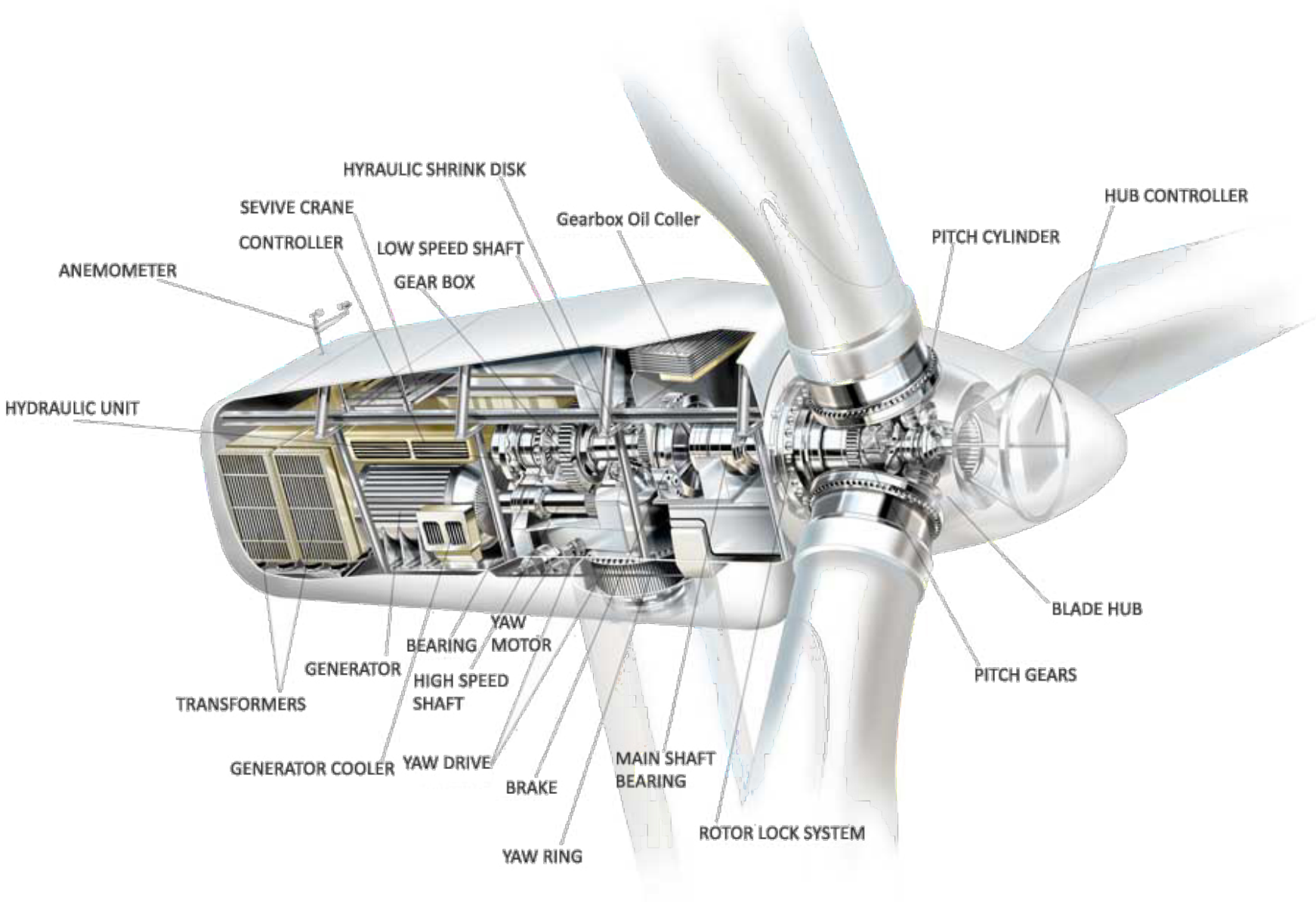
KNOW YOUR DATASET

In [1]:
```
#!pip install pillow
from PIL import Image
#!pip install requests
import requests
from io import BytesIO

response = requests.get('https://www.renewableenergyhub.co.uk/images/design/pages/wind-turbine-diagram.png')
Image.open(BytesIO(response.content))
```

Out[1]:



WIND TURBINE GENERATOR

---- ¶

WTG PRINCIPLE

Wind turbines work on a simple principle: instead of using electricity to make wind—like a fan—wind turbines use wind to make electricity. Wind turns the propeller-like blades of a turbine around a rotor, which spins a generator, which generates electricity

---

KNOW ABOUT WIND TURBINE PARTS

NACELLE

Sits atop the tower and contains the gear box, low- and high-speed shafts, generator, controller, and brake. Some nacelles are large enough for a helicopter to land on.

BLADES

Lifts and rotates when wind is blown over them, causing the rotor to spin. Most turbines have either two or three blades.

ROTOR

Blades and hub together form the rotor. Used to pass the Rotating energy from blades to Gearbox Low Speed Shaft.

LOW-SPEED SHAFT

It is connected rotor and used to Turns the low-speed shaft at about 30-60 rpm depends on Gearbox.

GEAR BOX

Connects the low-speed shaft to the high-speed shaft and increases the rotational speeds from about 30-60 rotations per minute (rpm), to about 1,000-1,800 rpm; this is the rotational speed required by most generators to produce electricity. The gear box is a costly (and heavy) part of the wind turbine.

HIGH-SPEED SHAFT

It is connected between Gearbox and Generator to Drive the generator at High speed to generate the electricity.

GENERATOR

Produces 50-cycle AC electricity; it is usually an Distributed Frequency induction generator (DFIG).

ANEMOMETER

Measures the wind speed and transmits wind speed data to the controller.

PITCH SYSTEM

Turns (or pitches) blades out of the wind to control the rotor speed, and to keep the rotor from turning in winds that are too high or too low to produce electricity.

WIND VANE

Measures wind direction and communicates with the yaw drive to orient the turbine properly with respect to the wind.

YAW DRIVE

Orients upwind turbines to keep them facing the wind when the direction changes. Downwind turbines don't require a yaw drive because the wind manually blows the rotor away from it.

TRANSFORMER

Transformer are used to transform the electrical voltage from one level to another level.

DATA DESCRIPTION

The dataset show the information of a single turbine generator in 10 minutual slots from last few years as shown below. The data says that the Turbine Generators Parameters like Power, Temperatures, Speeds and Position during Operation.

From this i am going to "Predict the Wind Speed trend" for next time slot and Checking the "relationship between the Average Active Power and Other Parameters for applying Linear Regression".

Variable	Definition	Data Type
Date	Date of a Observation Created	Format (DD-MM-YYYY)
Time	10 Minutes Time Slot of a Observation Created	Format (HH:MM:SS)
Avg_Active_Power	Average Power Generated by a turbine in a 10 Minute time slot	Integer
Avg_Ambient_Temp	Average Ambient Temperature Measured by a turbine during a 10 Minute time slot	Integer
Avg_Generator_Speed	Average Generator Speed Measured by a turbine during a 10 Minute time slot	Integer
Avg_Nacelle_Pos	Average Nacelle Position of a turbine as per wind direction during a 10 Minute time slot	Integer
Avg_Pitch_Angle	Average Pitch Angle of a turbine as per wind Speed during a 10 Minute time slot	Integer
Avg_Rotor_Speed	Average Rotor Speed of a turbine as per wind Speed during a 10 Minute time slot	Integer
Avg_Wind_Speed	Average Wind Speed of a turbine during a 10 Minute time slot	Integer
Bearing_DE_Temp	Average Bearing Drive End Temperature of a turbine during a 10 Minute time slot	Integer
Bearing_NDE_Temp	Average Non Bearing Drive End Temperature of a turbine during a 10 Minute time slot	Integer
Gearbox_bearing_Temp	Average Gearbox Bearing Temperature of a turbine during a 10 Minute time slot	Integer

Gearbox_oil_Temp	Average Gearbox Oil Temperature of a turbine during a 10 Minute time slot	Integer
Generator_winds_Temp_1	Average Generator winding R Phase Temperature of a turbine during a 10 Minute time slot	Integer
Generator_winds_Temp_2	Average Generator winding Y Phase Temperature of a turbine during a 10 Minute time slot	Integer
Generator_winds_Temp_3	Average Generator winding B Phase Temperature of a turbine during a 10 Minute time slot	Integer
Generator�s_sliprings_Temp	Average Generator's sliprings Temperature of a turbine during a 10 Minute time slot	Integer
Hydraulic_group_pressure	Average Hydraulic Group Pressure of a turbine during a 10 Minute time slot	Integer
Nacelle_Misalignment_Avg_Wind_Dir	Average Nacelle Misalignment & Wind Direction of a turbine during a 10 Minute time slot	Integer
Trafo_1_wind_Temp	Average Transformer winding R Phase Temperature of a turbine during a 10 Minute time slot	Integer
Trafo_2_wind_Temp	Average Transformer winding Y Phase Temperature of a turbine during a 10 Minute time slot	Integer
Trafo_3_wind_Temp	Average Transformer winding B Phase Temperature of a turbine during a 10 Minute time slot	Integer

IMPORT REQUIRED PACKAGES(LIBRARIES)

In [1]:
```
# !pip install pandas
import pandas as pd
# !pip install numpy
import numpy as np
# !pip install seaborn
import seaborn as sns
# !pip install matplotlib
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
# !pip install ipywidgets
import ipywidgets as widgets
from matplotlib.cm import get_cmap
```

IMPORT DATASET

Import dataset from google drive and assigned the dataset as "df" ---- It will take few minutes based on internet because its file size is large.

In [2]:
```
URL = 'https://drive.google.com/file/d/1haTtjbR90YnbAGls8QVdREmHNmx1a5mQ/view?usp=sharing'
path = 'https://drive.google.com/uc?export=download&id='+URL.split('/')[-2]
```

In [3]:
```
%%time
print('*' * 127)
print('WIND TURBINE GENERATOR PARAMETER DATASET')
Master = pd.read_csv(path,low_memory = False, encoding = 'cp1252', parse_dates=['Date'], skipinitialspace=True)
print('*' * 127)
```

```
***************************************************************************************************************************
********
WIND TURBINE GENERATOR PARAMETER DATASET
***************************************************************************************************************************
********
Wall time: 15.5 s
```

In [4]:
```
df = Master
```

EXPLORATORY DATA ANALYSIS

CONFIRM THE DATA TRANSFER FROM GOOGLE DRIVE TO NOTEBOOK

In [5]:
```
print('First Five Records of the Dataset')
df.head()  # As default shows the top 5 Rows.
```

```
First Five Records of the Dataset
```

Out[5]:

	Date	Avg_Active_Power	Avg_Ambient_Temp	Avg_Generator_Speed	Avg_Nacelle_Pos	Avg_Pitch_Angle	Avg_Rotor_Speed	Avg_Wind_Speed	Bea
0	2016-08-01	1487.3	25.1	1654.9	138.0	0.03	15.5	9.3	
1	2016-08-01	1647.6	25.1	1677.8	138.0	-0.31	15.7	9.3	
2	2016-08-01	1506.4	25.2	1661.4	138.0	-0.48	15.6	9.0	
3	2016-08-01	1240.9	25.1	1601.4	138.0	-0.56	15.0	8.3	
4	2016-08-01	1202.4	25.1	1607.1	138.0	-0.73	15.1	8.5	

5 rows × 21 columns

In [6]:
```
print('Last Five Records of the Dataset')
df.tail()  # As default shows the Last 5 Rows.
```

```
Last Five Records of the Dataset
```

Out[6]:

	Date	Avg_Active_Power	Avg_Ambient_Temp	Avg_Generator_Speed	Avg_Nacelle_Pos	Avg_Pitch_Angle	Avg_Rotor_Speed	Avg_Wind_Speed
245194	2021-04-30	205.1	32.2	1051.2	1.0	1.24	9.9	4.7
245195	2021-04-30	208.4	32.1	1063.5	1.0	1.33	10.0	4.7
245196	2021-04-30	228.0	31.8	1074.9	355.7	1.32	10.1	5.2
245197	2021-04-30	355.1	31.7	1106.6	354.0	0.17	10.4	5.7
245198	2021-04-30	374.3	31.6	1119.2	0.0	0.12	10.5	5.9

5 rows × 21 columns

Check for the Shape

In [7]:
```
print("The Data Frame having the Rows of '{}' and Columns of '{}'".format (df.shape[0],df.shape[1]))
```

```
The Data Frame having the Rows of '245199' and Columns of '21'
```

Check for the Detailed Information of the Dataset

In [8]:
```
print('Total_Columns: ', len(df.columns),'\n')
print(df.columns,'\n')
print('Shape :',df.shape)
```

```
Total_Columns:  21

Index(['Date', 'Avg_Active_Power', 'Avg_Ambient_Temp', 'Avg_Generator_Speed',
       'Avg_Nacelle_Pos', 'Avg_Pitch_Angle', 'Avg_Rotor_Speed',
       'Avg_Wind_Speed', 'Bearing_DE_Temp', 'Bearing_NDE_Temp',
       'Gearbox_bearing_Temp', 'Gearbox_oil_Temp', 'Generator_wind_Temp_1',
       'Generator_wind_Temp_2', 'Generator_wind_Temp_3',
       'Generator’s_sliprings_Temp', 'Hidraulic_group_pressure',
       'Nacelle_Misalignment_Avg_Wind_Dir', 'Trafo_1_wind_Temp',
       'Trafo_2_wind_Temp', 'Trafo_3_wind_Temp'],
      dtype='object')

Shape : (245199, 21)
```

In [9]:
```
df.rename({"Generator�s_sliprings_Temp":"Generators_sliprings_Temp", "Generator’s_sliprings_Temp":"Generators_sliprings_
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 245199 entries, 0 to 245198
Data columns (total 21 columns):
 #   Column                            Non-Null Count   Dtype
---  ------                            --------------   -----
 0   Date                              245199 non-null  datetime64[ns]
 1   Avg_Active_Power                  236405 non-null  float64
 2   Avg_Ambient_Temp                  236405 non-null  float64
 3   Avg_Generator_Speed               236405 non-null  float64
 4   Avg_Nacelle_Pos                   236395 non-null  float64
 5   Avg_Pitch_Angle                   236405 non-null  float64
 6   Avg_Rotor_Speed                   236405 non-null  float64
 7   Avg_Wind_Speed                    236405 non-null  float64
 8   Bearing_DE_Temp                   236405 non-null  float64
 9   Bearing_NDE_Temp                  236404 non-null  float64
 10  Gearbox_bearing_Temp              236405 non-null  float64
 11  Gearbox_oil_Temp                  236405 non-null  float64
 12  Generator_wind_Temp_1             236402 non-null  float64
 13  Generator_wind_Temp_2             236402 non-null  float64
 14  Generator_wind_Temp_3             236402 non-null  float64
 15  Generators_sliprings_Temp         236402 non-null  float64
 16  Hidraulic_group_pressure          236405 non-null  float64
 17  Nacelle_Misalignment_Avg_Wind_Dir 236405 non-null  float64
 18  Trafo_1_wind_Temp                 236402 non-null  float64
 19  Trafo_2_wind_Temp                 236401 non-null  float64
 20  Trafo_3_wind_Temp                 236401 non-null  float64
dtypes: datetime64[ns](1), float64(20)
memory usage: 39.3 MB
```

CHANGE THE DATA TYPE

In [11]:
```
df['Date'] = pd.to_datetime(df['Date'], errors='coerce')
```

Displays memory consumed by each column

In [12]:
```
print(df.memory_usage(),'\n')
print('Dataset uses {0} MB'.format(df.memory_usage().sum()/1024**2))
```

```
Index                                128
Date                             1961592
Avg_Active_Power                 1961592
Avg_Ambient_Temp                 1961592
Avg_Generator_Speed              1961592
Avg_Nacelle_Pos                  1961592
Avg_Pitch_Angle                  1961592
Avg_Rotor_Speed                  1961592
Avg_Wind_Speed                   1961592
Bearing_DE_Temp                  1961592
Bearing_NDE_Temp                 1961592
Gearbox_bearing_Temp             1961592
Gearbox_oil_Temp                 1961592
Generator_wind_Temp_1            1961592
Generator_wind_Temp_2            1961592
Generator_wind_Temp_3            1961592
Generators_sliprings_Temp        1961592
Hidraulic_group_pressure         1961592
Nacelle_Misalignment_Avg_Wind_Dir 1961592
Trafo_1_wind_Temp                1961592
Trafo_2_wind_Temp                1961592
Trafo_3_wind_Temp                1961592
dtype: int64

Dataset uses 39.285240173339844 MB
```

In [13]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 245199 entries, 0 to 245198
Data columns (total 21 columns):
 #   Column                          Non-Null Count   Dtype         
---  ------                          --------------   -----         
 0   Date                            245199 non-null  datetime64[ns]
 1   Avg_Active_Power                236405 non-null  float64       
 2   Avg_Ambient_Temp                236405 non-null  float64       
 3   Avg_Generator_Speed             236405 non-null  float64       
 4   Avg_Nacelle_Pos                 236395 non-null  float64       
 5   Avg_Pitch_Angle                 236405 non-null  float64       
 6   Avg_Rotor_Speed                 236405 non-null  float64       
 7   Avg_Wind_Speed                  236405 non-null  float64       
 8   Bearing_DE_Temp                 236405 non-null  float64       
 9   Bearing_NDE_Temp                236404 non-null  float64       
 10  Gearbox_bearing_Temp            236405 non-null  float64       
 11  Gearbox_oil_Temp                236405 non-null  float64       
 12  Generator_wind_Temp_1           236402 non-null  float64       
 13  Generator_wind_Temp_2           236402 non-null  float64       
 14  Generator_wind_Temp_3           236402 non-null  float64       
 15  Generators_sliprings_Temp       236402 non-null  float64       
 16  Hidraulic_group_pressure        236405 non-null  float64       
 17  Nacelle_Misalignment_Avg_Wind_Dir 236405 non-null  float64       
 18  Trafo_1_wind_Temp               236402 non-null  float64       
 19  Trafo_2_wind_Temp               236401 non-null  float64       
 20  Trafo_3_wind_Temp               236401 non-null  float64       
dtypes: datetime64[ns](1), float64(20)
memory usage: 39.3 MB
```

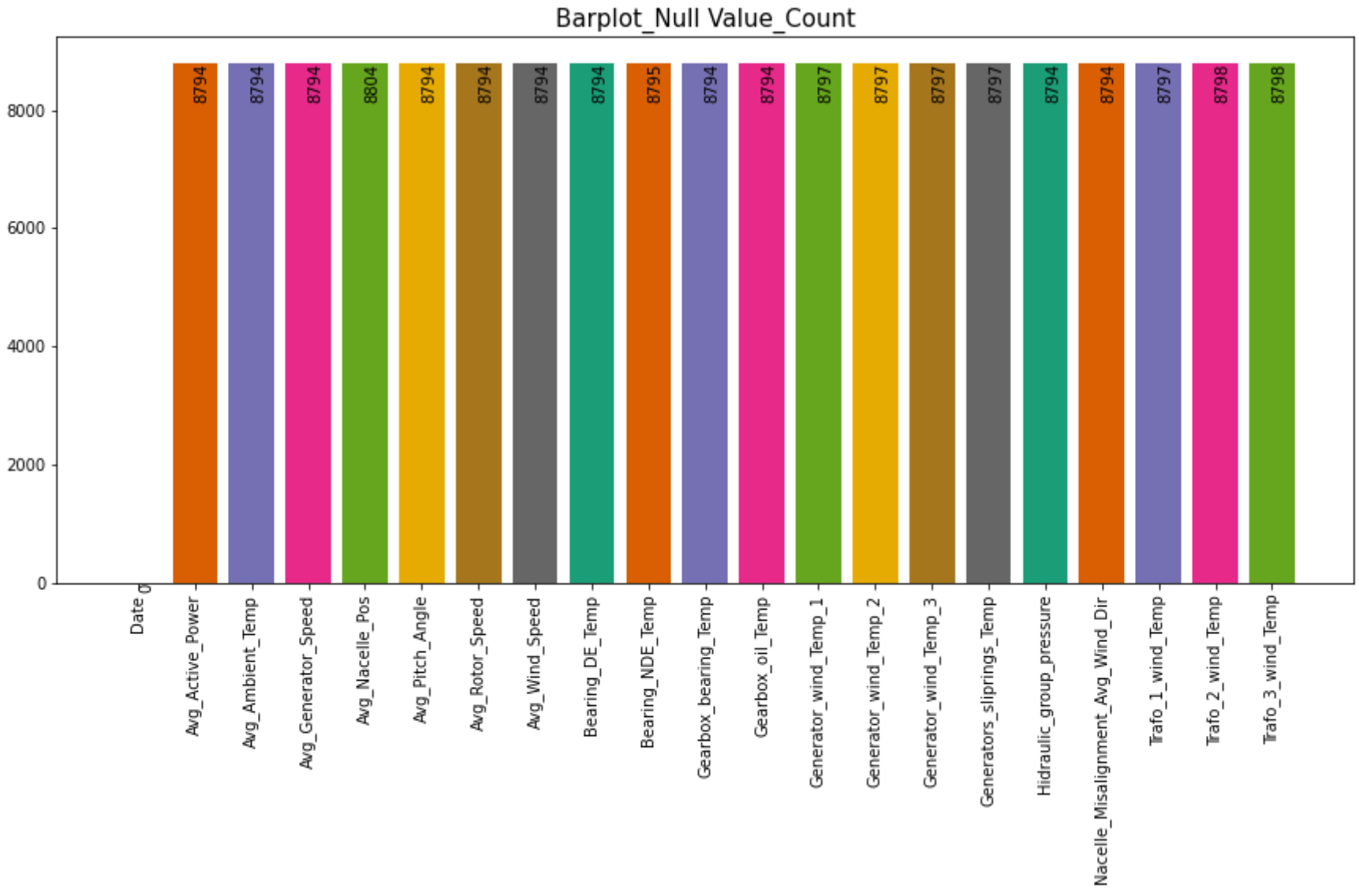
CHECK FOR NULL VALUES

In [14]:
```
df.isnull().sum()
```

Out[14]:
```
Date                                0
Avg_Active_Power                 8794
Avg_Ambient_Temp                 8794
Avg_Generator_Speed              8794
Avg_Nacelle_Pos                  8804
Avg_Pitch_Angle                  8794
Avg_Rotor_Speed                  8794
Avg_Wind_Speed                   8794
Bearing_DE_Temp                  8794
Bearing_NDE_Temp                 8795
Gearbox_bearing_Temp             8794
Gearbox_oil_Temp                 8794
Generator_wind_Temp_1            8797
Generator_wind_Temp_2            8797
Generator_wind_Temp_3            8797
Generators_sliprings_Temp        8797
Hidraulic_group_pressure         8794
Nacelle_Misalignment_Avg_Wind_Dir    8794
Trafo_1_wind_Temp                8797
Trafo_2_wind_Temp                8798
Trafo_3_wind_Temp                8798
dtype: int64
```

NULL VALUE COUNTS PLOT BEFORE TREATMENT

In [15]:
```
name = "Dark2"
cmap = get_cmap(name)
colors = cmap.colors
x = df.columns
y = df.isnull().sum()
plt.bar(x,y,color = colors, align = 'center')
plt.xticks(rotation=90)
plt.title('Barplot_Null Value_Count', fontsize = 15)
plt.subplots_adjust(left=0.4, bottom=0.1, right=2.3, top=1.3)
for i in range(len(x)):
    pos = y[i]
    string = '{:}'.format(pos)
    plt.text(i,pos,string,ha='left',color='black',rotation = 'vertical', va = 'top')
```



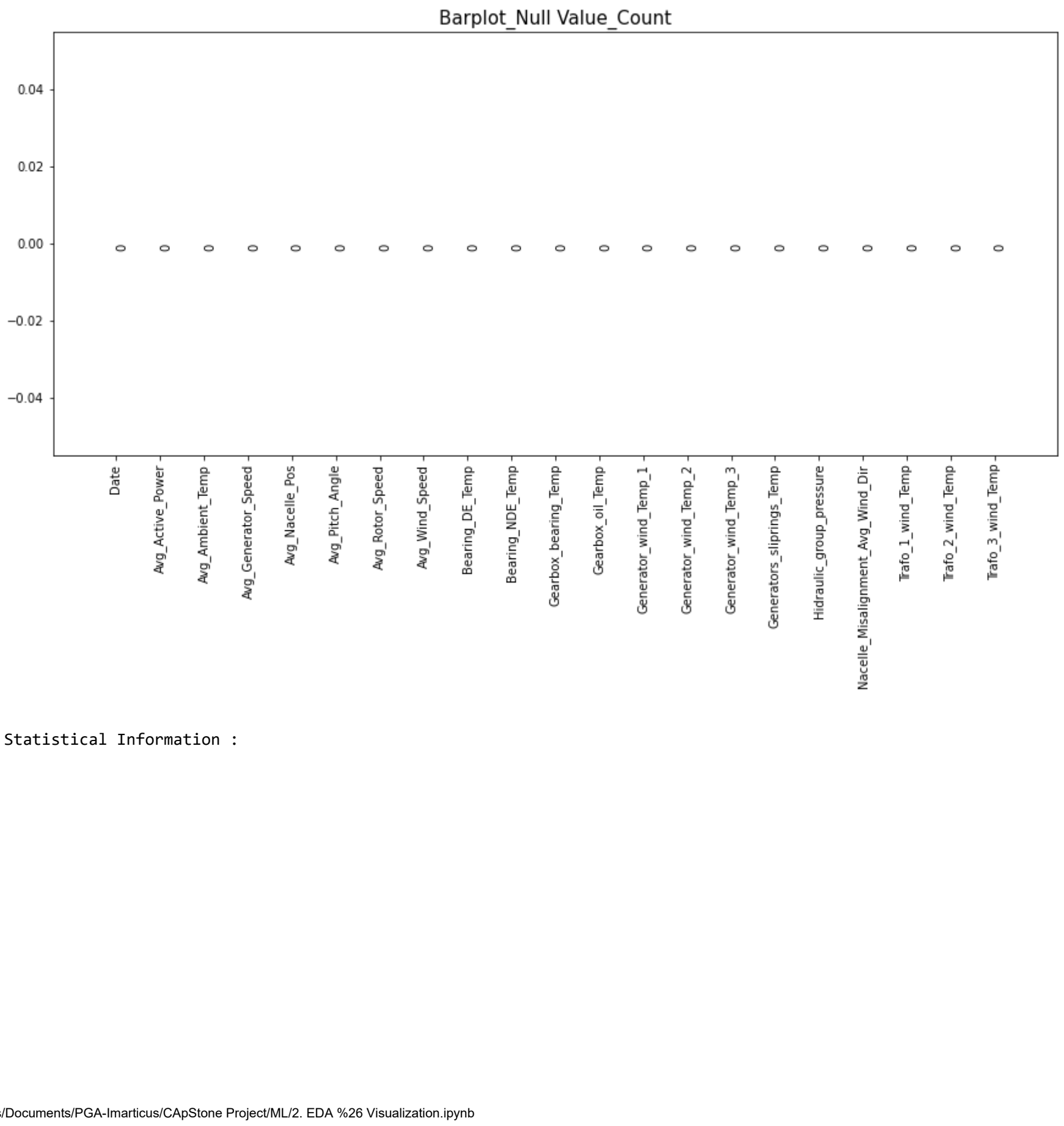
NULL VALUE TREATMENT

Every parameter of the Wind Turbine is a Numerical Variable. So, to avoid any Outliers effect on variables. I am doing Null values treatment with its Median.

In [16]:
```
constant_values = {'Avg_Active_Power': df.iloc[:,1].median(), 'Avg_Ambient_Temp': df.iloc[:,2].median(),
       'Avg_Generator_Speed': df.iloc[:,3].median(), 'Avg_Nacelle_Pos': df.iloc[:,4].median(), 'Avg_Pitch_Angle': df.iloc
       'Avg_Rotor_Speed': df.iloc[:,6].median(), 'Avg_Wind_Speed': df.iloc[:,7].median(), 'Bearing_DE_Temp': df.iloc[:,8
       'Bearing_NDE_Temp': df.iloc[:,9].median(), 'Gearbox_bearing_Temp': df.iloc[:,10].median(), 'Gearbox_oil_Temp': df
       'Generator_wind_Temp_1': df.iloc[:,12].median(), 'Generator_wind_Temp_2': df.iloc[:,13].median(),
       'Generator_wind_Temp_3': df.iloc[:,14].median(), "Generators_sliprings_Temp": df.iloc[:,15].median(),
       'Hidraulic_group_pressure': df.iloc[:,16].median(), 'Nacelle_Misalignment_Avg_Wind_Dir': df.iloc[:,17].median(),
       'Trafo_1_wind_Temp': df.iloc[:,18].median(), 'Trafo_2_wind_Temp': df.iloc[:,19].median(), 'Trafo_3_wind_Temp': df
df[['Avg_Active_Power', 'Avg_Ambient_Temp', 'Avg_Generator_Speed', 'Avg_Nacelle_Pos', 'Avg_Pitch_Angle', 'Avg_Rotor_Speed
    'Bearing_NDE_Temp', 'Gearbox_bearing_Temp', 'Gearbox_oil_Temp', 'Generator_wind_Temp_1', 'Generator_wind_Temp_2', 'G
    'Hidraulic_group_pressure', 'Nacelle_Misalignment_Avg_Wind_Dir',
    'Trafo_1_wind_Temp', 'Trafo_2_wind_Temp', 'Trafo_3_wind_Temp']] = df[['Avg_Active_Power', 'Avg_Ambient_Temp', 'Avg_G
                                                                          'Avg_Rotor_Speed', 'Avg_Wind_Speed', 'Bearing_
                                                                          'Gearbox_oil_Temp', 'Generator_wind_Temp_1', '
                                                                          "Generators_sliprings_Temp", 'Hidraulic_group_
                                                                          'Trafo_1_wind_Temp', 'Trafo_2_wind_Temp', 'Tra
```

NULL VALUE COUNTS PLOT AFTER TREATMENT

In [17]:
```
name = "Dark2"
cmap = get_cmap(name)
colors = cmap.colors
x = df.columns
y = df.isnull().sum()
plt.bar(x,y,color = colors, align = 'center')
plt.xticks(rotation=90)
plt.title('Barplot_Null Value_Count', fontsize = 15)
plt.subplots_adjust(left=0.4, bottom=0.1, right=2.3, top=1.3)
for i in range(len(x)):
    pos = y[i]
    string = '{:}'.format(pos)
    plt.text(i,pos,string,ha='left',color='black',rotation = 'vertical', va = 'top')
```



Statistical Information :

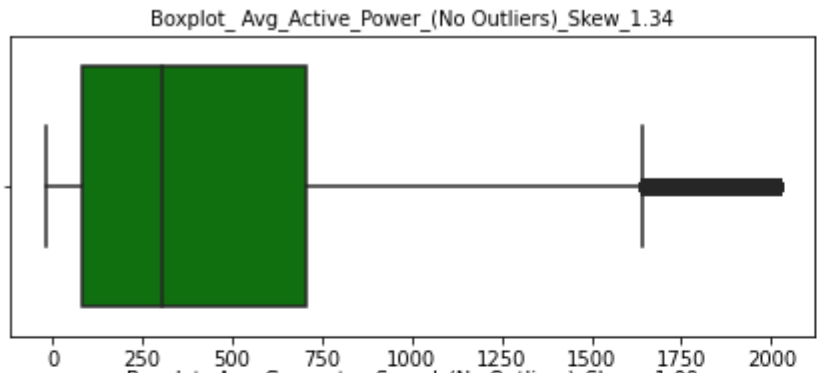
In [18]:
```
df.describe()
```

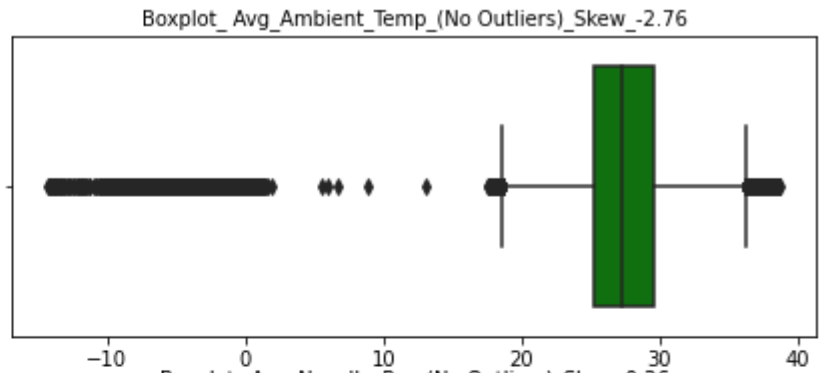
Out[18]:

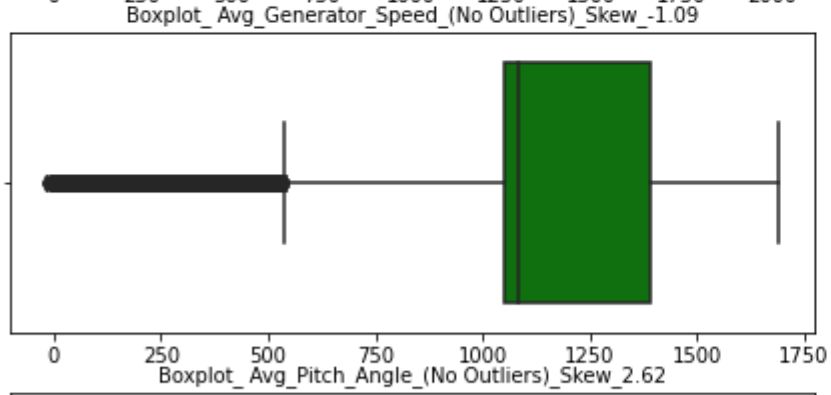
	Avg_Active_Power	Avg_Ambient_Temp	Avg_Generator_Speed	Avg_Nacelle_Pos	Avg_Pitch_Angle	Avg_Rotor_Speed	Avg_Wind_Speed	Bearir
count	245199.000000	245199.000000	245199.00000	245199.000000	245199.000000	245199.000000	245199.000000	24
mean	499.378702	27.357988	1090.59523	160.795747	9.678610	10.239833	5.853972	
std	551.593183	4.459187	439.39702	97.220176	23.826089	4.121497	2.518097	
min	-17.300000	-14.300000	-18.10000	0.000000	-1.010000	0.000000	0.100000	
25%	84.600000	25.200000	1049.90000	83.000000	0.000000	9.800000	4.100000	
50%	305.900000	27.200000	1081.70000	165.400000	0.710000	10.200000	5.600000	
75%	707.950000	29.600000	1392.60000	203.000000	2.840000	13.100000	7.300000	
max	2019.900000	38.700000	1689.70000	360.000000	87.100000	15.900000	24.700000	

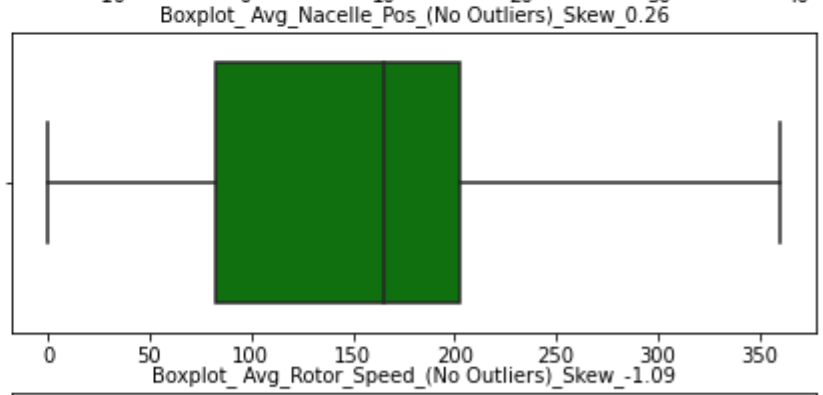
CHECK FOR OUTLIER VALUES USING BOX PLOTS

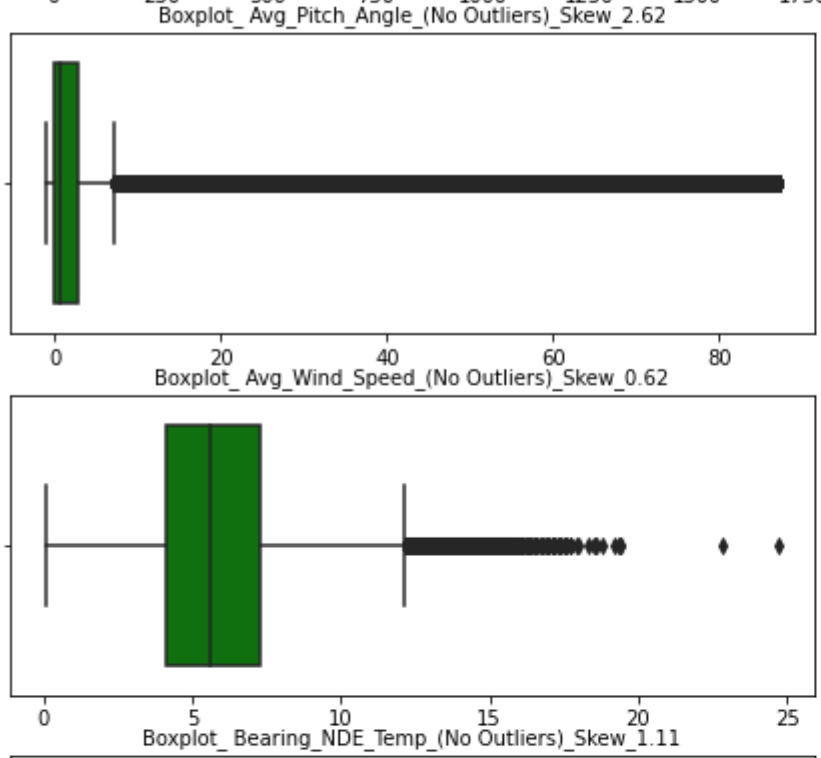
In [19]:
```
plt.subplot(8,2,1)
sns.boxplot(df.iloc[:,1], color = 'green')
plt.xlabel("")
plt.title('Boxplot_ '+ df.iloc[:,1].name +'_(No Outliers)_Skew_' + str(round(df.iloc[:,1].skew(),2)), fontsize = 10)
plt.subplot(8,2,2)
sns.boxplot(df.iloc[:,2], color = 'green')
plt.xlabel("")
plt.title('Boxplot_ '+ df.iloc[:,2].name +'_(No Outliers)_Skew_' + str(round(df.iloc[:,2].skew(),2)), fontsize = 10)
plt.subplot(8,2,3)
sns.boxplot(df.iloc[:,3], color = 'green')
plt.xlabel("")
plt.title('Boxplot_ '+ df.iloc[:,3].name +'_(No Outliers)_Skew_' + str(round(df.iloc[:,3].skew(),2)), fontsize = 10)
plt.subplot(8,2,4)
sns.boxplot(df.iloc[:,4], color = 'green')
plt.xlabel("")
plt.title('Boxplot_ '+ df.iloc[:,4].name +'_(No Outliers)_Skew_' + str(round(df.iloc[:,4].skew(),2)), fontsize = 10)
plt.subplot(8,2,5)
sns.boxplot(df.iloc[:,5], color = 'green')
plt.xlabel("")
plt.title('Boxplot_ '+ df.iloc[:,5].name +'_(No Outliers)_Skew_' + str(round(df.iloc[:,5].skew(),2)), fontsize = 10)
plt.subplot(8,2,6)
sns.boxplot(df.iloc[:,6], color = 'green')
plt.xlabel("")
plt.title('Boxplot_ '+ df.iloc[:,6].name +'_(No Outliers)_Skew_' + str(round(df.iloc[:,6].skew(),2)), fontsize = 10)
plt.subplot(8,2,7)
sns.boxplot(df.iloc[:,7], color = 'green')
plt.xlabel("")
plt.title('Boxplot_ '+ df.iloc[:,7].name +'_(No Outliers)_Skew_' + str(round(df.iloc[:,7].skew(),2)), fontsize = 10)
plt.subplot(8,2,8)
sns.boxplot(df.iloc[:,8], color = 'green')
plt.xlabel("")
plt.title('Boxplot_ '+ df.iloc[:,8].name +'_(No Outliers)_Skew_' + str(round(df.iloc[:,8].skew(),2)), fontsize = 10)
plt.subplot(8,2,9)
sns.boxplot(df.iloc[:,9], color = 'green')
plt.xlabel("")
plt.title('Boxplot_ '+ df.iloc[:,9].name +'_(No Outliers)_Skew_' + str(round(df.iloc[:,9].skew(),2)), fontsize = 10)
plt.subplot(8,2,10)
sns.boxplot(df.iloc[:,10], color = 'green')
plt.xlabel("")
plt.title('Boxplot_ '+ df.iloc[:,10].name +'_(No Outliers)_Skew_' + str(round(df.iloc[:,10].skew(),2)), fontsize = 10)

