PROJECT REPORT

On

Decision Support System Model for PSUs Closing Data Forecasting

Department of Computer Science & Engineering

Dr B R Ambedkar National Institute of Technology,

Jalandhar

Submitted To

Dr. Nonita Sharma

Assistant Professor

Submitted By

Rohit Kumar | 21203019

Suneeta Singh | 21203028

Jahnavi | 21903044

Sanjana Saxena | 21903045

ABSTRACT

Stock market prediction is assembling previous historical data of the shares and after assembling, it is trying to forecast the future value of that particular company's share. This prediction of the stock using historical time series data is based on daily factor.
Hence it becomes convenient to predict the data using past few years.
Basically, our project proposes the model for forecasting the PSUs closing data using BSE; i.e. Bombay Stock Exchange; for past 4 years.

Purpose: *The main purpose of this project is to predict mostly accurate prediction and from this it can generate the profits for the investors. As we know, the time series model works best for the consistent data (regular time intervals) that in turn help us to forecast the future.*

Objective: *The objective of our model is to predict a nearing value of the share and detect the pattern from 4 years of BSE data.*

As we choose the PSUs data from the Bombay Stock Exchange, so the result can be very useful for Government, investors and as well as for all the stakeholders.

TABLE OF CONTENTS

List of Figures

1. <u>INTRODUCTION</u>

1.1.Time Series Forecasting-

Basically, the time series forecasting can be defined as a process in which future can be predicted by making use of given data inhand and past data. The main purpose of working is to have a clear picture of future based on past and present scenario and accordingly decision can be made.

Once, the future trends are obtained accurately, then the business and financial houses can propose a plan accordingly and can solve most of the problem that will be faced in upcoming time.

With the help of time series forecasting, models can be built based on historical analysis and which can further be used in decision making for future. It must be noted that, the more meaningful data we will have, the more accurate our forecasting will be.

1.2.Application of Time Series Forecasting-

- Time Series forecasting plays a vital role in business and financial field as it helps us to gather information related to monthly revenue. There exist policymakers and experts that make use of time series forecasting in order to reach to a conclusion about the economic growth, market sustainability, production etc.
- It's also used in economic forecasting to have an insight of trend that will be witnessed in future. Gross Domestic product (GDP) values can also be obtained through analysis.
- Weather Forecasting again uses the concept of time series forecasting to predict the weather of any location by tracing the past and present record. Eg:- In order to predict the level of humidity of Calcutta for the next month , we need an in depth analysis of data related to previous and present month. The plot obtained after analysis will help us to know about the trend of level of humidity for the next month.
- In the medical field, time series forecasting had also made an irreplaceable position by tracking the health record of the patient accurately. The most common example is the use of ECGs machines through which one can monitor the cardiac position of any patient by recording the pulse rate of heart.

1.3.Time Series data-

When any value is associated with the time, where time can be in any format, that is referred to as time series data. The important characteristic of time series data is that it is recorded over regular time interval, i.e. it should be consistent in nature.

The data we encountered on our excel sheet during the initial stage was the time series data but one drawback with the data was that it was inconsistent in nature and therefore we did preprocessing of the data to make it meaningful and easy to read.

1.4.Flowchart about DSS-

To facilitate the project flow chart is necessary.

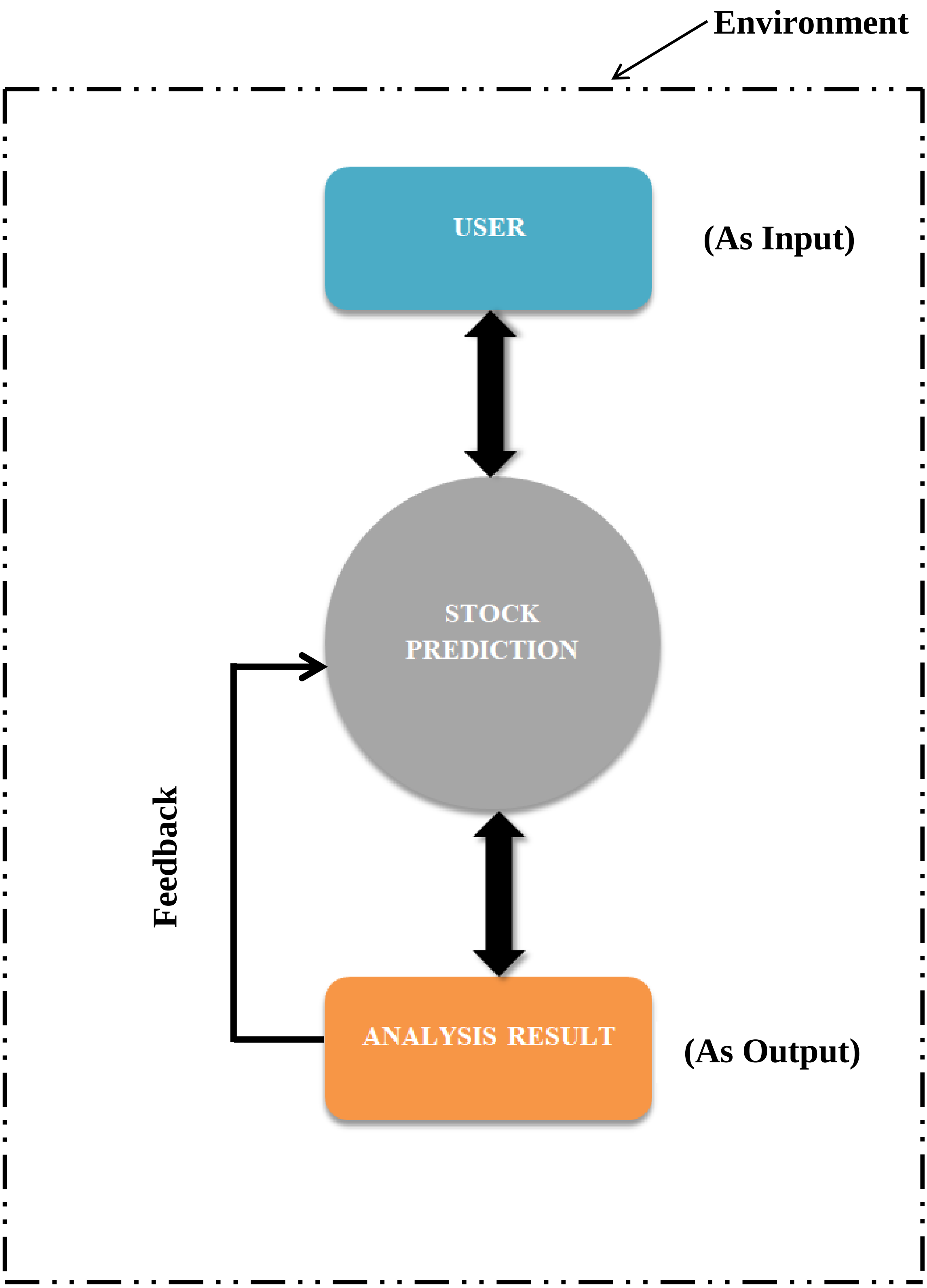


Fig 1: Flowchart of DSS

1.5.About Project-

In our project, we worked on the historical data of PSU (Public Sector Undertaking) from 01-01-2017 to 01-01-2021 on daily basis and our main aim was to predict the future trend. To get started, firstly we downloaded the data from BSE (Bombay Stock Exchange) site whose indices are in Sensex. After downloading it, we did preprocessing of our collected data to make in consistent in nature. During this process, we converted the multivariate data into univariant and performed various steps using different function like "Match". From all these steps, we obtained the data in sorted and meaningful manner which became easy to be read. And finally the data we acquired was ready for analysis.

