DAILY POWER CONSUMPTION DATASET ANALYSIS AND FORECASTING

Summer Project Report

Gourab Ghosh
M.Sc. in Big Data Analytics, Department of Computer Science
Ramakrishna Mission Vivekananda Educational And Research Institute
Belur Math, Howrah
Pin-711202, West Bengal
May 12, 2021

ACKNOWLEDGEMENT

This is to acknowledge the support and help that has been received while doing this Project, **DAILY POWER CONSUMPTION DATASET ANALYSIS AND FORECASTING** from our Teacher, Dr. Sudipta Das. It is very kind of him to help me in various stages while doing the project and to provied necessary informations which are of geart help. I am very much grateful to him and thanks him for his cordial coordination.
I would also like to thank my Institution and faculty members for helping me in my project. I want to express my gratitude to Br.Mrimay Mj, Br.Tamal Mj, Dr. Aditya Bagchi and Swathy Prabhu Maharaj, H.O.D of Computer Science Dept, RKMVERI Belur, for extending their support
At last but not the least I would like to express my thanks to my parents without whom this project would not have been possible.

A **Project Report** submitted by **GOURAB GHOSH**
to Ramakrishna Mission Vivekananda Educational and Research Institute for the course of **SUMMER PROJECT** in **M.Sc. in Big Data Analytics**

TABLE OF CONTENT

1 INTRODUCTION

A **time series** is a series of data points indexed or listed or graphed in time order. Most commonly, a time series is a sequence taken at successive equally spaced points in time.
Time Series analysis can be useful to see how a given asset, security or economic variable changes over time. Time series analysis comprises methods for analysing time series data in order to extract meaningful statistics and other characteristics of the data. Time series forecasting is the use of a model to predict future values based on previously observed values.
A number of different notations are in use for time-series analysis. A common notation specifying a **time series X** that is indexed by the natural numbers is written as **X = [X1, X2, ...]**.Another common notation is **Y = [Yt: t belongs to T]**, where **T is the index set**.

2 OVERVIEW OF PROJECT

Daily Power Consumption Data is a dataset of a household where we have different columns, such as **Global active power**, **Global reactive power**, **Voltage**, **Global intensity**. Our time series analysis interest lies in these column categories. For analysis and forecasting of time series, we have to go through following stages:

- At first, we have to read the dataset and check if there is any missing value in it or not. Depending on that we have to prepare that accordingly.
- We have to then do Data Processing and prepare the variables,**Global active power**,**Global reactive power**,**Voltage**,**Global intensity** for time series analysis accordingly.
- We have to **visualize** our dataset then and try to get meaningful idea and important vision to work on.
- For time series, an important part is **Stationarity Checking**. We have to then check the Stationarity of our time series data and make them stationary if not.
- After Stationarity Checking, **ACF PACF plots** will be drawn and from there we can get the idea of our time series model and their components.
- Then we will try to check if there is any **causal relationship** between the time series and will do the **Granger's Causality Test** for that.
- After **casual relationship** checking, we have to apply our time series model accordingly.
- At the last, we will **forecast our Time Series** and **visualize it with future predictions**.

3 DATA COLLECTION

We are going to work on **Daily Power Consumption Dataset**. An overview of the dataset is like as follows:

```
In [5]: df=pd.read_csv(r'C:\Users\GOURAV\Downloads\household_power_consumption.txt', delimiter=';')
        print('Number of rows and columns:', df.shape)
        df

        Number of rows and columns: (2075259, 9)
```

Out[5]:

	Date	Time	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3
0	16/12/2006	17:24:00	4.216	0.418	234.840	18.400	0.000	1.000	17.0000
1	16/12/2006	17:25:00	5.360	0.436	233.630	23.000	0.000	1.000	16.0000
2	16/12/2006	17:26:00	5.374	0.498	233.290	23.000	0.000	2.000	17.0000
3	16/12/2006	17:27:00	5.388	0.502	233.740	23.000	0.000	1.000	17.0000
4	16/12/2006	17:28:00	3.666	0.528	235.680	15.800	0.000	1.000	17.0000
...	...	...	...	...	...	...	...	...	...
2075254	26/11/2010	20:58:00	0.9460	0.0000	240.4300	4.0000	0.0000	0.0000	0.0000
2075255	26/11/2010	20:59:00	0.9440	0.0000	240.0000	4.0000	0.0000	0.0000	0.0000
2075256	26/11/2010	21:00:00	0.9380	0.0000	239.8200	3.8000	0.0000	0.0000	0.0000
2075257	26/11/2010	21:01:00	0.9340	0.0000	239.7000	3.8000	0.0000	0.0000	0.0000
2075258	26/11/2010	21:02:00	0.9320	0.0000	239.5500	3.8000	0.0000	0.0000	0.0000

2075259 rows × 9 columns

Figure 1: Dataset Overview

Here, we can see that the dataset has **9 columns** and total **2075259 rows**.
There are several categories, such as **four(4)** in the dataset and they are:**Global active power**,**Global reactive power**,**Voltage**,**Global intensity**. Our aim is to analysis and forecast these different 4 categories. We can clearly understand that we have **hourly based data** for all the categories. So we have to transform it to **daily data** for our working convenience and so our next task will be **Data Processing**.

4 DATA PREPARATION

At the very beginning I have combine the Date and Time column to one column, **"DateTime"** column so that I can make the categories as Time Series. Then I have checked if there is any **null value** or missing value in the dataset and I found that there is some, so I remove them from the dataset.

```
In [6]: df['date_time'] = pd.to_datetime(df['Date'] + ' ' + df['Time'])

In [7]: df['Global_active_power'] = pd.to_numeric(df['Global_active_power'], errors='coerce')
        df['Global_reactive_power'] = pd.to_numeric(df['Global_reactive_power'], errors='coerce')
        df['Voltage'] = pd.to_numeric(df['Voltage'], errors='coerce')
        df['Global_intensity'] = pd.to_numeric(df['Global_intensity'], errors='coerce')

In [8]: df['date_time']=pd.to_datetime(df['date_time'])

In [9]: df=df.loc[:,['date_time','Global_active_power', 'Global_reactive_power', 'Voltage','Global_intensity']]
        df.sort_values('date_time', inplace=True, ascending=True)
        df = df.reset_index(drop=True)

In [10]: df.head()
```

Out[10]:

	date_time	Global_active_power	Global_reactive_power	Voltage	Global_intensity
0	2006-12-16 17:24:00	4.2160	0.4180	234.8400	18.4000
1	2006-12-16 17:25:00	5.3600	0.4360	233.6300	23.0000
2	2006-12-16 17:26:00	5.3740	0.4980	233.2900	23.0000
3	2006-12-16 17:27:00	5.3880	0.5020	233.7400	23.0000
4	2006-12-16 17:28:00	3.6660	0.5280	235.6800	15.8000

```
In [11]: df.isnull().sum()

Out[11]: date_time                0
         Global_active_power      25979
         Global_reactive_power    25979
         Voltage                  25979
         Global_intensity         25979
         dtype: int64

In [12]: df = df.dropna(subset=['Global_active_power'])
         df = df.dropna(subset=['Global_reactive_power'])
         df = df.dropna(subset=['Voltage'])
         df = df.dropna(subset=['Global_intensity'])

In [13]: df.isnull().sum()

Out[13]: date_time                0
         Global_active_power      0
         Global_reactive_power    0
         Voltage                  0
         Global_intensity         0
         dtype: int64
```