plt.subplots_adjust(left=0.45, bottom=0, right=2.5, top=4.9)
```

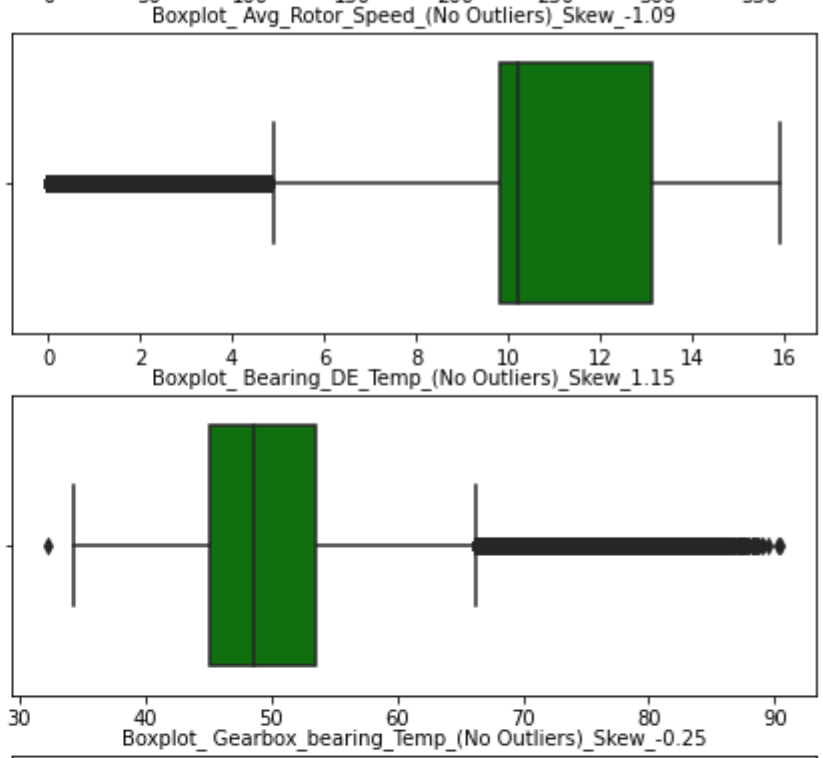

Boxplot_ Avg_Active_Power_(No Outliers)_Skew_1.34
0
250
500
750
1000
1250
1500
1750
2000

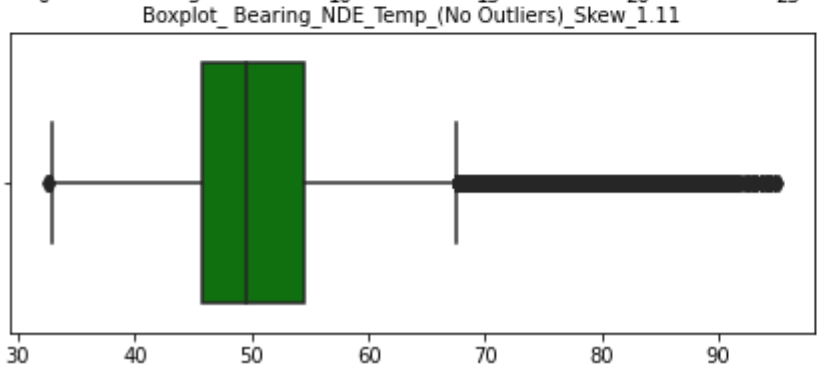

Boxplot_ Avg_Ambient_Temp_(No Outliers)_Skew_-2.76
−10
0
10
20
30
40

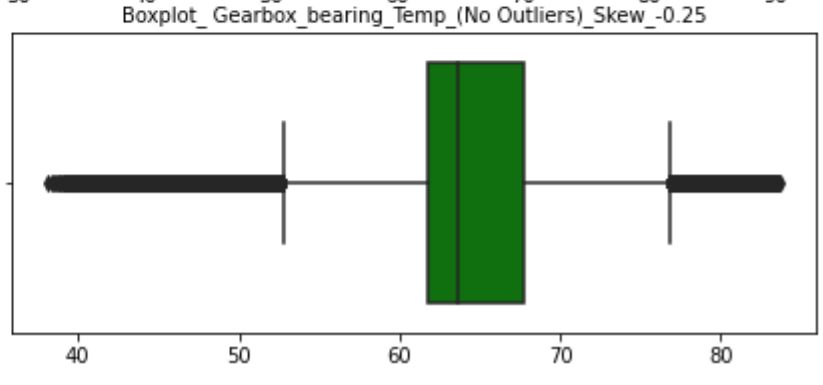

Boxplot_ Avg_Generator_Speed_(No Outliers)_Skew_-1.09
0
250
500
750
1000
1250
1500
1750


Boxplot_ Avg_Nacelle_Pos_(No Outliers)_Skew_0.26
0
50
100
150
200
250
300
350

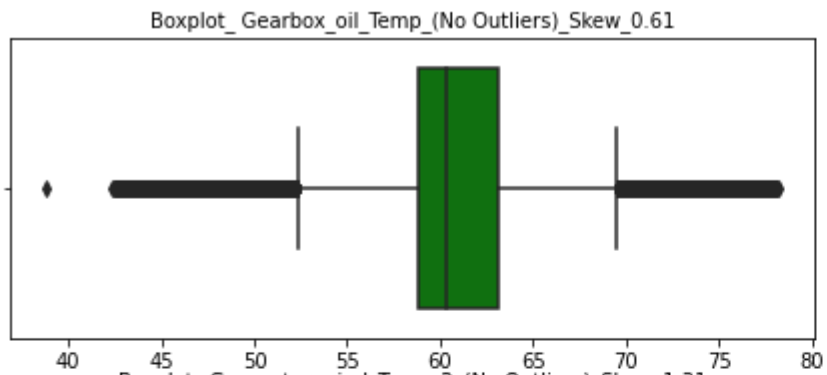

Boxplot_ Avg_Pitch_Angle_(No Outliers)_Skew_2.62
0
20
40
60
80
Boxplot_ Avg_Wind_Speed_(No Outliers)_Skew_0.62
0
5
10
15
20
25

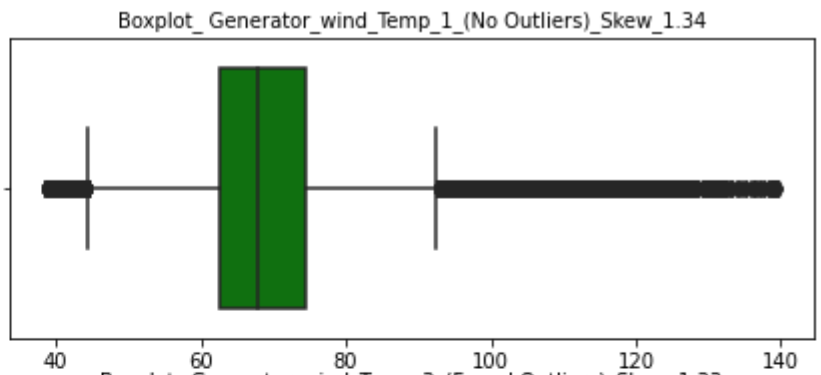

Boxplot_ Avg_Rotor_Speed_(No Outliers)_Skew_-1.09
0
2
4
6
8
10
12
14
16
Boxplot_ Bearing_DE_Temp_(No Outliers)_Skew_1.15
30
40
50
60
70
80
90

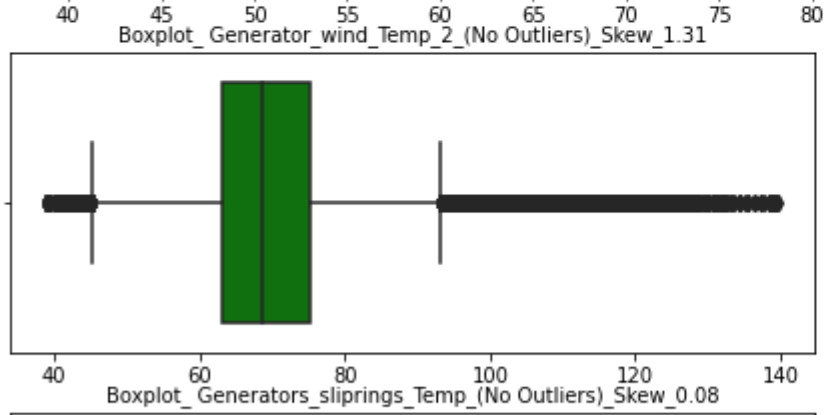

Boxplot_ Bearing_NDE_Temp_(No Outliers)_Skew_1.11
30
40
50
60
70
80
90

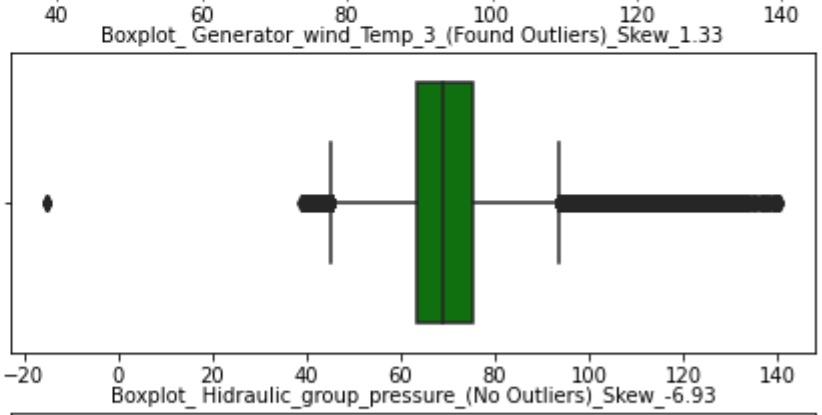

Boxplot_ Gearbox_bearing_Temp_(No Outliers)_Skew_-0.25
40
50
60
70
80

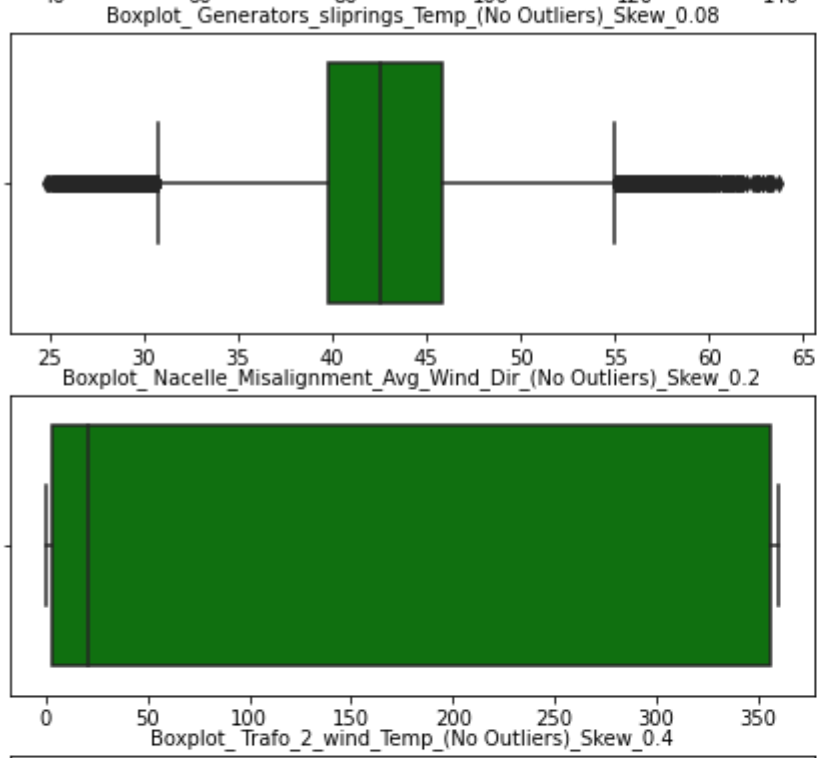
In [20]:
```
plt.subplot(8,2,1)
sns.boxplot(df.iloc[:,11], color = 'green')
plt.xlabel("")
plt.title('Boxplot_ '+ df.iloc[:,11].name +'_(No Outliers)_Skew_' + str(round(df.iloc[:,11].skew(),2)), fontsize = 10)
plt.subplot(8,2,2)
sns.boxplot(df.iloc[:,12], color = 'green')
plt.xlabel("")
plt.title('Boxplot_ '+ df.iloc[:,12].name +'_(No Outliers)_Skew_' + str(round(df.iloc[:,12].skew(),2)), fontsize = 10)
plt.subplot(8,2,3)
sns.boxplot(df.iloc[:,13], color = 'green')
plt.xlabel("")
plt.title('Boxplot_ '+ df.iloc[:,13].name +'_(No Outliers)_Skew_' + str(round(df.iloc[:,13].skew(),2)), fontsize = 10)
plt.subplot(8,2,4)
sns.boxplot(df.iloc[:,14], color = 'green')
plt.xlabel("")
plt.title('Boxplot_ '+ df.iloc[:,14].name +'_(Found Outliers)_Skew_' + str(round(df.iloc[:,14].skew(),2)), fontsize = 10
plt.subplot(8,2,5)
sns.boxplot(df.iloc[:,15], color = 'green')
plt.xlabel("")
plt.title('Boxplot_ '+ df.iloc[:,15].name +'_(No Outliers)_Skew_' + str(round(df.iloc[:,15].skew(),2)), fontsize = 10)
plt.subplot(8,2,6)
sns.boxplot(df.iloc[:,16], color = 'green')
plt.xlabel("")
plt.title('Boxplot_ '+ df.iloc[:,16].name +'_(No Outliers)_Skew_' + str(round(df.iloc[:,16].skew(),2)), fontsize = 10)
plt.subplot(8,2,7)
sns.boxplot(df.iloc[:,17], color = 'green')
plt.xlabel("")
plt.title('Boxplot_ '+ df.iloc[:,17].name +'_(No Outliers)_Skew_' + str(round(df.iloc[:,17].skew(),2)), fontsize = 10)
plt.subplot(8,2,8)
sns.boxplot(df.iloc[:,18], color = 'green')
plt.xlabel("")
plt.title('Boxplot_ '+ df.iloc[:,18].name +'_(No Outliers)_Skew_' + str(round(df.iloc[:,18].skew(),2)), fontsize = 10)
plt.subplot(8,2,9)
sns.boxplot(df.iloc[:,19], color = 'green')
plt.xlabel("")
plt.title('Boxplot_ '+ df.iloc[:,19].name +'_(No Outliers)_Skew_' + str(round(df.iloc[:,19].skew(),2)), fontsize = 10)
plt.subplot(8,2,10)
sns.boxplot(df.iloc[:,20], color = 'green')
plt.xlabel("")
plt.title('Boxplot_ '+ df.iloc[:,20].name +'_(No Outliers)_Skew_' + str(round(df.iloc[:,20].skew(),2)), fontsize = 10)

plt.subplots_adjust(left=0.45, bottom=0, right=2.5, top=4.9)
```

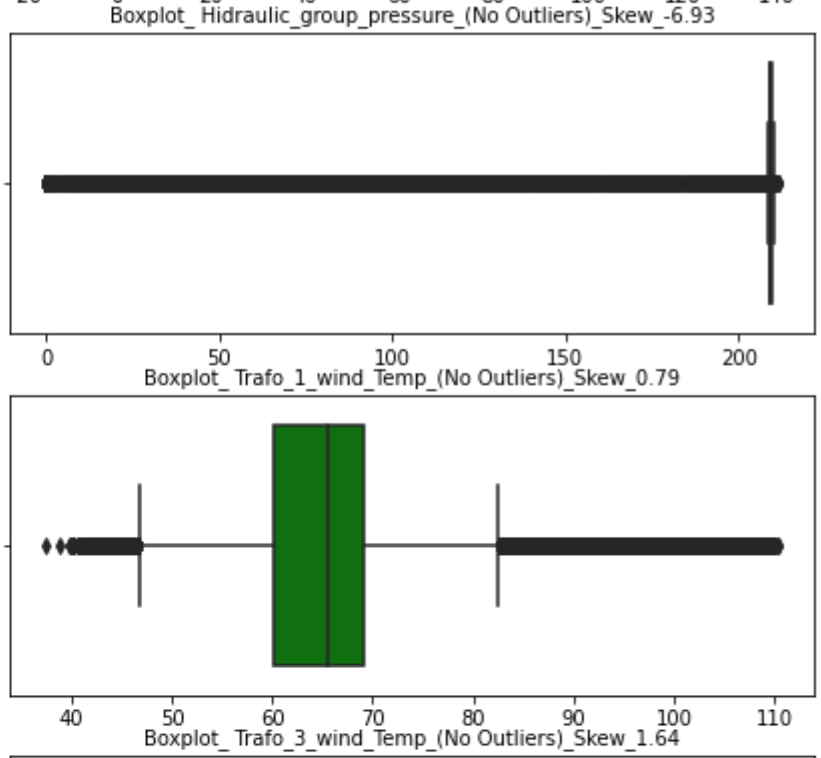












From above box plots i found that the data is almost not having the outliers except one variable that is Generat or Winding Temperature-3. So, we need Outliers Treatment as using Quantiles.

Check for the Quartile Ranges

In [21]:
```
print('Lower Limit - 5% :', df.iloc[:,14].quantile(0.05), '\n Upper Limit - 95% :', df.iloc[:,14].quantile(0.95))
```