In order to predict the future data and trend, we followed the given steps:

- **Analyzing the time series data by plotting it:** In this, we first extracted the few initial reading from excel sheet, made changes in the date column by converting it to datetime index and obtained the plot of our readings.
- **Converting time series into stationary form:** To check whether our data is stationary or not, we applied the dickey fuller test, which showed that the data was not stationary and then we carried out differencing. The first differencing, second differencing and seasonal differencing was calculated which was then checked against p-value.
- **Plotted ACF and PACF:** we plotted ACF and PACF which was used further during the construction of ARIMA model.
- **Construction of ARIMA Model:** During this step, ARIMA model was made on basis of given seasonal data by giving certain parameters as input. Once the model was obtained we checked for the error in the model by using 'resid' method. The plot obtained depicted that our model was perfect has it showed distribution of errors around zero.
- **Made the prediction on the basis of model created:** In order to predict the future, we added more months to our dataset and obtained two data frame which we concatenated further to one. Finally we obtained the model prediction line in orange which depicted the future value….Thus we became successful in predicting the future using historical data.

2. METHODOLOGY

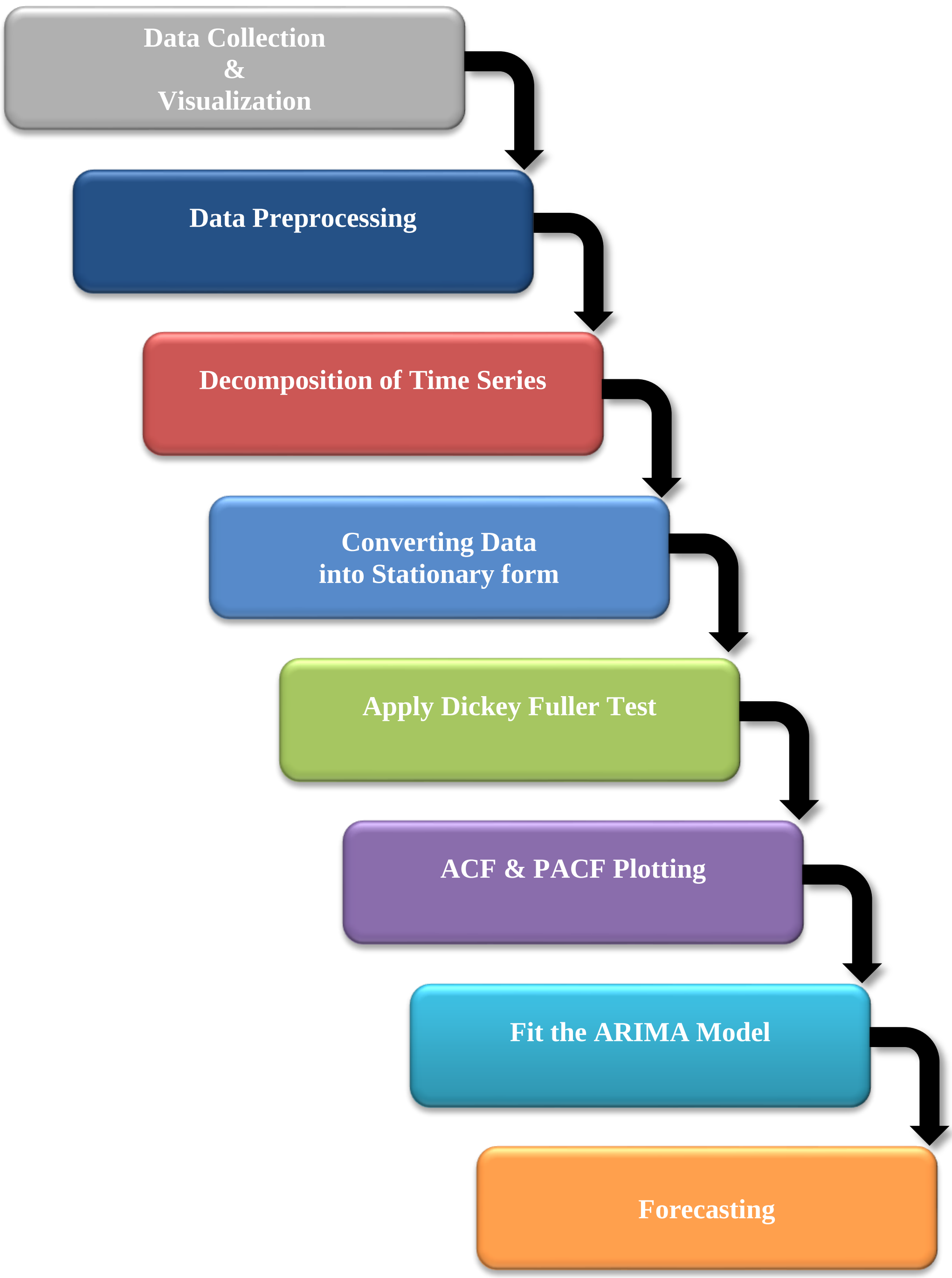
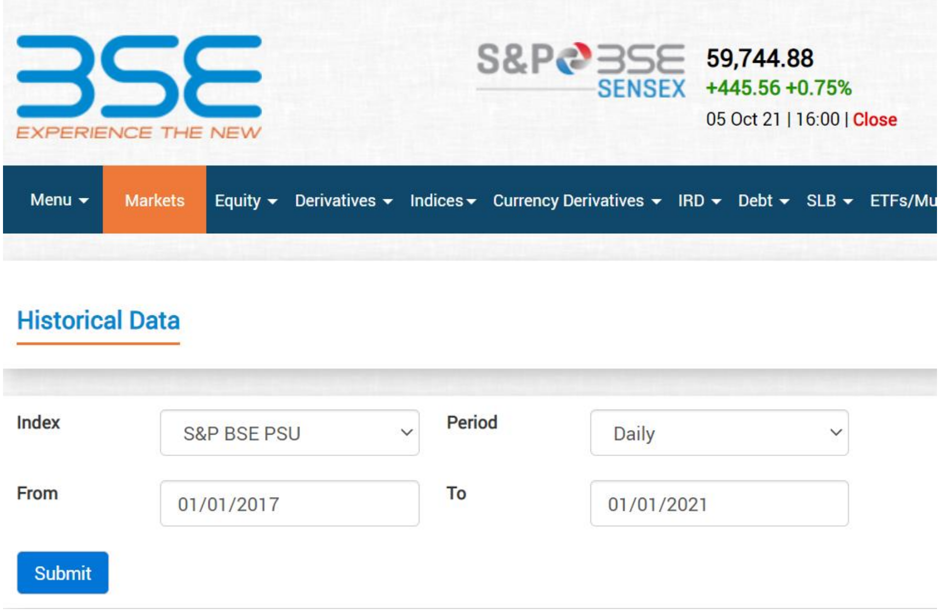


Fig 2: Flowchart of Methodology

2.1.Data Collection-

In this stage, we will collect the data and feed it into an excel file. The data used for this project is collect 4 years (01-01-2017 – 01-01-2021) of S&P BSE PSU data from BSE India, which you can download from: https://www.bseindia.com/Indices/IndexArchiveData.html.
We chose to use the Closing Value for the project.



