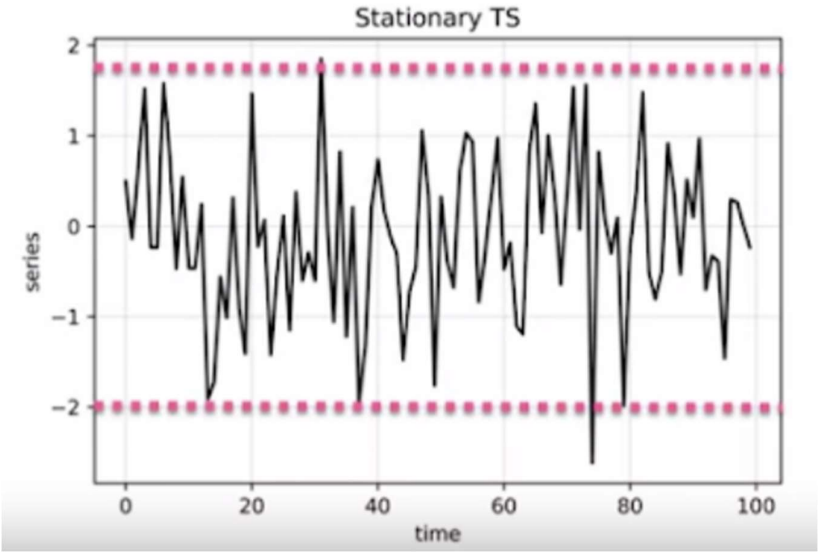
- A stationary time series will have a constant mean throughout the entire series.



- As an example, if we were to draw the mean of the series, this holds as the mean throughout all of the time.
- A good example where the mean wouldn't be constant is if we had some type of trend. With an upward or downward trend, for example, the mean at the end of our series would be noticeably higher or lower than the mean at the beginning of the series.

Constant Variance:

- A stationary time series will have a constant variance throughout the entire series.



b) Methods to check stationarity:

- Visualizing the line plot of the time series
- Performing statistical tests on the time series (Augmented Dickey Fuller Test and KPSS Test)

Augmented Dickey Fuller Test*:*

- The ADF test uses hypothesis testing to check for stationarity.
- It has a null hypothesis and an alternative hypothesis. The null hypothesis of this test is that the times series is non-stationary. The alternative hypothesis is that the time series is stationary.
- The ADF test has an important parameter known as the p-value that determines whether a time series is stationary. The time series is stationary when the p-value is less than 0.05.

Kwiatkowski–Phillips–Schmidt–Shin (KPSS) test:

- KPSS test also uses hypothesis testing to check the stationarity of time series.
- The null hypothesis is that the time series is stationary. The alternative hypothesis is that the time series is non-stationary.

- Here, when p-value is less than 0.05, we reject null hypothesis, hence the time series is non-stationary
- If p-value is greater than 0.05, we accept the null hypothesis, hence the time series is stationary.

c) Test Results:

We performed both, Augmented Dickey Fuller Test and Kwiatkowski Phillips Schmidt Shin (KPSS) test on our data to check whether it is stationary or not.

ADF Test: -

Null Hypothesis - Data is not stationary

Alternate Hypothesis - Data is stationary

KPSS Test: -

Null Hypothesis - Data is stationary

Alternate Hypothesis - Data is not stationary

The following are the test results. From the results, we found out that our data is non-stationary.

Out[30]:

	adf	kpss
Test Statistic	-0.309904	7.218406
p-value	0.92409	0.01
Numbers of lags	26	26
decision	Non-Stationary	Non-Stationary
Critical Value (1%)	-3.433388	0.739
Critical Value (5%)	-2.862882	0.463
Critical Value (10%)	-2.567484	0.347
Critical Value (2.5%)	NaN	0.574

3.3 Data Cleaning and Pre-processing:

3.3.1 Handling null values:

- The data collected, consists of zero null values, hence we do not have to deal with them.

3.3.2 Handling outliers:

- The data consists of very few outliers in the year 2016 and 2021, which are not affecting the results that much, so there is no need to remove them.

3.3.3 Making the time series Stationary:

The following techniques were used to make the time series stationary.

- One time differencing
- Seasonal differencing
- Log transformation
- Square root transformation
- Subtracting the Moving Average
- Seasonal Decompose
- hp_filter

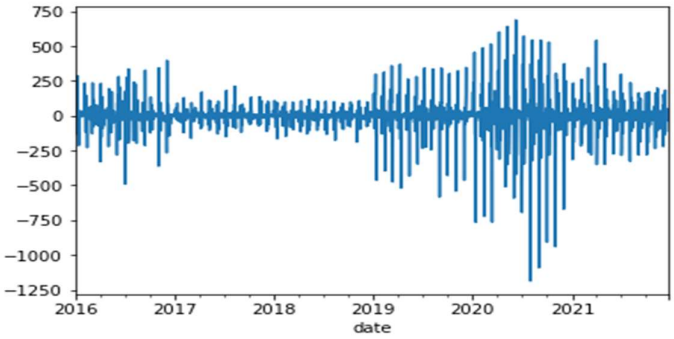
1. **One time differencing:**

   Differencing is a process of making the time series stationary. It makes the variance, the covariance, and the mean of the time series constant. It also reduces the repetitive cycles and the seasonality components. The differencing technique finds the difference between the current time series value and the previous value. We may get the difference between the time series values once but still not make the time series stationary. In this case, we need to find the difference multiple times until the time series becomes stationary.