Figure 2: Data Preparation I

Then as we can clearly see that the dataset has **hourly DateTime**, we have to make it **daily DateTime** for our working convenience. Then again we do null value checking and remove them from the dataset.

```
In [15]: df = df.set_index('date_time')
         df.index

Out[15]: DatetimeIndex(['2006-12-16 17:24:00', '2006-12-16 17:25:00',
                        '2006-12-16 17:26:00', '2006-12-16 17:27:00',
                        '2006-12-16 17:28:00', '2006-12-16 17:29:00',
                        '2006-12-16 17:30:00', '2006-12-16 17:31:00',
                        '2006-12-16 17:32:00', '2006-12-16 17:33:00',
                        ...
                        '2010-12-11 23:50:00', '2010-12-11 23:51:00',
                        '2010-12-11 23:52:00', '2010-12-11 23:53:00',
                        '2010-12-11 23:54:00', '2010-12-11 23:55:00',
                        '2010-12-11 23:56:00', '2010-12-11 23:57:00',
                        '2010-12-11 23:58:00', '2010-12-11 23:59:00'],
                       dtype='datetime64[ns]', name='date_time', length=2049280, freq=None)

In [16]: df= df.resample('D').mean()
         df.head()

Out[16]:
```

date_time	Global_active_power	Global_reactive_power	Voltage	Global_intensity
2006-12-16	3.0535	0.0882	236.2438	13.0828
2006-12-17	2.3545	0.1569	240.0870	9.9990
2006-12-18	1.5304	0.1124	241.2317	6.4217
2006-12-19	1.1571	0.1048	241.9993	4.9264
2006-12-20	1.5457	0.1118	242.3081	6.4674

```
In [17]: df.isnull().sum()

Out[17]: Global_active_power      24
         Global_reactive_power    24
         Voltage                  24
         Global_intensity         24
         dtype: int64

In [18]: df = df.dropna(subset=['Global_active_power'])
         df = df.dropna(subset=['Global_reactive_power'])
         df = df.dropna(subset=['Voltage'])
         df = df.dropna(subset=['Global_intensity'])

In [19]: df.isnull().sum()

Out[19]: Global_active_power      0
         Global_reactive_power    0
         Voltage                  0
         Global_intensity         0
         dtype: int64
```

Figure 3: Data Preparation II

5 DATA VISUALIZATION

After transforming the categories into different time series dataset and **indexing over DateTime**,we have to **visualize them differently** and should try to find out some meaningful information that would help us in further analysis and forecast.

Plot of **4 diffferent categories** together.

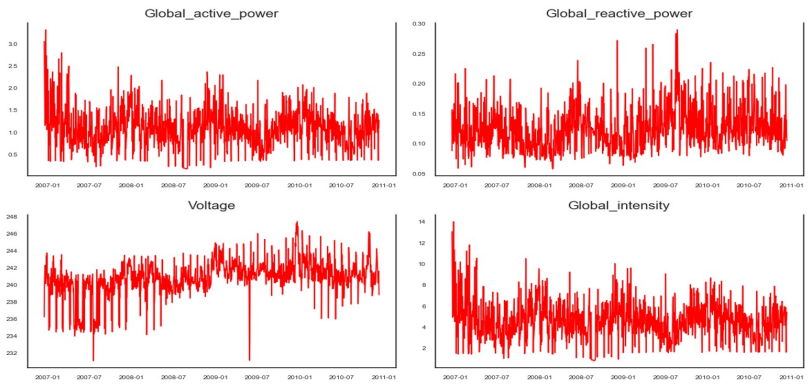

Figure 4: Visualization of Categories I

Global Active Power category visualize into different **Time Series** Components.

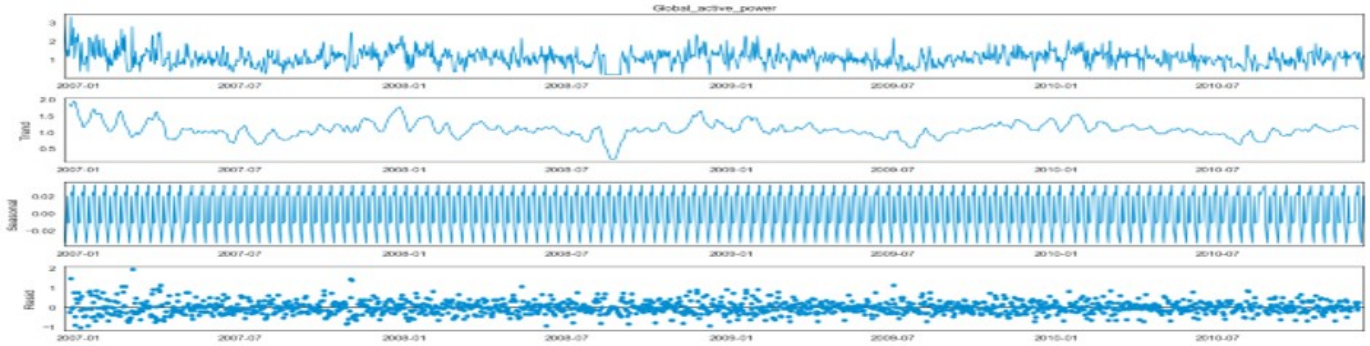

Figure 5: Time Series Component Breakdown for Global Active Power I

Global Reactive Power category visualize into different **Time Series** Components.

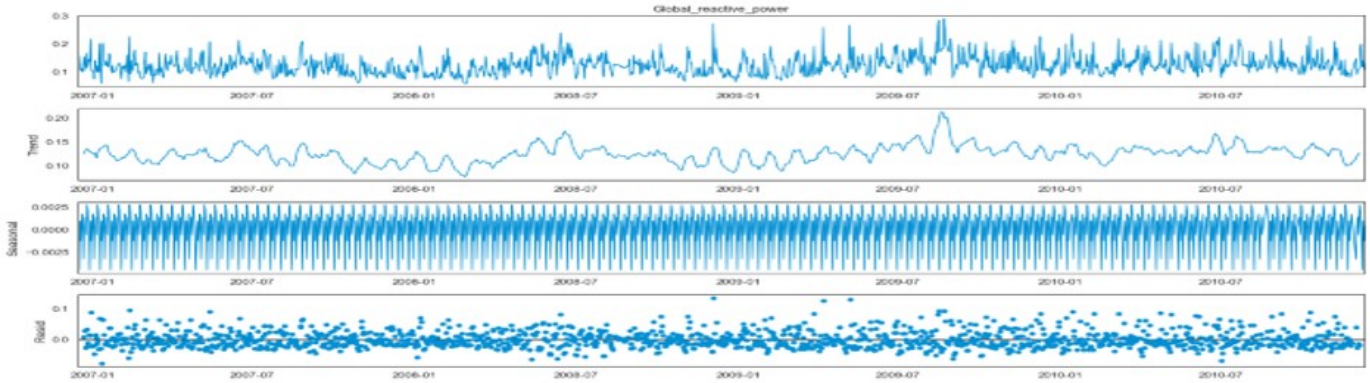

Figure 6: Time Series Component Breakdown for Global Reactive Power I

Voltage category visualize into different **Time Series** Components.

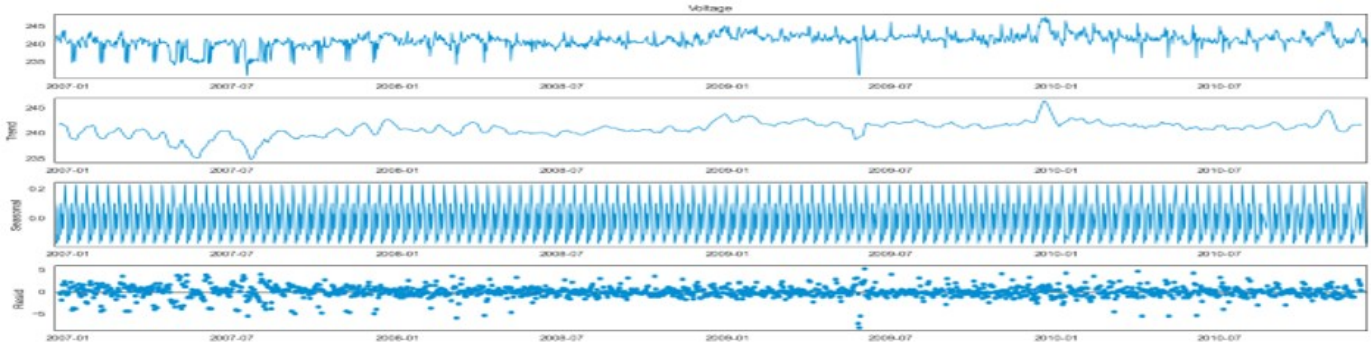

Figure 7: Time Series Component Breakdown for Voltage I

Global Intensity category visualize into different **Time Series** Components.