2.2.Data Preprocessing-

The data we are using for our project is a time series data. It has a constant time interval but on Saturday and Sunday stock market are closed that's why the downloaded data is not in the form of constant time. So for this, we have to do the preprocessing of the data.
STEP 1: Check the data of date that it is in date format or not.
STEP 2: Add the dates of Saturday and Sunday that are missing.
STEP 3: Now use the MATCH function for data filling in the available ones and write NOT AVAILABLE for the Saturday and Sunday.
=MATCH(E2,$A$2:$A$993,0)
Here 0 for exact value.
STEP 4: Now sort the values of Date to oldest to newest.
STEP 5: In the 5th step of preprocessing, we already know that not available values of Saturday and Sunday is as same as their previous (because Saturday and Sunday stock markets are closed); now using following commands, we fill the blank values same as their previous:
: Press Ctrl+G
: Goto Special -> Blanks
: (=B2) -> Press Ctrl+ Enter.
The blank spaces are filled by their previous values.
Now the preprocessed data is ready.

2.3.Original Graphs

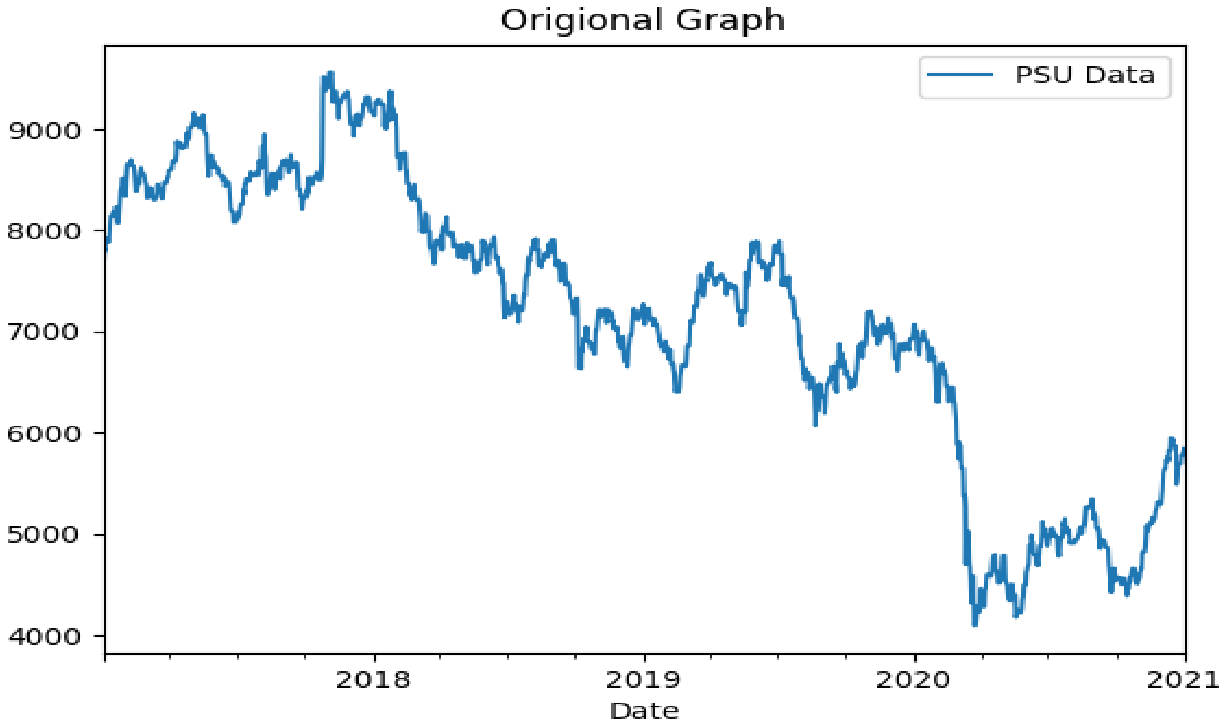


Fig 3: Original Graph

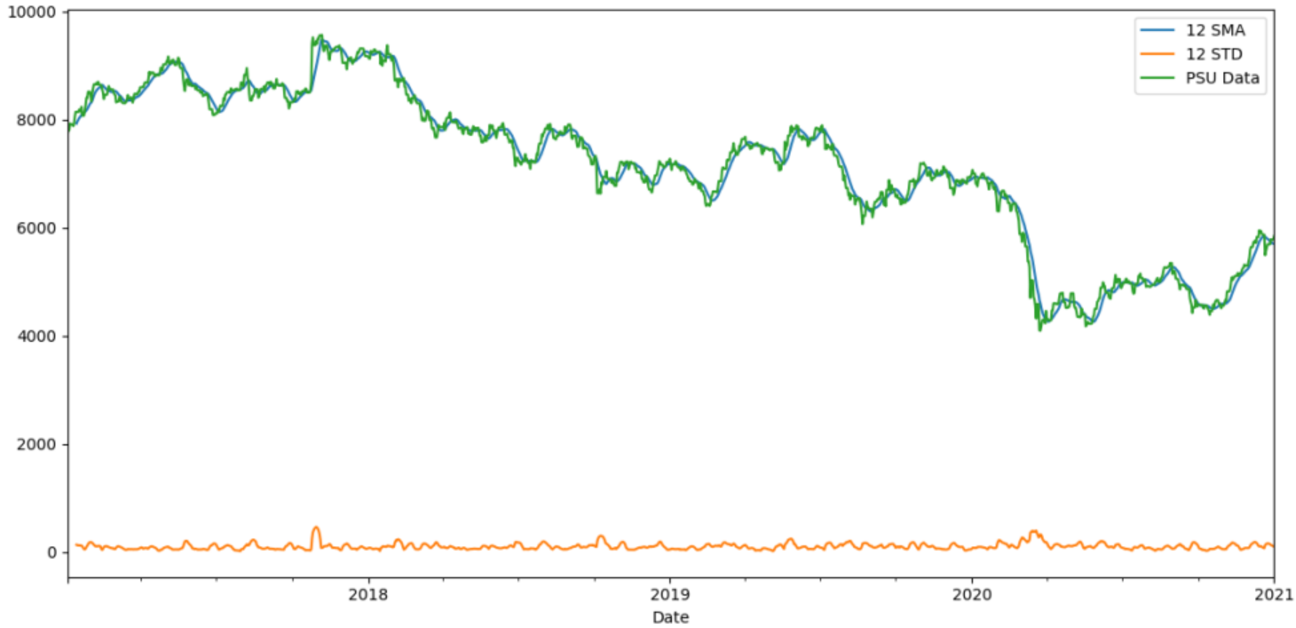


Fig 4: Graph of Visualizing the Data

2.4.Time Series Decomposition-

For the decomposition, statsmodels are used.