   The result for one time differencing is as follows, and after one time difference the data has become stationary.

```
1  # Cyclic component
2  check_stationarity(df_cyclic)
```

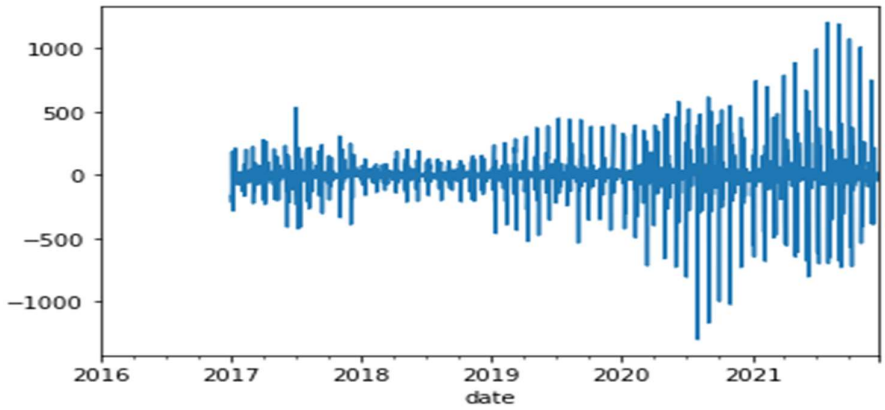
```
('Accept Null Hypothesis, Data is Stationary',
 'Rejest Null Hypothesis, Data is Stationary')
```




```
In [39]:  1 # adfuller test on One difference
          2 check_stationarity(df['price one difference'].dropna())

Out[39]: ('Accept Null Hypothesis, Data is Stationary',
          'Rejest Null Hypothesis, Data is Stationary')
```

2. Seasonal Differencing

The difference using seasonal period, in this case which was 365 days, and the results were stationary.



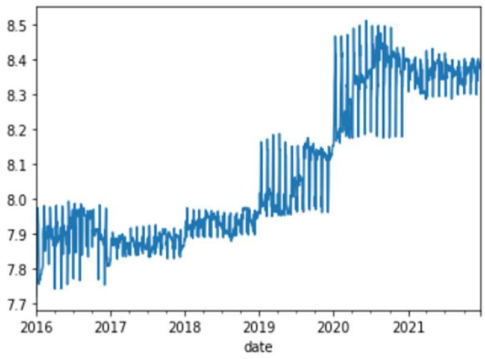

```
In [87]:  1 check_stationarity(df['two_time_diff'].dropna())

Out[87]: ('Accept Null Hypothesis, Data is Stationary',
          'Rejest Null Hypothesis, Data is Stationary')
```

3.Log transformation

In this method, the entire data is converted into its logarithmic form and then the stationarity tests are performed. The results after log transformation were ‘non-stationary.’

```
In [47]:  1 df['log'].plot()
```

```
Out[47]: <AxesSubplot:xlabel='date'>
```




```
In [48]:  1 # Checking for Stationarity for Log transformed data
          2 check_stationarity(df['log'])
```

```
Out[48]: ('Reject Null Hypothesis, Data is Non Stationary',
          'Accept Null Hypothesis, Data is Non Stationary')
```

4.Square root transformation

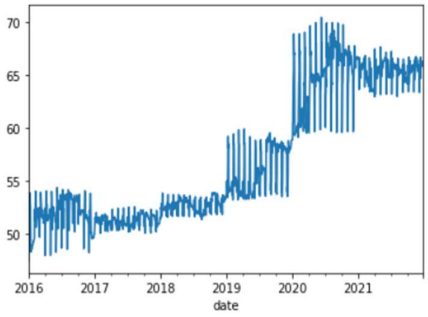
In this method, square root of the entire data is taken and then the stationarity tests are performed. The results after square root transformation were 'non-stationary'.

Square root transformation

```
[49]:  1 df['sqrt']=np.sqrt(df['price'])
```

```
[50]:  1 df['sqrt'].plot()
```

```
Out[50]: <AxesSubplot:xlabel='date'>
```

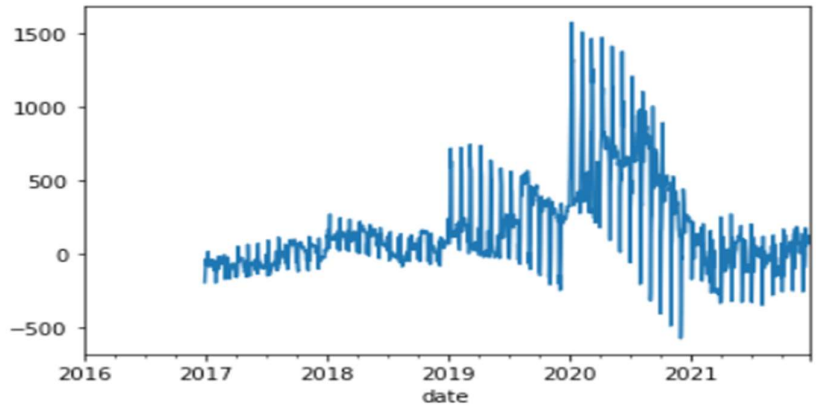



```
[51]:  1 # Checking Stationarity for Square root transformed data
       2 check_stationarity(df['sqrt'])
```

```
Out[51]: ('Reject Null Hypothesis, Data is Non Stationary',
          'Accept Null Hypothesis, Data is Non Stationary')
```

5. Subtracting the moving average

In this method, the moving average of a suitable window size is subtracted from the original time series, and new data is generated. The tests are performed on this new data. The results after subtracting moving average were 'non-stationary'.




```
1  check_stationarity(df['subtract_MA'].dropna())
```

```
('Reject Null Hypothesis, Data is Non Stationary',
 'Accept Null Hypothesis, Data is Non Stationary')
```