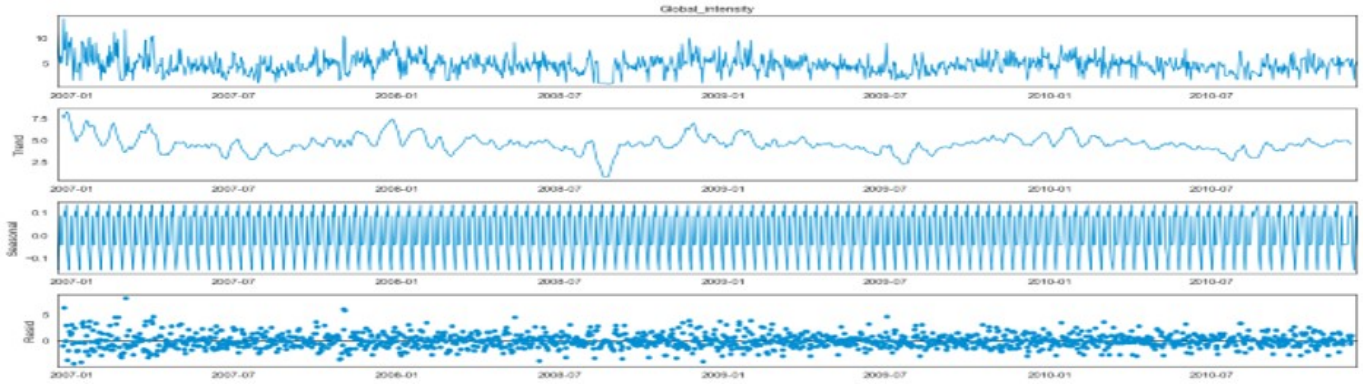

Figure 8: Time Series Component Breakdown for Global Intensity I

We can find in the data visualization that the data for each category is **not centred to Zero (0)**. So we have to make them **Zero(0) centred** and visualize them.

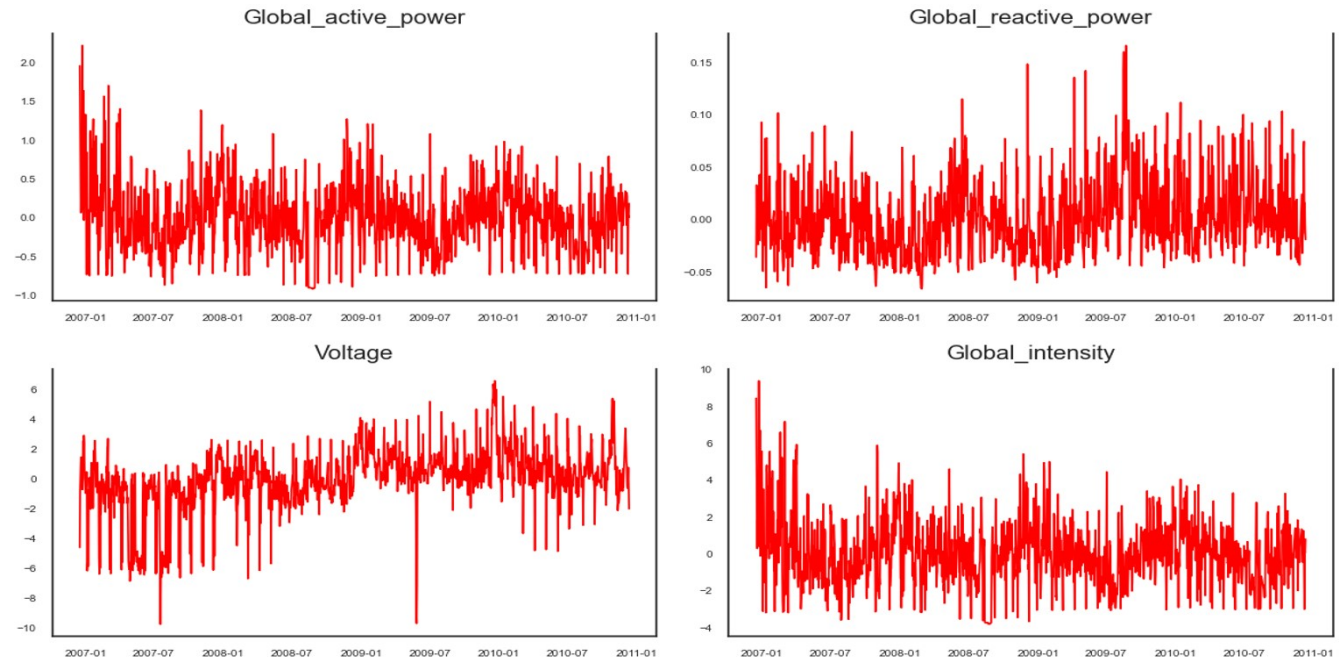

Figure 9: Visualization of Categories II

Global Active Power category visualize into different **Time Series** Components.

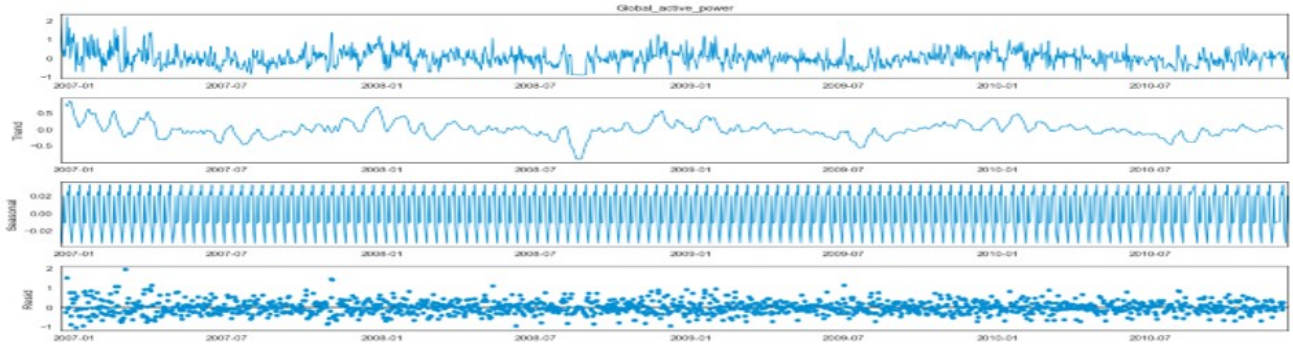

Figure 10: Time Series Component Breakdown for Global Active Power II

Global Reactive Power category visualize into different **Time Series** Components.

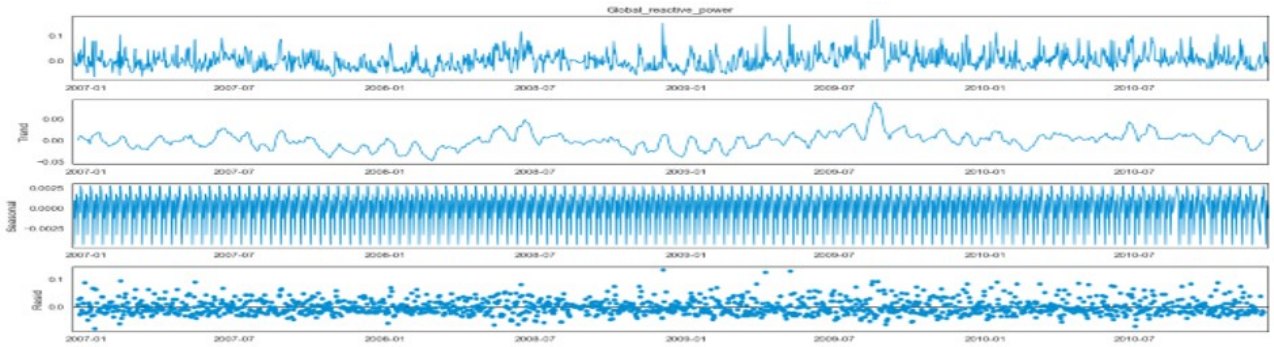

Figure 11: Time Series Component Breakdown for Global Reactive Power II

Voltage category visualize into different **Time Series** Components.