```
Lower Limit - 5% : 51.8
 Upper Limit - 95% : 105.3
```

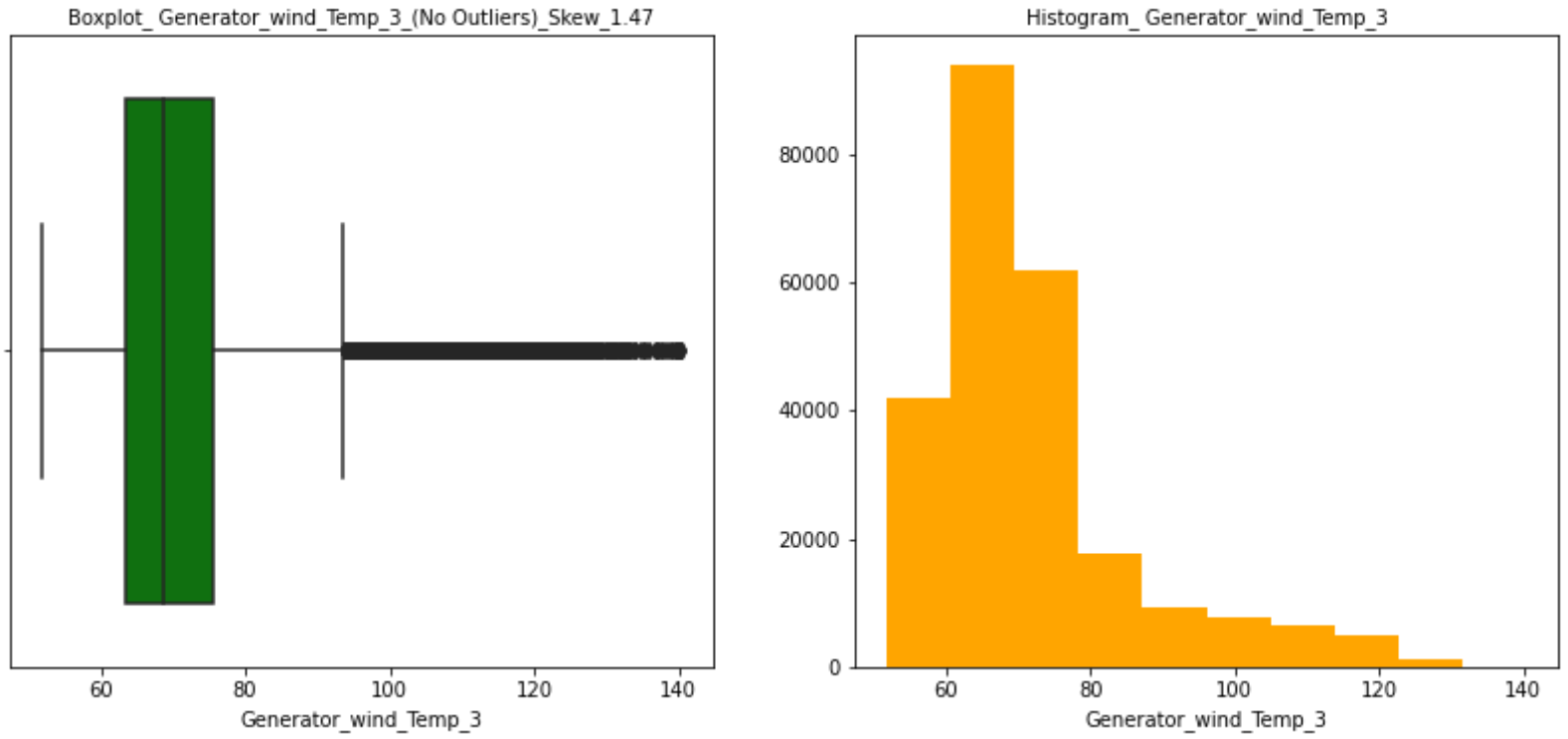
Replace the Outliers with its Quartile ranges

In [22]:
```
df.iloc[:,14] = np.where(df.iloc[:,14] < df.iloc[:,14].quantile(0.05), df.iloc[:,14].quantile(0.05), df.iloc[:,14])
df.iloc[:,14].describe()
```

Out[22]:
```
count    245199.000000
mean         71.662278
std          14.424620
min          51.800000
25%          63.400000
50%          68.800000
75%          75.500000
max         140.400000
Name: Generator_wind_Temp_3, dtype: float64
```

Box Plot and Histogram plot for re-checking the Outliers

In [23]:
```
plt.subplot(1,2,1)
sns.boxplot(df.iloc[:,14], color = 'green')
plt.xlabel(df.iloc[:,14].name, fontsize = 10)
plt.title('Boxplot_ '+ df.iloc[:,14].name +'_(No Outliers)_Skew_' + str(round(df.iloc[:,14].skew(),2)), fontsize = 10)
plt.subplot(1,2,2)
plt.hist(df.iloc[:,14], color = 'orange')
plt.xlabel(df.iloc[:,14].name, fontsize = 10)
plt.title('Histogram_ '+ df.iloc[:,14].name, fontsize = 10)
plt.subplots_adjust(left=0.4, bottom=0.1, right=2.2, top=1.2)
```



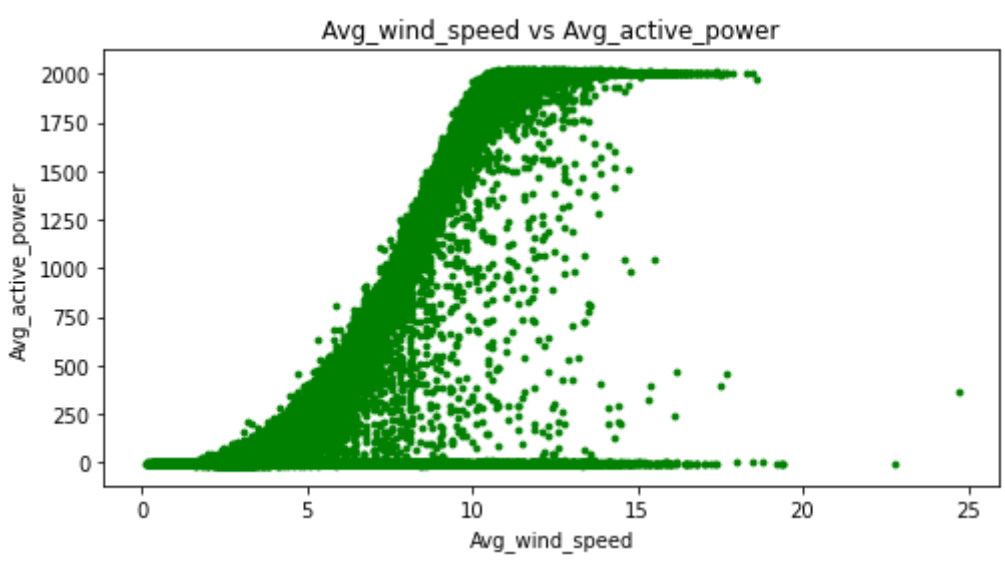
In [24]:
```
df.to_csv('Cleaned_df.csv')
```

VISUALIZATION

In [25]:
```
def Scatter_Plot(X,Y,Size,Color):
        fig = plt.figure(figsize=(8,4))

        plt.scatter(x = X,
                    y = Y, s = float(Size), color = Color)

        plt.xlabel(X.name.capitalize())
        plt.ylabel(Y.name.capitalize())

        plt.title("%s vs %s"%(X.name.capitalize(), Y.name.capitalize()))
```

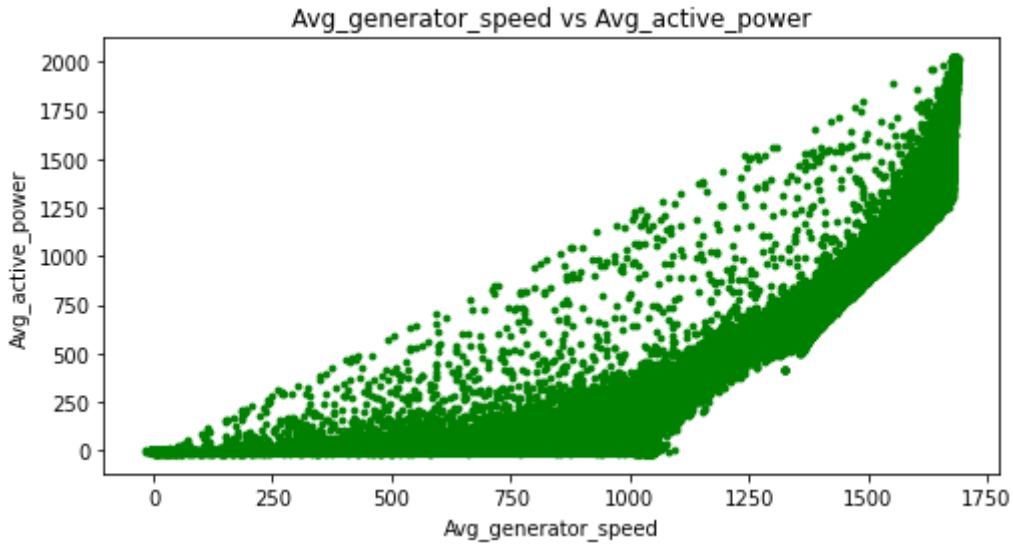
POWER CURVE - Average Wind Speed Vs Average Actvie Power

In [26]:
```
Scatter_Plot(df.iloc[:,7],df.iloc[:,1], 8, 'green')
```



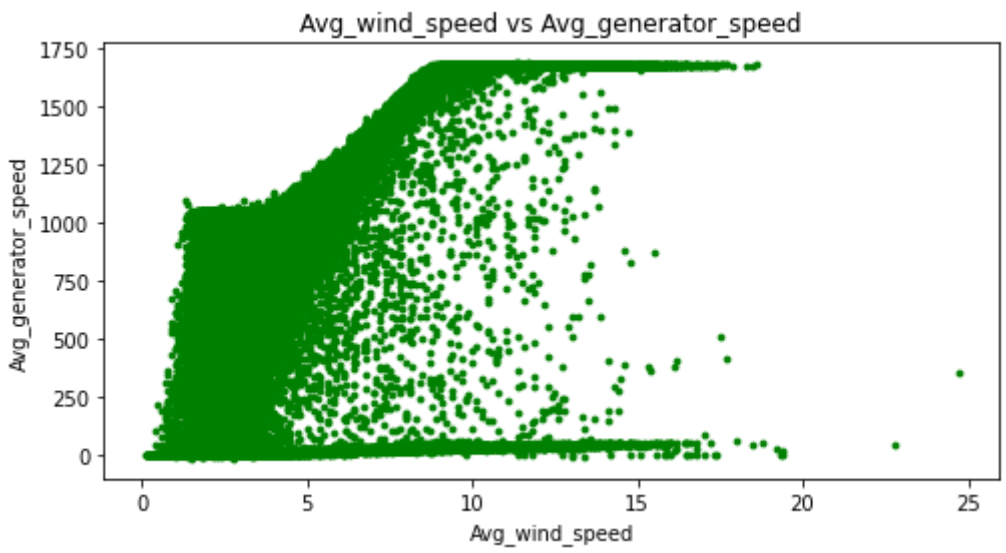
Average Generator Speed Vs Average Actvie Power

In [27]:
```
Scatter_Plot(df.iloc[:,3],df.iloc[:,1], 8, 'green')
```



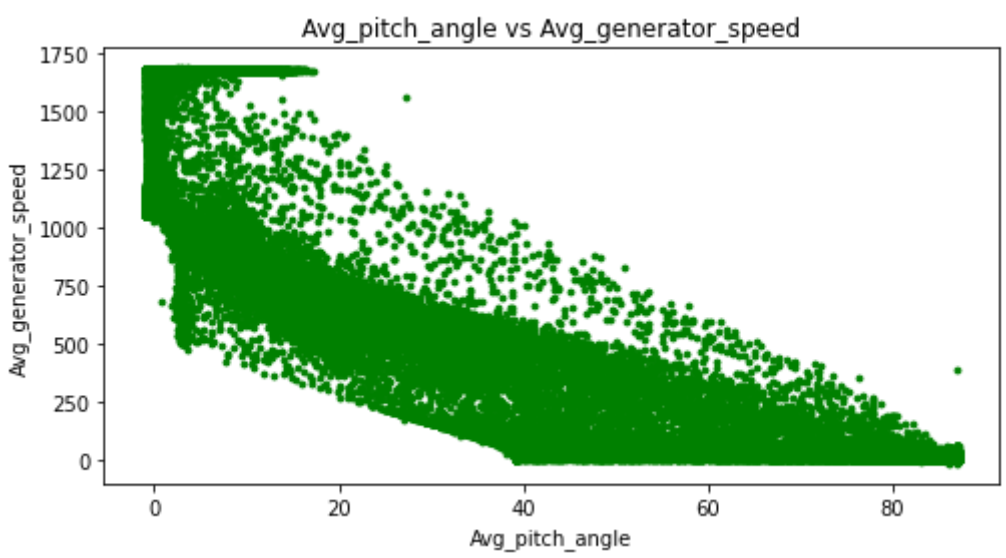
Average Generator Speed Vs Average Wind Speed

In [28]:
```
Scatter_Plot(df.iloc[:,7],df.iloc[:,3], 8, 'green')
```



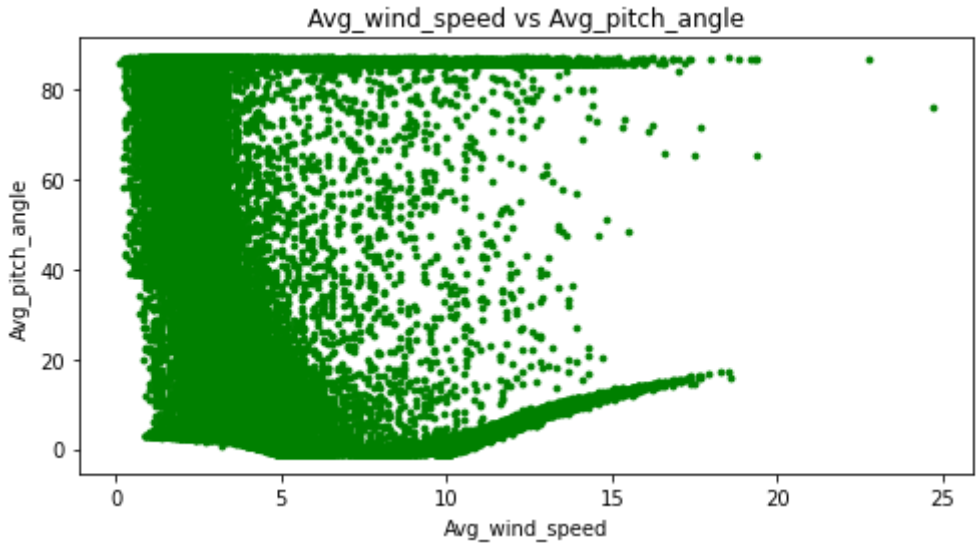
Average Pitch Angle Vs Average Generator Speed

In [29]:
```
Scatter_Plot(df.iloc[:,5],df.iloc[:,3], 8, 'green')
```



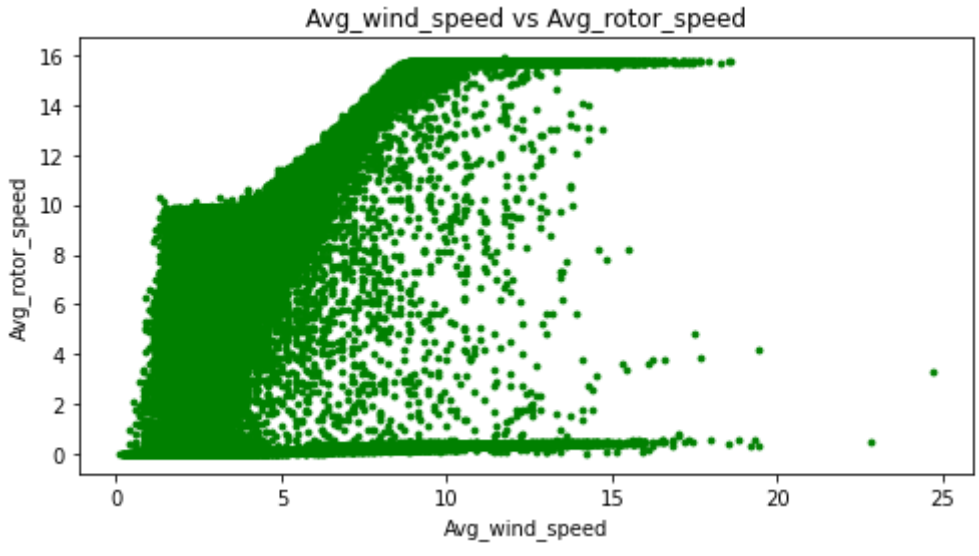
Average Pitch Angle Vs Average Wind Speed

In [30]:
```
Scatter_Plot(df.iloc[:,7],df.iloc[:,5], 8, 'green')
```



Average Wind Speed Vs Average Rotor Speed

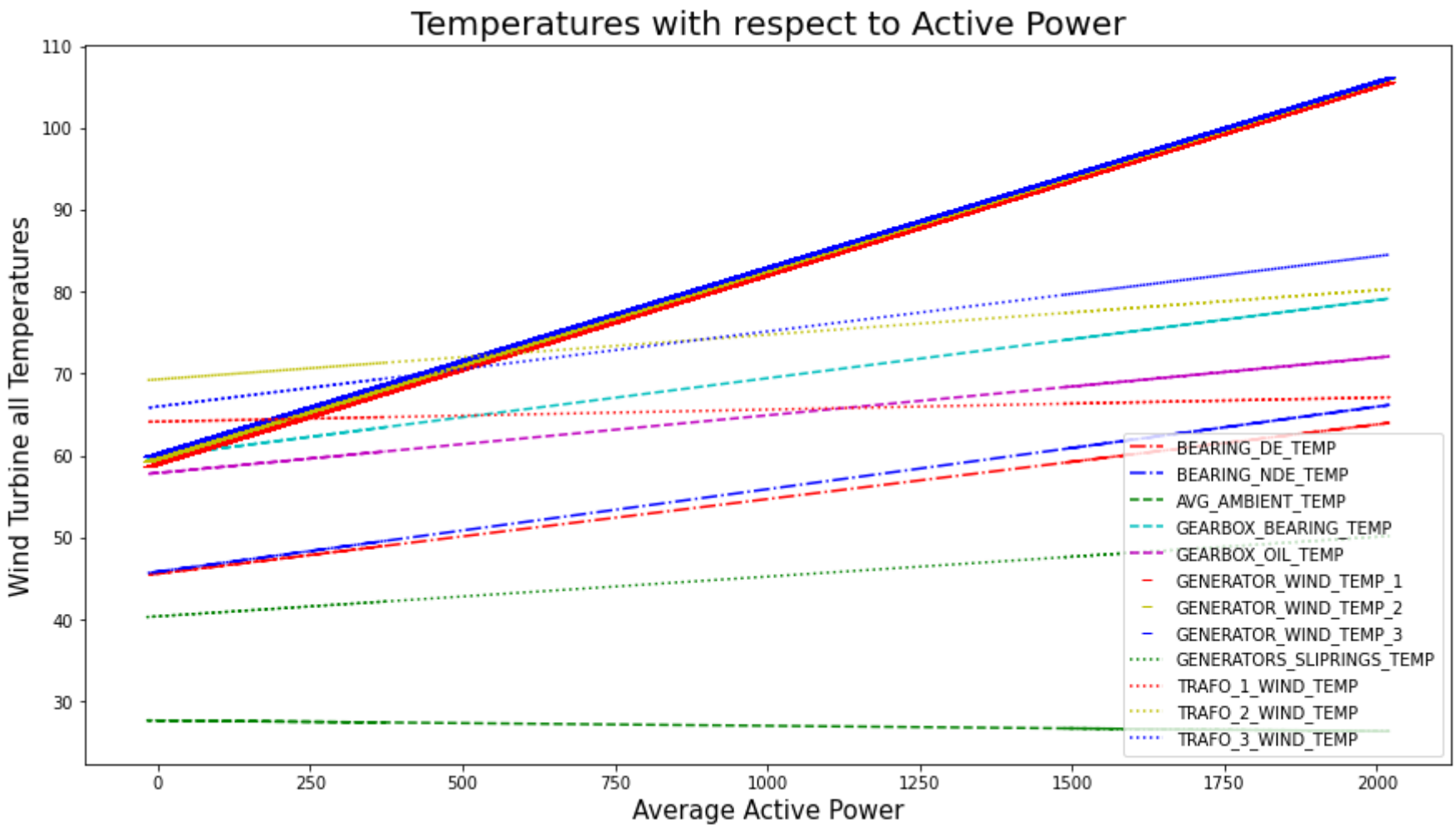
In [31]:
```
Scatter_Plot(df.iloc[:,7],df.iloc[:,6], 8, 'green')
```



All Temperature Parameters Vs Active Power

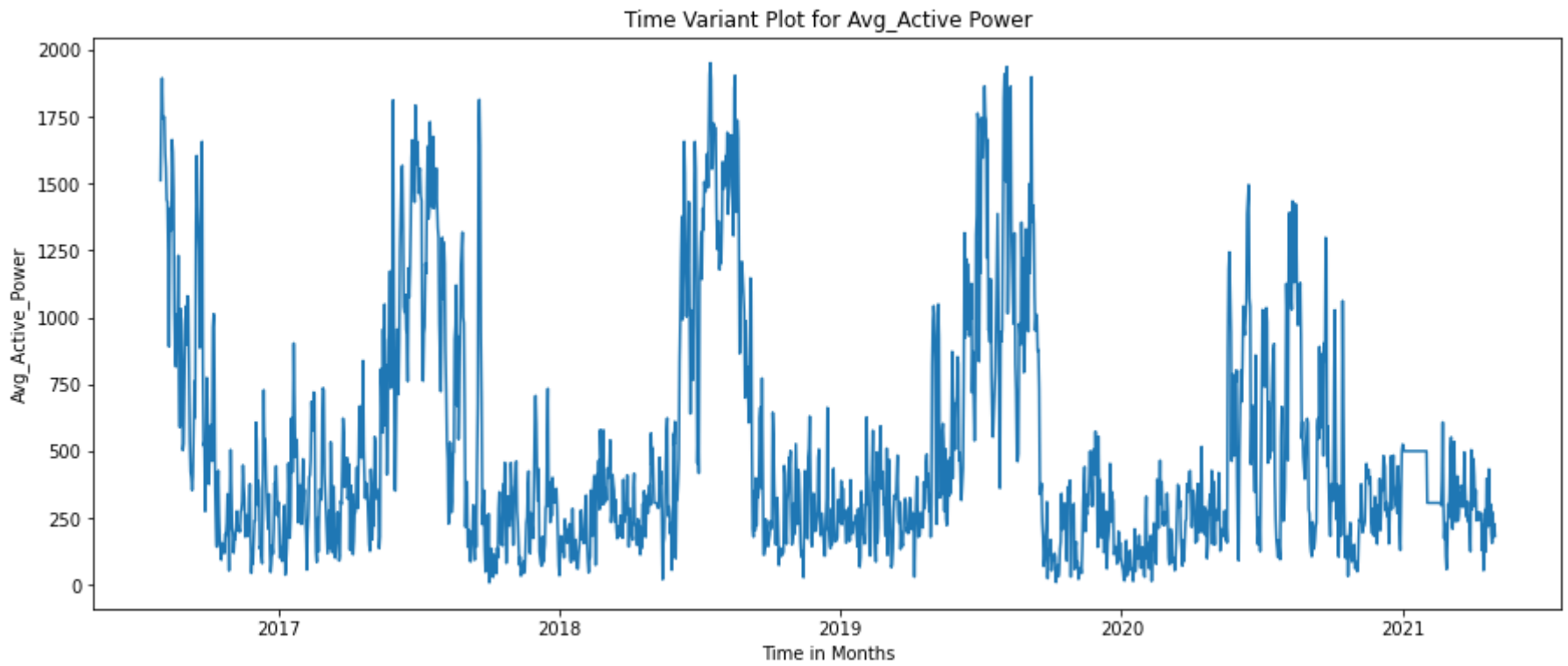
In [32]:
```
# line 1 points

x1 = df.iloc[:,1]
y1 = df.iloc[:,8]

z1 = np.polyfit(x1, y1, 1)
p1 = np.poly1d(z1)
plt.plot(x1,p1(x1),"r-.", label = df.iloc[:,8].name.upper())

# line 2 points
x2 = df.iloc[:,1]
y2 = df.iloc[:,9]

z2 = np.polyfit(x2, y2, 1)
p2 = np.poly1d(z2)
plt.plot(x2,p2(x2),"b-.",label = df.iloc[:,9].name.upper())

# line 3 points
x3 = df.iloc[:,1]
y3 = df.iloc[:,2]

z3 = np.polyfit(x3, y3, 1)
p3 = np.poly1d(z3)
plt.plot(x3,p3(x3),"g--", label = df.iloc[:,2].name.upper())

# line 4 points
x4 = df.iloc[:,1]
y4 = df.iloc[:,10]

z4 = np.polyfit(x4, y4, 1)
p4 = np.poly1d(z4)
plt.plot(x4,p4(x4),"c--",label = df.iloc[:,10].name.upper())

# line 5 points
x5 = df.iloc[:,1]
y5 = df.iloc[:,11]

z5 = np.polyfit(x5, y5, 1)
p5 = np.poly1d(z5)
plt.plot(x5,p5(x5),"m--", label = df.iloc[:,11].name.upper())

# line 6 points
x6 = df.iloc[:,1]
y6 = df.iloc[:,12]

z6 = np.polyfit(x6, y6, 1)
p6 = np.poly1d(z6)
plt.plot(x6,p6(x6),"r_",label = df.iloc[:,12].name.upper())

# line 7 points
x7 = df.iloc[:,1]
y7 = df.iloc[:,13]

z7 = np.polyfit(x7, y7, 1)
p7 = np.poly1d(z7)
plt.plot(x7,p7(x7),"y_",label = df.iloc[:,13].name.upper())

# line 8 points
x8 = df.iloc[:,1]
y8 = df.iloc[:,14]

z8 = np.polyfit(x8, y8, 1)
p8 = np.poly1d(z8)
plt.plot(x8,p8(x8),"b_",label = df.iloc[:,14].name.upper())

# line 9 points
x9 = df.iloc[:,1]
y9 = df.iloc[:,15]

z9 = np.polyfit(x9, y9, 1)
p9 = np.poly1d(z9)
plt.plot(x9,p9(x9),"g:",label = df.iloc[:,15].name.upper())

# line 10 points
x10 = df.iloc[:,1]
y10 = df.iloc[:,18]

z10 = np.polyfit(x10, y10, 1)
p10 = np.poly1d(z10)
plt.plot(x10,p10(x10),"r:",label = df.iloc[:,18].name.upper())

# line 11 points
x11 = df.iloc[:,1]
y11 = df.iloc[:,19]

z11 = np.polyfit(x11, y11, 1)
p11 = np.poly1d(z11)
plt.plot(x11,p11(x11),"y:",label = df.iloc[:,19].name.upper())
```

```
# line 12 points
x12 = df.iloc[:,1]
y12 = df.iloc[:,20]

z12 = np.polyfit(x12, y12, 1)
p12 = np.poly1d(z12)
plt.plot(x12,p12(x12),"b:",label = df.iloc[:,20].name.upper())