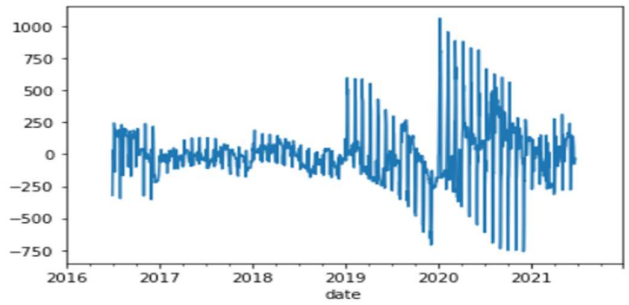
6. Seasonal decompose

In this method, the time series is first decomposed into its components (trend, seasonality and residuals) and then the trend and seasonality components are removed from the original data. Tests are performed on this new data. The results obtained were 'stationary'.

```
1  temp_2=seasonal_decompose_mul.observed - seasonal_decompose_mul.trend
```

```
1  temp_2.plot()
```

```
<AxesSubplot:xlabel='date'>
```




```
1  check_stationarity(temp_2.dropna())
```

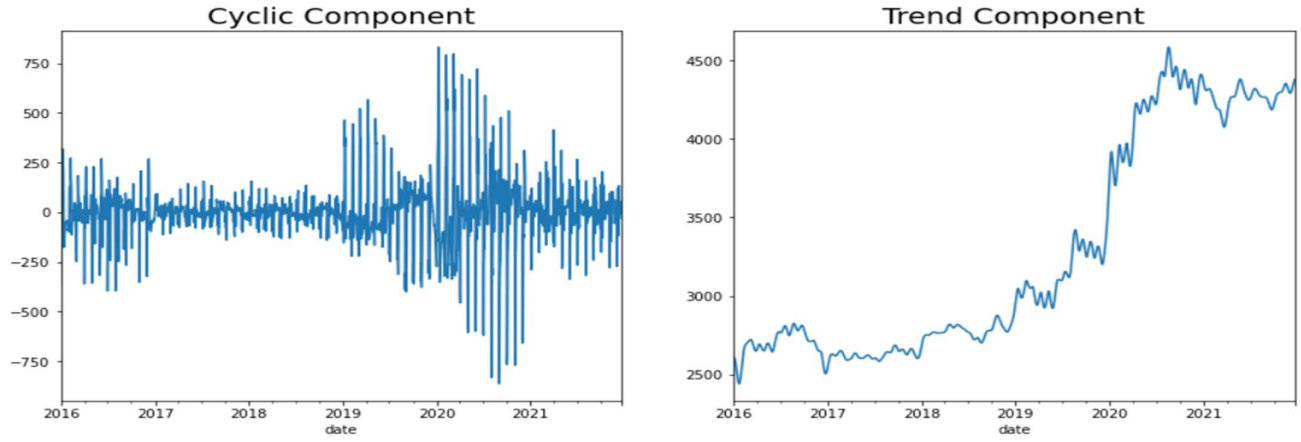
```
('Accept Null Hypothesis, Data is Stationary',
 'Rejest Null Hypothesis, Data is Stationary')
```

7. hp_filter

In this method, the Trend and the Cyclic component of the original dataset is first separated, and then tests are performed on the cyclic component of the time series. The results after performing this method were ‘Stationary’.

```
from statsmodels.tsa.filters.hp_filter import hpfilter
df_cyclic, df_trend = hpfilter(df)
```

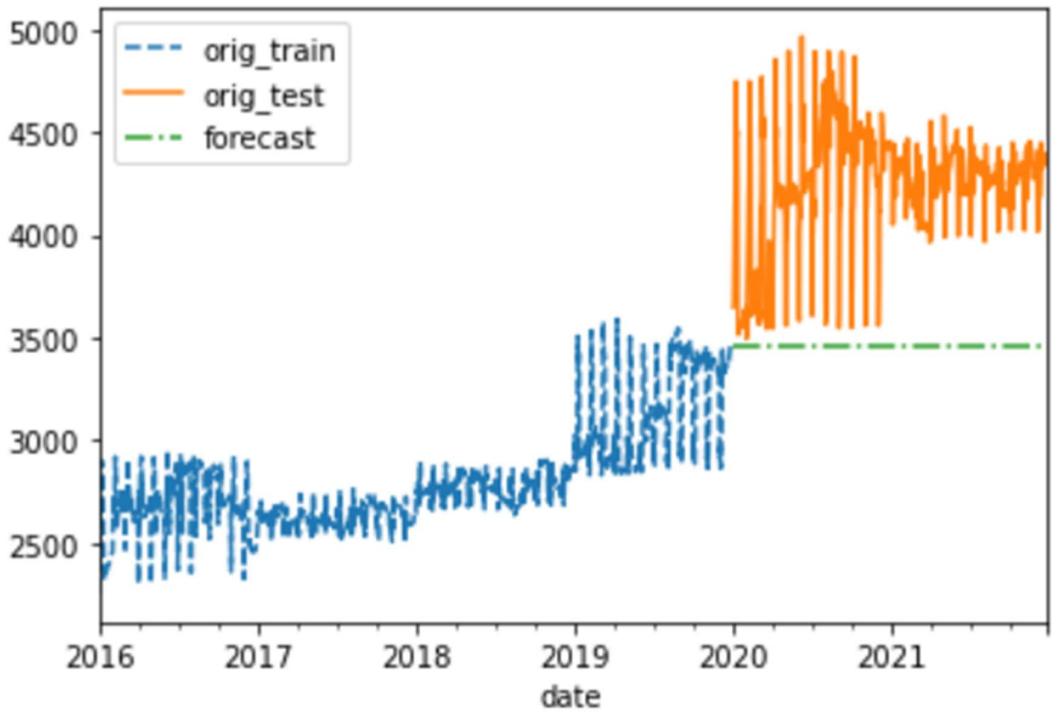
```
fig, ax = plt.subplots(1,2, figsize=(15,6))
df_cyclic.plot(ax=ax[0], title='Cyclic Component')
df_trend.plot(ax=ax[1], title='Trend Component')
ax[0].title.set_size(20); ax[1].title.set_size(20)
```