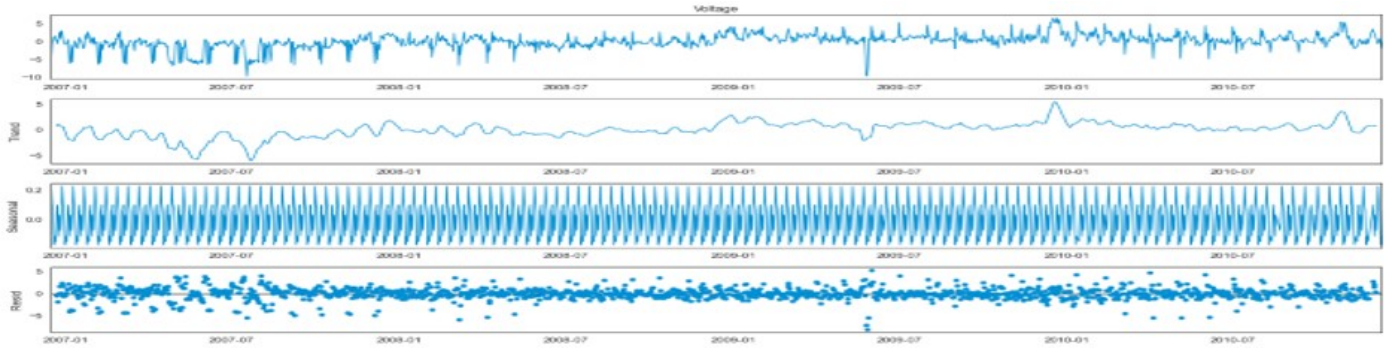

Figure 12: Time Series Component Breakdown for VoltageII

Global Intensity category visualize into different **Time Series** Components.

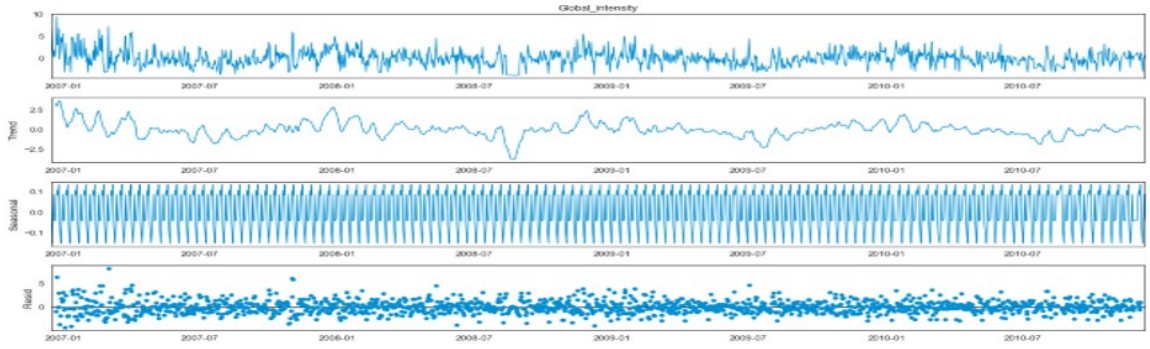

Figure 13: Time Series Component Breakdown for Global IntensityII

6 STATIONARITY CHECKING

In a time series analysis, A **stationary time series** is one whose properties **do not depend on the time** at which the series is observed. Thus, **time series with trends, or with seasonality, are not stationary** — the trend and seasonality will affect the value of the time series at different times.**Summary statistics** calculated on the time series are **consistent over time**, like the mean or the variance of the observations. In statistics, the **Dickey–Fuller test** tests **the null hypothesis that a unit root is present in an autoregressive model**. The **alternative hypothesis** is different depending on which version of the test is used, but is usually **stationarity or trend-stationarity**.

H0=non-stationary process VS H1=stationary process.
If **p value greater than alpha**,fail to reject H0,i.e., **non-stationary process** and **differencing(d)** has to be done to on the dataset.
If **p value less than alpha**,reject H0,i.e., **stationary process**.

Augmented Dickey-Fuller Test on "Global Active Power"

Null Hypothesis: Data has unit root. Non-Stationary.
Significance level = 0.05
Test Statistics = -8.3277
No. Lags Chosen = 9
Critical value at 0.01 = -3.435
Critical value at 0.05 = -2.864
Critical value at 0.1 = -2.568
•**P-Value = 0.0. Rejecting Null Hypothesis.**
•**Series is Stationary.**

Augmented Dickey-Fuller Test on "Global Reactive Power"

Null Hypothesis: Data has unit root. Non-Stationary.
Significance level = 0.05
Test Statistics =-6.7658
No. Lags Chosen = 13
Critical value at 0.01 = -3.435
Critical value at 0.05 = -2.864
Critical value at 0.1 = -2.568
•**P-Value = 0.0. Rejecting Null Hypothesis.**
•**Series is Stationary.**

Augmented Dickey-Fuller Test on "Voltage"

Null Hypothesis: Data has unit root. Non-Stationary.
Significance level = 0.05
Test Statistics = -5.4486
No. Lags Chosen = 11
Critical value at 0.01 = -3.435
Critical value at 0.05 = -2.864
Critical value at 0.1 = -2.568
•**P-Value = 0.0. Rejecting Null Hypothesis.**
•**Series is Stationary.**

Augmented Dickey-Fuller Test on "Global Intensity"

Null Hypothesis: Data has unit root. Non-Stationary.
Significance level = 0.05
Test Statistics = -8.3961
No. Lags Chosen = 9
Critical value at 0.01 = -3.435
Critical value at 0.05 = -2.864
Critical value at 0.1 = -2.568
•**P-Value = 0.0. Rejecting Null Hypothesis.**
•**Series is Stationary.**

7 ACF PACF PLOTS

Autocorrelation Function(ACF) and **Partial Autocorrelation Function(PACF)** are two important parts in Time Series analysis.**ACF plot** is merely a bar chart of the coefficients of correlation between a time series and lags of itself.The **PACF plot** is a plot of the partial correlation coefficients between the series and lags of itself.
ACF and **PACF plots** allow one to determine if there is any **correlation between time series and its lag** or **dependence of time series and its lag over time.**

From the **acf pacf plot of Global Active Power**, we can see that **the lag points are correlated between themselves and they share a relationship.**

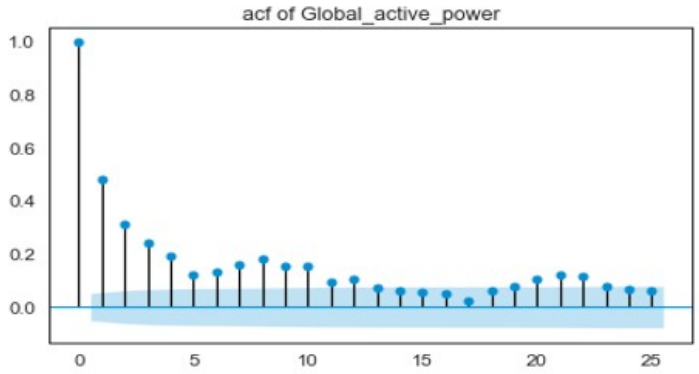

Figure 14: acf plot of Global Active Power

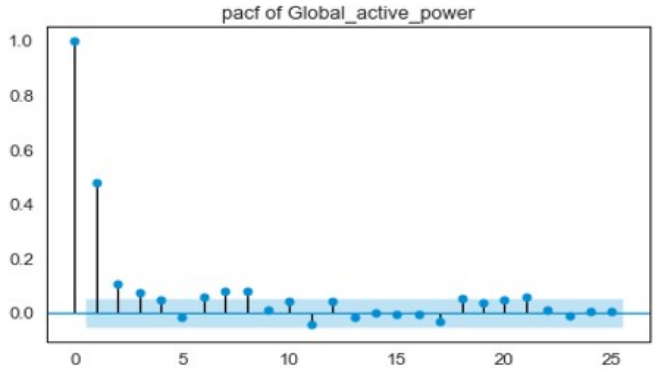

Figure 15: pacf plot of Global Active Power

From the **acf pacf plot of Global Reactive Power**, we can see that **the lag points are correlated between themselves and they share a relationship.**