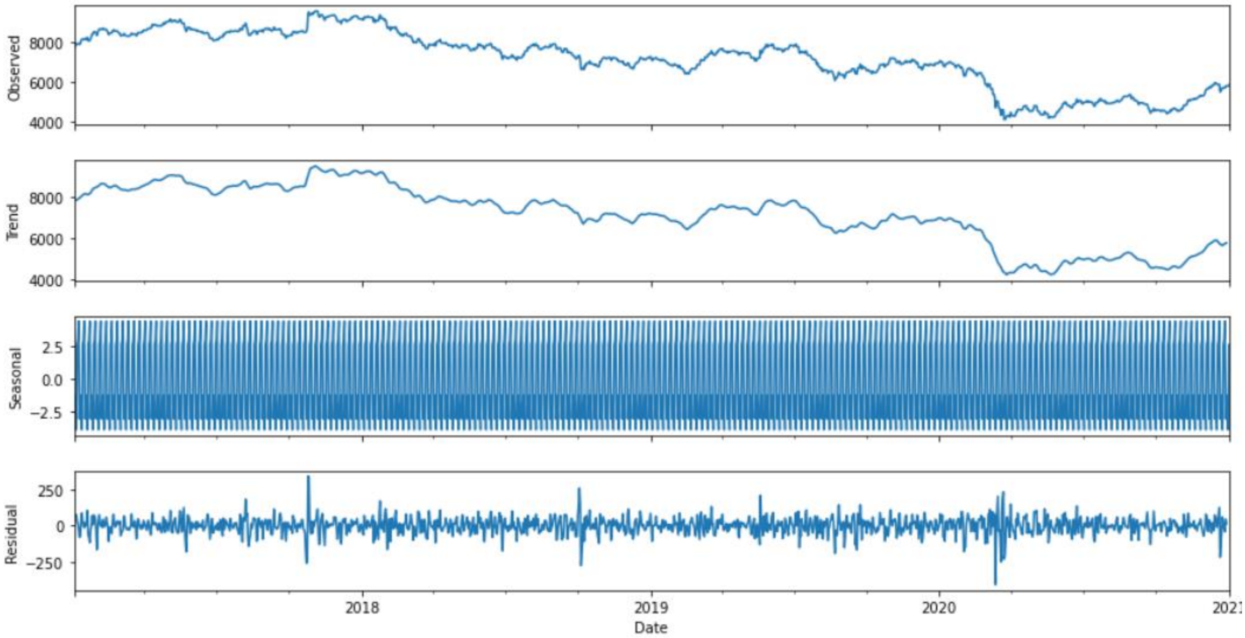


Fig 5: Graph of Time Series Decomposition

2.5.Check the Stationarity of Time Series-

Firstly the given Data is tested that it is stationary or not, by the use of Dickey Fuller Test.

```
#converting the data into stationary form
#dicky fuller test
from statsmodels.tsa.stattools import adfuller
fuller_test = adfuller (df['PSU Data'])
fuller_test
```

2.6.Value of Dickey Fuller Test-

```
#Test p value
def test_p_value(data):
   fuller_test = adfuller(data)
print( ' p-value: ',fuller_test[1])
if fuller_test[1] <= 0.05:
  print( 'Reject null hypothesis,data is stationary')
else:
    print('Do not reject null hypothesis, data is not stationary')
```

```
 p-value:  0.7946549550701474
Do not reject null hypothesis, data is not stationary
```

As you see, the Data is not stationary, So that the differencing is required.

2.7.Differencing-

2.7.1. First Order Differencing

- After plotting First Difference the p-value is: 0.0
- We can see that it rejects null hypothesis and the data is stationary.
- In First difference, we got the data in Stationary form that is why we do not need to do second order differencing.

```
test_p_value(df['First_diff'].dropna())
```

```
p_value :  0.0
Reject null hypothesis,data is stationary
```

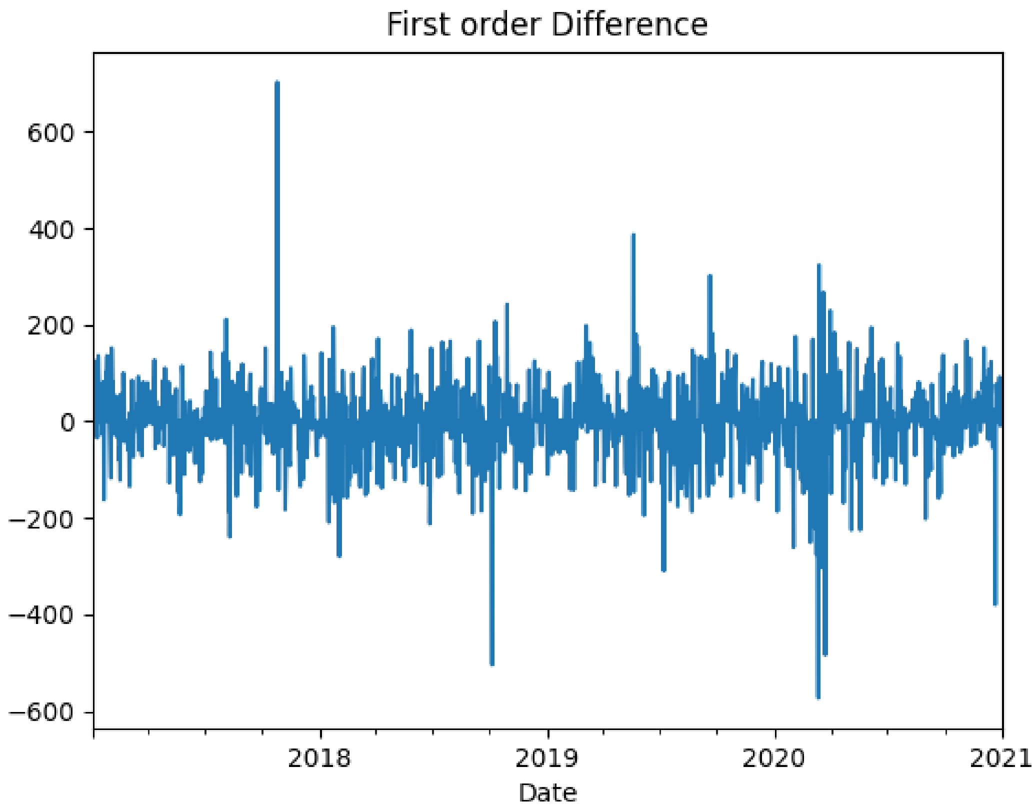


Fig 6: Graph of First Order Difference

2.7.2. Seasonal Differencing-

```
#FSeasonal difference
df['Seasonal_diff']=df['PSU Data'] - df['PSU Data'].shift(12)
df['Seasonal_diff'].plot()
plt.title("Seasonal Difference")
test_p_value(df['Seasonal_diff'].dropna())
result= adfuller(df['Seasonal_diff'].dropna())
print('p-value :',result[1])
if result[1] <= 0.05:
  print( 'Reject null hypothesis,data is stationary')
else:
    print('Do not reject null hypothesis, data is not stationary
```

```
p-value : 2.2793798515297124e-06
Reject null hypothesis,data is stationary
```