# Set the y & x axis label of the current axis.
plt.ylabel('Wind Turbine all Temperatures', fontsize = 15)
plt.xlabel('Average Active Power', fontsize = 15)
# Set a title of the current axes.
plt.title('Temperatures with respect to Active Power', fontsize = 20)
# show a legend on the plot
plt.legend()
# Display a figure.
plt.subplots_adjust(left=0.5, bottom=0, right=2.4, top=1.5)
plt.show()
```



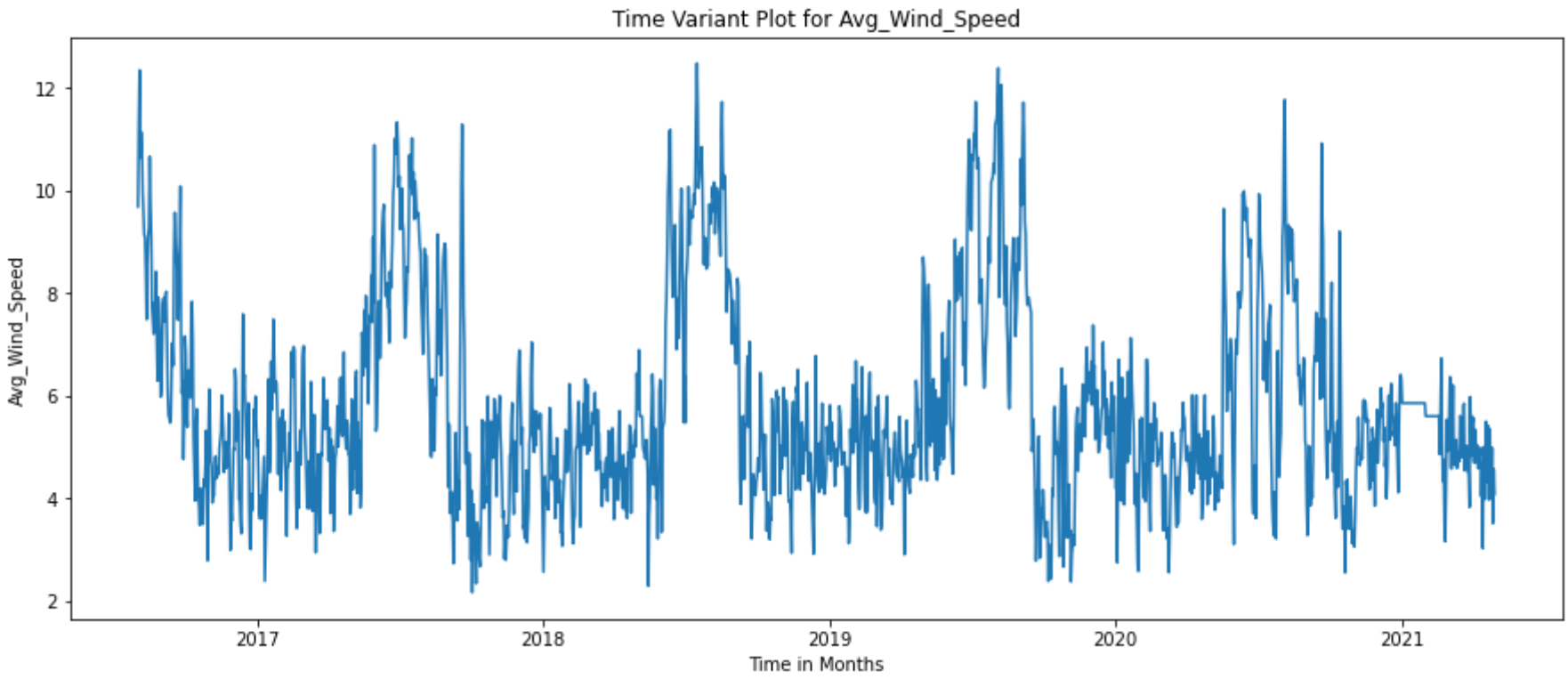
ACTIVE POWER TREND DURING A TIME PERIOD

In [33]:
```
pf1 = df[['Date','Avg_Active_Power']]
pf1 = pf1.set_index('Date')
y1 = pf1['Avg_Active_Power'].resample('D').mean()
y1 = y1.fillna(y1.mean())
plt.figure(figsize=(15,6))
plt.plot(y1)
plt.title('Time Variant Plot for Avg_Active Power')
plt.xlabel("Time in Days")
plt.ylabel("Avg_Active_Power")
plt.show()
```



WIND SPEED TREND OVER A TIME PERIOD

In [34]:
```
pf2 = df[['Date','Avg_Wind_Speed']]
pf2 = pf2.set_index('Date')
y2 = pf2['Avg_Wind_Speed'].resample('D').mean()
y2 = y2.fillna(y2.mean())
plt.figure(figsize=(15,6))
plt.plot(y2)
plt.title('Time Variant Plot for Avg_Wind_Speed')
plt.xlabel("Time in Days")
plt.ylabel("Avg_Wind_Speed")
plt.show()
```



WIND ROSE PLOTS

In [35]:
```
pf3 = df[['Avg_Wind_Speed', 'Avg_Nacelle_Pos','Nacelle_Misalignment_Avg_Wind_Dir', 'Avg_Active_Power']]
pf3
```

Out[35]:

	Avg_Wind_Speed	Avg_Nacelle_Pos	Nacelle_Misalignment_Avg_Wind_Dir	Avg_Active_Power
0	9.3	138.0	357.0	1487.3
1	9.3	138.0	356.9	1647.6
2	9.0	138.0	356.5	1506.4
3	8.3	138.0	356.5	1240.9
4	8.5	138.0	356.2	1202.4
...	...	...	...	...
245194	4.7	1.0	0.6	205.1
245195	4.7	1.0	3.9	208.4
245196	5.2	355.7	355.1	228.0
245197	5.7	354.0	3.8	355.1
245198	5.9	0.0	2.6	374.3

245199 rows × 4 columns

In [36]:
```
pf3['Direction'] = np.where((pf3['Avg_Nacelle_Pos']>337.5) & (pf3['Avg_Nacelle_Pos']<=22.5),"0N","0N")

pf3['Direction'] = np.where((pf3['Avg_Nacelle_Pos']>22.5) & (pf3['Avg_Nacelle_Pos']<=67.5),"1NE",pf3['Direction'])

pf3['Direction'] = np.where((pf3['Avg_Nacelle_Pos']>67.5) & (pf3['Avg_Nacelle_Pos']<=112.5),"2E",pf3['Direction'])

pf3['Direction'] = np.where((pf3['Avg_Nacelle_Pos']>112.5) & (pf3['Avg_Nacelle_Pos']<=157.7),"3SE",pf3['Direction'])

pf3['Direction'] = np.where((pf3['Avg_Nacelle_Pos']>157.7) & (pf3['Avg_Nacelle_Pos']<=202.5),"4S",pf3['Direction'])

pf3['Direction'] = np.where((pf3['Avg_Nacelle_Pos']>202.5) & (pf3['Avg_Nacelle_Pos']<=247.5),"5SW",pf3['Direction'])

pf3['Direction'] = np.where((pf3['Avg_Nacelle_Pos']>247.5) & (pf3['Avg_Nacelle_Pos']<=292.5),"6W",pf3['Direction'])

pf3['Direction'] = np.where((pf3['Avg_Nacelle_Pos']>292.5) & (pf3['Avg_Nacelle_Pos']<=337.5),"7NW",pf3['Direction'])
```

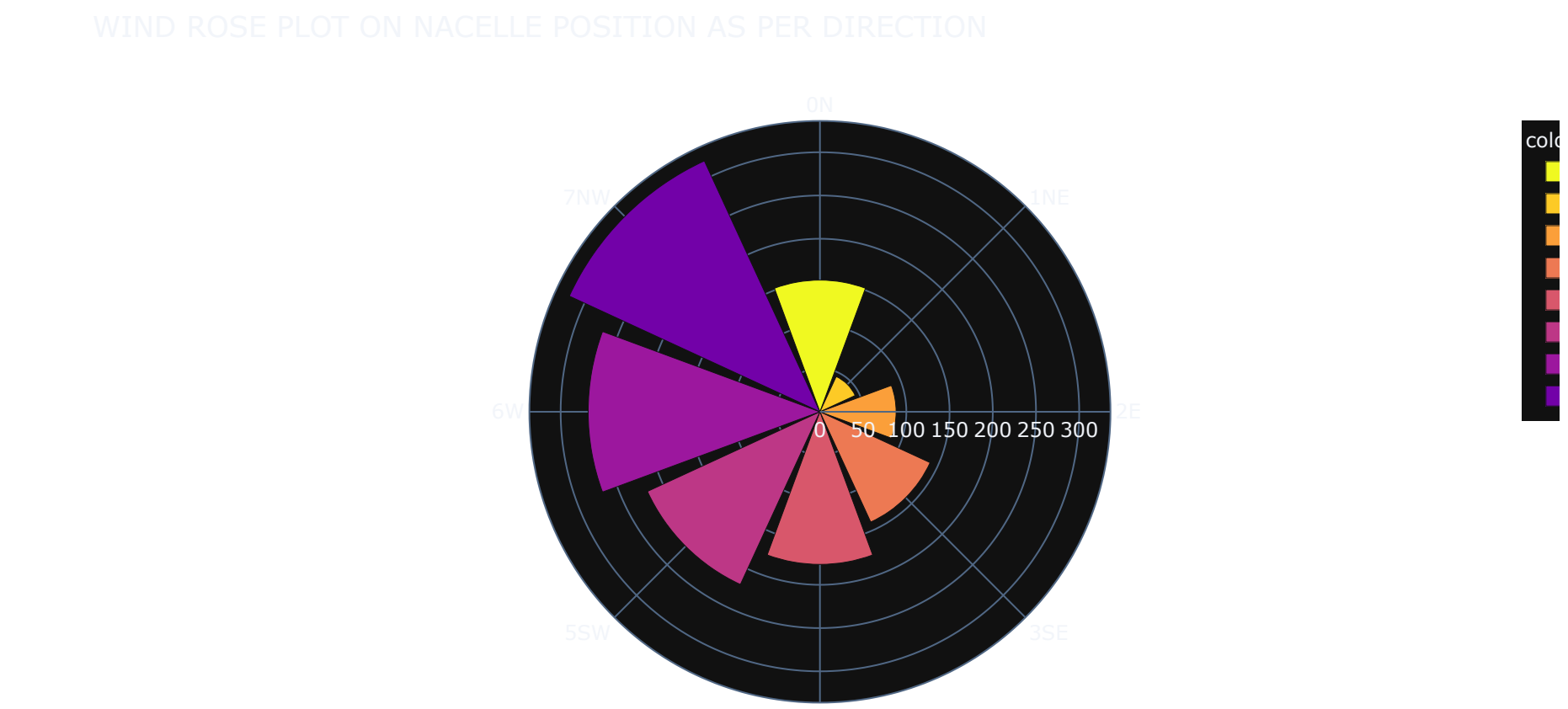
In [37]:
```
pf4 = pf3.groupby('Direction').mean()
pf4
```

Out[37]:

Direction	Avg_Wind_Speed	Avg_Nacelle_Pos	Nacelle_Misalignment_Avg_Wind_Dir	Avg_Active_Power
0N	5.329365	152.206170	165.683403	302.621934
1NE	4.756077	44.900122	163.068175	255.076733
2E	4.908596	88.145161	164.629859	291.300842
3SE	6.365296	140.367936	150.401662	653.891086
4S	7.220002	176.387014	161.040035	798.247221
5SW	5.285135	219.814881	187.714309	439.715250
6W	3.880825	268.179053	169.902159	156.240819
7NW	4.361006	319.378438	156.757856	198.279020

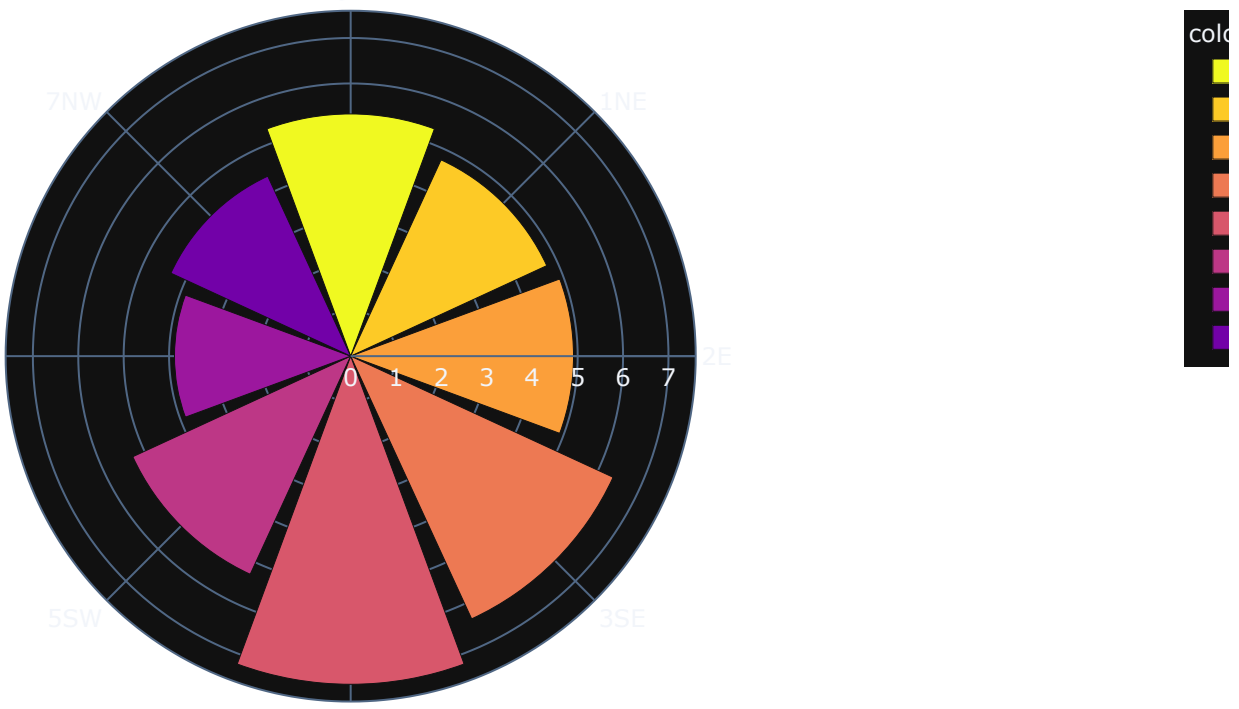
NACELLE POSITION AS PER WIND DIRECTION

In [38]:
```
# !pip install plotly
import plotly.express as px
fig = px.bar_polar(pf4, r="Avg_Nacelle_Pos", theta=pf4.index.values,
                   color=pf4.index.values, template="plotly_dark",
                   color_discrete_sequence= px.colors.sequential.Plasma_r, title = 'WIND ROSE PLOT ON NACELLE POSITION AS
fig.show()
```



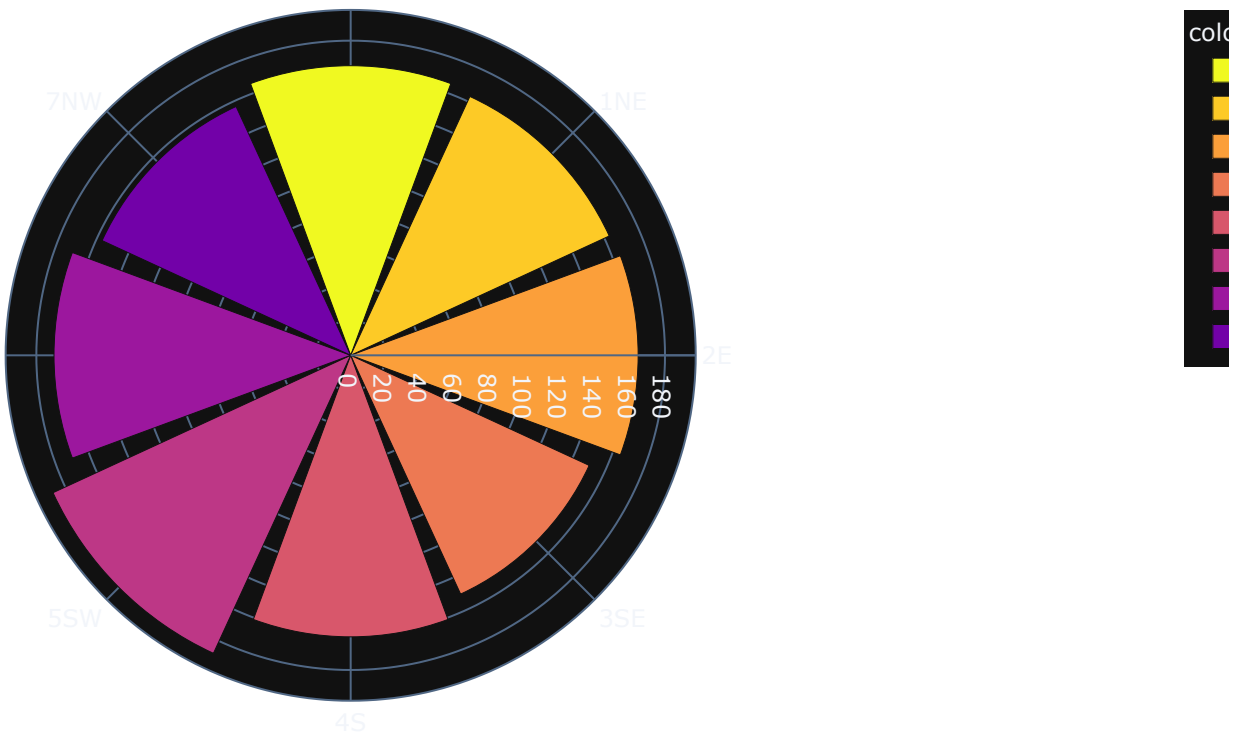
AVERAGE WIND DIRECTION

In [39]:
```
fig = px.bar_polar(pf4, r="Avg_Wind_Speed", theta=pf4.index.values,
                   color=pf4.index.values, template="plotly_dark",
                   color_discrete_sequence= px.colors.sequential.Plasma_r, title = 'WIND ROSE PLOT ON AVERAGE WIND DIREC
fig.show()
```



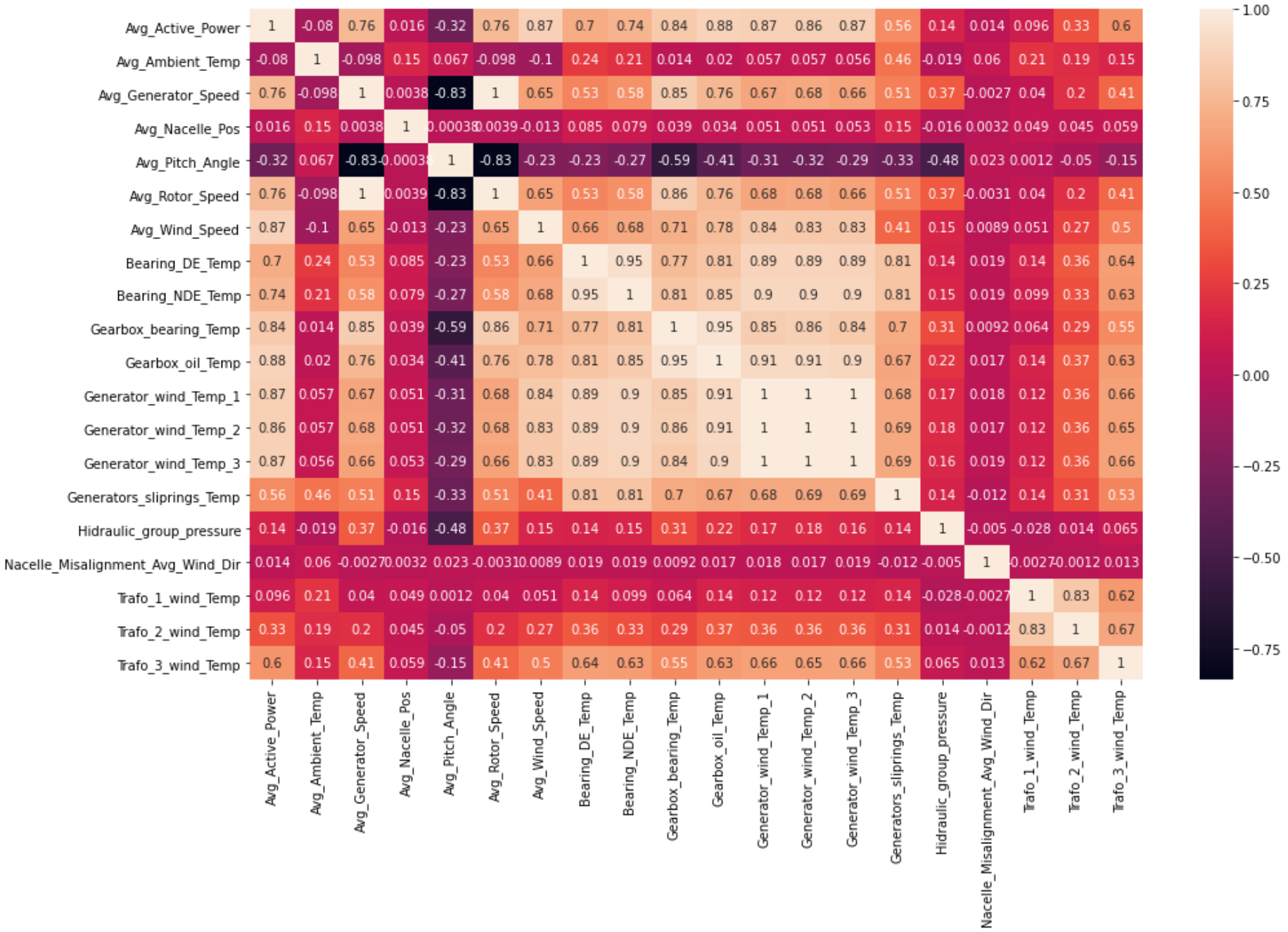
AVERAGE NACELLE MISALIGNMENT WITH RESPECT TO WIND DIRECTION

In [40]:
```
fig = px.bar_polar(pf4, r="Nacelle_Misalignment_Avg_Wind_Dir", theta=pf4.index.values,
                   color=pf4.index.values, template="plotly_dark",
                   color_discrete_sequence= px.colors.sequential.Plasma_r, title = 'AVERAGE NACELLE MISALIGNMENT WITH RES
fig.show()
```



CORRELATION PLOT

In [41]:
```
sns.heatmap(df.corr(), annot = True)
plt.subplots_adjust(left=0.8, bottom=0, right=2.8, top=1.8)
plt.show()
```



Thank You.....

MODEL BUILDING -- LINEAR REGRESSION

SPLIT THE TRAIN AND TEST DATASET

In [10]:
```
# !pip install pandas
import pandas as pd
# !pip install numpy
import numpy as np
# !pip install seaborn
import seaborn as sns
# !pip install matplotlib
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
# !pip install ipywidgets
import ipywidgets as widgets
```

READ THE CLEANED DATASET

In [11]:
```
df = pd.read_csv('Cleaned_df.csv')
df = df.drop(['Unnamed: 0'],axis = 1)
```

SPLIT THE DATASET INTO X AND Y

In [12]:
```
x = df.iloc[:,2:]
y = df.iloc[:,1]
```

SPLIT THE DATASET INTO TRAIN AND TEST OF X AND Y

In [13]:
```
# !pip install sklearn
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y,train_size = 0.8, random_state = 42)
```

CHECK FOR SHAPE OF TRAIN AND TEST SETS

In [14]:
```
print('Shape of x_train: ',x_train.shape)
print('Shape of x_test: ',x_test.shape)
print('Shape of y_train: ',y_train.shape)
print('Shape of y_test: ',y_test.shape)
```

```
Shape of x_train:  (196159, 19)
Shape of x_test:  (49040, 19)
Shape of y_train:  (196159,)
Shape of y_test:  (49040,)
```

FIT THE MODELS

In [15]:
```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr = lr.fit(x_train,y_train)
```

PREDICT THE MODEL

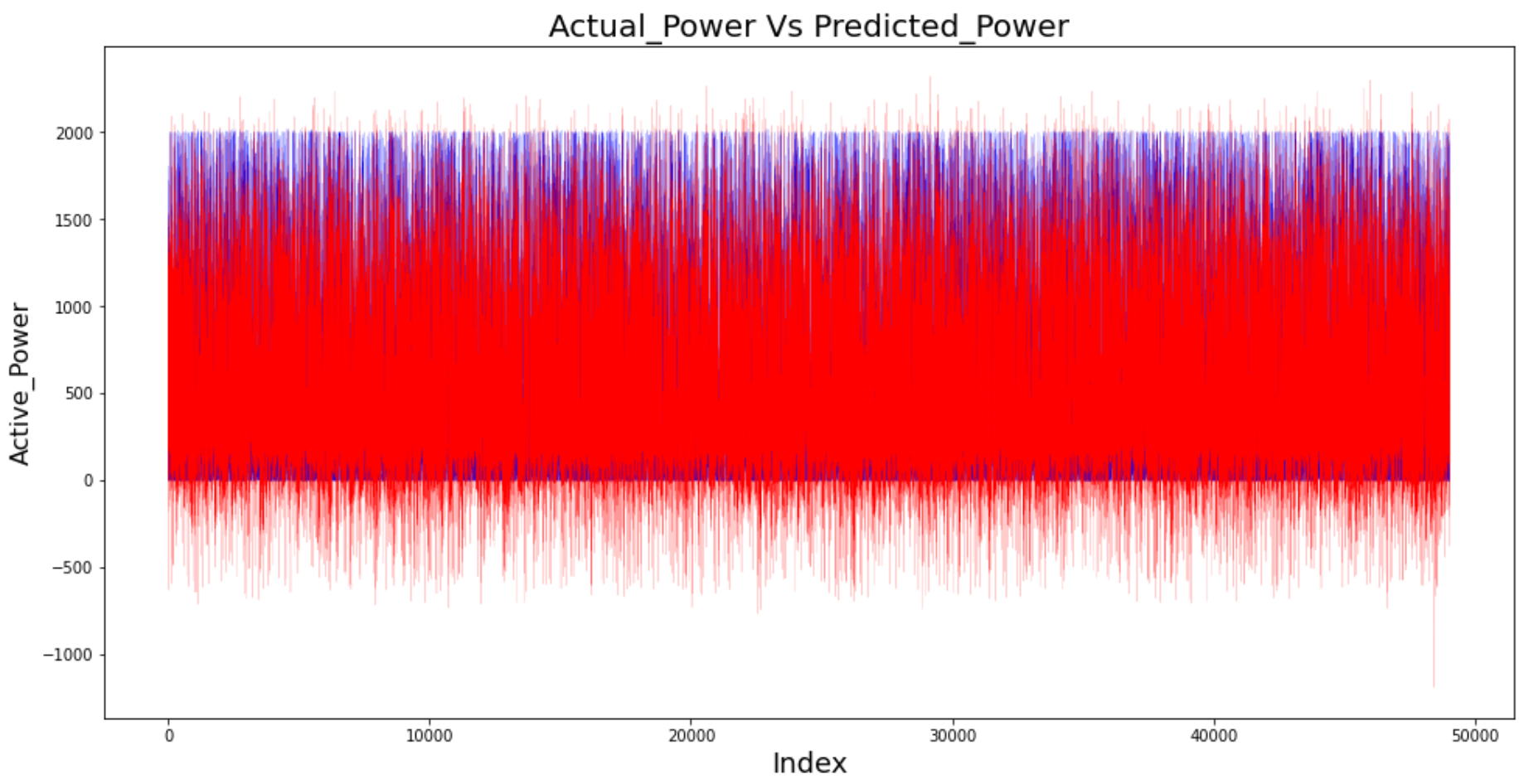
In [16]:
```
y_pred = lr.predict(x_test)
```

MODEL-1

In [17]:
```
# !pip install statmodels
import statsmodels.api as sm
x_train_sm = x_train

x_train_sm = sm.add_constant(x_train_sm)

mlm = sm.OLS(y_train,x_train_sm).fit()

mlm.params
print(mlm.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:       Avg_Active_Power   R-squared:                       0.938
Model:                            OLS   Adj. R-squared:                  0.938
Method:                 Least Squares   F-statistic:                 1.552e+05
Date:                Mon, 10 May 2021   Prob (F-statistic):               0.00
Time:                        00:40:34   Log-Likelihood:            -1.2446e+06
No. Observations:              196159   AIC:                         2.489e+06
Df Residuals:                  196139   BIC:                         2.489e+06
Df Model:                          19
Covariance Type:            nonrobust
=====================================================================================================
                                        coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------------------
const                             -3124.6801      9.735   -320.962      0.000   -3143.761   -3105.599
Avg_Ambient_Temp                     -4.8097      0.095    -50.819      0.000      -4.995      -4.624
Avg_Generator_Speed                  -0.0036      0.077     -0.047      0.963      -0.154       0.146
Avg_Nacelle_Pos                   5.365e-05      0.003      0.016      0.987      -0.006       0.006
Avg_Pitch_Angle                      12.3866      0.045    275.231      0.000      12.298      12.475
Avg_Rotor_Speed                     116.5211      8.183     14.239      0.000     100.483     132.560
Avg_Wind_Speed                       32.1814      0.325     99.074      0.000      31.545      32.818
Bearing_DE_Temp                      -9.4580      0.153    -61.715      0.000      -9.758      -9.158
Bearing_NDE_Temp                     -8.7528      0.158    -55.406      0.000      -9.062      -8.443
Gearbox_bearing_Temp                  2.9521      0.256     11.512      0.000       2.449       3.455
Gearbox_oil_Temp                     26.9376      0.341     78.954      0.000      26.269      27.606
Generator_wind_Temp_1               116.9005      2.156     54.210      0.000     112.674     121.127
Generator_wind_Temp_2              -144.2204      2.128    -67.784      0.000    -148.391    -140.050
Generator_wind_Temp_3                34.7784      0.289    120.193      0.000      34.211      35.345
Generators_sliprings_Temp            14.6944      0.154     95.160      0.000      14.392      14.997
Hidraulic_group_pressure              0.2188      0.014     15.428      0.000       0.191       0.247
Nacelle_Misalignment_Avg_Wind_Dir     0.0027      0.002      1.479      0.139      -0.001       0.006
Trafo_1_wind_Temp                    -1.9114      0.084    -22.684      0.000      -2.077      -1.746
Trafo_2_wind_Temp                     0.4971      0.071      7.003      0.000       0.358       0.636
Trafo_3_wind_Temp                     3.8216      0.074     51.661      0.000       3.677       3.967
==============================================================================
Omnibus:                    34340.376   Durbin-Watson:                   1.998
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           263120.746
Skew:                           0.636   Prob(JB):                         0.00
Kurtosis:                       8.530   Cond. No.                     3.86e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.86e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