3.4 Model Creation and hyperparameter tuning:

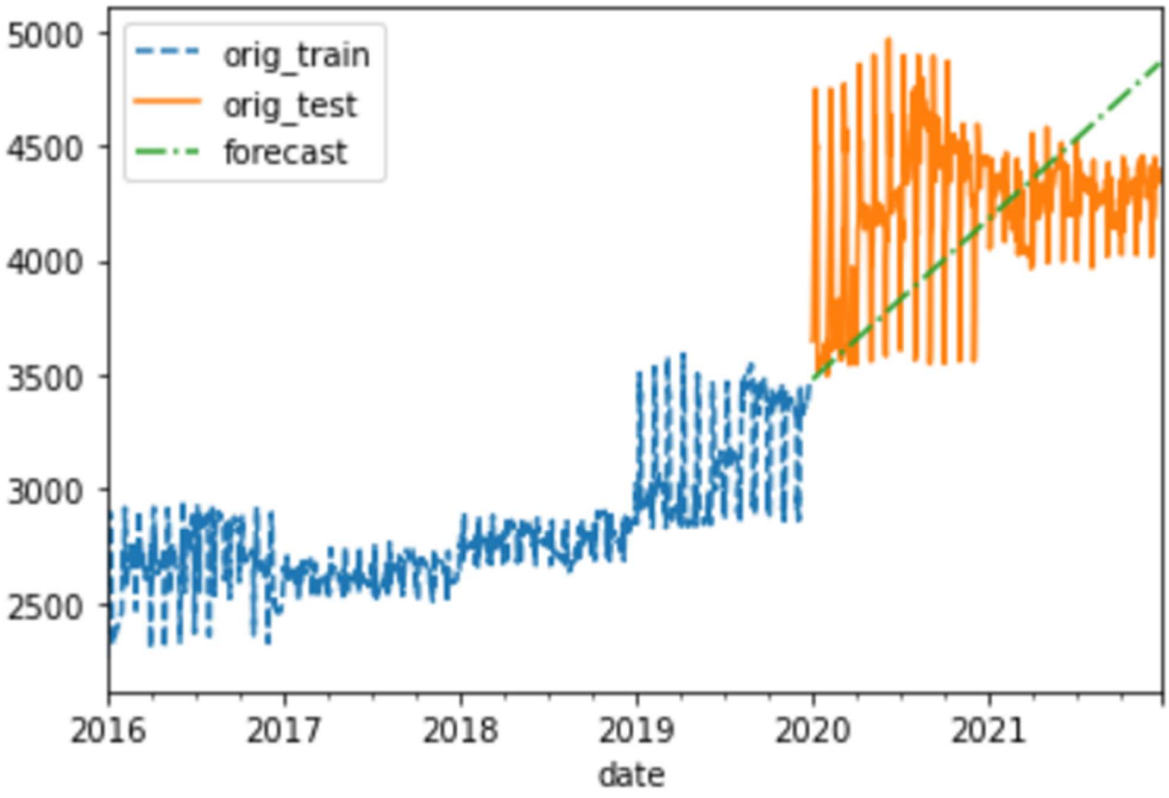
3.4.1 Simple Exponential Smoothing:

- Single Exponential Smoothing, also called Simple Exponential Smoothing, is a time series forecasting method for univariate data without a trend or seasonality.
- It requires a single parameter, called alpha (a), also called the smoothing factor or smoothing coefficient.
- This parameter controls the rate at which the influence of observations at prior time steps decays exponentially.
- Alpha is often set to a value between 0 and 1.
- Large values mean that the model pays attention mainly to the most recent past observations, whereas smaller values mean more of the history is taken into account when making a prediction.
- We did the hyperparameter tuning for different values of 'alpha' and the best results were obtained for alpha= 0.2653.