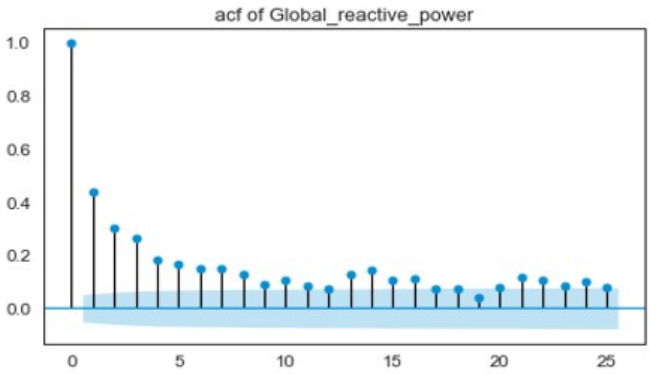

Figure 16: acf plot of Global Reactive Power

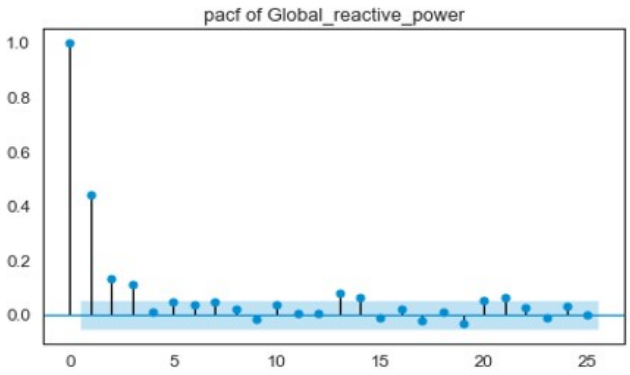

Figure 17: pacf plot of Global Reactive Power

From the **acf pacf plot of Voltage**, we can see that **the lag points are correlated between themselves and they share a relationship.**

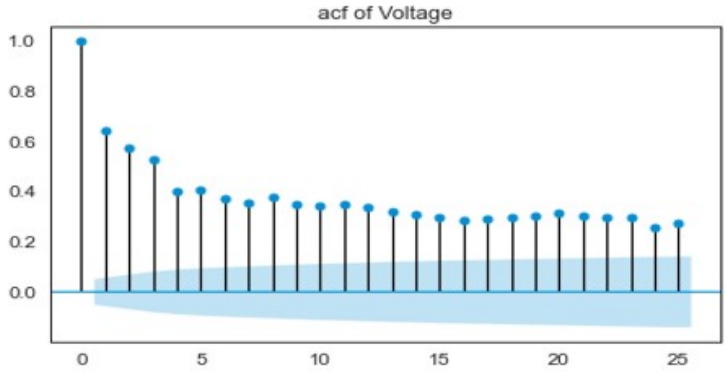

Figure 18: acf plot of Voltage

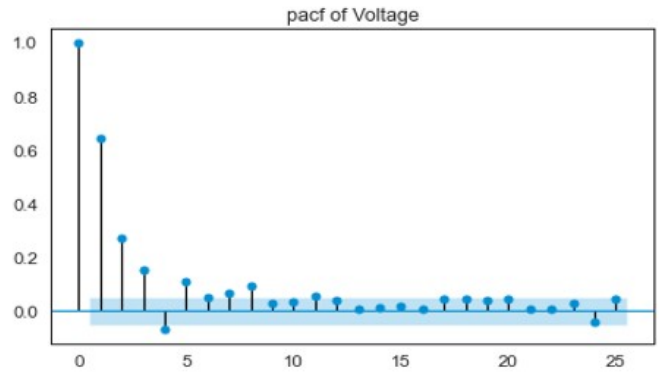

Figure 19: pacf plot of Voltage

From the **acf pacf plot of Global Intensity**, we can see that **the lag points are correlated between themselves and they share a relationship.**

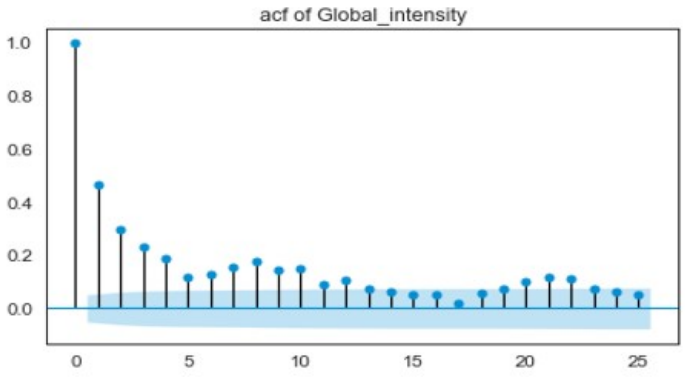

Figure 20: acf plot of Global Intensity

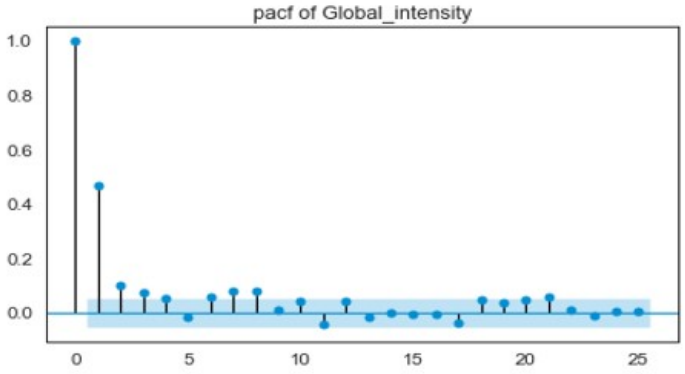

Figure 21: pacf plot of Global Intensity

8 CAUSALITY CHECKING OF TIME SERIES

Causality concerns relationships where **a change in one variable necessarily results in a change in another variable**. There are **three conditions for causality: covariation, temporal precedence, and control for "third variables".** The latter comprise alternative explanations for the observed causal relationship.

The Granger causality test is a statistical hypothesis test for determining whether one time series is useful for forecasting another. If probability value is less than any level, then the hypothesis would be rejected at that level. Granger causality is a statistical concept of causality that is based on prediction. According to Granger causality, **if a signal X1 "Granger-causes" (or "G-causes") a signal X2, then past values of X1 should contain information that helps predict X2 above and beyond the information contained in past values of X2 alone.**

When time series X Granger-causes time series Y, the patterns in X are approximately repeated in Y after some time lag. Thus, past values of X can be used for the prediction of future values of Y. A time series X is said to Granger-cause Y if it can be shown, usually through a series of t-tests and F-tests on lagged values of X (and with lagged values of Y also included), that those X values provide statistically significant information about future values of Y.

H0: Xt does not granger causes Yt VS H1: Xt granger causes Yt. If **p value more than alpha(0.05),fail to reject H0,i.e., not have granger causes** . If **p value less than alpha(0.05),reject H0,i.e., have granger causes.**

Granger Causality Test is one-way test. Here,we test on X variable granger causes Y but not the other way round. And so,for that we should keep in mind the order we put random variables in. The first one put, will be Y,dependent variable and second one put, will be X,independent variable,which will granger-cause.