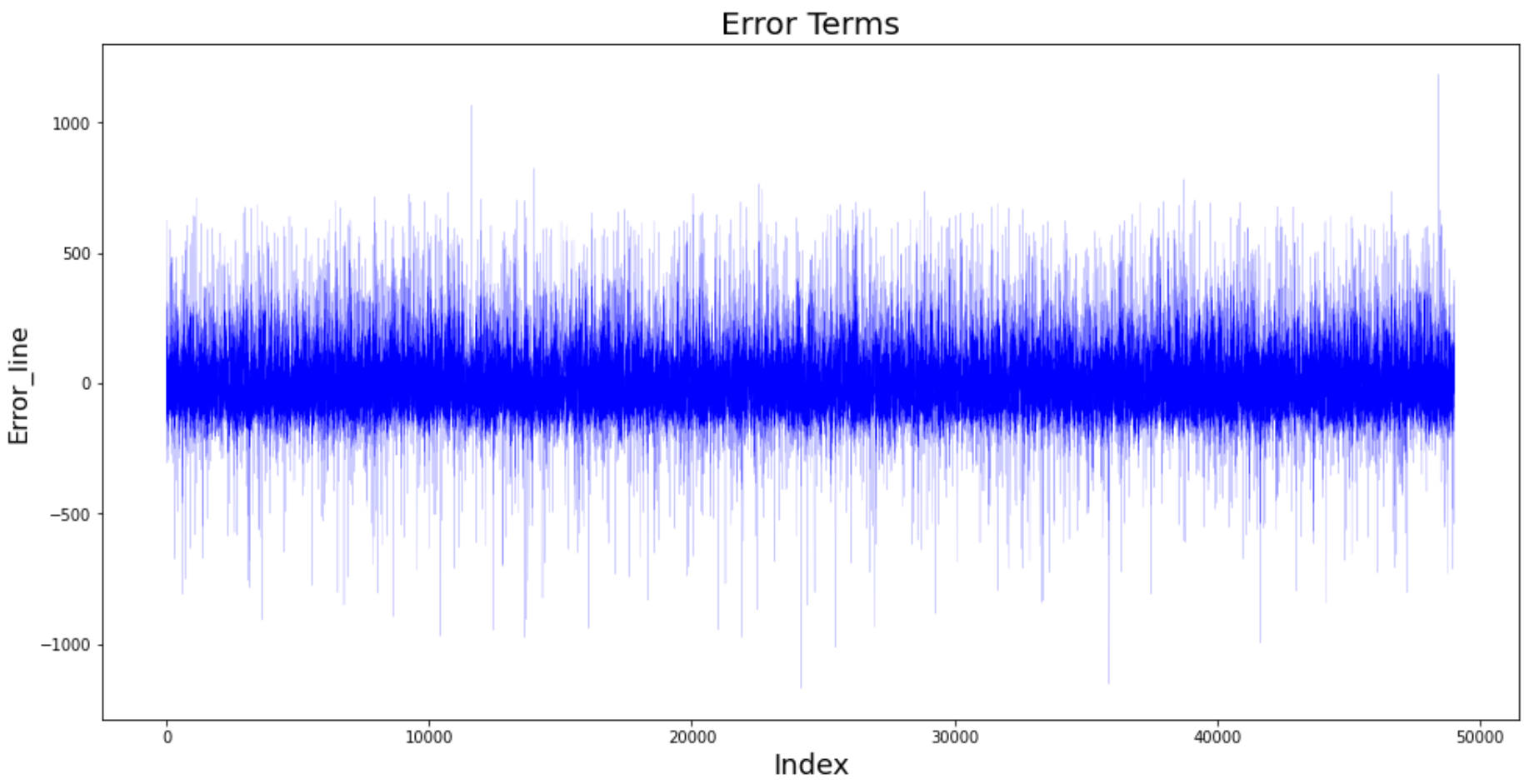
In [18]:
```
#Actual vs Predicted
c = [i for i in range(1,49041,1)]
plt.plot(c,y_test, color="blue", linewidth=0.1, linestyle="-")
plt.plot(c,y_pred, color="red",  linewidth=0.1, linestyle="-")
plt.title('Actual_Power Vs Predicted_Power', fontsize=20)
plt.xlabel('Index', fontsize=18)
plt.ylabel('Active_Power', fontsize=16)
plt.subplots_adjust(left=0.4, bottom=0.1, right=2.5, top=1.6)
```



In [19]:
```
from statsmodels.stats.outliers_influence import variance_inflation_factor
# VIF dataframe
vif_data = pd.DataFrame()
vif_data["feature"] = x_train.columns
# calculating VIF for each feature
vif_data["VIF"] = [variance_inflation_factor(x_train.values, i)
                          for i in range(len(x_train.columns))]

print(vif_data)
```

	feature	VIF
0	Avg_Ambient_Temp	70.956154
1	Avg_Generator_Speed	83644.989518
2	Avg_Nacelle_Pos	3.898009
3	Avg_Pitch_Angle	13.630874
4	Avg_Rotor_Speed	84271.377809
5	Avg_Wind_Speed	44.248870
6	Bearing_DE_Temp	615.214488
7	Bearing_NDE_Temp	669.374913
8	Gearbox_bearing_Temp	2756.496545
9	Gearbox_oil_Temp	2659.267301
10	Generator_wind_Temp_1	231529.737063
11	Generator_wind_Temp_2	232227.893434
12	Generator_wind_Temp_3	4572.275104
13	Generators_sliprings_Temp	450.132150
14	Hidraulic_group_pressure	81.460160
15	Nacelle_Misalignment_Avg_Wind_Dir	1.930151
16	Trafo_1_wind_Temp	311.304992
17	Trafo_2_wind_Temp	273.175199
18	Trafo_3_wind_Temp	285.184301

In [20]:
```
c = [i for i in range(1,49041,1)]

plt.plot(c,y_test-y_pred, color="blue", linewidth=0.1, linestyle="-")
plt.title('Error Terms', fontsize=20)               # Plot heading
plt.xlabel('Index', fontsize=18)                       # X-label
plt.ylabel('Error_line', fontsize=16)
plt.subplots_adjust(left=0.4, bottom=0.1, right=2.5, top=1.6)
```



In [21]:
```
from sklearn.metrics import mean_squared_error, r2_score
mse = mean_squared_error(y_test, y_pred)
r_squared = r2_score(y_test, y_pred)
print('Mean_Squared_Error :' ,mse)
print('r_square_value :', round(r_squared,3),"% Variance of the Active Power is Explained by the Wind Speed, Pitch Angle
```

```
Mean_Squared_Error : 18890.960947349187
r_square_value : 0.938 % Variance of the Active Power is Explained by the Wind Speed, Pitch Angle and Rotor Speed
```

MODEL-2

In [22]:
```
x1_train = x_train[['Avg_Ambient_Temp', 'Avg_Generator_Speed', 'Avg_Pitch_Angle', 'Avg_Rotor_Speed',
                    'Avg_Wind_Speed', 'Bearing_DE_Temp', 'Bearing_NDE_Temp', 'Gearbox_bearing_Temp', 'Gearbox_oil_Temp',
                    'Generator_wind_Temp_1', 'Generator_wind_Temp_2', 'Generator_wind_Temp_3', 'Generators_sliprings_Tem
                    'Hidraulic_group_pressure', 'Nacelle_Misalignment_Avg_Wind_Dir', 'Trafo_1_wind_Temp', 'Trafo_2_wind_
                    'Trafo_3_wind_Temp']]
x1_test = x_test[['Avg_Ambient_Temp', 'Avg_Generator_Speed', 'Avg_Pitch_Angle', 'Avg_Rotor_Speed',
                    'Avg_Wind_Speed', 'Bearing_DE_Temp', 'Bearing_NDE_Temp', 'Gearbox_bearing_Temp', 'Gearbox_oil_Temp',
                    'Generator_wind_Temp_1', 'Generator_wind_Temp_2', 'Generator_wind_Temp_3', 'Generators_sliprings_Tem
                    'Hidraulic_group_pressure', 'Nacelle_Misalignment_Avg_Wind_Dir', 'Trafo_1_wind_Temp', 'Trafo_2_wind_
                    'Trafo_3_wind_Temp']]
```

Avg_Nacelle_Pos -- varible is removed because of its significant value is more than 0.05.

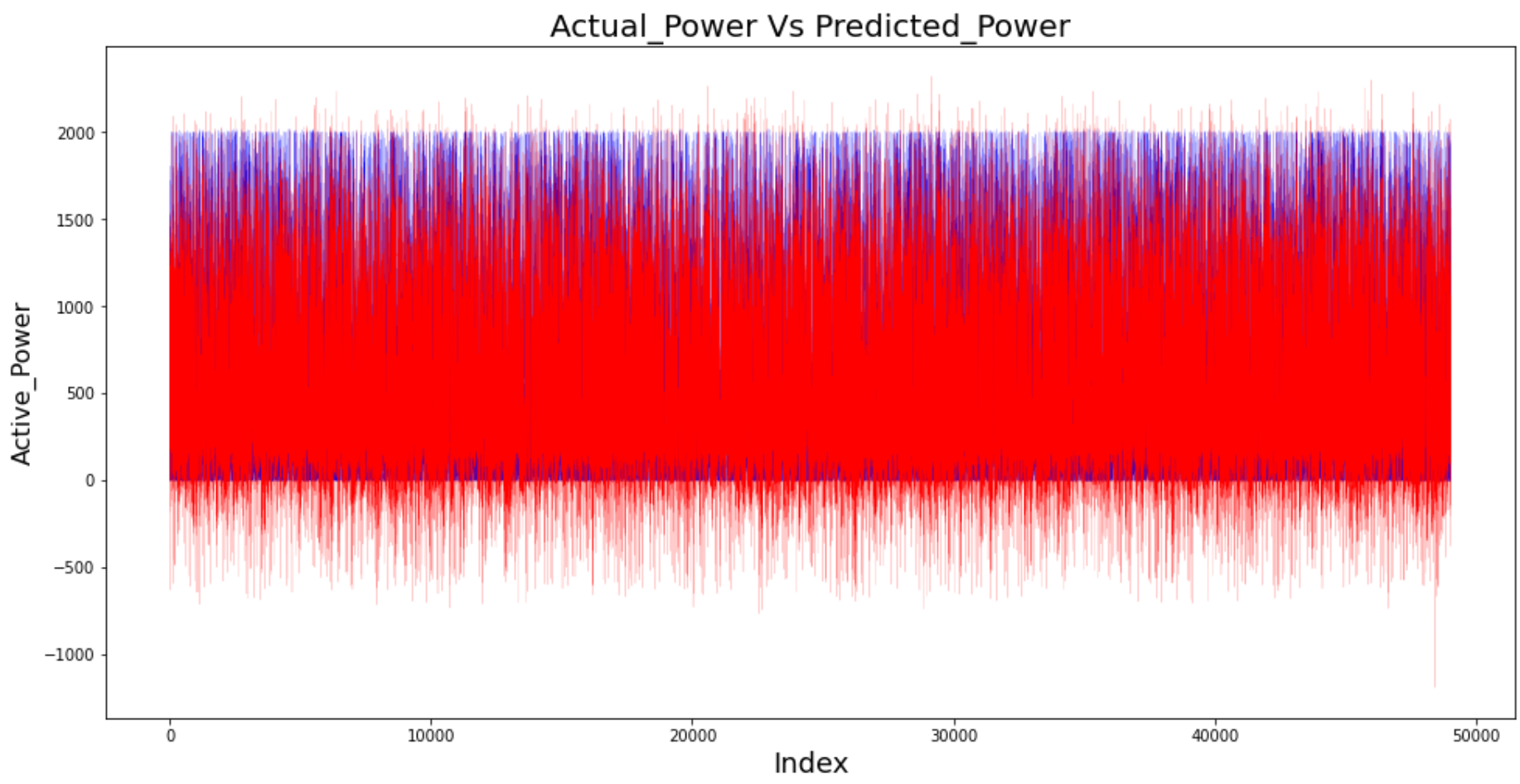
In [23]:
```
lr1 = lr.fit(x1_train,y_train)
```

In [24]:
```
y_pred1 = lr1.predict(x1_test)
```

In [25]:
```
import statsmodels.api as sm
x1_train_sm = x1_train

x1_train_sm = sm.add_constant(x1_train_sm)

mlm1 = sm.OLS(y_train,x1_train_sm).fit()


mlm1.params
print(mlm1.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:       Avg_Active_Power   R-squared:                       0.938
Model:                            OLS   Adj. R-squared:                  0.938
Method:                 Least Squares   F-statistic:                 1.639e+05
Date:                Mon, 10 May 2021   Prob (F-statistic):               0.00
Time:                        00:40:49   Log-Likelihood:            -1.2446e+06
No. Observations:              196159   AIC:                         2.489e+06
Df Residuals:                  196140   BIC:                         2.489e+06
Df Model:                          18
Covariance Type:            nonrobust
=================================================================================================
                                    coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------------------
const                         -3124.6778      9.734   -320.999      0.000   -3143.757   -3105.599
Avg_Ambient_Temp                 -4.8097      0.095    -50.888      0.000      -4.995      -4.624
Avg_Generator_Speed              -0.0036      0.077     -0.047      0.963      -0.154       0.146
Avg_Pitch_Angle                  12.3866      0.045    275.319      0.000      12.298      12.475
Avg_Rotor_Speed                 116.5212      8.183     14.239      0.000     100.483     132.560
Avg_Wind_Speed                   32.1813      0.325     99.090      0.000      31.545      32.818
Bearing_DE_Temp                  -9.4580      0.153    -61.722      0.000      -9.758      -9.158
Bearing_NDE_Temp                 -8.7529      0.158    -55.415      0.000      -9.062      -8.443
Gearbox_bearing_Temp              2.9522      0.256     11.512      0.000       2.450       3.455
Gearbox_oil_Temp                 26.9375      0.341     78.975      0.000      26.269      27.606
Generator_wind_Temp_1           116.8993      2.155     54.242      0.000     112.675     121.123
Generator_wind_Temp_2          -144.2192      2.126    -67.825      0.000    -148.387    -140.052
Generator_wind_Temp_3            34.7784      0.289    120.207      0.000      34.211      35.345
Generators_sliprings_Temp        14.6946      0.154     95.511      0.000      14.393      14.996
Hidraulic_group_pressure          0.2188      0.014     15.428      0.000       0.191       0.247
Nacelle_Misalignment_Avg_Wind_Dir 0.0027      0.002      1.479      0.139      -0.001       0.006
Trafo_1_wind_Temp                -1.9114      0.084    -22.687      0.000      -2.077      -1.746
Trafo_2_wind_Temp                 0.4971      0.071      7.003      0.000       0.358       0.636
Trafo_3_wind_Temp                 3.8216      0.074     51.661      0.000       3.677       3.967
==============================================================================
Omnibus:                    34340.705   Durbin-Watson:                   1.998
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           263122.547
Skew:                           0.636   Prob(JB):                         0.00
Kurtosis:                       8.530   Cond. No.                     3.83e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.83e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

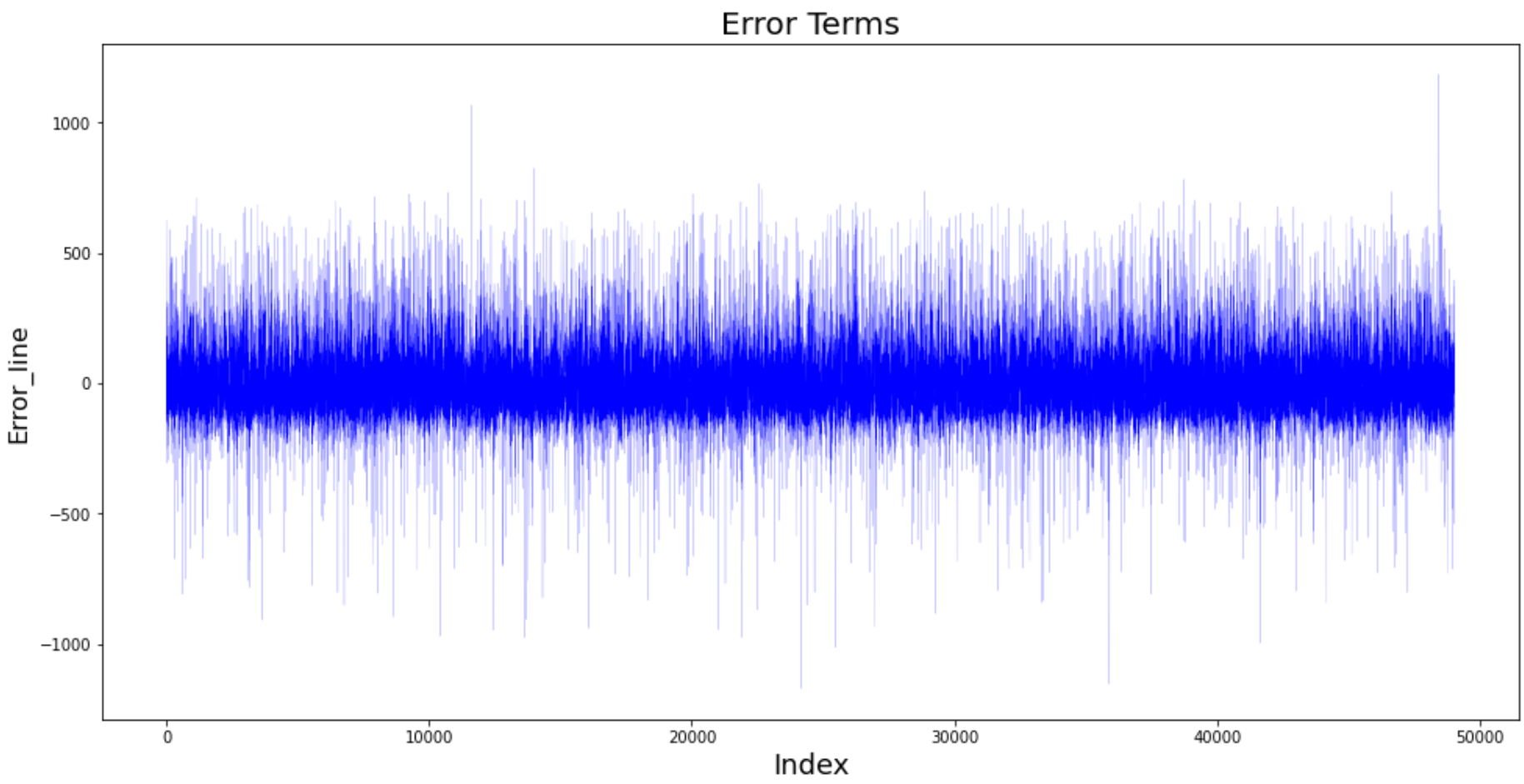
In [26]:
```
#Actual vs Predicted
c = [i for i in range(1,49041,1)]
plt.plot(c,y_test, color="blue", linewidth=0.1, linestyle="-")
plt.plot(c,y_pred1, color="red",  linewidth=0.1, linestyle="-")
plt.title('Actual_Power Vs Predicted_Power', fontsize=20)
plt.xlabel('Index', fontsize=18)
plt.ylabel('Active_Power', fontsize=16)
plt.subplots_adjust(left=0.4, bottom=0.1, right=2.5, top=1.6)
```



In [27]:
```
from statsmodels.stats.outliers_influence import variance_inflation_factor
# VIF dataframe
vif_data = pd.DataFrame()
vif_data["feature"] = x1_train.columns
# calculating VIF for each feature
vif_data["VIF"] = [variance_inflation_factor(x1_train.values, i)
                          for i in range(len(x1_train.columns))]

print(vif_data)
```

	feature	VIF
0	Avg_Ambient_Temp	70.755721
1	Avg_Generator_Speed	83644.968373
2	Avg_Pitch_Angle	13.620940
3	Avg_Rotor_Speed	84271.360161
4	Avg_Wind_Speed	44.235271
5	Bearing_DE_Temp	615.113078
6	Bearing_NDE_Temp	669.097227
7	Gearbox_bearing_Temp	2756.379755
8	Gearbox_oil_Temp	2658.394396
9	Generator_wind_Temp_1	231153.373715
10	Generator_wind_Temp_2	231860.090841
11	Generator_wind_Temp_3	4571.090312
12	Generators_sliprings_Temp	446.646861
13	Hidraulic_group_pressure	81.459161
14	Nacelle_Misalignment_Avg_Wind_Dir	1.930145
15	Trafo_1_wind_Temp	311.207540
16	Trafo_2_wind_Temp	273.126137
17	Trafo_3_wind_Temp	285.184025

In [28]:
```
c = [i for i in range(1,49041,1)]

plt.plot(c,y_test-y_pred, color="blue", linewidth=0.1, linestyle="-")
plt.title('Error Terms', fontsize=20)              # Plot heading
plt.xlabel('Index', fontsize=18)                      # X-label
plt.ylabel('Error_line', fontsize=16)
plt.subplots_adjust(left=0.4, bottom=0.1, right=2.5, top=1.6)
```



In [29]:
```
from sklearn.metrics import mean_squared_error, r2_score
mse = mean_squared_error(y_test, y_pred1)
r_squared = r2_score(y_test, y_pred1)
print('Mean_Squared_Error :' ,mse)
print('r_square_value :', round(r_squared,3),"% Variance of the Active Power is Explained by the Wind Speed, Pitch Angle
```

```
Mean_Squared_Error : 18890.96745365323
r_square_value : 0.938 % Variance of the Active Power is Explained by the Wind Speed, Pitch Angle and Rotor Speed
```

MODEL-3

In [30]:
```
x2_train = x_train[['Avg_Ambient_Temp', 'Avg_Generator_Speed', 'Avg_Pitch_Angle', 'Avg_Rotor_Speed',
                    'Avg_Wind_Speed', 'Bearing_DE_Temp', 'Bearing_NDE_Temp', 'Gearbox_bearing_Temp', 'Gearbox_oil_Temp',
                    'Generator_wind_Temp_1', 'Generator_wind_Temp_2', 'Generator_wind_Temp_3', 'Generators_sliprings_Tem
                    'Hidraulic_group_pressure', 'Trafo_1_wind_Temp', 'Trafo_2_wind_Temp',
                    'Trafo_3_wind_Temp']]
x2_test = x_test[['Avg_Ambient_Temp', 'Avg_Generator_Speed', 'Avg_Pitch_Angle', 'Avg_Rotor_Speed',
                    'Avg_Wind_Speed', 'Bearing_DE_Temp', 'Bearing_NDE_Temp', 'Gearbox_bearing_Temp', 'Gearbox_oil_Temp',
                    'Generator_wind_Temp_1', 'Generator_wind_Temp_2', 'Generator_wind_Temp_3', 'Generators_sliprings_Tem
                    'Hidraulic_group_pressure', 'Trafo_1_wind_Temp', 'Trafo_2_wind_Temp',
                    'Trafo_3_wind_Temp']]
```

Nacelle_Misalignment_Avg_Wind_Dir varible is removed because of its significant value is more than 0.05.

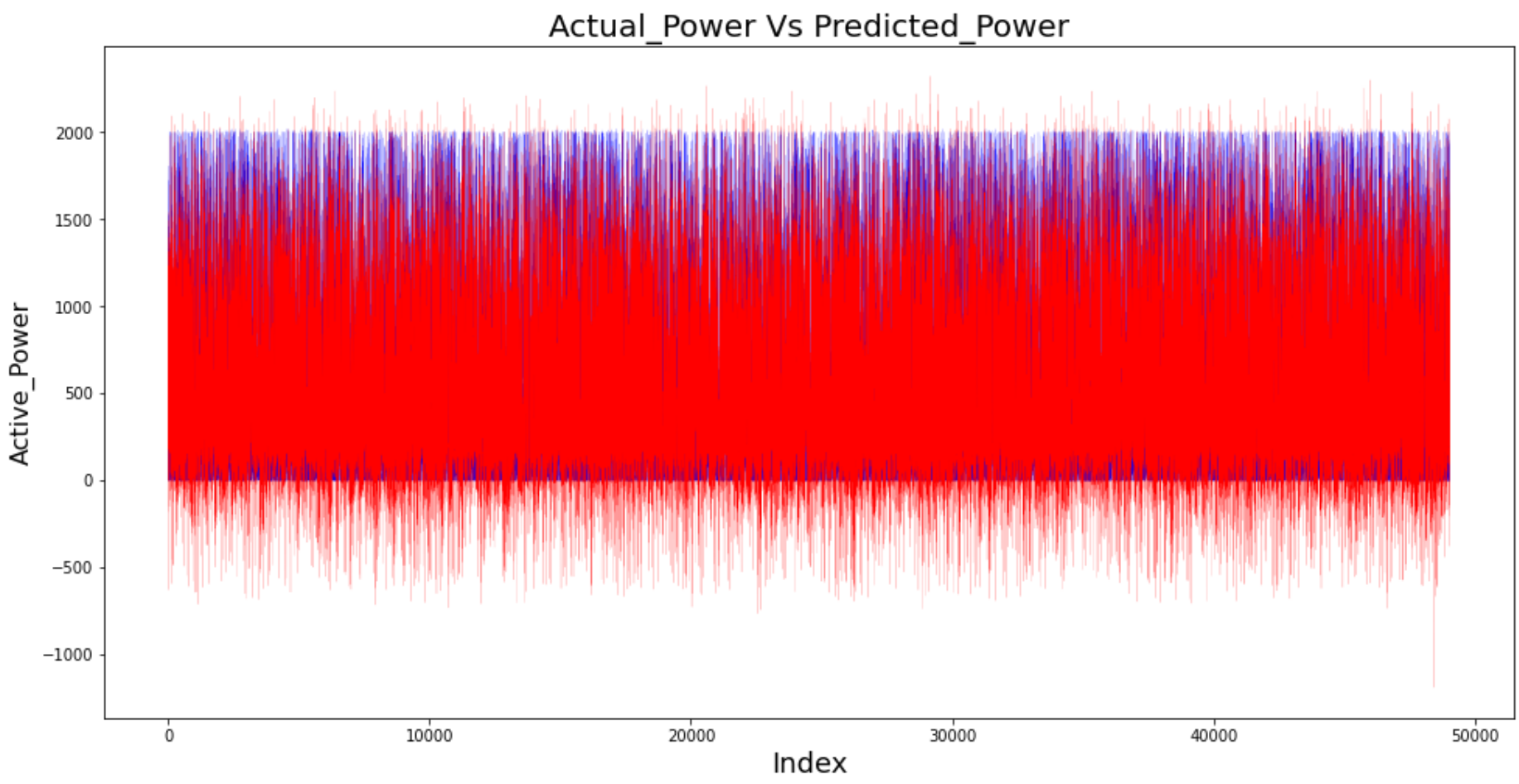
In [31]:
```
lr2 = lr.fit(x2_train,y_train)
```

In [32]:
```
y_pred2 = lr2.predict(x2_test)
```

In [33]:
```
import statsmodels.api as sm
x2_train_sm = x2_train

x2_train_sm = sm.add_constant(x2_train_sm)

mlm2 = sm.OLS(y_train,x2_train_sm).fit()

mlm2.params
print(mlm2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:       Avg_Active_Power   R-squared:                       0.938
Model:                            OLS   Adj. R-squared:                  0.938
Method:                 Least Squares   F-statistic:                 1.735e+05
Date:                Mon, 10 May 2021   Prob (F-statistic):               0.00
Time:                        00:41:03   Log-Likelihood:            -1.2446e+06
No. Observations:              196159   AIC:                         2.489e+06
Df Residuals:                  196141   BIC:                         2.489e+06
Df Model:                          17
Covariance Type:            nonrobust
================================================================================================
                                   coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
const                        -3124.4585      9.733   -321.013      0.000   -3143.535   -3105.382
Avg_Ambient_Temp                -4.7949      0.094    -51.016      0.000      -4.979      -4.611
Avg_Generator_Speed              0.0012      0.076      0.015      0.988      -0.149       0.151
Avg_Pitch_Angle                 12.3884      0.045    275.462      0.000      12.300      12.477
Avg_Rotor_Speed                116.0305      8.176     14.191      0.000     100.005     132.056
Avg_Wind_Speed                  32.1640      0.325     99.101      0.000      31.528      32.800
Bearing_DE_Temp                 -9.4560      0.153    -61.711      0.000      -9.756      -9.156
Bearing_NDE_Temp                -8.7499      0.158    -55.401      0.000      -9.060      -8.440
Gearbox_bearing_Temp             2.9580      0.256     11.536      0.000       2.455       3.461
Gearbox_oil_Temp                26.9373      0.341     78.975      0.000      26.269      27.606
Generator_wind_Temp_1          116.9224      2.155     54.254      0.000     112.699     121.146
Generator_wind_Temp_2         -144.2520      2.126    -67.844      0.000    -148.419    -140.085
Generator_wind_Temp_3           34.7897      0.289    120.287      0.000      34.223      35.357
Generators_sliprings_Temp       14.6734      0.153     95.790      0.000      14.373      14.974
Hidraulic_group_pressure         0.2189      0.014     15.435      0.000       0.191       0.247
Trafo_1_wind_Temp               -1.9121      0.084    -22.696      0.000      -2.077      -1.747
Trafo_2_wind_Temp                0.4958      0.071      6.986      0.000       0.357       0.635
Trafo_3_wind_Temp                3.8223      0.074     51.673      0.000       3.677       3.967
==============================================================================
Omnibus:                    34340.512   Durbin-Watson:                   1.998
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           263101.815
Skew:                           0.636   Prob(JB):                         0.00
Kurtosis:                       8.529   Cond. No.                     3.80e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.8e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