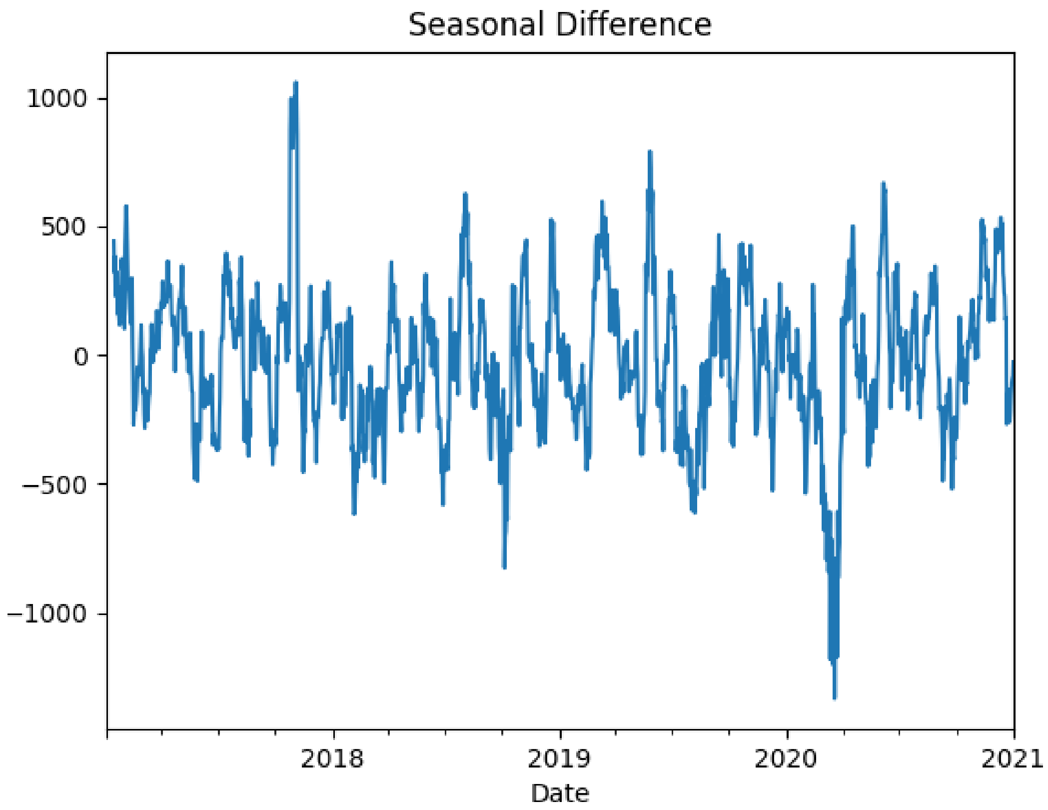


Fig 7: Graph of Seasonal Differencing

2.8.Check the Residuals-

- To know about the residuals values or error, the ‘resid’ method can be called on the results.
- The Plot of the residual point can be created.

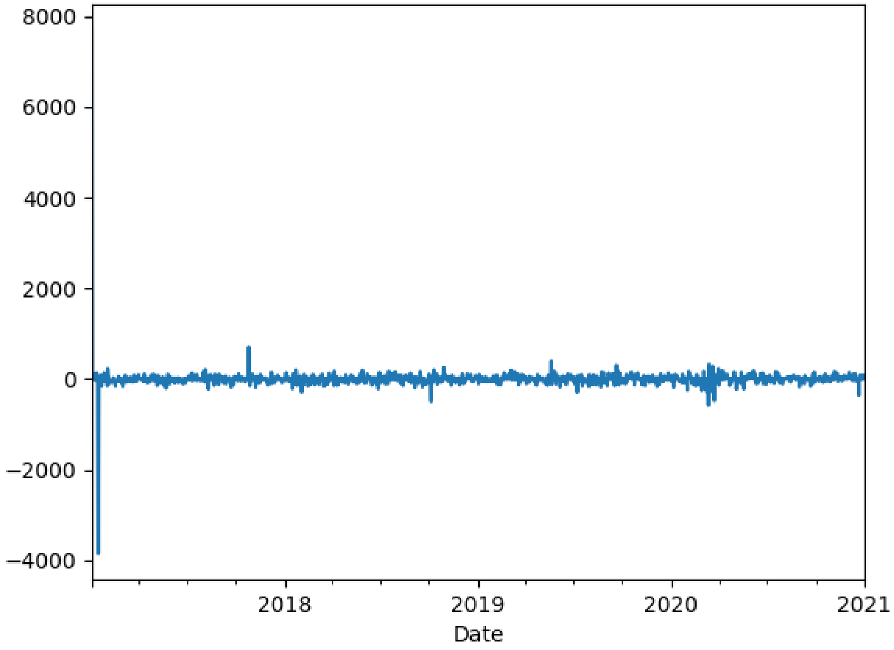


Fig 8: Graph of Checking the Residuals

The Distributed can be clearly seen by plotting the KDE. And after observing the graph below we can see that the graph is distributed around 0 which is good.

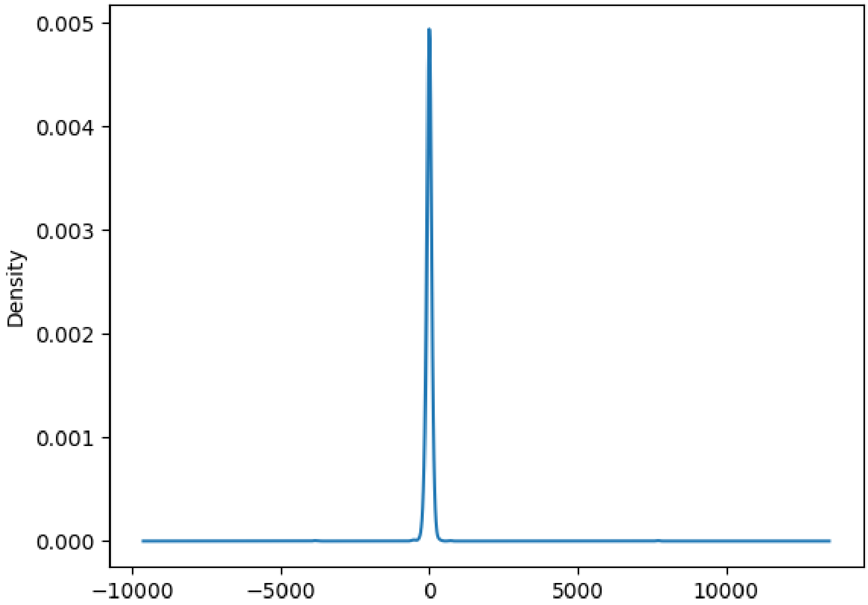


Fig 9: Graph of Distribution

2.9.ACF and PACF Tools-

ACF and PACF plots allow determining the AR and MA components of ARIMA Model.
ARIMA model has order (p,d,q) in which p=autoregressive term(AR), d= differencing term, q= moving term(MA).
We have taken here ARIMA Model of order (0,1,0), where we are using first order differencing
And seasonal order (1,1,1,12),there are the shifting will happen by entire season that is 12.