We have divided dataset into 2 parts, i.e., **Train Set and Test Set.**
Test Set has last 15 variables and Train Set has the rest of the dataset. It will help us in future to see how good our forecasting is and to verify the prediction results.
We will apply **Granger Causality Test on the Train Dataset and will see the output**:

Variable	GlobalActive(x)	GlobalReactive(x)	Voltage(x)	GlobalIntensity(x)
GlobalActive(y)	1.000	0.000	0.000	0.000
GlobalReactive(y)	0.000	1.000	0.000	0.000
Voltage(y)	0.000	0.006	1.000	0.000
GlobalIntensity(y)	0.000	0.000	0.000	1.000

The above table can be read like this, as:
The row are the Response (y) and the columns are the predictor series (x).
For example, if we take the value **0.0060 in (row 3, column 2)**, it refers to the **p-value of GlobalReactive(x) causing Voltage(y)**. Whereas, the value **0.0000 in (row 2, column 1)** refers to the **p-value of GlobalActive(x) causing GlobalReactive(y).**

If a given p-value is less than significance level (0.05), then, the corresponding x series (column) causes the y (row).
For example, p-value of 0.0060 at (row 3, column 2) represents the p-value of the Grangers Causality test for GlobalReactive(x) causing Volatge(y), which is less that the significance level of 0.05.
So, we can **reject the null hypothesis and conclude GlobalReactive(x) causes Voltage(y).**
Looking at the p-values in the above table, it is pretty much clear that all the variables (time series) in the system are interchangeably causing each other. This makes this system of multi time series a good candidate for using **VAR models to forecast.**

9 COINTEGRATION TEST

Cointegration Test helps to establish **the presence of a statistically significant connection between two or more time series.**
To understand Cointegration, we have to understand **'order of integration' (d)** first.
Order of integration(d) is nothing but the number of differencing required to make a non-stationary time series stationary.
Now, when we have two or more time series, and there exists a linear combination of them that has an order of integration (d) less than that of the individual series, then the collection of series is said to be cointegrated.
When two or more time series are cointegrated, it means they in the long run, have a statistically significant relationship.
This is the basic premise on which Vector Autoregression(VAR) models is based on. So, **it's necessary to implement the cointegration test before starting to build VAR models.**
Soren Johanssen devised a procedure to implement the cointegration test.

Name	Test Statistic greater than C(0.95)	Signif
Global Active Power	416.97 greater than 40.1749	**True**
Global Reactive Power	273.07 greater than 24.2761	**True**
Voltage	151.82 greater than 12.3212	**True**
Global Intensity	53.27 greater than 4.1296	**True**

The above shows that all **four(4) variables have statistically significant relationship between themselves** and can be used to **build VAR model.**

10 VECTOR AUTOREGRESSION MODEL(VAR MODEL)

Vector Autoregression (VAR) is a multivariate forecasting algorithm that is used when two or more time series influence each other.
That means, the **basic requirements** in order to use VAR are:
1)There should be at least two time series (variables).
2)The time series should influence each other.

The reason behind it is called **'Autoregressive'** is that, each variable (Time Series) is modeled as a function of the past values, that is the predictors are nothing but the lags (time delayed value) of the series.
Difference between VAR and other Autoregressive models like AR, ARMA or ARIMA is that, the latter Autoregressive models are **uni-directional**, where, the predictors influence the Y and not vice-versa. Whereas, Vector Autoregression (VAR) is **bi-directional**. That is, the variables influence each other.

11 FORECASTING WITH VAR MODEL

In **Autoregression models**, the time series is modeled as a **linear combination of it's own lags**. That is, the past values of the series are used to forecast the current and future.

A typical **AR(p) model equation** looks something like this:

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \cdots + \beta_p Y_{t-p} + \epsilon_t$$

Figure 22: AR model order p

where **a is the intercept**,a constant term of the equation and **b1, b2 are the coefficients of the lags of Y till order p**.
Order 'p' means, up to p-lags of Y is used and they are the predictors in the equation. The **et is the error**, which is considered as white noise.

So, how does a **VAR model** look like?
In the **VAR model**, each variable is modeled as a linear combination of past values of itself and the past values of other variables in the system. Since we have multiple time series that influence each other, it is **modeled as a system of equations with one equation per variable (time series).**
That is, if **we have 4 time series that influence each other, we will have a system of 4 equations.**

How are the equations exactly framed then?
Let's suppose, we have two variables (Time series) Y1 and Y2, and we need to forecast the values of these variables at time (t). To calculate Y1(t), VAR will use the past values of both Y1 as well as Y2. Likewise, to compute Y2(t), the past values of both Y1 and Y2 be used.
For example, the system of equations for a VAR(1) model with two time series (variables 'Y1' and 'Y2') are as follows:

$$Y_{1,t} = \alpha_1 + \beta_{11,1} Y_{1,t-1} + \beta_{12,1} Y_{2,t-1} + \epsilon_{1,t}$$
$$Y_{2,t} = \alpha_2 + \beta_{21,1} Y_{1,t-1} + \beta_{22,1} Y_{2,t-1} + \epsilon_{2,t}$$

Figure 23: VAR model of order 1

Where, **Y1,t-1 and Y2,t-1 are the first lag of time series Y1 and Y2 respectively.**
The above equations are referred to as a **VAR(1) model**, because, each equation is of order 1, that is, it contains up to one lag of each of the predictors (Y1 and Y2).
Likewise, the **second order VAR(2) model** for two variables would include up to two lags for each variable (Y1 and Y2).

$$Y_{1,t} = \alpha_1 + \beta_{11,1} Y_{1,t-1} + \beta_{12,1} Y_{2,t-1} + \beta_{11,2} Y_{1,t-2} + \beta_{12,2} Y_{2,t-2} + \epsilon_{1,t}$$
$$Y_{2,t} = \alpha_2 + \beta_{21,1} Y_{1,t-1} + \beta_{22,1} Y_{2,t-1} + \beta_{21,2} Y_{1,t-2} + \beta_{22,2} Y_{2,t-2} + \epsilon_{2,t}$$

Figure 24: VAR model of order 2

12 ORDER SELECTION (p) OF VAR MODEL

Select the best order of equation for VAR Model is of most importance. As the forecasting of time series is depending on this order of equation, we should choose the **order (p) of VAR Model** quite carefully.
To select the **right order of the VAR model**, we iteratively fit increasing orders of VAR model and pick the order that gives a model with **the least AIC**. Though the usual practice is to look at the AIC, we can also check other **best fit** comparison estimates of **BIC, AICC**.

Lag Order	**AIC**	**BIC**
1	14.833526973001838	14.75996793183535
2	14.932116142386851	14.799634915228198
3	14.989432711642113	14.797962539887424
4	15.009971919719641	14.75944592104978
5	15.037293811318353	14.727644979383086
6	15.05452804648703	14.685689250588164
7	15.077943267918107	14.649847252691591
8	**15.108313391671755**	**14.620892776768763**
9	15.10910005138919	14.562287331156153
10	15.10977474292819	14.503502286085789

In the above table, we can see that the **AIC converges at lag 8**, **the BIC is also amongst the lowest there**. So we will build our **VAR Model with lag 8** and will forecast time series.

13 FITTING OF VAR MODEL

After the selection of **best order (p) of equation for VAR Model**, the fitting is done on the model and we get **set of equations for each time series combining the other time series as well.**

```
In [39]: res = model.fit(8)
         print(res.summary())
```

```
  Summary of Regression Results
==================================
Model:                         VAR
Method:                        OLS
Date:           Thu, 24, Dec, 2020
Time:                     20:04:25
--------------------------------------------------------------------
No. of Equations:         4.00000    BIC:                   -14.6209
Nobs:                     1425.00    HQIC:                  -14.9263
Log likelihood:           2808.72    FPE:                2.74509e-07
AIC:                     -15.1083    Det(Omega_mle):     2.50488e-07
--------------------------------------------------------------------
Results for equation Global_active_power
================================================================================
                               coefficient       std. error           t-stat            prob
--------------------------------------------------------------------------------
const                           -0.002929         0.008960           -0.327           0.744
L1.Global_active_power          -0.903843         1.001630           -0.902           0.367
L1.Global_reactive_power        -2.337874         0.410893           -5.690           0.000
L1.Voltage                       0.040501         0.007625            5.311           0.000
L1.Global_intensity              0.321405         0.241658            1.330           0.184
L2.Global_active_power           0.019185         1.021765            0.019           0.985
L2.Global_reactive_power         0.128723         0.443900            0.290           0.772
L2.Voltage                      -0.001915         0.008247           -0.232           0.816
L2.Global_intensity             -0.000741         0.246549           -0.003           0.998
L3.Global_active_power          -0.601869         1.023797           -0.588           0.557
L3.Global_reactive_power        -0.275415         0.444478           -0.620           0.535
L3.Voltage                       0.017144         0.008364            2.050           0.040
L3.Global_intensity              0.157814         0.247090            0.639           0.523
L4.Global_active_power          -0.239091         1.026017           -0.233           0.816
L4.Global_reactive_power        -0.023030         0.446414           -0.052           0.959
L4.Voltage                      -0.020843         0.008373           -2.489           0.013
L4.Global_intensity              0.068088         0.247621            0.275           0.783
L5.Global_active_power           0.041839         1.024305            0.041           0.967
L5.Global_reactive_power         0.907034         0.447669            2.026           0.043
L5.Voltage                       0.010648         0.008366            1.273           0.203
L5.Global_intensity             -0.018984         0.247208           -0.077           0.939
L6.Global_active_power           3.171028         1.016536            3.119           0.002
L6.Global_reactive_power         0.876879         0.445166            1.970           0.049
L6.Voltage                      -0.039434         0.008360           -4.717           0.000
L6.Global_intensity             -0.762440         0.245335           -3.108           0.002
L7.Global_active_power          -2.459202         1.013622           -2.426           0.015
L7.Global_reactive_power        -0.330166         0.444411           -0.743           0.458
L7.Voltage                      -0.000880         0.008332           -0.106           0.916
L7.Global_intensity              0.596744         0.244606            2.440           0.015
L8.Global_active_power          -0.936331         0.977343           -0.958           0.338
L8.Global_reactive_power        -1.282861         0.411290           -3.119           0.002
L8.Voltage                       0.021824         0.007802            2.797           0.005
L8.Global_intensity              0.252978         0.235794            1.073           0.283
================================================================================
```