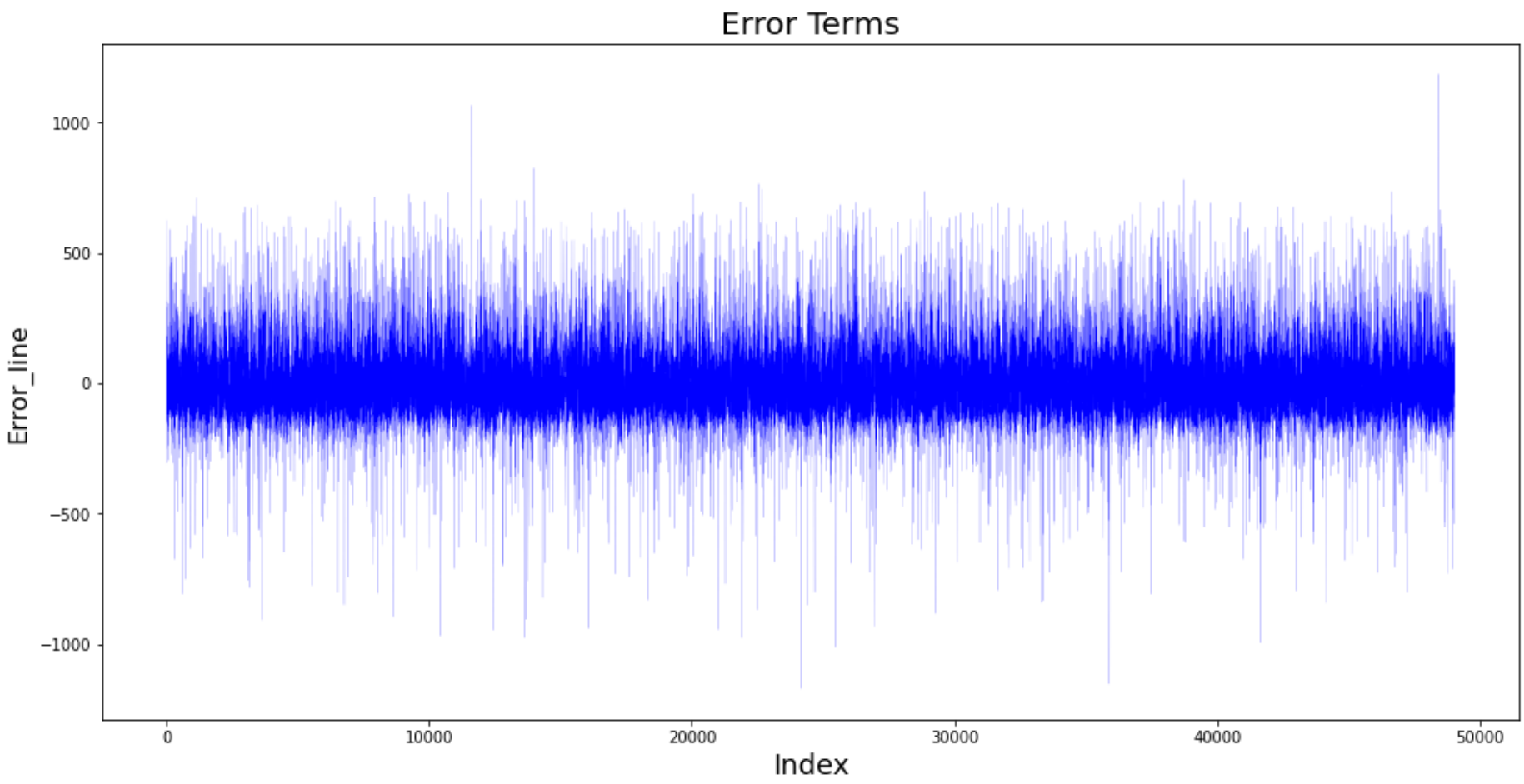
In [34]:
```
#Actual vs Predicted
c = [i for i in range(1,49041,1)]
plt.plot(c,y_test, color="blue", linewidth=0.1, linestyle="-")
plt.plot(c,y_pred2, color="red",  linewidth=0.1, linestyle="-")
plt.title('Actual_Power Vs Predicted_Power', fontsize=20)
plt.xlabel('Index', fontsize=18)
plt.ylabel('Active_Power', fontsize=16)
plt.subplots_adjust(left=0.4, bottom=0.1, right=2.5, top=1.6)
```



In [35]:
```
from statsmodels.stats.outliers_influence import variance_inflation_factor
# VIF dataframe
vif_data = pd.DataFrame()
vif_data["feature"] = x2_train.columns
# calculating VIF for each feature
vif_data["VIF"] = [variance_inflation_factor(x2_train.values, i)
                          for i in range(len(x2_train.columns))]

print(vif_data)
```

	feature	VIF
0	Avg_Ambient_Temp	69.958895
1	Avg_Generator_Speed	83497.589288
2	Avg_Pitch_Angle	13.609280
3	Avg_Rotor_Speed	84131.728964
4	Avg_Wind_Speed	44.178478
5	Bearing_DE_Temp	615.044433
6	Bearing_NDE_Temp	669.020778
7	Gearbox_bearing_Temp	2755.936095
8	Gearbox_oil_Temp	2657.976214
9	Generator_wind_Temp_1	231150.900292
10	Generator_wind_Temp_2	231849.033783
11	Generator_wind_Temp_3	4567.485073
12	Generators_sliprings_Temp	442.858267
13	Hidraulic_group_pressure	81.452678
14	Trafo_1_wind_Temp	311.202159
15	Trafo_2_wind_Temp	273.080192
16	Trafo_3_wind_Temp	285.172116

In [36]:
```
c = [i for i in range(1,49041,1)]

plt.plot(c,y_test-y_pred2, color="blue", linewidth=0.1, linestyle="-")
plt.title('Error Terms', fontsize=20)              # Plot heading
plt.xlabel('Index', fontsize=18)                      # X-label
plt.ylabel('Error_line', fontsize=16)
plt.subplots_adjust(left=0.4, bottom=0.1, right=2.5, top=1.6)
```



In [37]:
```
from sklearn.metrics import mean_squared_error, r2_score
mse = mean_squared_error(y_test, y_pred2)
r_squared = r2_score(y_test, y_pred2)
print('Mean_Squared_Error :' ,mse)
print('r_square_value :', round(r_squared,3),"% Variance of the Active Power is Explained by the Wind Speed, Pitch Angle
```

```
Mean_Squared_Error : 18889.494881377803
r_square_value : 0.938 % Variance of the Active Power is Explained by the Wind Speed, Pitch Angle and Rotor Speed
```

MODEL-4

In [38]:
```
x3_train = x_train[['Avg_Ambient_Temp', 'Avg_Pitch_Angle', 'Avg_Rotor_Speed',
                    'Avg_Wind_Speed', 'Bearing_DE_Temp', 'Bearing_NDE_Temp', 'Gearbox_bearing_Temp', 'Gearbox_oil_Temp',
                    'Generator_wind_Temp_1', 'Generator_wind_Temp_2', 'Generator_wind_Temp_3', 'Generators_sliprings_Tem
                    'Hidraulic_group_pressure', 'Trafo_1_wind_Temp', 'Trafo_2_wind_Temp',
                    'Trafo_3_wind_Temp']]
x3_test = x_test[['Avg_Ambient_Temp', 'Avg_Pitch_Angle', 'Avg_Rotor_Speed',
                    'Avg_Wind_Speed', 'Bearing_DE_Temp', 'Bearing_NDE_Temp', 'Gearbox_bearing_Temp', 'Gearbox_oil_Temp',
                    'Generator_wind_Temp_1', 'Generator_wind_Temp_2', 'Generator_wind_Temp_3', 'Generators_sliprings_Tem
                    'Hidraulic_group_pressure', 'Trafo_1_wind_Temp', 'Trafo_2_wind_Temp',
                    'Trafo_3_wind_Temp']]
```

'Avg_Generator_Speed' varible is removed because of its significant value is more than 0.05.

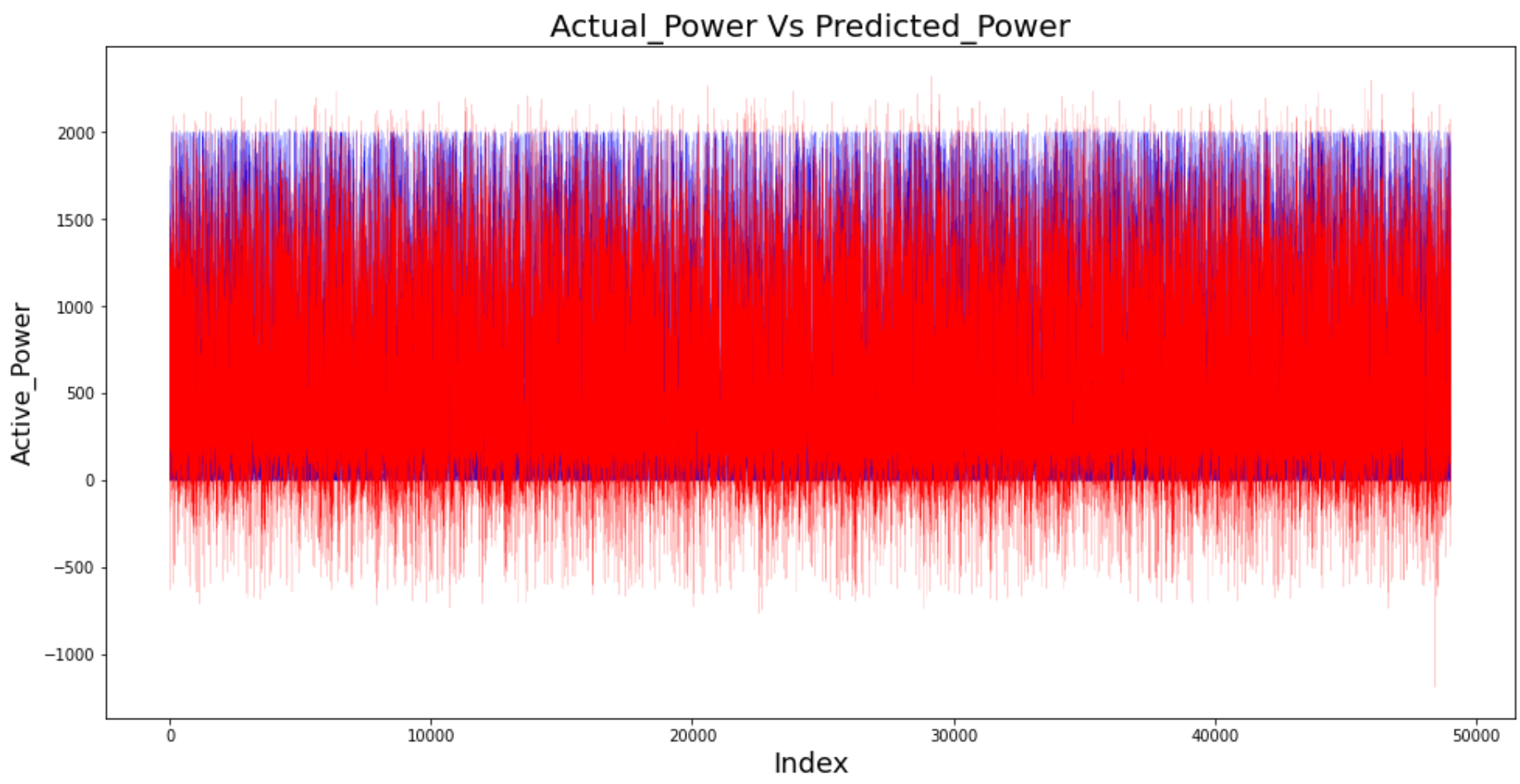
In [39]:
```
lr3 = lr.fit(x3_train,y_train)
```

In [40]:
```
y_pred3 = lr3.predict(x3_test)
```

In [41]:
```
import statsmodels.api as sm
x3_train_sm = x3_train

x3_train_sm = sm.add_constant(x3_train_sm)

mlm3 = sm.OLS(y_train,x3_train_sm).fit()

mlm3.params
print(mlm3.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:       Avg_Active_Power   R-squared:                       0.938
Model:                            OLS   Adj. R-squared:                  0.938
Method:                 Least Squares   F-statistic:                 1.844e+05
Date:                Mon, 10 May 2021   Prob (F-statistic):               0.00
Time:                        00:41:14   Log-Likelihood:            -1.2446e+06
No. Observations:              196159   AIC:                         2.489e+06
Df Residuals:                  196142   BIC:                         2.489e+06
Df Model:                          16
Covariance Type:            nonrobust
================================================================================================
                                   coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
const                        -3124.4575      9.733   -321.021      0.000   -3143.534   -3105.381
Avg_Ambient_Temp                -4.7949      0.094    -51.025      0.000      -4.979      -4.611
Avg_Pitch_Angle                 12.3884      0.045    275.466      0.000      12.300      12.477
Avg_Rotor_Speed                116.1561      0.332    349.570      0.000     115.505     116.807
Avg_Wind_Speed                  32.1635      0.323     99.522      0.000      31.530      32.797
Bearing_DE_Temp                 -9.4560      0.153    -61.714      0.000      -9.756      -9.156
Bearing_NDE_Temp                -8.7500      0.158    -55.406      0.000      -9.060      -8.440
Gearbox_bearing_Temp             2.9578      0.256     11.550      0.000       2.456       3.460
Gearbox_oil_Temp                26.9375      0.341     79.005      0.000      26.269      27.606
Generator_wind_Temp_1          116.9223      2.155     54.255      0.000     112.698     121.146
Generator_wind_Temp_2         -144.2517      2.126    -67.845      0.000    -148.419    -140.084
Generator_wind_Temp_3           34.7897      0.289    120.288      0.000      34.223      35.357
Generators_sliprings_Temp       14.6733      0.153     95.838      0.000      14.373      14.973
Hidraulic_group_pressure         0.2189      0.014     15.436      0.000       0.191       0.247
Trafo_1_wind_Temp               -1.9121      0.084    -22.696      0.000      -2.077      -1.747
Trafo_2_wind_Temp                0.4958      0.071      6.989      0.000       0.357       0.635
Trafo_3_wind_Temp                3.8223      0.074     51.687      0.000       3.677       3.967
==============================================================================
Omnibus:                    34340.835   Durbin-Watson:                   1.998
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           263102.847
Skew:                           0.636   Prob(JB):                         0.00
Kurtosis:                       8.529   Cond. No.                     9.34e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 9.34e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

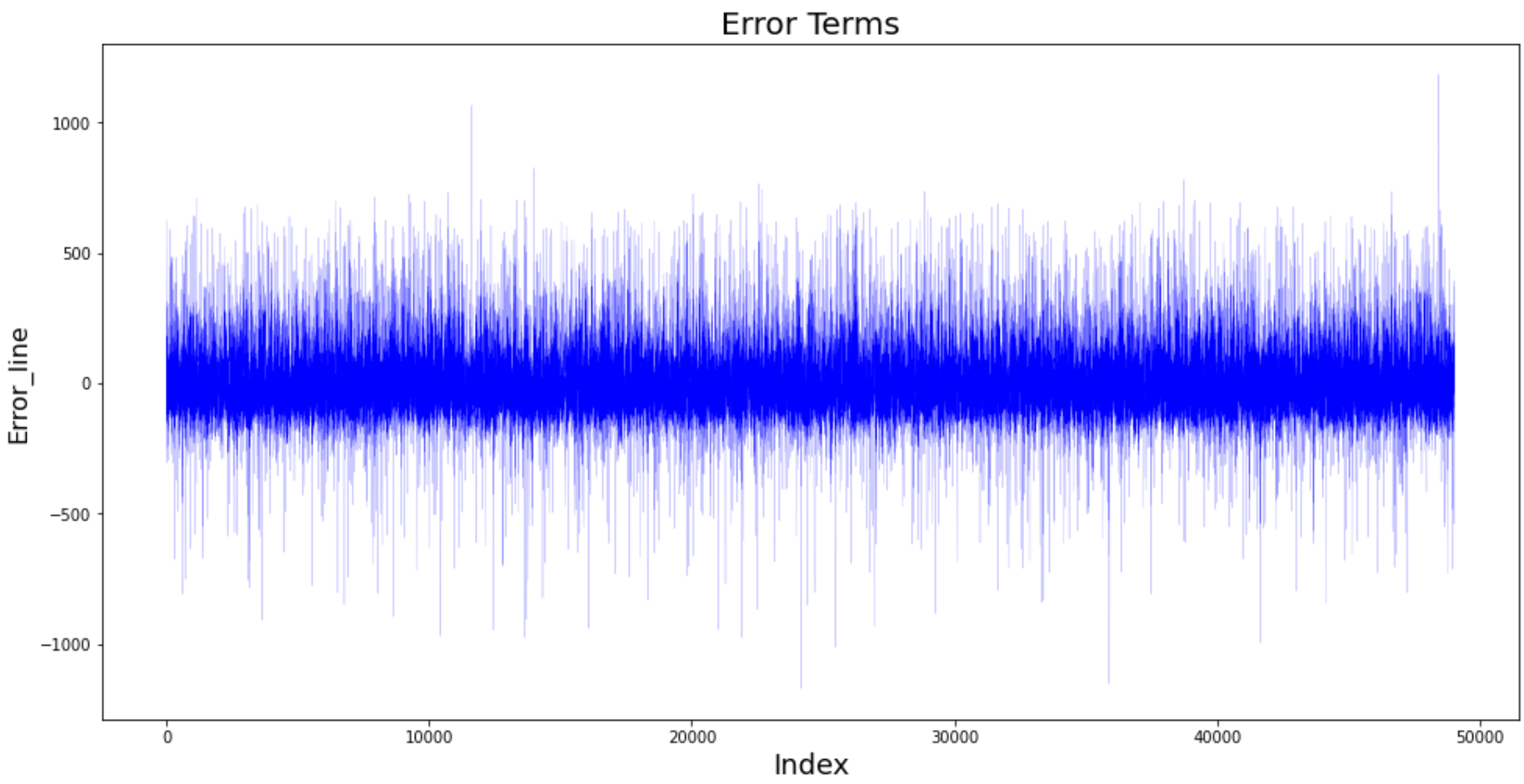
In [42]:
```
#Actual vs Predicted
c = [i for i in range(1,49041,1)]
plt.plot(c,y_test, color="blue", linewidth=0.1, linestyle="-")
plt.plot(c,y_pred3, color="red",  linewidth=0.1, linestyle="-")
plt.title('Actual_Power Vs Predicted_Power', fontsize=20)
plt.xlabel('Index', fontsize=18)
plt.ylabel('Active_Power', fontsize=16)
plt.subplots_adjust(left=0.4, bottom=0.1, right=2.5, top=1.6)
```



In [43]:
```
from statsmodels.stats.outliers_influence import variance_inflation_factor
# VIF dataframe
vif_data = pd.DataFrame()
vif_data["feature"] = x3_train.columns
# calculating VIF for each feature
vif_data["VIF"] = [variance_inflation_factor(x3_train.values, i)
                          for i in range(len(x3_train.columns))]

print(vif_data)
```

```
                     feature           VIF
0            Avg_Ambient_Temp     69.934265
1             Avg_Pitch_Angle     13.608803
2             Avg_Rotor_Speed    137.958877
3              Avg_Wind_Speed     43.805196
4             Bearing_DE_Temp    614.971703
5            Bearing_NDE_Temp    668.889859
6        Gearbox_bearing_Temp   2748.459206
7            Gearbox_oil_Temp   2653.258020
8       Generator_wind_Temp_1  231139.296838
9       Generator_wind_Temp_2  231830.976611
10      Generator_wind_Temp_3    4567.470518
11  Generators_sliprings_Temp     442.432227
12   Hidraulic_group_pressure      81.429600
13          Trafo_1_wind_Temp     311.197941
14          Trafo_2_wind_Temp     272.863521
15          Trafo_3_wind_Temp     285.012433
```

In [44]:
```
c = [i for i in range(1,49041,1)]

plt.plot(c,y_test-y_pred3, color="blue", linewidth=0.1, linestyle="-")
plt.title('Error Terms', fontsize=20)              # Plot heading
plt.xlabel('Index', fontsize=18)                      # X-label
plt.ylabel('Error_line', fontsize=16)
plt.subplots_adjust(left=0.4, bottom=0.1, right=2.5, top=1.6)
```



In [45]:
```
from sklearn.metrics import mean_squared_error, r2_score
mse = mean_squared_error(y_test, y_pred3)
r_squared = r2_score(y_test, y_pred3)
print('Mean_Squared_Error :' ,mse)
print('r_square_value :', round(r_squared,3),"% Variance of the Active Power is Explained by the Wind Speed, Pitch Angle
```

```
Mean_Squared_Error : 18889.474403181845
r_square_value : 0.938 % Variance of the Active Power is Explained by the Wind Speed, Pitch Angle and Rotor Speed
```

MODEL-5

In [46]:
```
x4_train = x_train[['Avg_Ambient_Temp', 'Avg_Pitch_Angle', 'Avg_Rotor_Speed', 'Avg_Wind_Speed', 'Bearing_DE_Temp',
                    'Gearbox_bearing_Temp', 'Gearbox_oil_Temp', 'Generator_wind_Temp_1',
                    'Generator_wind_Temp_2', 'Generator_wind_Temp_3', 'Generators_sliprings_Temp',
                    'Hidraulic_group_pressure', 'Trafo_1_wind_Temp', 'Trafo_2_wind_Temp',
                    'Trafo_3_wind_Temp']]
x4_test = x_test[['Avg_Ambient_Temp', 'Avg_Pitch_Angle', 'Avg_Rotor_Speed', 'Avg_Wind_Speed', 'Bearing_DE_Temp',
                    'Gearbox_bearing_Temp', 'Gearbox_oil_Temp', 'Generator_wind_Temp_1',
                    'Generator_wind_Temp_2', 'Generator_wind_Temp_3', 'Generators_sliprings_Temp',
                    'Hidraulic_group_pressure', 'Trafo_1_wind_Temp', 'Trafo_2_wind_Temp',
                    'Trafo_3_wind_Temp']]
```

'Bearing_NDE_Temp' & 'Bearing_DE_Temp' These varibles are Auto Correlated each other. So, i removed one variable ('Bearing_NDE_Temp')

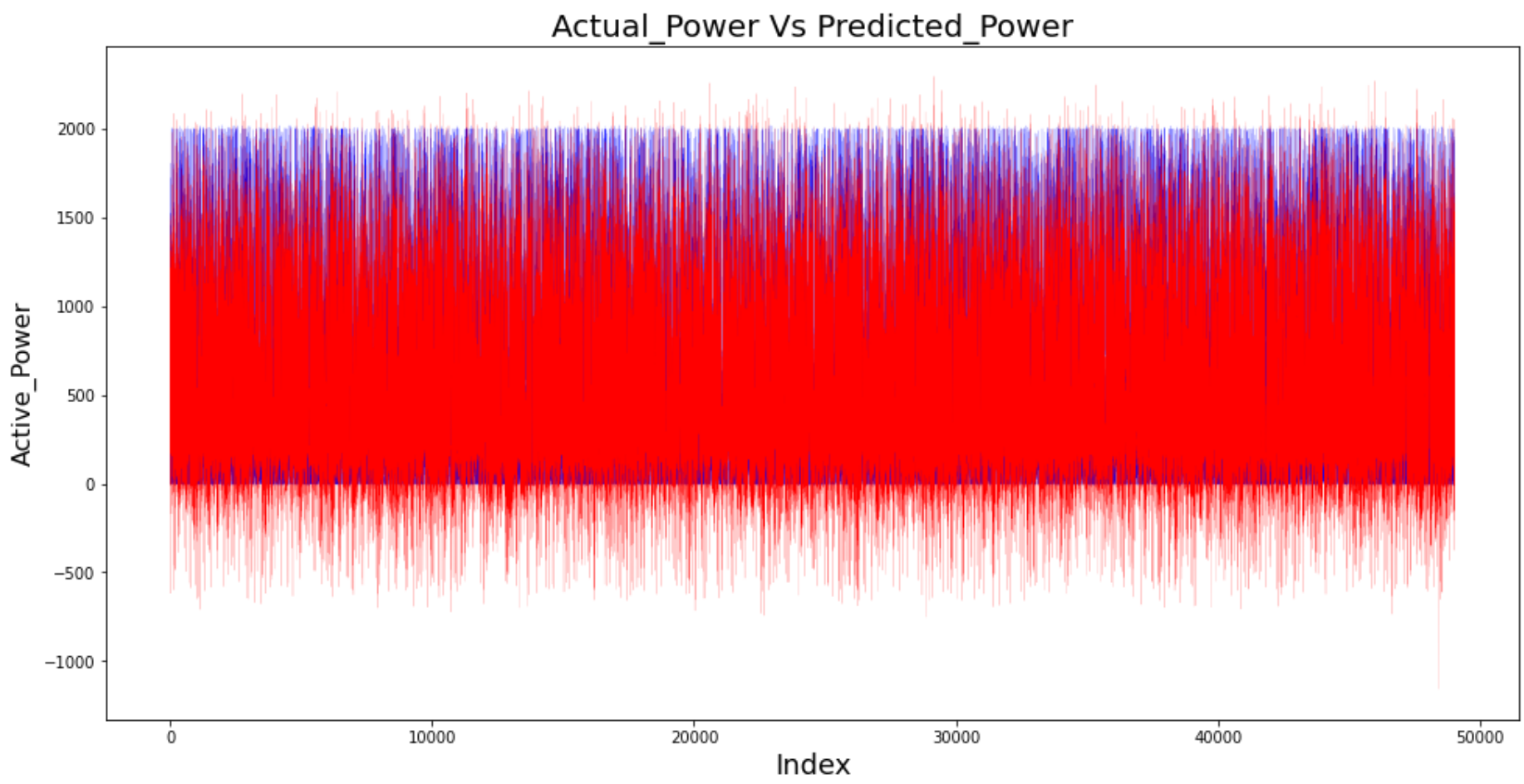
In [47]:
```
lr4 = lr.fit(x4_train,y_train)
```

In [48]:
```
y_pred4 = lr4.predict(x4_test)
```

In [49]:
```
import statsmodels.api as sm
x4_train_sm = x4_train

x4_train_sm = sm.add_constant(x4_train_sm)

mlm4 = sm.OLS(y_train,x4_train_sm).fit()

mlm4.params
print(mlm4.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:       Avg_Active_Power   R-squared:                       0.937
Model:                            OLS   Adj. R-squared:                  0.937
Method:                 Least Squares   F-statistic:                 1.934e+05
Date:                Mon, 10 May 2021   Prob (F-statistic):               0.00
Time:                        00:41:23   Log-Likelihood:            -1.2461e+06
No. Observations:              196159   AIC:                         2.492e+06
Df Residuals:                  196143   BIC:                         2.492e+06
Df Model:                          15
Covariance Type:            nonrobust
================================================================================================
                                   coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
const                        -3052.2801      9.720   -314.006      0.000   -3071.332   -3033.228
Avg_Ambient_Temp                -4.7923      0.095    -50.603      0.000      -4.978      -4.607
Avg_Pitch_Angle                 12.3403      0.045    272.326      0.000      12.252      12.429
Avg_Rotor_Speed                116.4816      0.335    347.893      0.000     115.825     117.138
Avg_Wind_Speed                  32.7520      0.326    100.614      0.000      32.114      33.390
Bearing_DE_Temp                -14.3852      0.126   -114.421      0.000     -14.632     -14.139
Gearbox_bearing_Temp             2.7047      0.258     10.482      0.000       2.199       3.211
Gearbox_oil_Temp                25.0503      0.342     73.269      0.000      24.380      25.720
Generator_wind_Temp_1          117.0581      2.172     53.898      0.000     112.801     121.315
Generator_wind_Temp_2         -144.9050      2.143    -67.627      0.000    -149.105    -140.705
Generator_wind_Temp_3           34.3993      0.291    118.053      0.000      33.828      34.970
Generators_sliprings_Temp       12.7109      0.150     84.675      0.000      12.417      13.005
Hidraulic_group_pressure         0.2117      0.014     14.814      0.000       0.184       0.240
Trafo_1_wind_Temp               -1.3993      0.084    -16.582      0.000      -1.565      -1.234
Trafo_2_wind_Temp                0.5641      0.071      7.891      0.000       0.424       0.704
Trafo_3_wind_Temp                3.3687      0.074     45.481      0.000       3.224       3.514
==============================================================================
Omnibus:                    32465.610   Durbin-Watson:                   1.998
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           253064.039
Skew:                           0.582   Prob(JB):                         0.00
Kurtosis:                       8.441   Cond. No.                     9.11e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 9.11e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

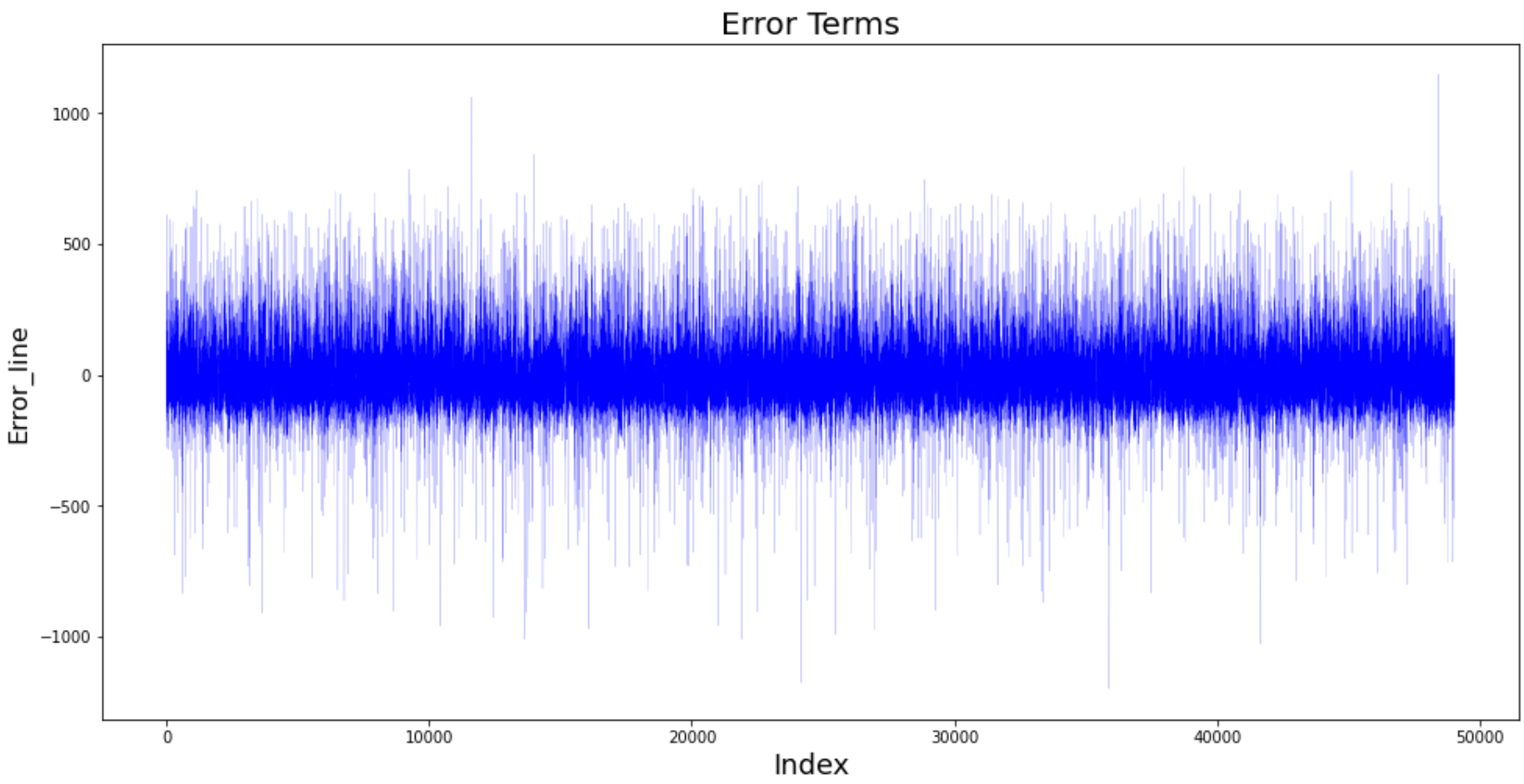
In [50]:
```
#Actual vs Predicted
c = [i for i in range(1,49041,1)]
plt.plot(c,y_test, color="blue", linewidth=0.1, linestyle="-")
plt.plot(c,y_pred4, color="red",  linewidth=0.1, linestyle="-")
plt.title('Actual_Power Vs Predicted_Power', fontsize=20)
plt.xlabel('Index', fontsize=18)
plt.ylabel('Active_Power', fontsize=16)
plt.subplots_adjust(left=0.4, bottom=0.1, right=2.5, top=1.6)
```