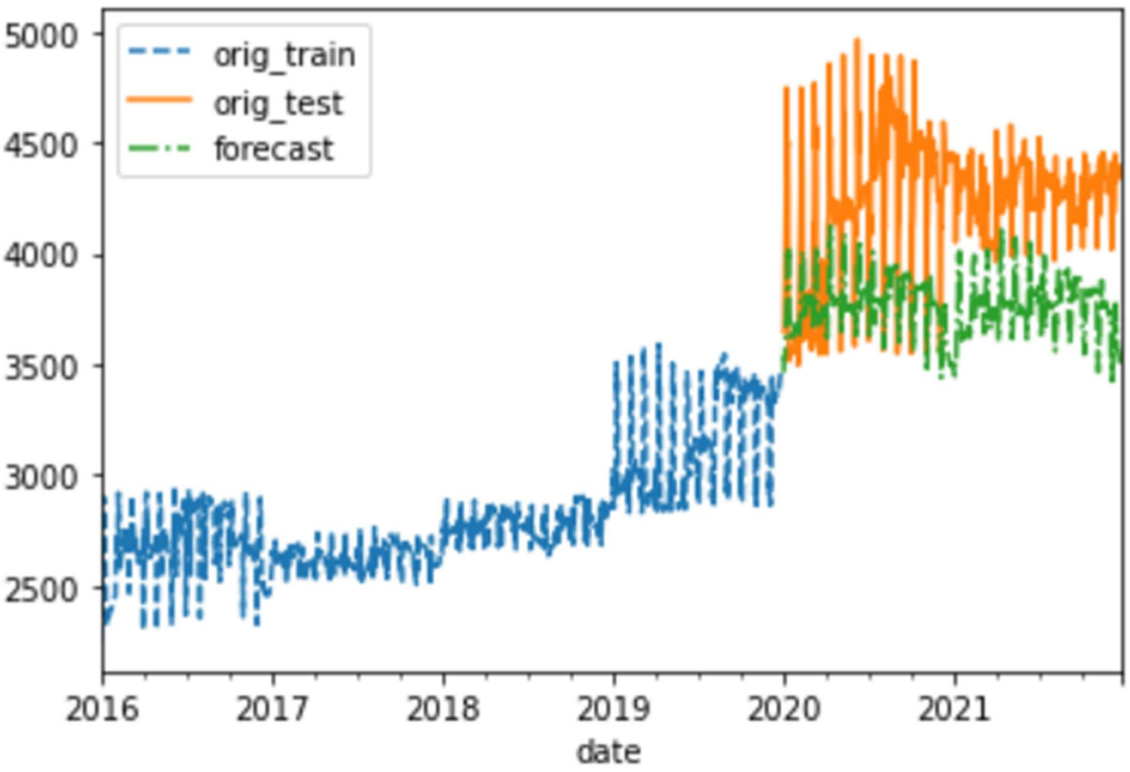
3.4.2 Double Exponential Smoothing:

- Double Exponential Smoothing is an extension to Exponential Smoothing that explicitly adds support for trends in the univariate time series.
- In addition to the alpha parameter for controlling the smoothing factor for the level, a smoothing factor is added to control the decay of the influence of the change in a trend, called beta (b).
- The method supports trends that change in different ways: an additive and a multiplicative, depending on whether the trend is linear or exponential respectively.
- After doing the hyperparameter tuning, the best results were obtained for alpha=0.91 and beta=0.91.



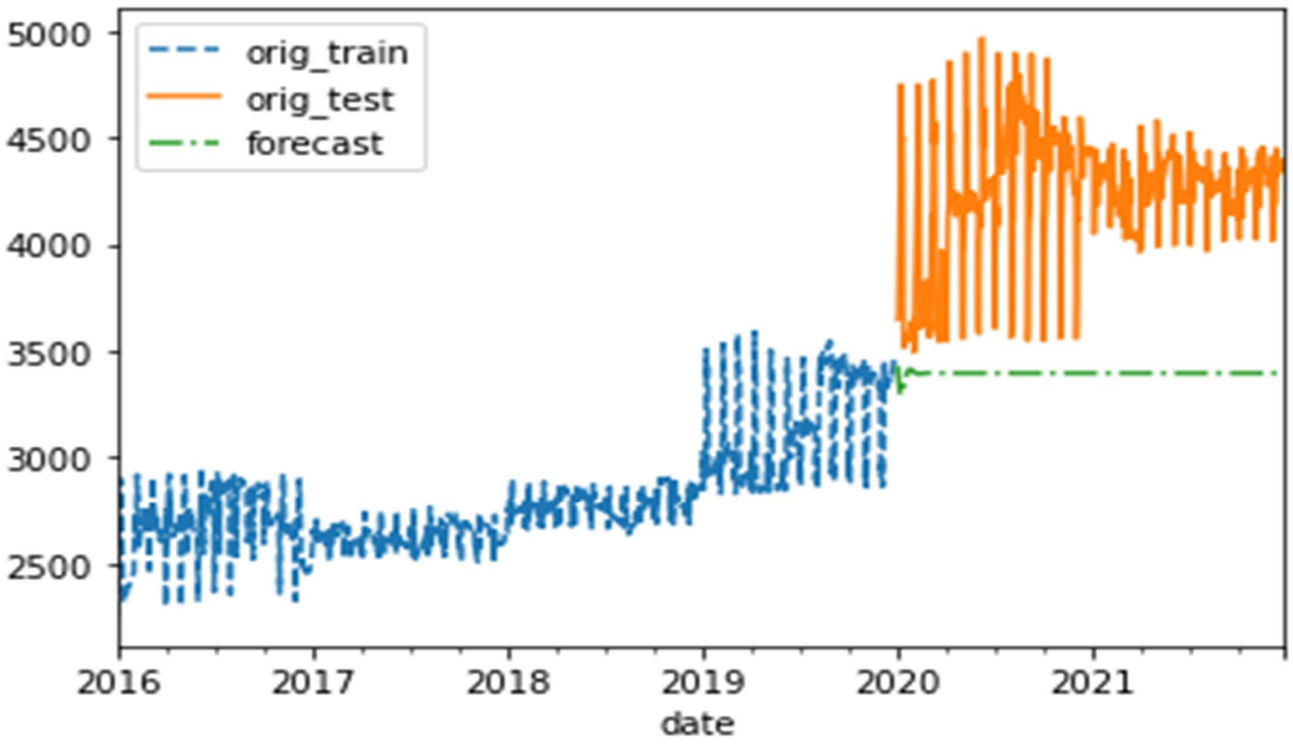
3.4.3 Holt Winter's Exponential Smoothing:

- Triple Exponential Smoothing is an extension of Exponential Smoothing that explicitly adds support for seasonality to the univariate time series.
- This method is sometimes called Holt-Winters Exponential Smoothing.
- In addition to the alpha and beta smoothing factors, a new parameter is added called gamma (g), which controls the influence on the seasonal component.
- As with the trend, the seasonality may be modelled as either an additive or multiplicative process, for a linear or exponential change in the seasonality.
- The hyperparameters fro which the Holt Winter's model was performing best were Trend ='Additive', Seasonality = 'Multiplicative', Seasonal period = 366.