2.9.1. First difference ACF Plot

- The First difference ACF Plot has shown below:
- And we can observe that our dataset is stationary after first order differencing.

```
#plotting the ACF and PACF plot
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
first_diff = plot_acf(df['First_diff'].dropna())
plt.title("First Difference ACF")
```

```
Text(0.5, 1.0, 'First Difference ACF')
```

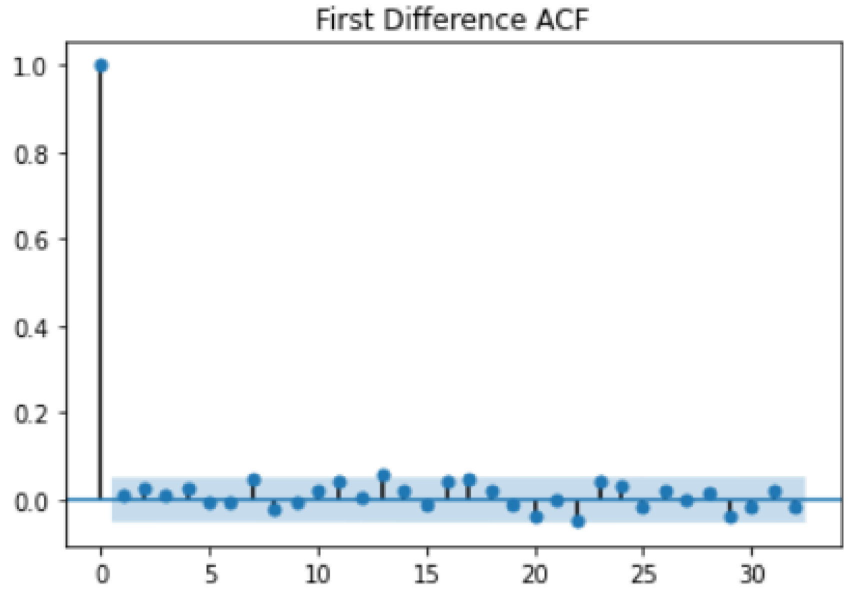


Fig 10: Graph of First Difference ACF

2.9.2. Second difference PACF Plot

- The Second order differencing graph has shown below and we can see that there are more gradually decreasing in the data, which will appropriate.

```
sec_diff = plot_pacf(df['Second_diff'].dropna())
plt.title("Second Difference ACF")
```

```
Text(0.5, 1.0, 'Second Difference ACF')
```

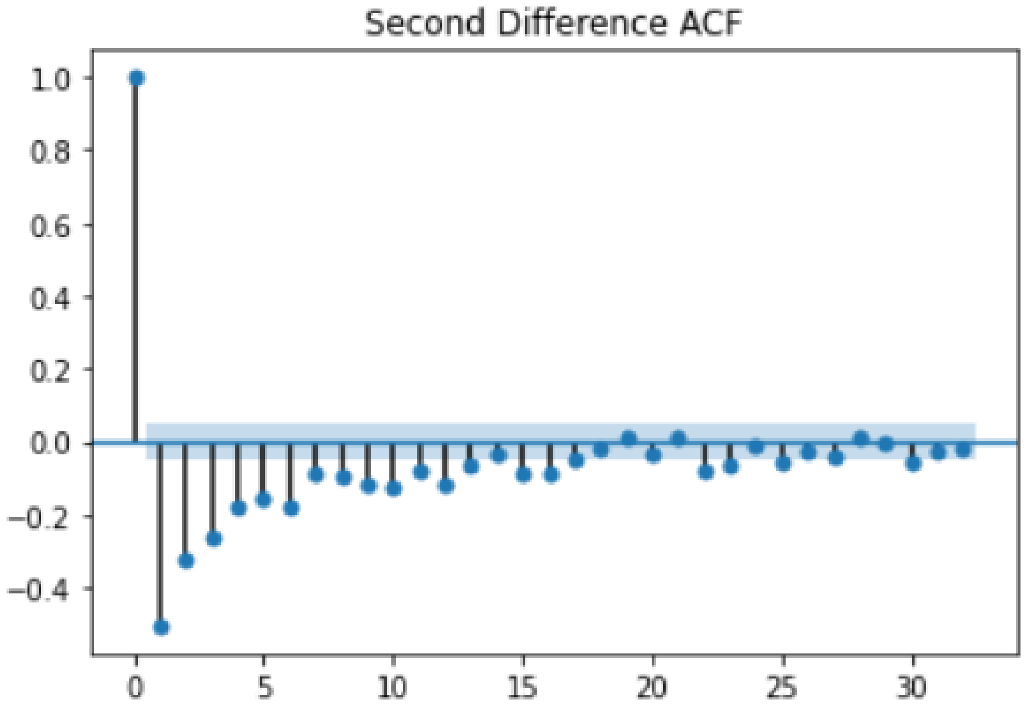


Fig 11: Graph of Second Order PACF

2.9.3. Seasonal difference ACF Plot

- By plotting the autocorrelation seasonal graph, we can see that more number of positive observations in graph. And some data stored within our given region. Graph is gradually decreasing till every 12 lags.

```
p1 = plot_acf(df['Seasonal_diff'].dropna())
plt.title("Autocorrelation Seasonal Difference")
```

```
Text(0.5, 1.0, 'Autocorrelation Seasonal Difference')
```