Figure 25: model fitting of VAR model with lag 8

We have got **Results for equations for other three(3) variables** as well. As the **lag choosen here is 8, we get coefficients upto that order for each variable contributing and a constant term as well.**

14 CHECK FOR SERIAL CORRELATION OF RESIDUALS

After **the fitting of VAR Model**, we can check for **Serial Correlation of Residuals (Errors) using Durbin Watson Statistic.**
Serial correlation of residuals is used to check if there is any leftover pattern in the residuals (errors).

What does that mean?
If there is any correlation left in the residuals, then, there is some pattern in the time series that is still left to be explained by the model. In that case, the typical course of action is to either increase the order of the model or induce more predictors into the system or look for a different algorithm to model the time series.
So, **checking for serial correlation** is to ensure that the model is sufficiently able to explain the variances and patterns in the time series.
A common way of **checking for serial correlation of errors** can be measured using the **Durbin Watson's Statistic.**

$$DW = \frac{\sum_{t=2}^{T}((e_t - e_{t-1})^2)}{\sum_{t=1}^{T} e_t^2}$$

Figure 26: Durbin Watson's Statistic

The value of this **statistic can vary between 0 and 4.**
The closer it is to the value 2, then there is no significant serial correlation. The closer to 0, there is a positive serial correlation, and **the closer it is to 4 implies negative serial correlation.**

RESULTS:

Global Active Power : 1.99
Global Reactive Power : 2.0
Voltage : 2.0
Global Intensity : 1.99

So, we can now say that there is **no significant relatinship left between the variables** and all have been exlpored by our **VAR model.**

15 FORECASTING RESULTS WITH VAR MODEL

In order to forecast, the **VAR model** expects up to the lag order number of observations from the past data. This is because, the terms in the VAR model are essentially the lags of the various time series in the dataset, so we need to provide it as many of the previous values as indicated by the lag order used by the model.

```
In [41]: pred = res.forecast(df_zm.values, 15)
         pred_df = pd.DataFrame(pred, index=df.index[-15:], columns = df.columns)
         pred_df
```

Out[41]:

date_time	Global_active_power	Global_reactive_power	Voltage	Global_intensity
2010-11-23	0.1169	-0.0078	-0.2699	0.4802
2010-11-24	0.1430	-0.0029	-0.9932	0.6084
2010-11-25	0.0304	0.0023	-0.6088	0.1426
2010-11-26	-0.1233	0.0006	-0.2559	-0.4986
2010-12-01	-0.1826	-0.0043	-0.4641	-0.7518
2010-12-02	-0.0286	0.0011	-0.3883	-0.1067
2010-12-03	0.0030	0.0004	-0.4599	0.0285
2010-12-04	0.0041	-0.0033	-0.4707	0.0290
2010-12-05	0.0324	-0.0015	-0.2921	0.1411
2010-12-06	0.0173	-0.0016	-0.4084	0.0819
2010-12-07	0.0052	-0.0013	-0.3346	0.0301
2010-12-08	-0.0101	-0.0014	-0.2890	-0.0354
2010-12-09	-0.0297	-0.0014	-0.3224	-0.1162
2010-12-10	-0.0160	-0.0002	-0.2840	-0.0589
2010-12-11	-0.0141	-0.0006	-0.2890	-0.0508

Figure 27: Forecasting with VAR model

The forecasts are generated but it is on the scale of the training data used by the model. So, we have to bring it back up to its original scale, which our input dataset has.
For that we have to **inverse the forecasts that we get from the training data and then sum it up with the test data**, so that we can find **our actual prediction of our time series dataset.**

```
In [42]: pred_inverse = pred_df.cumsum()

         # inverse the difference values and assigning to variable 'f'
         f = pred_inverse + X_test
         print(f)
```

```
            Global_active_power  Global_reactive_power  Voltage  \
date_time
2010-11-23               0.1204                -0.0361  -0.5308
2010-11-24               0.4153                -0.0425  -2.0686
2010-11-25               0.1921                -0.0516  -1.1721
2010-11-26               0.2532                -0.0360  -2.6732
2010-12-01               0.3286                 0.0122   0.8229
2010-12-02               0.1784                -0.0430  -0.1903
2010-12-03               0.0095                -0.0312  -1.2629
2010-12-04              -0.1295                -0.0282  -2.7408
2010-12-05               0.3167                -0.0282  -3.3227
2010-12-06               0.2917                 0.0480  -5.0939
2010-12-07              -0.3037                 0.0565  -5.5636
2010-12-08              -0.7164                -0.0260  -5.2764
2010-12-09               0.0056                -0.0050  -4.7740
2010-12-10              -0.0330                -0.0268  -6.4170
2010-12-11               0.1315                -0.0407  -8.1288

            Global_intensity
date_time
2010-11-23            0.5249
2010-11-24            1.7085
2010-11-25            0.7710
2010-11-26            1.0592
2010-12-01            1.3274
2010-12-02            0.6505
2010-12-03           -0.0068
2010-12-04           -0.5367
2010-12-05            1.3304
2010-12-06            1.3697
2010-12-07           -1.0506
2010-12-08           -2.8594
2010-12-09            0.1563
2010-12-10            0.0231
2010-12-11            0.7257
```

Figure 28: Getting actual results of Predictions

The forecasts are back to the original scale. Now we can **visualize them** and also will **evaluate them to see how good they are.**

16 VISUALIZE THE FORECASTING RESULTS

After **the fitting of VAR Model and forecasting of our training dataset**, we should **visualize the forecasting results** and see how they are looking like in comparison of actual data.