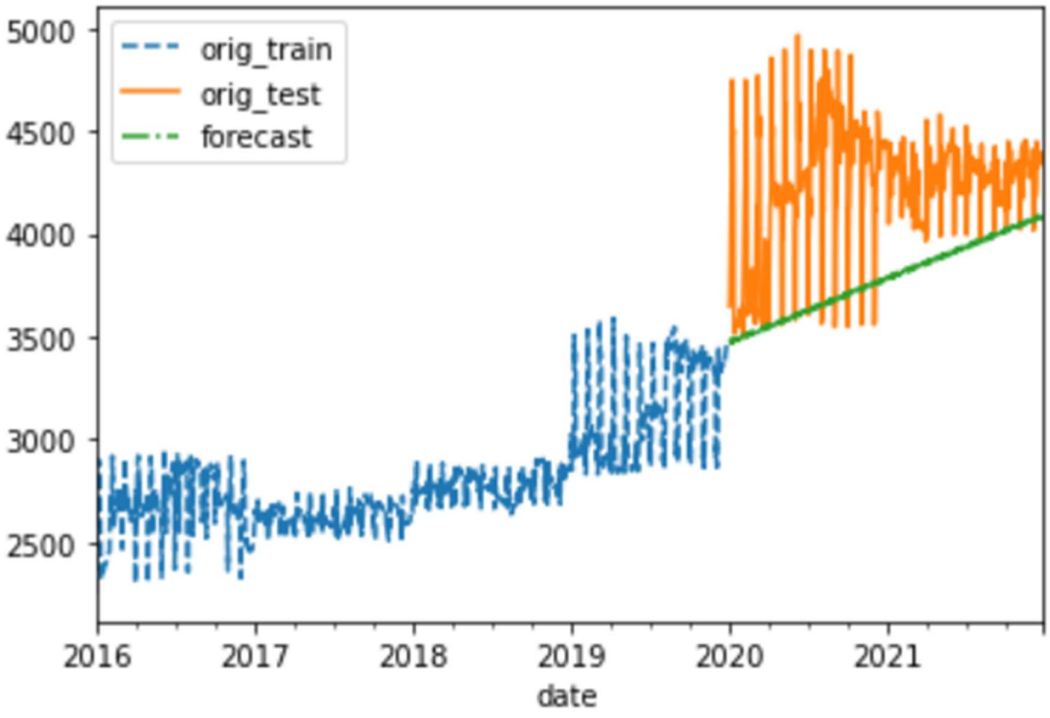
3.4.4 ARIMA (Auto Regressive Integrated Moving Average):

- The ARMA model needs a stationary time series.
- The ARIMA model adds automatic differencing to the ARMA model. It has an additional parameter that you can set to the number of times that the time series needs to be differenced. For example, an ARMA (1,1) that needs to be differenced one time would result in the following notation: ARIMA (1, 1, 1). The first 1 is for the AR order, the second one is for the differencing, and the third 1 is for the MA order. ARIMA (1, 0, 1) would be the same as ARMA (1, 1).
- The Hyperparameters for which the ARIMA model was performing the best were p=2, q=1, d=2.\



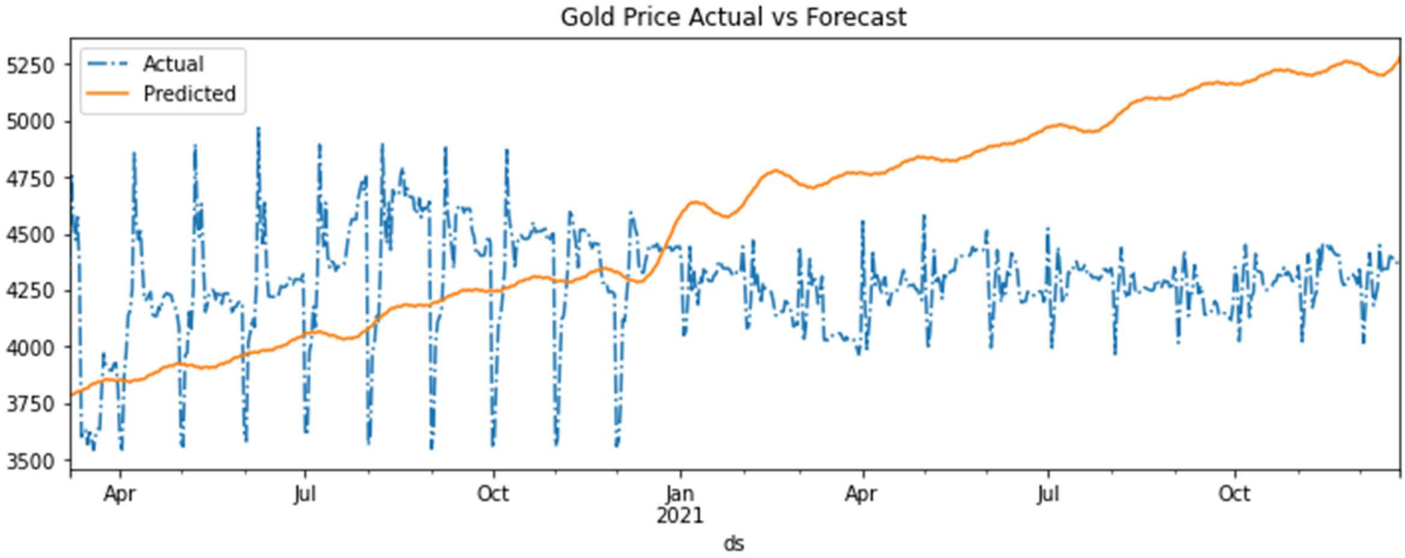
3.4.5 SARIMA (Seasonal Autoregressive Integrated Moving Average):