In [51]:
```
from statsmodels.stats.outliers_influence import variance_inflation_factor
# VIF dataframe
vif_data = pd.DataFrame()
vif_data["feature"] = x4_train.columns
# calculating VIF for each feature
vif_data["VIF"] = [variance_inflation_factor(x4_train.values, i)
                           for i in range(len(x4_train.columns))]

print(vif_data)
```

	feature	VIF
0	Avg_Ambient_Temp	69.931381
1	Avg_Pitch_Angle	13.608564
2	Avg_Rotor_Speed	137.953312
3	Avg_Wind_Speed	43.748536
4	Bearing_DE_Temp	411.876170
5	Gearbox_bearing_Temp	2742.710813
6	Gearbox_oil_Temp	2652.418117
7	Generator_wind_Temp_1	230826.581554
8	Generator_wind_Temp_2	231643.128321
9	Generator_wind_Temp_3	4566.958546
10	Generators_sliprings_Temp	421.338858
11	Hidraulic_group_pressure	81.347775
12	Trafo_1_wind_Temp	306.452596
13	Trafo_2_wind_Temp	272.843233
14	Trafo_3_wind_Temp	281.161026

In [52]:
```
c = [i for i in range(1,49041,1)]

plt.plot(c,y_test-y_pred4, color="blue", linewidth=0.1, linestyle="-")
plt.title('Error Terms', fontsize=20)              # Plot heading
plt.xlabel('Index', fontsize=18)                      # X-label
plt.ylabel('Error_line', fontsize=16)
plt.subplots_adjust(left=0.4, bottom=0.1, right=2.5, top=1.6)
```



In [53]:
```
from sklearn.metrics import mean_squared_error, r2_score
mse = mean_squared_error(y_test, y_pred4)
r_squared = r2_score(y_test, y_pred4)
print('Mean_Squared_Error :' ,mse)
print('r_square_value :', round(r_squared,3),"% Variance of the Active Power is Explained by the Wind Speed, Pitch Angle
```

```
Mean_Squared_Error : 19200.959868118574
r_square_value : 0.937 % Variance of the Active Power is Explained by the Wind Speed, Pitch Angle and Rotor Speed
```

MODEL-6

In [54]:
```
x5_train = x_train[['Avg_Ambient_Temp', 'Avg_Pitch_Angle', 'Avg_Rotor_Speed', 'Avg_Wind_Speed', 'Bearing_DE_Temp',
                    'Gearbox_bearing_Temp', 'Gearbox_oil_Temp', 'Generator_wind_Temp_1',
                    'Generator_wind_Temp_3', 'Generators_sliprings_Temp', 'Hidraulic_group_pressure', 'Trafo_1_wind_Temp
                    'Trafo_2_wind_Temp', 'Trafo_3_wind_Temp']]
x5_test = x_test[['Avg_Ambient_Temp', 'Avg_Pitch_Angle', 'Avg_Rotor_Speed', 'Avg_Wind_Speed', 'Bearing_DE_Temp',
                    'Gearbox_bearing_Temp', 'Gearbox_oil_Temp', 'Generator_wind_Temp_1',
                    'Generator_wind_Temp_3', 'Generators_sliprings_Temp', 'Hidraulic_group_pressure', 'Trafo_1_wind_Temp
                    'Trafo_2_wind_Temp', 'Trafo_3_wind_Temp']]
```

'Generator_wind_Temp_1', 'Generator_wind_Temp_2' These varibles are Auto Correlated each other. So, i removed one variable ('Generator_wind_Temp_2')

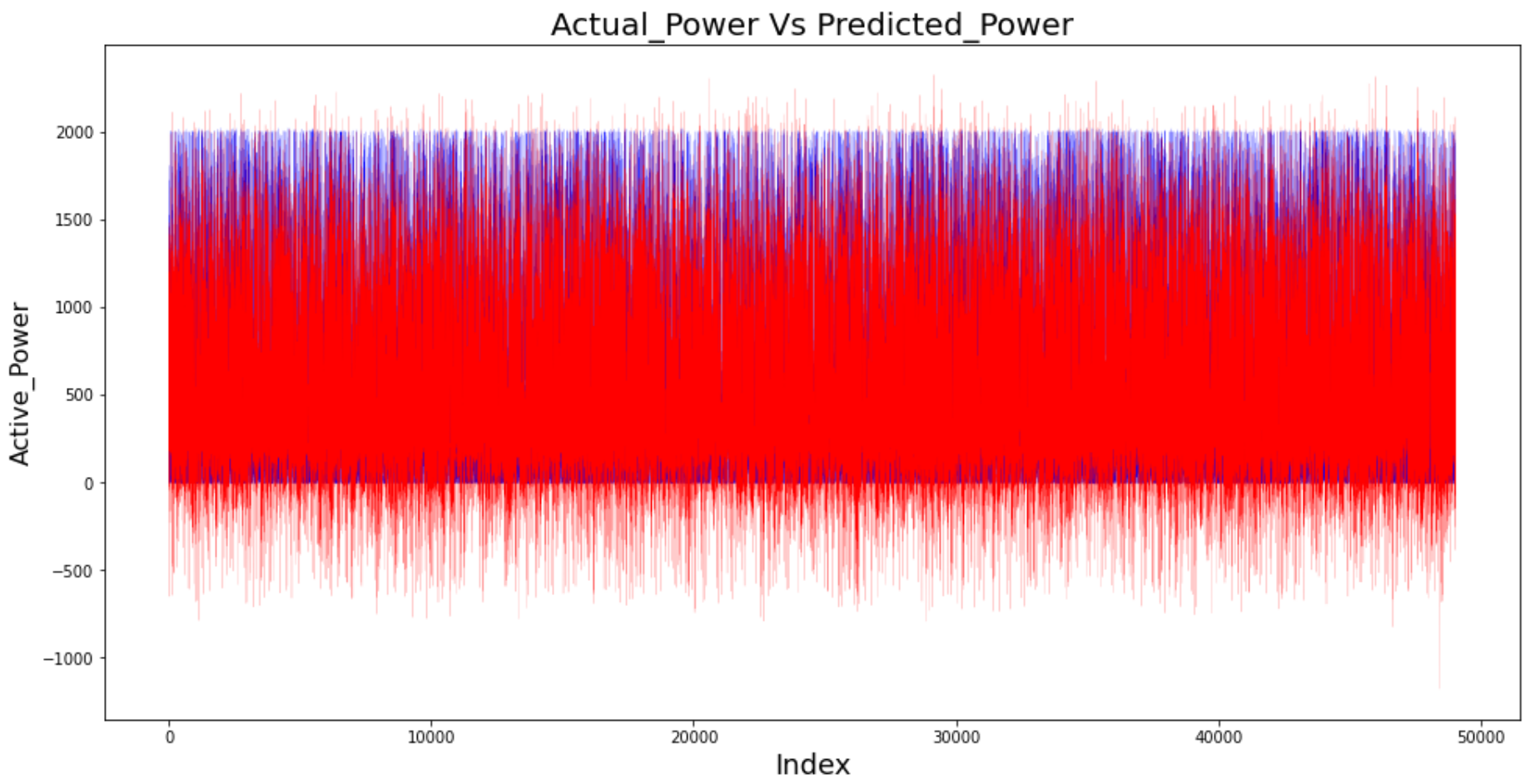
In [55]:
```
lr5 = lr.fit(x5_train,y_train)
```

In [56]:
```
y_pred5 = lr5.predict(x5_test)
```

In [57]:
```
import statsmodels.api as sm
x5_train_sm = x5_train

x5_train_sm = sm.add_constant(x5_train_sm)

mlm5 = sm.OLS(y_train,x5_train_sm).fit()

mlm5.params
print(mlm5.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:       Avg_Active_Power   R-squared:                       0.935
Model:                            OLS   Adj. R-squared:                  0.935
Method:                 Least Squares   F-statistic:                 2.022e+05
Date:                Mon, 10 May 2021   Prob (F-statistic):               0.00
Time:                        00:41:31   Log-Likelihood:            -1.2484e+06
No. Observations:              196159   AIC:                         2.497e+06
Df Residuals:                  196144   BIC:                         2.497e+06
Df Model:                          14
Covariance Type:            nonrobust
================================================================================================
                                   coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
const                        -3214.7902      9.528   -337.409      0.000   -3233.465   -3196.116
Avg_Ambient_Temp                -5.2072      0.096    -54.469      0.000      -5.395      -5.020
Avg_Pitch_Angle                 13.5097      0.042    318.841      0.000      13.427      13.593
Avg_Rotor_Speed                116.8875      0.339    345.162      0.000     116.224     117.551
Avg_Wind_Speed                  38.0114      0.320    118.876      0.000      37.385      38.638
Bearing_DE_Temp                -13.9962      0.127   -110.167      0.000     -14.245     -13.747
Gearbox_bearing_Temp             1.1050      0.260      4.251      0.000       0.596       1.614
Gearbox_oil_Temp                26.3377      0.345     76.270      0.000      25.661      27.014
Generator_wind_Temp_1          -28.4262      0.302    -94.229      0.000     -29.017     -27.835
Generator_wind_Temp_3           34.8105      0.295    118.121      0.000      34.233      35.388
Generators_sliprings_Temp       13.4166      0.151     88.567      0.000      13.120      13.714
Hidraulic_group_pressure         0.2348      0.014     16.246      0.000       0.206       0.263
Trafo_1_wind_Temp               -2.0116      0.085    -23.700      0.000      -2.178      -1.845
Trafo_2_wind_Temp                0.9973      0.072     13.847      0.000       0.856       1.139
Trafo_3_wind_Temp                3.7910      0.075     50.777      0.000       3.645       3.937
==============================================================================
Omnibus:                    38016.002   Durbin-Watson:                   2.000
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           287712.537
Skew:                           0.732   Prob(JB):                         0.00
Kurtosis:                       8.750   Cond. No.                     8.53e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 8.53e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

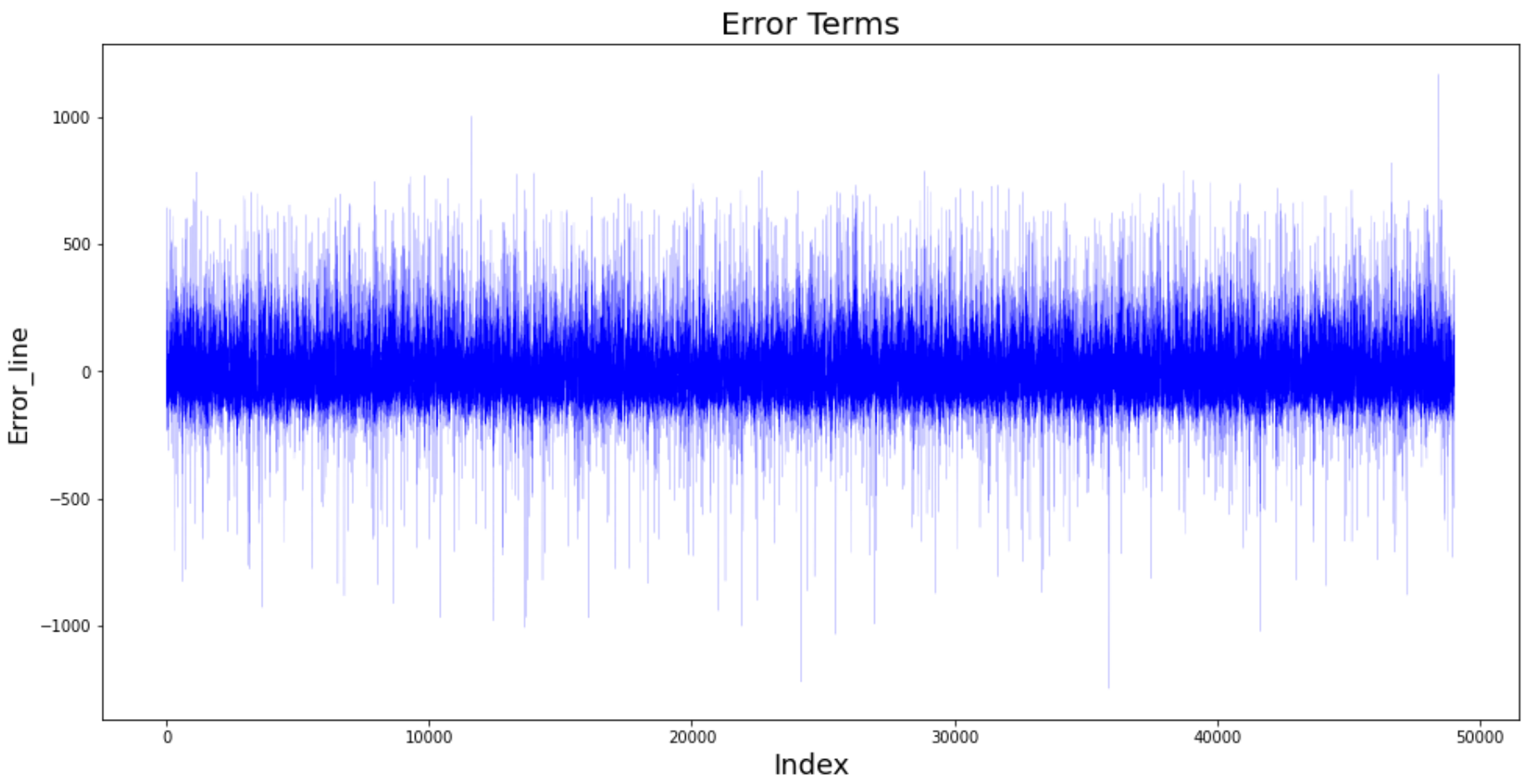
In [58]:
```
#Actual vs Predicted
c = [i for i in range(1,49041,1)]
plt.plot(c,y_test, color="blue", linewidth=0.1, linestyle="-")
plt.plot(c,y_pred5, color="red",  linewidth=0.1, linestyle="-")
plt.title('Actual_Power Vs Predicted_Power', fontsize=20)
plt.xlabel('Index', fontsize=18)
plt.ylabel('Active_Power', fontsize=16)
plt.subplots_adjust(left=0.4, bottom=0.1, right=2.5, top=1.6)
```



In [59]:
```
from statsmodels.stats.outliers_influence import variance_inflation_factor
# VIF dataframe
vif_data = pd.DataFrame()
vif_data["feature"] = x5_train.columns
# calculating VIF for each feature
vif_data["VIF"] = [variance_inflation_factor(x5_train.values, i)
                           for i in range(len(x5_train.columns))]

print(vif_data)
```

```
                     feature         VIF
0            Avg_Ambient_Temp   69.503414
1             Avg_Pitch_Angle   11.769708
2             Avg_Rotor_Speed  137.732875
3              Avg_Wind_Speed   41.224769
4             Bearing_DE_Temp  411.291111
5        Gearbox_bearing_Temp  2737.457889
6            Gearbox_oil_Temp  2601.629653
7       Generator_wind_Temp_1  4492.552730
8       Generator_wind_Temp_3  4566.868338
9   Generators_sliprings_Temp   420.397298
10  Hidraulic_group_pressure    81.121386
11         Trafo_1_wind_Temp  300.566284
12         Trafo_2_wind_Temp  269.594255
13         Trafo_3_wind_Temp  278.373809
```

In [60]:
```
c = [i for i in range(1,49041,1)]

plt.plot(c,y_test-y_pred5, color="blue", linewidth=0.1, linestyle="-")
plt.title('Error Terms', fontsize=20)              # Plot heading
plt.xlabel('Index', fontsize=18)                      # X-label
plt.ylabel('Error_line', fontsize=16)
plt.subplots_adjust(left=0.4, bottom=0.1, right=2.5, top=1.6)
```



In [61]:
```
from sklearn.metrics import mean_squared_error, r2_score
mse = mean_squared_error(y_test, y_pred5)
r_squared = r2_score(y_test, y_pred5)
print('Mean_Squared_Error :' ,mse)
print('r_square_value :', round(r_squared,3),"% Variance of the Active Power is Explained by the Wind Speed, Pitch Angle
```

```
Mean_Squared_Error : 19596.03455227658
r_square_value : 0.935 % Variance of the Active Power is Explained by the Wind Speed, Pitch Angle and Rotor Speed
```

MODEL-7

In [62]:
```
x6_train = x_train[['Avg_Ambient_Temp', 'Avg_Pitch_Angle', 'Avg_Rotor_Speed', 'Avg_Wind_Speed', 'Bearing_DE_Temp',
                    'Gearbox_bearing_Temp', 'Gearbox_oil_Temp', 'Generator_wind_Temp_1', 'Generators_sliprings_Temp',
                    'Hidraulic_group_pressure', 'Trafo_1_wind_Temp', 'Trafo_2_wind_Temp', 'Trafo_3_wind_Temp']]
x6_test = x_test[['Avg_Ambient_Temp', 'Avg_Pitch_Angle', 'Avg_Rotor_Speed', 'Avg_Wind_Speed', 'Bearing_DE_Temp',
                    'Gearbox_bearing_Temp', 'Gearbox_oil_Temp', 'Generator_wind_Temp_1', 'Generators_sliprings_Temp',
                    'Hidraulic_group_pressure', 'Trafo_1_wind_Temp', 'Trafo_2_wind_Temp', 'Trafo_3_wind_Temp']]
```

'Generator_wind_Temp_1', 'Generator_wind_Temp_3' These varibles are Auto Correlated each other. So, i removed one variable ('Generator_wind_Temp_3')

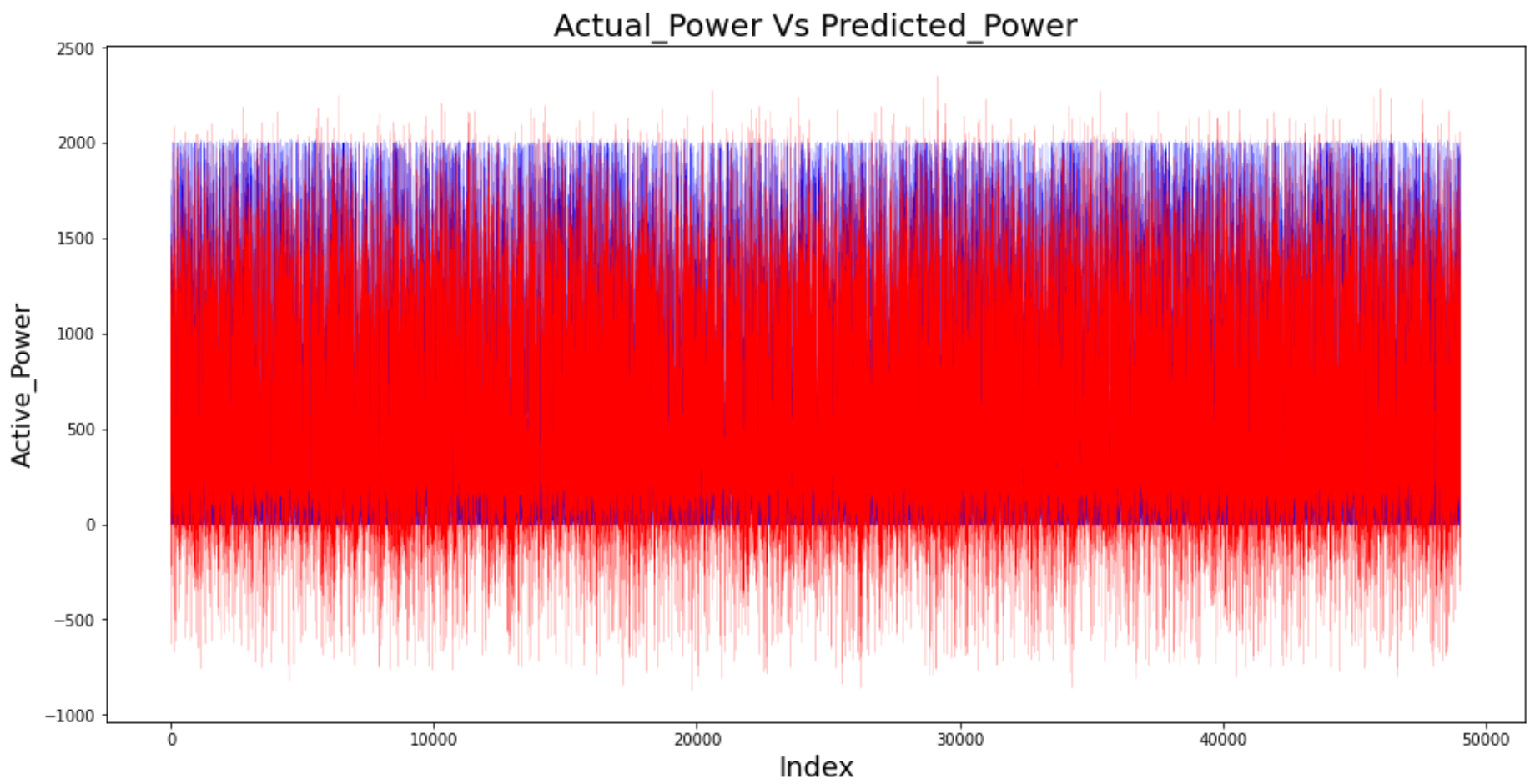
In [63]:
```
lr6 = lr.fit(x6_train,y_train)
```

In [64]:
```
y_pred6 = lr6.predict(x6_test)
```

In [65]:
```
import statsmodels.api as sm
x6_train_sm = x6_train

x6_train_sm = sm.add_constant(x6_train_sm)

mlm6 = sm.OLS(y_train,x6_train_sm).fit()

mlm6.params
print(mlm6.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:       Avg_Active_Power   R-squared:                       0.931
Model:                            OLS   Adj. R-squared:                  0.931
Method:                 Least Squares   F-statistic:                 2.023e+05
Date:                Mon, 10 May 2021   Prob (F-statistic):               0.00
Time:                        00:41:37   Log-Likelihood:            -1.2551e+06
No. Observations:              196159   AIC:                         2.510e+06
Df Residuals:                  196145   BIC:                         2.510e+06
Df Model:                          13
Covariance Type:            nonrobust
================================================================================================
                                   coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
const                        -3102.5963      9.812   -316.211      0.000   -3121.827   -3083.365
Avg_Ambient_Temp                -6.5052      0.098    -66.187      0.000      -6.698      -6.313
Avg_Pitch_Angle                 14.1497      0.043    325.337      0.000      14.064      14.235
Avg_Rotor_Speed                120.3763      0.349    344.772      0.000     119.692     121.061
Avg_Wind_Speed                  36.8071      0.331    111.278      0.000      36.159      37.455
Bearing_DE_Temp                -14.6500      0.131   -111.524      0.000     -14.907     -14.393
Gearbox_bearing_Temp            -2.0452      0.268     -7.643      0.000      -2.570      -1.521
Gearbox_oil_Temp                27.2033      0.357     76.133      0.000      26.503      27.904
Generator_wind_Temp_1            5.8439      0.086     68.316      0.000       5.676       6.012
Generators_sliprings_Temp       16.2227      0.155    104.770      0.000      15.919      16.526
Hidraulic_group_pressure         0.2097      0.015     14.018      0.000       0.180       0.239
Trafo_1_wind_Temp               -2.4433      0.088    -27.840      0.000      -2.615      -2.271
Trafo_2_wind_Temp                1.2977      0.074     17.420      0.000       1.152       1.444
Trafo_3_wind_Temp                4.3404      0.077     56.280      0.000       4.189       4.492
==============================================================================
Omnibus:                    44311.715   Durbin-Watson:                   2.000
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           302008.341
Skew:                           0.917   Prob(JB):                         0.00
Kurtosis:                       8.795   Cond. No.                     8.21e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 8.21e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

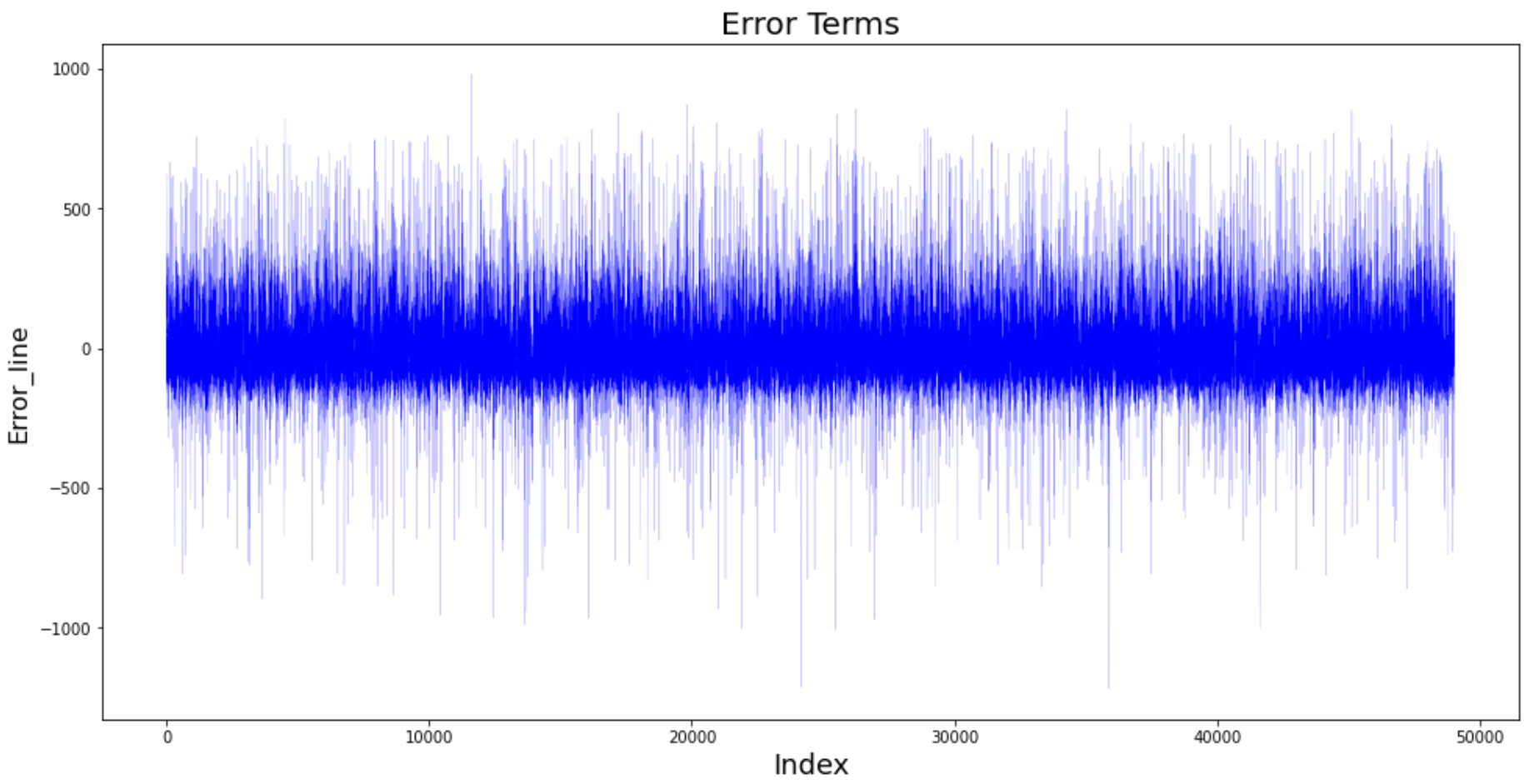
In [66]:
```
#Actual vs Predicted
c = [i for i in range(1,49041,1)]
plt.plot(c,y_test, color="blue", linewidth=0.1, linestyle="-")
plt.plot(c,y_pred6, color="red",  linewidth=0.1, linestyle="-")
plt.title('Actual_Power Vs Predicted_Power', fontsize=20)
plt.xlabel('Index', fontsize=18)
plt.ylabel('Active_Power', fontsize=16)
plt.subplots_adjust(left=0.4, bottom=0.1, right=2.5, top=1.6)
```