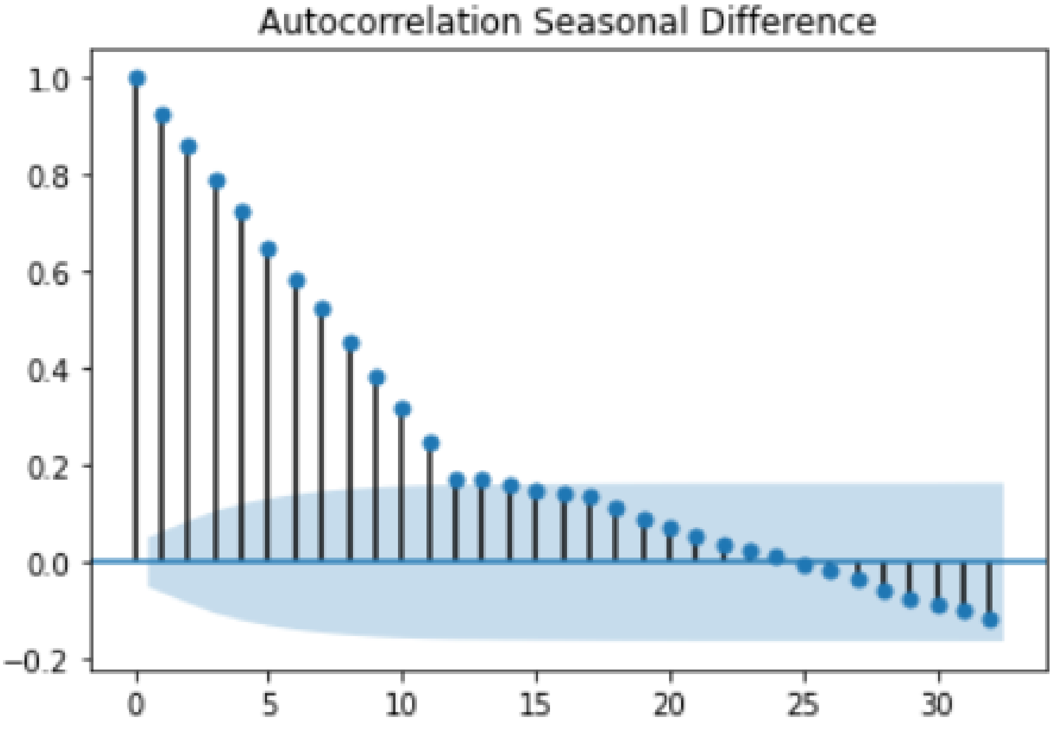


Fig 12: Graph of Seasonal Difference of ACF

2.9.4. Seasonal Difference PACF Plot

- By observing Partial Autocorrelation graph we can see that more number of dataset stored in within given range and also some other has high value which is out of our range. Graph is almost constant till every 12 lags.

```
p2 = plot_pacf(df['Seasonal_diff'].dropna())
plt.title("Partial Autocorrelation Seasonal Difference")
```

```
Text(0.5, 1.0, 'Partial Autocorrelation Seasonal Difference')
```

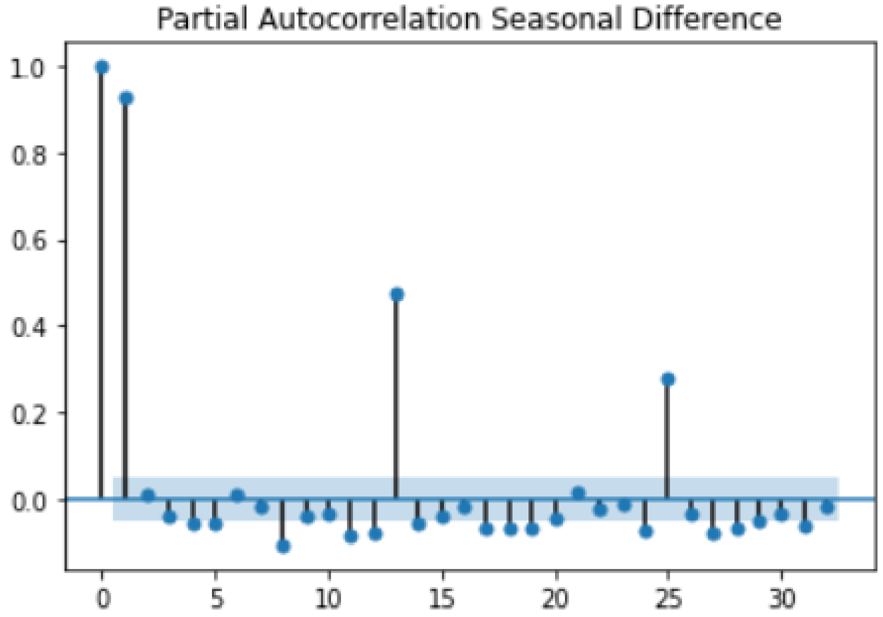


Fig 13: Graph of Seasonal Difference of PACF

2.10. Forecasting Model-

By predicting the values, the model's performance can be ascertained. First, we can predict the data present and then move onto predicting future data.

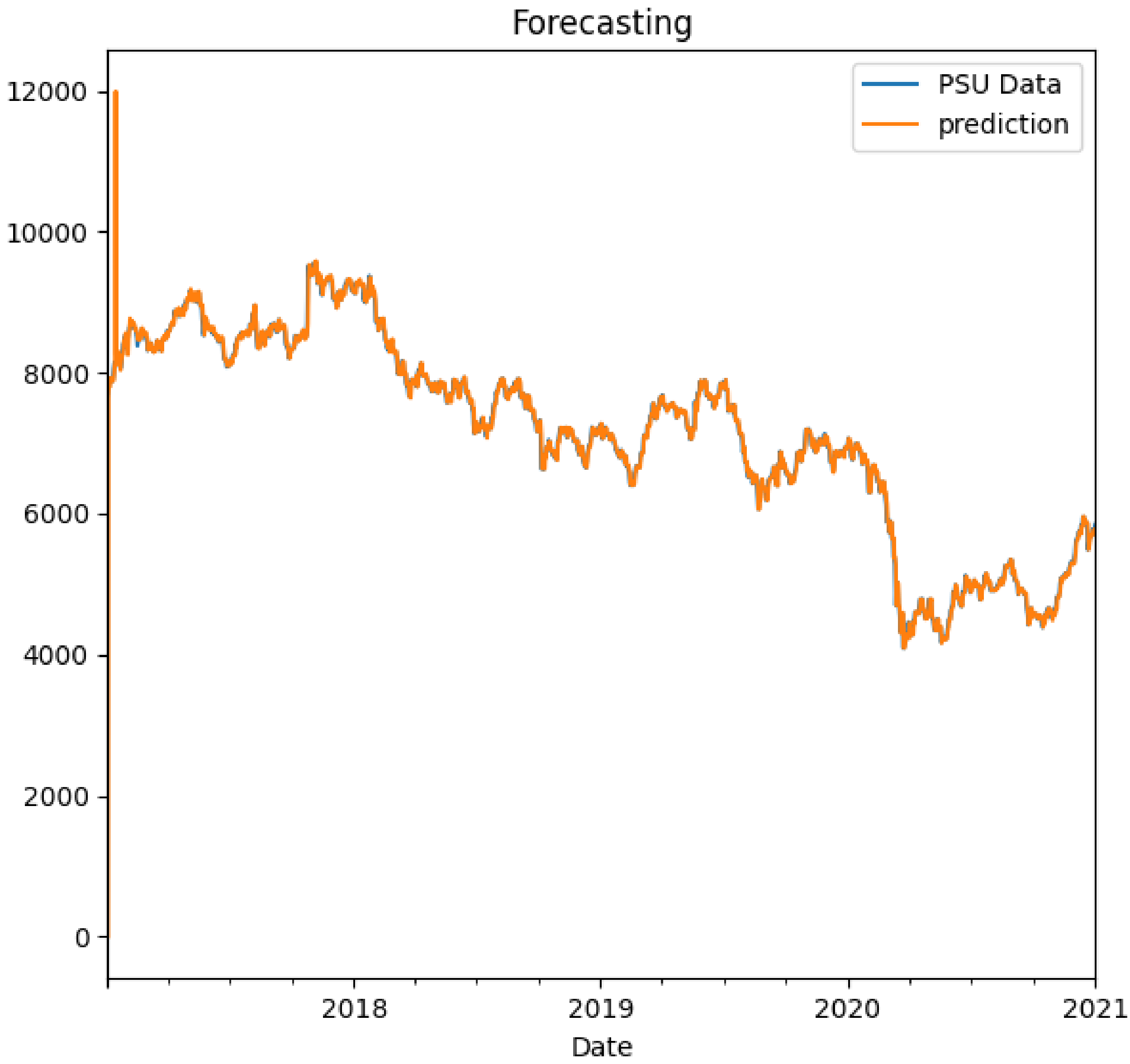


Fig 14: Graph of Forecasting Model

As seen from the above plot, the model does a good job is predicting the present data. Now to predict for the future, we can add more months to the dataset with null values and predict for it. This can be done using pandas. The last index is obtained which is the last date and a month offset is added to it which starts from 1 and goes up to 24.