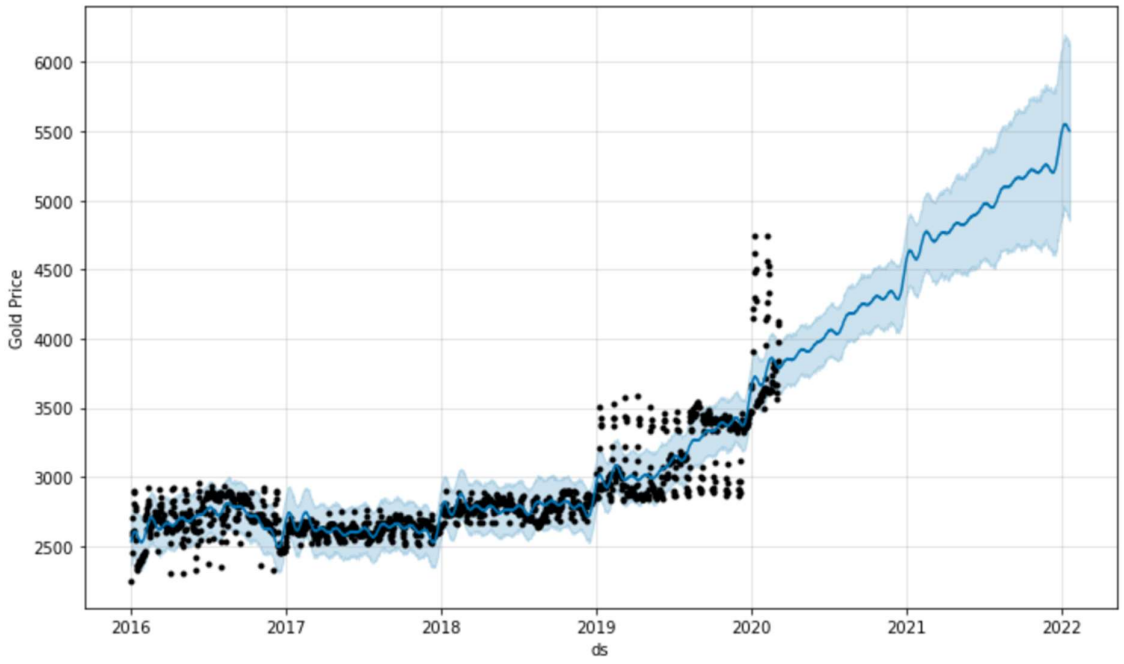
- SARIMA adds seasonal effects into the ARIMA model. If seasonality is present in your time series, it is very important to use it in your forecast.
- SARIMA notation is quite a bit more complex than ARIMA, as each of the components receives a seasonal parameter on top of the regular parameter.
- For example, let's consider the ARIMA (p, d, q) as seen before. In SARIMA notation, this becomes SARIMA (p, d, q, m).
- m is simply the number of observations per year: monthly data has m=12, quarterly data has m=4 etc. The small letters (p, d, q) represent the non-seasonal orders. The capital letters (P, D, Q) represent the seasonal orders.
- The hyperparameters for which the SARIMA model was performing the best were, order = (2,1,2) and seasonal order = (0,1,1,12)



3.4.6 FB Prophet

- Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects.
- It works best with time series that have strong seasonal effects and several seasons of historical data.
- Prophet is robust to missing data and shifts in the trend, and typically handles outliers well.
- Prophet is the facebook's open-source tool for making the time series predictions.
- Prophet decomposes the time series into trend, seasonality and holiday effect.





3.5 Model Evaluation:

The models are trained on the initial 70% of the time series, and tested on the remaining 30%. They try to generate a linear function while assigning coefficients to each of the parameters. These regression coefficients are then used to make predictions. The performance of our models is evaluated based on the following error metrics:

– Mean Relative Error (MRE): -

$$\mathbf{MRE} = \frac{1}{n}\sum \frac{|Pt* - Pt|}{Pt}$$

– Mean absolute Error (MAE): -

$$\mathbf{MAE} = \frac{1}{n}\sum |Pt* - Pt|$$

– Root Mean Square Error (RMSE): -

$$\mathbf{RMSE} = \frac{1}{n}\sqrt{\sum (Pt* - Pt)2}$$

3.6 Observation Table:

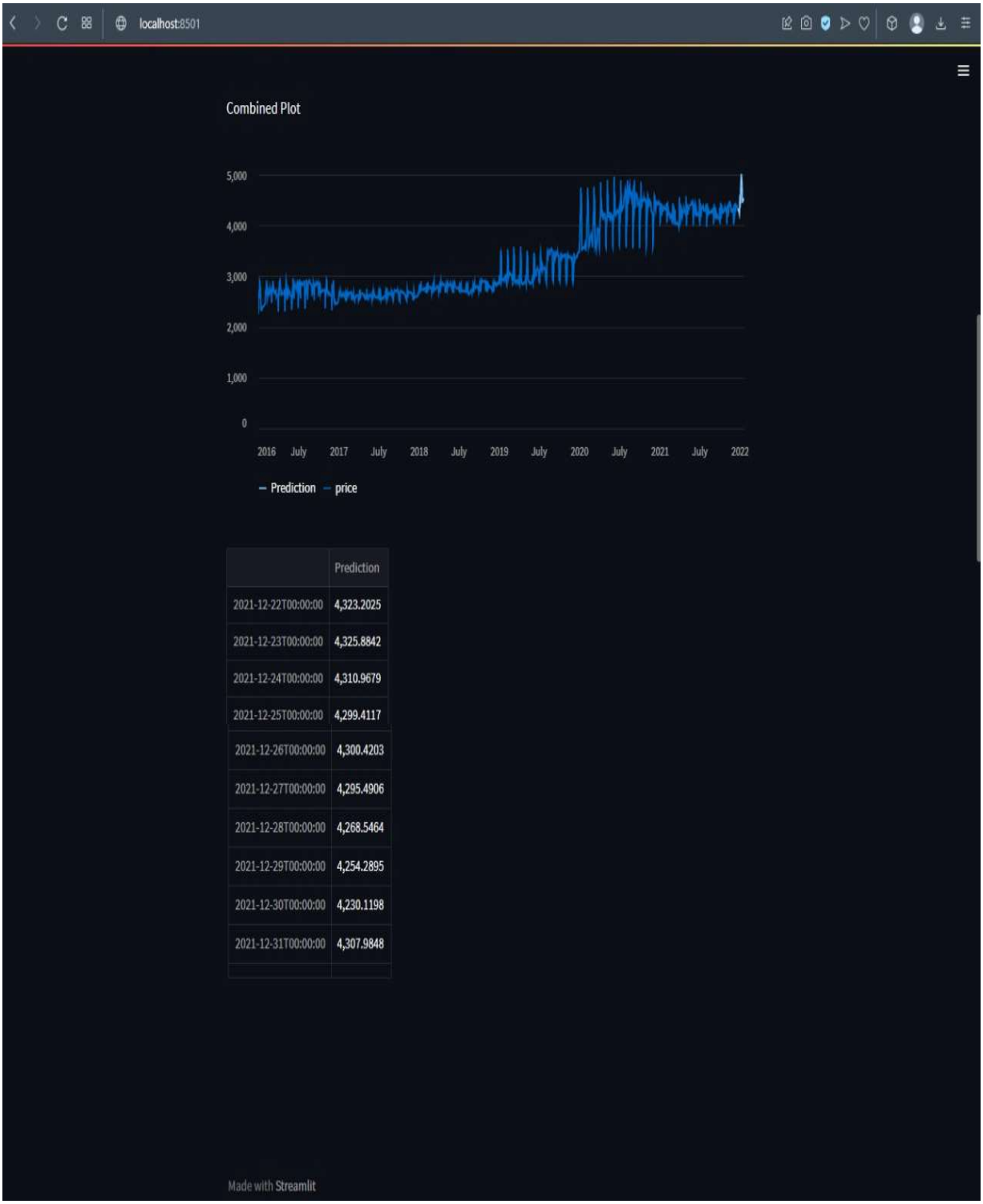
Model	Params	MAPE	RMSE	RMPSE	AIC
Simple exp	Alpha =0.2653	0.22	-	2.332	-
Double exp	Alpha =0.91 Beta =0.91	0.086		1.059	
Holt (Trial 1)	Trend = 'add' Seasonality = 'mul' Seasonal_periods = 365	0.145	677.08	1.548	12931.81
Holt (Trial 2)	Trend = 'add' Seasonality = 'add' Seasonal_periods = 365	•	•	•	13215.034
Holt (Trial 3)	Trend = 'add' Seasonality = 'mul' Seasonal_periods = 365 Transformation = cube root	•	•	0.3633	-5802.679
Holt (Trial 4)	Trend = 'add' Seasonality = 'mul' Seasonal_periods = 366	0.1314	540.123	1.445	
ARIMA	Order (2,1,2)	0.2178	634.33	2.3757	16798.92
SARIMAX	Order (2,1,2) Seasonal order (0,1,1,12)	0.1535	715.175	1.629	16528.14
FB Prophet	•	•	•	•	•

3.7 Finalizing the model:

After performing the analysis and hyperparameter tuning for all the models, model evaluation was done using scores such as MAPE, RMSE, AIC etc. From the result table's we can see, the minimum MAPE score is obtained from the Holt Winter's Model and that is 1.8%. Hence, we finalised the Holt Winter's Model for making the predictions.

3.8 Deployment:

The deployment of the project was done using Streamlit.



	Prediction
2021-12-22T00:00:00	4,323.2025
2021-12-23T00:00:00	4,325.8842
2021-12-24T00:00:00	4,310.9679
2021-12-25T00:00:00	4,299.4117
2021-12-26T00:00:00	4,300.4203
2021-12-27T00:00:00	4,295.4906
2021-12-28T00:00:00	4,268.5464
2021-12-29T00:00:00	4,254.2895
2021-12-30T00:00:00	4,230.1198
2021-12-31T00:00:00	4,307.9848

4. Conclusion

This study evaluated the use of 6 different parametric and nonparametric time series analysis and forecasting techniques, using the daily gold price data from 2016 to 2021. The report consists of the detailed analysis of the data along with model building and comparison of their results. Using various evaluation scores such as AIC, MAPE, RMSE, RMPSE we concluded that, Holt Winter's Model is performing best for the data, giving minimum MAPE and AIC scores compared to other models. Finally, we forecasted the Gold Prices for the next 30 days and deployed the model on streamlit.