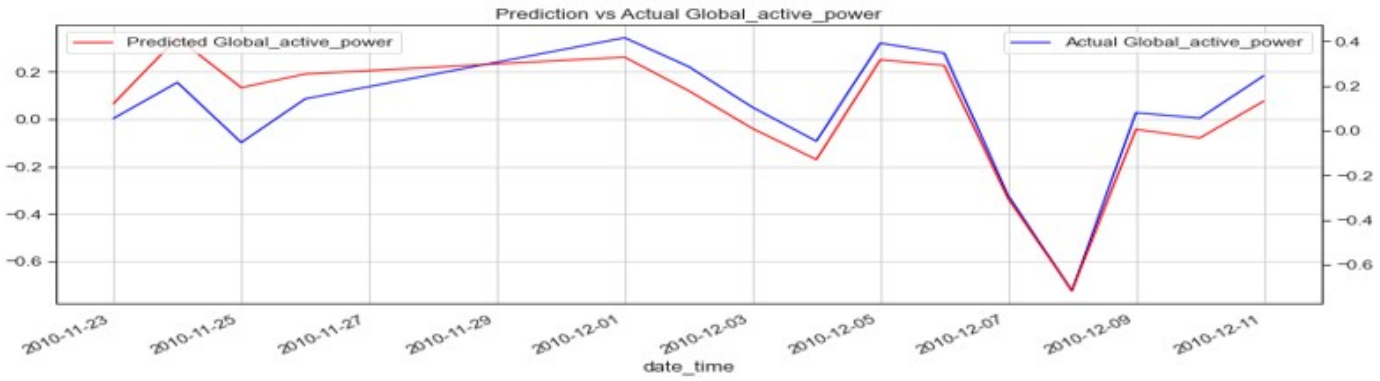

Figure 29: Visualization of results for Global Active Power

From the above figure, we can see that **the Predicted and Actual of Global Active Power is quite close to each other and the model is fitted well.**

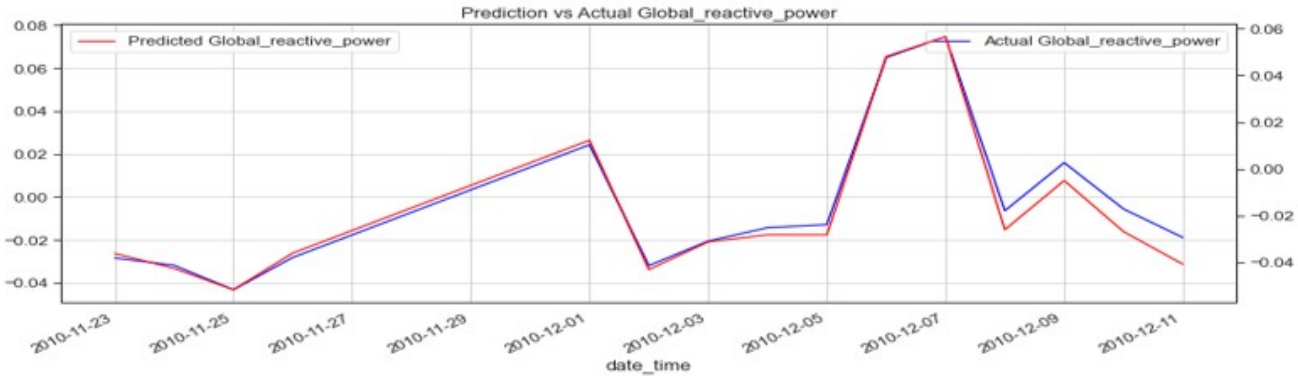

Figure 30: Visualization of results for Global Reactive Power

From the above figure, we can see that **the Predicted and Actual of Global Reactive Power is quite close to each other and the model is fitted well.**

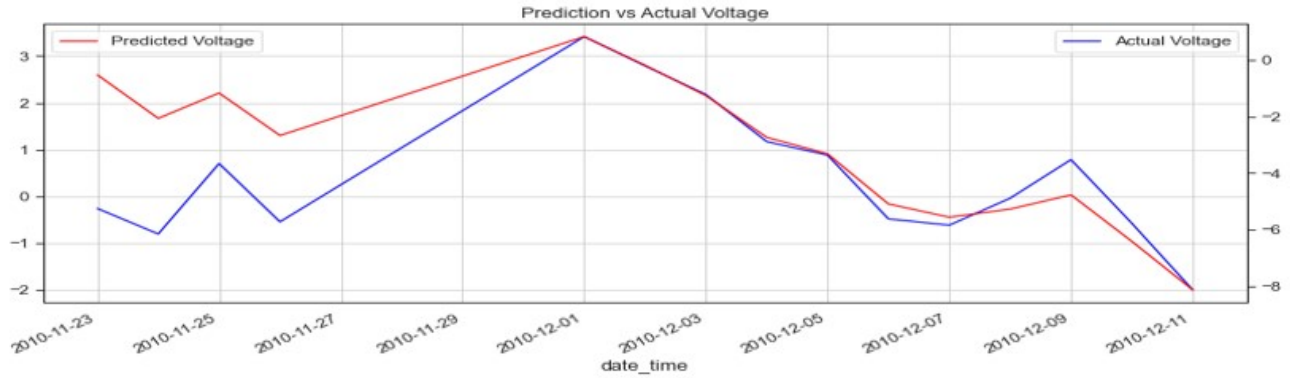

Figure 31: Visualization of results for Voltage

From the above figure, we can see that **the Predicted and Actual of Voltage is not that much of close upto certain point of time due to may be some error but after that both the curves are almost overlapping each other. So we can say that the model is fitted well.**

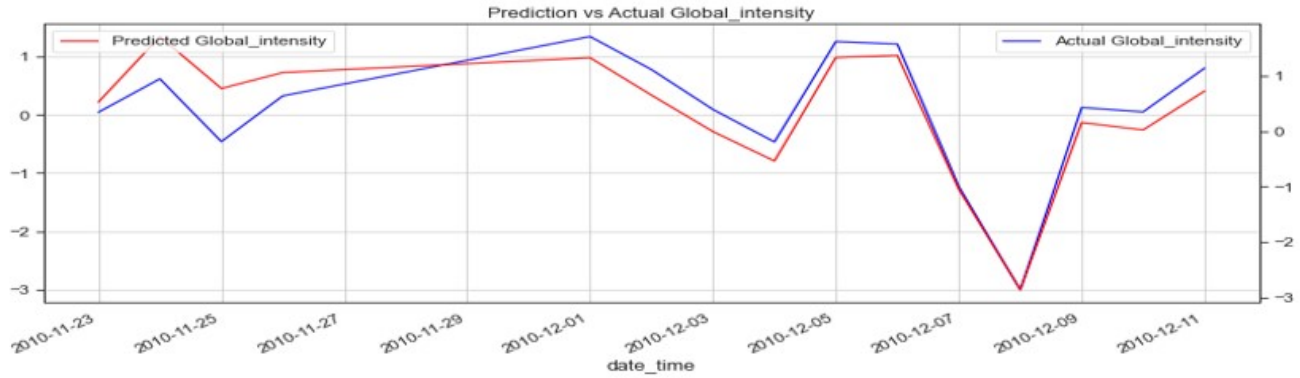

Figure 32: Visualization of results for Global Intensity

From the above figure, we can see that **the Predicted and Actual of Global Intensity is quite close to each other and the model is fitted well.**

17 EVALUATE THE FORECASTING RESULTS

At the very end, we should **evaluate our Forecasting Results**, so that we can understand how good our fitted model is and how good forecasting is done. For the evaluation, we will use **three(3)** measures, namely — **Mean Absolute Error(MAE), Mean Squared Error(MSE)** and **Root Mean Squared Error(RMSE).**

Catgeory	**MAE**	**MSE**	**RMSE**
Global Active Power	0.075117	0.013589	0.116574
Global Reactive Power	0.014438	0.000233	0.015253
Voltage	3.665285	16.386988	4.048084
Global Intensity	0.302574	0.239380	0.489265

From the table, we can say that **our model fitting is quite good** for **Global Active Power, Global Reactive Power, Global Intensity** and the **forecasting results are quite close with the actual values of time series.** For **Voltage**, **there is error** and that is due to the model is unable to fit it properly at the early stages, **the model does well at later stages** and that is why **it also fitts well and has quite small amount of error,** which is marginal.

18 CONCLUSION

In **Univariate Time Series Analysis**, Conventional Forecasting Approaches are good, they are suitable enough. But for **Multivariate Time Series Analysis**, this kind of approach should be taken. In **Univariate**, focus is mainly on one(1) variable, we do forecasting and prediction, keeping in mind the effect of that variable only in the time span. But in **Multivariate**, our approach is to do forecasting and prediction of one(1) variable considering not only that variable but also other associated variables as well. So, it is more suitable and appropiate enough in real life situation.

We often forget to implement the effect of one variable on another in case of prediction. But **VAR model** keeps that in mind and predicts variable accordingly. Inter connecting variables where **they are correlated, past values of them effect the others and it happens for all**. So it is not simple case of forecasting, prediction here becomes complex and **VAR** has its very important role to play.