The Extra Dates are:
[Timestamp ('2021-02-01 00:00:00'),
Timestamp ('2021-03-01 00:00:00'),
Timestamp ('2021-04-01 00:00:00'),
Timestamp ('2021-05-01 00:00:00'),
Timestamp ('2021-06-01 00:00:00'),
Timestamp ('2021-07-01 00:00:00'),
Timestamp ('2021-08-01 00:00:00'),
Timestamp ('2021-09-01 00:00:00'),
Timestamp ('2021-10-01 00:00:00'),
Timestamp ('2021-11-01 00:00:00'),
Timestamp ('2021-12-01 00:00:00'),
Timestamp ('2022-01-01 00:00:00'),
Timestamp ('2022-02-01 00:00:00'),
Timestamp ('2022-03-01 00:00:00'),
Timestamp ('2022-04-01 00:00:00'),
Timestamp ('2022-05-01 00:00:00'),
Timestamp ('2022-06-01 00:00:00'),
Timestamp ('2022-07-01 00:00:00'),
Timestamp ('2022-08-01 00:00:00'),
Timestamp ('2022-09-01 00:00:00'),
Timestamp ('2022-10-01 00:00:00'),
Timestamp ('2022-11-01 00:00:00'),
Timestamp ('2022-12-01 00:00:00')]

Now another data frame is created with these extra future date values as an index and the rest of the column values as null.

	PSU Data	First_diff	Second_diff	Seasonal_diff
2021-02-01	NaN	NaN	NaN	NaN
2021-03-01	NaN	NaN	NaN	NaN
2021-04-01	NaN	NaN	NaN	NaN
2021-05-01	NaN	NaN	NaN	NaN
2021-06-01	NaN	NaN	NaN	NaN

Now the original data frame and the future data frame is concatenated into a single data frame for forecasting.

2.11. Final Result-

The final plotting of our forecasting result has shown below:

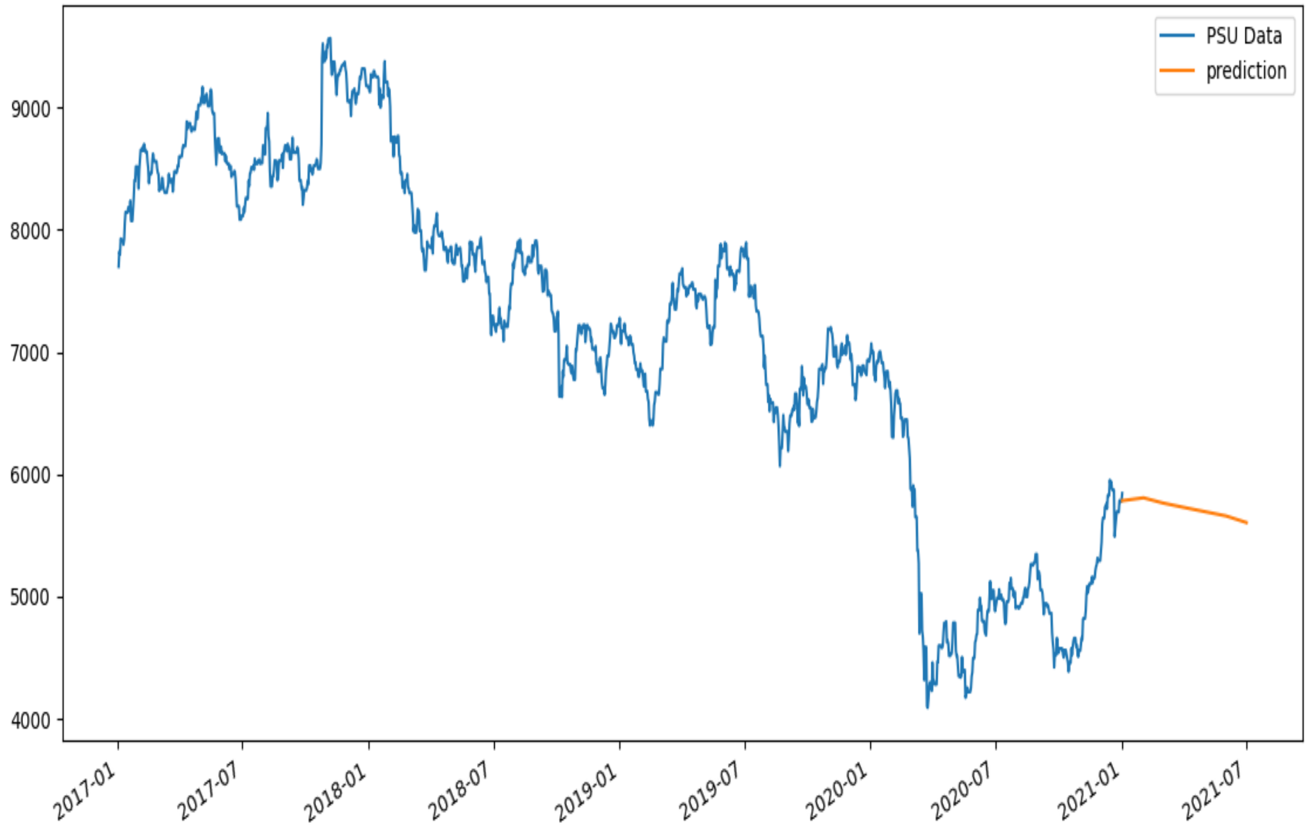


Fig 15: Final Graph

The Prediction can be seen clearly for the future values.

3. CONCLUSION

ARIMA model was successfully used to predict the future values for the PSU Dataset which is a seasonal dataset with non-stationary behavior. Using the proper steps, the data was converted to the stationary form and the prediction model was built.

REFERENCES

1) Kulkarni M., Jadha A. & Dhingra D., **"Time Series Data Analysis for Stock Market Prediction",** *3rd International Conference On Innovative Computing And Communication* (ICICC-2020).
   Available on: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3563111

2) Turban Efraim, ***Decision support systems and intelligent systems***, 1998.
3) Stock Market Forecasting
4) Electricity production forecasting using ARIMA model in Python

Thank You

For Your Attention