In [67]:
```
from statsmodels.stats.outliers_influence import variance_inflation_factor
# VIF dataframe
vif_data = pd.DataFrame()
vif_data["feature"] = x6_train.columns
# calculating VIF for each feature
vif_data["VIF"] = [variance_inflation_factor(x6_train.values, i)
                          for i in range(len(x6_train.columns))]

print(vif_data)
```

```
                     feature         VIF
0           Avg_Ambient_Temp    68.670256
1            Avg_Pitch_Angle    11.569321
2            Avg_Rotor_Speed   136.876007
3             Avg_Wind_Speed    41.172345
4            Bearing_DE_Temp   410.596555
5       Gearbox_bearing_Temp  2695.365135
6           Gearbox_oil_Temp  2568.161972
7      Generator_wind_Temp_1   289.828345
8  Generators_sliprings_Temp   408.753682
9   Hidraulic_group_pressure    81.101311
10         Trafo_1_wind_Temp   300.300197
11         Trafo_2_wind_Temp   269.403294
12         Trafo_3_wind_Temp   277.515705
```

In [68]:
```
c = [i for i in range(1,49041,1)]

plt.plot(c,y_test-y_pred6, color="blue", linewidth=0.1, linestyle="-")
plt.title('Error Terms', fontsize=20)              # Plot heading
plt.xlabel('Index', fontsize=18)                      # X-label
plt.ylabel('Error_line', fontsize=16)
plt.subplots_adjust(left=0.4, bottom=0.1, right=2.5, top=1.6)
```



In [69]:
```
from sklearn.metrics import mean_squared_error, r2_score
mse = mean_squared_error(y_test, y_pred6)
r_squared = r2_score(y_test, y_pred6)
print('Mean_Squared_Error :' ,mse)
print('r_square_value :', round(r_squared,3),"% Variance of the Active Power is Explained by the Wind Speed, Pitch Angle
```

```
Mean_Squared_Error : 20977.0031614993
r_square_value : 0.931 % Variance of the Active Power is Explained by the Wind Speed, Pitch Angle and Rotor Speed
```

MODEL-8

In [70]:
```
x7_train = x_train[['Avg_Ambient_Temp', 'Avg_Pitch_Angle', 'Avg_Rotor_Speed', 'Avg_Wind_Speed', 'Bearing_DE_Temp',
                    'Gearbox_bearing_Temp', 'Gearbox_oil_Temp', 'Generator_wind_Temp_1', 'Generators_sliprings_Temp',
                    'Hidraulic_group_pressure', 'Trafo_2_wind_Temp', 'Trafo_3_wind_Temp']]
x7_test = x_test[['Avg_Ambient_Temp', 'Avg_Pitch_Angle', 'Avg_Rotor_Speed', 'Avg_Wind_Speed', 'Bearing_DE_Temp',
                    'Gearbox_bearing_Temp', 'Gearbox_oil_Temp', 'Generator_wind_Temp_1', 'Generators_sliprings_Temp',
                    'Hidraulic_group_pressure', 'Trafo_2_wind_Temp', 'Trafo_3_wind_Temp']]
```

'Trafo_1_wind_Temp', 'Trafo_2_wind_Temp' These varibles are Auto Correlated each other. So, i removed one variable ('Trafo_1_wind_Temp')

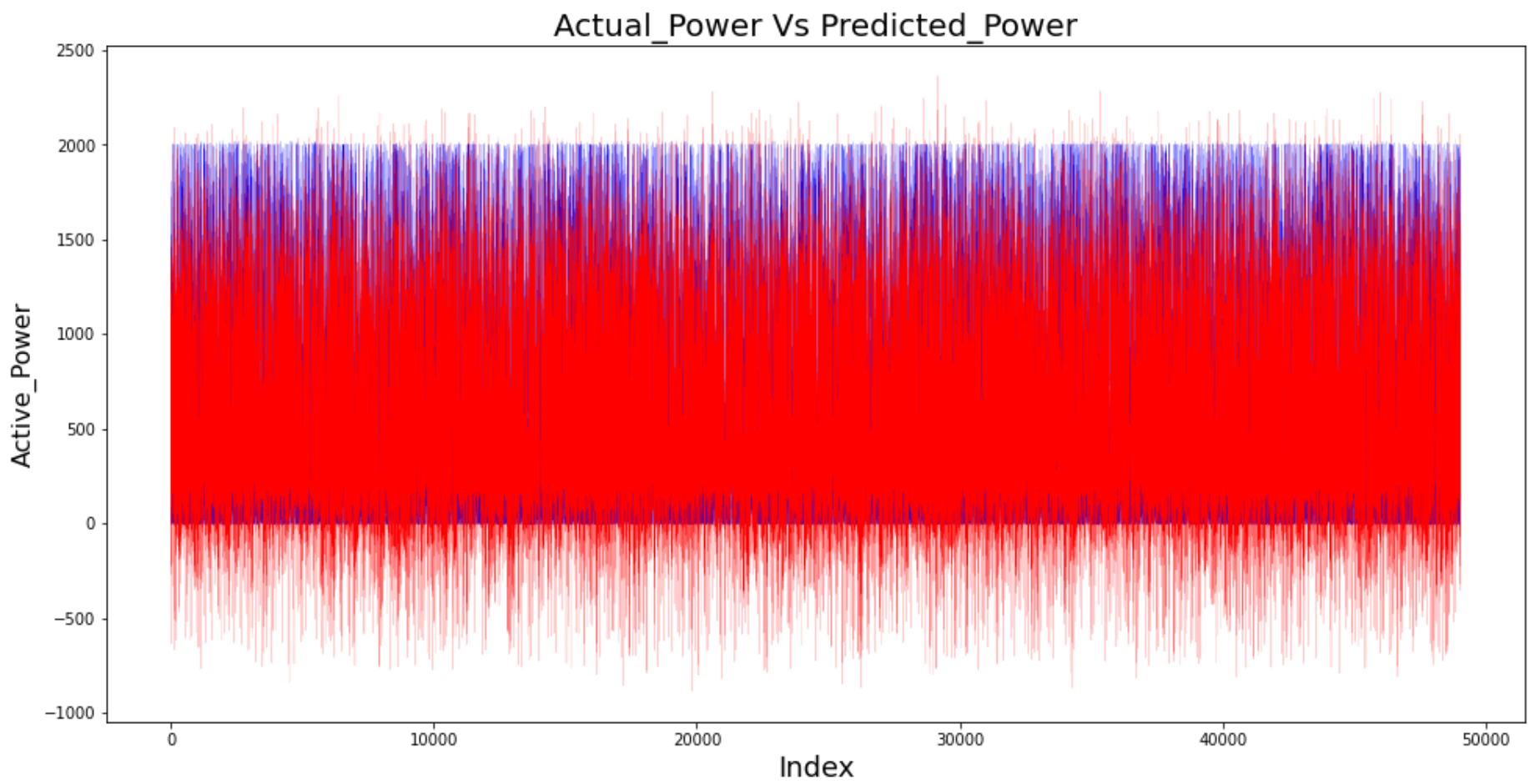
In [71]:
```
lr7 = lr.fit(x7_train,y_train)
```

In [72]:
```
y_pred7 = lr7.predict(x7_test)
```

In [73]:
```
import statsmodels.api as sm
x7_train_sm = x7_train

x7_train_sm = sm.add_constant(x7_train_sm)

mlm7 = sm.OLS(y_train,x7_train_sm).fit()

mlm7.params
print(mlm7.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:       Avg_Active_Power   R-squared:                       0.930
Model:                            OLS   Adj. R-squared:                  0.930
Method:                 Least Squares   F-statistic:                 2.182e+05
Date:                Mon, 10 May 2021   Prob (F-statistic):               0.00
Time:                        00:41:44   Log-Likelihood:            -1.2555e+06
No. Observations:              196159   AIC:                         2.511e+06
Df Residuals:                  196146   BIC:                         2.511e+06
Df Model:                          12
Covariance Type:            nonrobust
================================================================================================
                                   coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
const                        -3139.2723      9.742   -322.237      0.000   -3158.367   -3120.178
Avg_Ambient_Temp                -6.8940      0.097    -70.721      0.000      -7.085      -6.703
Avg_Pitch_Angle                 14.3159      0.043    331.652      0.000      14.231      14.400
Avg_Rotor_Speed                121.1415      0.349    347.360      0.000     120.458     121.825
Avg_Wind_Speed                  36.6781      0.331    110.681      0.000      36.029      37.328
Bearing_DE_Temp                -14.3146      0.131   -109.217      0.000     -14.571     -14.058
Gearbox_bearing_Temp            -1.6877      0.268     -6.302      0.000      -2.213      -1.163
Gearbox_oil_Temp                27.1097      0.358     75.726      0.000      26.408      27.811
Generator_wind_Temp_1            6.0710      0.085     71.155      0.000       5.904       6.238
Generators_sliprings_Temp       16.4537      0.155    106.206      0.000      16.150      16.757
Hidraulic_group_pressure         0.2275      0.015     15.196      0.000       0.198       0.257
Trafo_2_wind_Temp               -0.2415      0.050     -4.828      0.000      -0.340      -0.143
Trafo_3_wind_Temp                3.3085      0.068     48.824      0.000       3.176       3.441
==============================================================================
Omnibus:                    45147.168   Durbin-Watson:                   1.999
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           312774.572
Skew:                           0.932   Prob(JB):                         0.00
Kurtosis:                       8.898   Cond. No.                     7.91e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 7.91e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

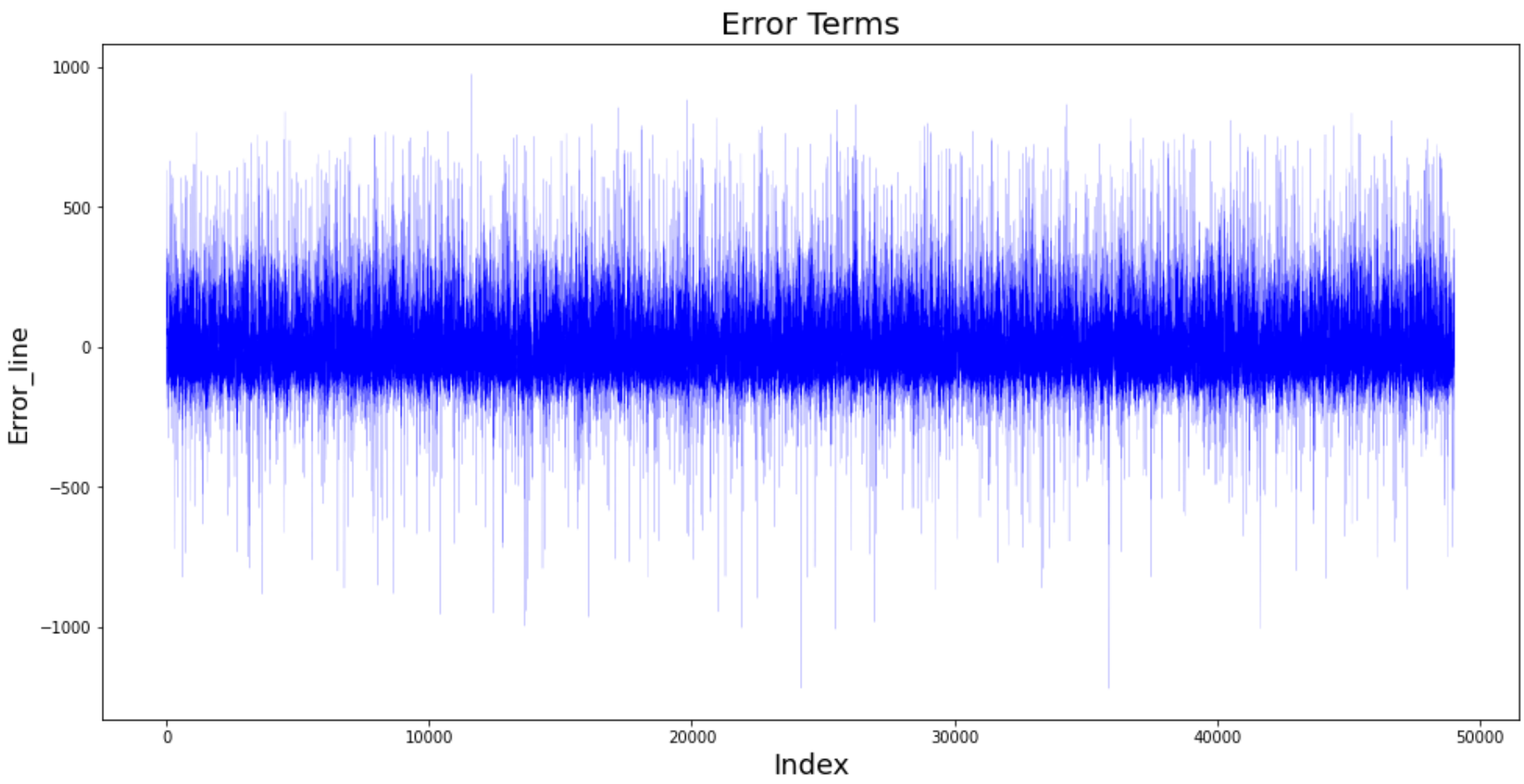
In [74]:
```
#Actual vs Predicted
c = [i for i in range(1,49041,1)]
plt.plot(c,y_test, color="blue", linewidth=0.1, linestyle="-")
plt.plot(c,y_pred7, color="red",  linewidth=0.1, linestyle="-")
plt.title('Actual_Power Vs Predicted_Power', fontsize=20)
plt.xlabel('Index', fontsize=18)
plt.ylabel('Active_Power', fontsize=16)
plt.subplots_adjust(left=0.4, bottom=0.1, right=2.5, top=1.6)
```



In [75]:
```
from statsmodels.stats.outliers_influence import variance_inflation_factor
# VIF dataframe
vif_data = pd.DataFrame()
vif_data["feature"] = x7_train.columns
# calculating VIF for each feature
vif_data["VIF"] = [variance_inflation_factor(x7_train.values, i)
                          for i in range(len(x7_train.columns))]

print(vif_data)
```

```
                      feature         VIF
0            Avg_Ambient_Temp    67.113971
1             Avg_Pitch_Angle    11.360998
2             Avg_Rotor_Speed   135.742639
3              Avg_Wind_Speed    41.169139
4             Bearing_DE_Temp   407.308201
5        Gearbox_bearing_Temp  2679.419095
6            Gearbox_oil_Temp  2524.926123
7       Generator_wind_Temp_1   281.508267
8   Generators_sliprings_Temp   408.074206
9    Hidraulic_group_pressure    81.100535
10          Trafo_2_wind_Temp   121.867959
11          Trafo_3_wind_Temp   214.358740
```

In [76]:
```
c = [i for i in range(1,49041,1)]

plt.plot(c,y_test-y_pred7, color="blue", linewidth=0.1, linestyle="-")
plt.title('Error Terms', fontsize=20)              # Plot heading
plt.xlabel('Index', fontsize=18)                      # X-label
plt.ylabel('Error_line', fontsize=16)
plt.subplots_adjust(left=0.4, bottom=0.1, right=2.5, top=1.6)
```



In [77]:
```
from sklearn.metrics import mean_squared_error, r2_score
mse = mean_squared_error(y_test, y_pred7)
r_squared = r2_score(y_test, y_pred7)
print('Mean_Squared_Error :' ,mse)
print('r_square_value :', round(r_squared,3),"% Variance of the Active Power is Explained by the Wind Speed, Pitch Angle
```

```
Mean_Squared_Error : 21034.474288578378
r_square_value : 0.93 % Variance of the Active Power is Explained by the Wind Speed, Pitch Angle and Rotor Speed
```

MODEL-9

In [78]:
```
x8_train = x_train[['Avg_Ambient_Temp', 'Avg_Pitch_Angle', 'Avg_Rotor_Speed', 'Avg_Wind_Speed', 'Bearing_DE_Temp',
                    'Gearbox_oil_Temp', 'Generator_wind_Temp_1', 'Generators_sliprings_Temp',
                    'Hidraulic_group_pressure', 'Trafo_2_wind_Temp', 'Trafo_3_wind_Temp']]
x8_test = x_test[['Avg_Ambient_Temp', 'Avg_Pitch_Angle', 'Avg_Rotor_Speed', 'Avg_Wind_Speed', 'Bearing_DE_Temp',
                    'Gearbox_oil_Temp', 'Generator_wind_Temp_1', 'Generators_sliprings_Temp',
                    'Hidraulic_group_pressure', 'Trafo_2_wind_Temp', 'Trafo_3_wind_Temp']]
```

'Gearbox_bearing_Temp', 'Gearbox_oil_Temp' These varibles are Auto Correlated each other. So, i removed one variable ('Gearbox_bearing_Temp')

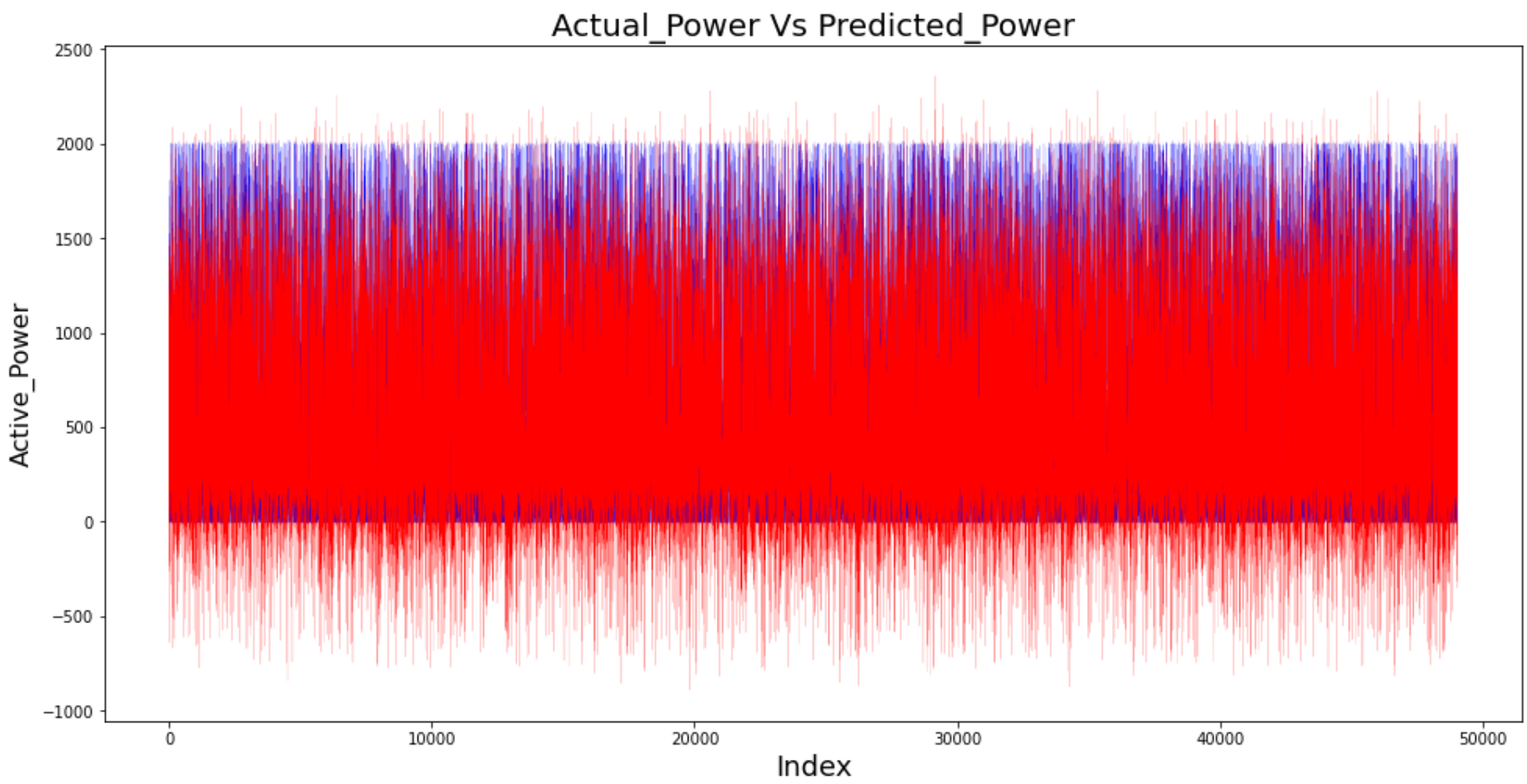
In [79]:
```
lr8 = lr.fit(x8_train,y_train)
```

In [80]:
```
y_pred8 = lr8.predict(x8_test)
```

In [81]:
```
import statsmodels.api as sm
x8_train_sm = x8_train

x8_train_sm = sm.add_constant(x8_train_sm)

mlm8 = sm.OLS(y_train,x8_train_sm).fit()

mlm8.params
print(mlm8.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:       Avg_Active_Power   R-squared:                       0.930
Model:                            OLS   Adj. R-squared:                  0.930
Method:                 Least Squares   F-statistic:                 2.380e+05
Date:                Mon, 10 May 2021   Prob (F-statistic):               0.00
Time:                        00:41:49   Log-Likelihood:            -1.2555e+06
No. Observations:              196159   AIC:                         2.511e+06
Df Residuals:                  196147   BIC:                         2.511e+06
Df Model:                          11
Covariance Type:            nonrobust
================================================================================================
                                   coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
const                        -3126.5294      9.531   -328.042      0.000   -3145.210   -3107.849
Avg_Ambient_Temp                -6.8243      0.097    -70.454      0.000      -7.014      -6.634
Avg_Pitch_Angle                 14.3342      0.043    332.803      0.000      14.250      14.419
Avg_Rotor_Speed                120.5063      0.334    360.910      0.000     119.852     121.161
Avg_Wind_Speed                  36.9414      0.329    112.361      0.000      36.297      37.586
Bearing_DE_Temp                -14.3849      0.131   -110.143      0.000     -14.641     -14.129
Gearbox_oil_Temp                25.3162      0.217    116.567      0.000      24.890      25.742
Generator_wind_Temp_1            6.0969      0.085     71.535      0.000       5.930       6.264
Generators_sliprings_Temp       16.2095      0.150    108.055      0.000      15.915      16.504
Hidraulic_group_pressure         0.2131      0.015     14.399      0.000       0.184       0.242
Trafo_2_wind_Temp               -0.2131      0.050     -4.277      0.000      -0.311      -0.115
Trafo_3_wind_Temp                3.3676      0.067     50.175      0.000       3.236       3.499
==============================================================================
Omnibus:                    44892.103   Durbin-Watson:                   1.999
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           317595.110
Skew:                           0.920   Prob(JB):                         0.00
Kurtosis:                       8.956   Cond. No.                     7.50e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 7.5e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

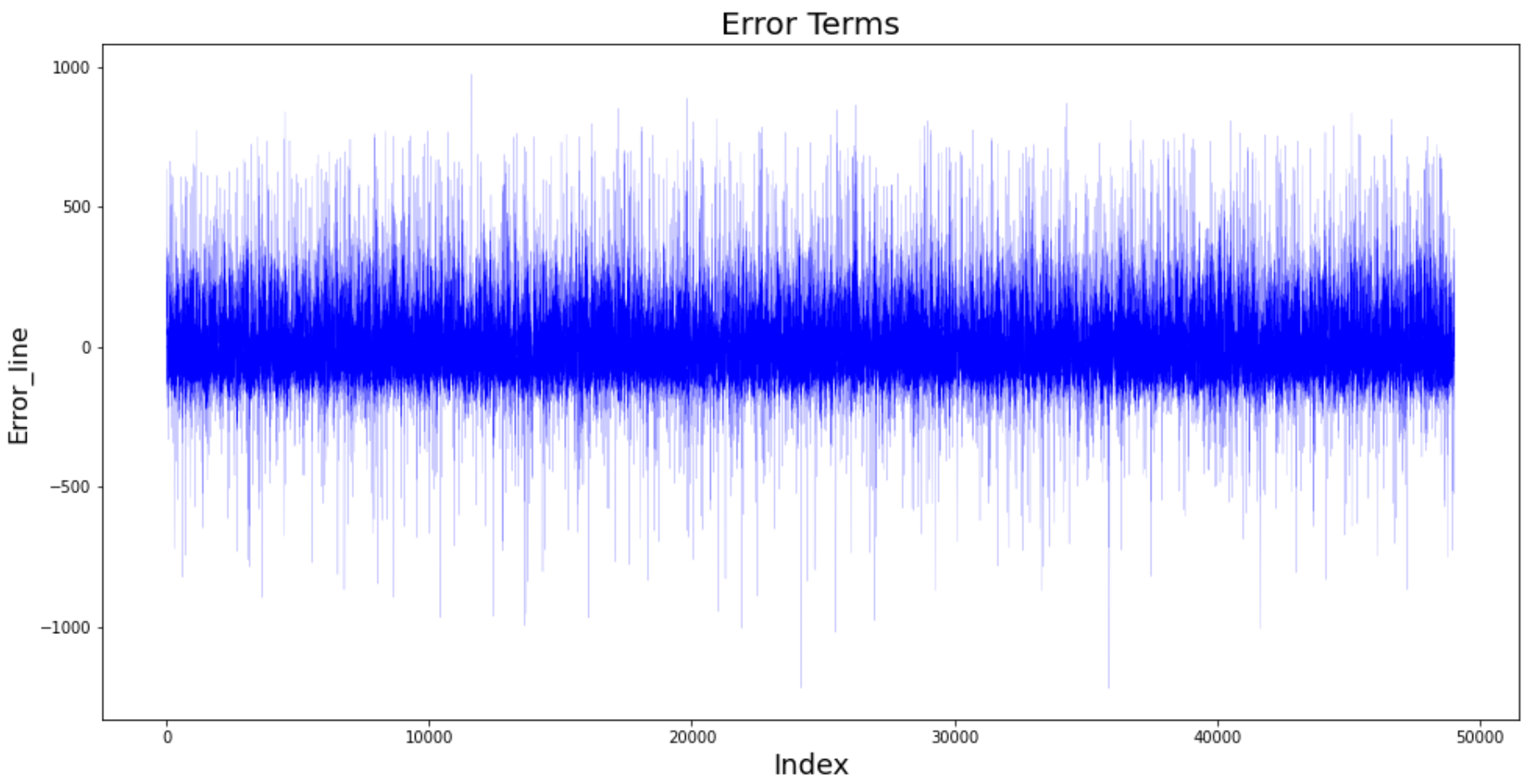
In [82]:
```
#Actual vs Predicted
c = [i for i in range(1,49041,1)]
plt.plot(c,y_test, color="blue", linewidth=0.1, linestyle="-")
plt.plot(c,y_pred8, color="red",  linewidth=0.1, linestyle="-")
plt.title('Actual_Power Vs Predicted_Power', fontsize=20)
plt.xlabel('Index', fontsize=18)
plt.ylabel('Active_Power', fontsize=16)
plt.subplots_adjust(left=0.4, bottom=0.1, right=2.5, top=1.6)
```



In [83]:
```
from statsmodels.stats.outliers_influence import variance_inflation_factor
# VIF dataframe
vif_data = pd.DataFrame()
vif_data["feature"] = x8_train.columns
# calculating VIF for each feature
vif_data["VIF"] = [variance_inflation_factor(x8_train.values, i)
                          for i in range(len(x8_train.columns))]

print(vif_data)
```

```
                     feature         VIF
0           Avg_Ambient_Temp   65.952156
1            Avg_Pitch_Angle   11.302001
2            Avg_Rotor_Speed  122.102926
3             Avg_Wind_Speed   40.566746
4            Bearing_DE_Temp  404.377687
5          Gearbox_oil_Temp   495.460649
6      Generator_wind_Temp_1  280.816467
7   Generators_sliprings_Temp  385.166722
8    Hidraulic_group_pressure   80.314856
9          Trafo_2_wind_Temp  120.734064
10         Trafo_3_wind_Temp  210.089479
```

In [84]:
```
c = [i for i in range(1,49041,1)]

plt.plot(c,y_test-y_pred8, color="blue", linewidth=0.1, linestyle="-")
plt.title('Error Terms', fontsize=20)              # Plot heading
plt.xlabel('Index', fontsize=18)                      # X-label
plt.ylabel('Error_line', fontsize=16)
plt.subplots_adjust(left=0.4, bottom=0.1, right=2.5, top=1.6)
```



In [85]:
```
from sklearn.metrics import mean_squared_error, r2_score
mse = mean_squared_error(y_test, y_pred8)
r_squared = r2_score(y_test, y_pred8)
print('Mean_Squared_Error :' ,mse)
print('r_square_value :', round(r_squared,3),"% Variance of the Active Power is Explained by the Wind Speed, Pitch Angle
```

```
Mean_Squared_Error : 21034.049515555536
r_square_value : 0.93 % Variance of the Active Power is Explained by the Wind Speed, Pitch Angle and Rotor Speed
```

MODEL-10

In [86]:
```
x9_train = x_train[['Avg_Ambient_Temp', 'Avg_Pitch_Angle', 'Avg_Rotor_Speed', 'Avg_Wind_Speed', 'Bearing_DE_Temp',
                    'Generator_wind_Temp_1', 'Generators_sliprings_Temp', 'Gearbox_oil_Temp']]
x9_test = x_test[['Avg_Ambient_Temp', 'Avg_Pitch_Angle', 'Avg_Rotor_Speed', 'Avg_Wind_Speed', 'Bearing_DE_Temp',
                  'Generator_wind_Temp_1', 'Generators_sliprings_Temp', 'Gearbox_oil_Temp']]
```

'Hidraulic_group_pressure', 'Trafo_2_wind_Temp', 'Trafo_3_wind_Temp' These varibles are Auto Correlated. So, i removed the variable

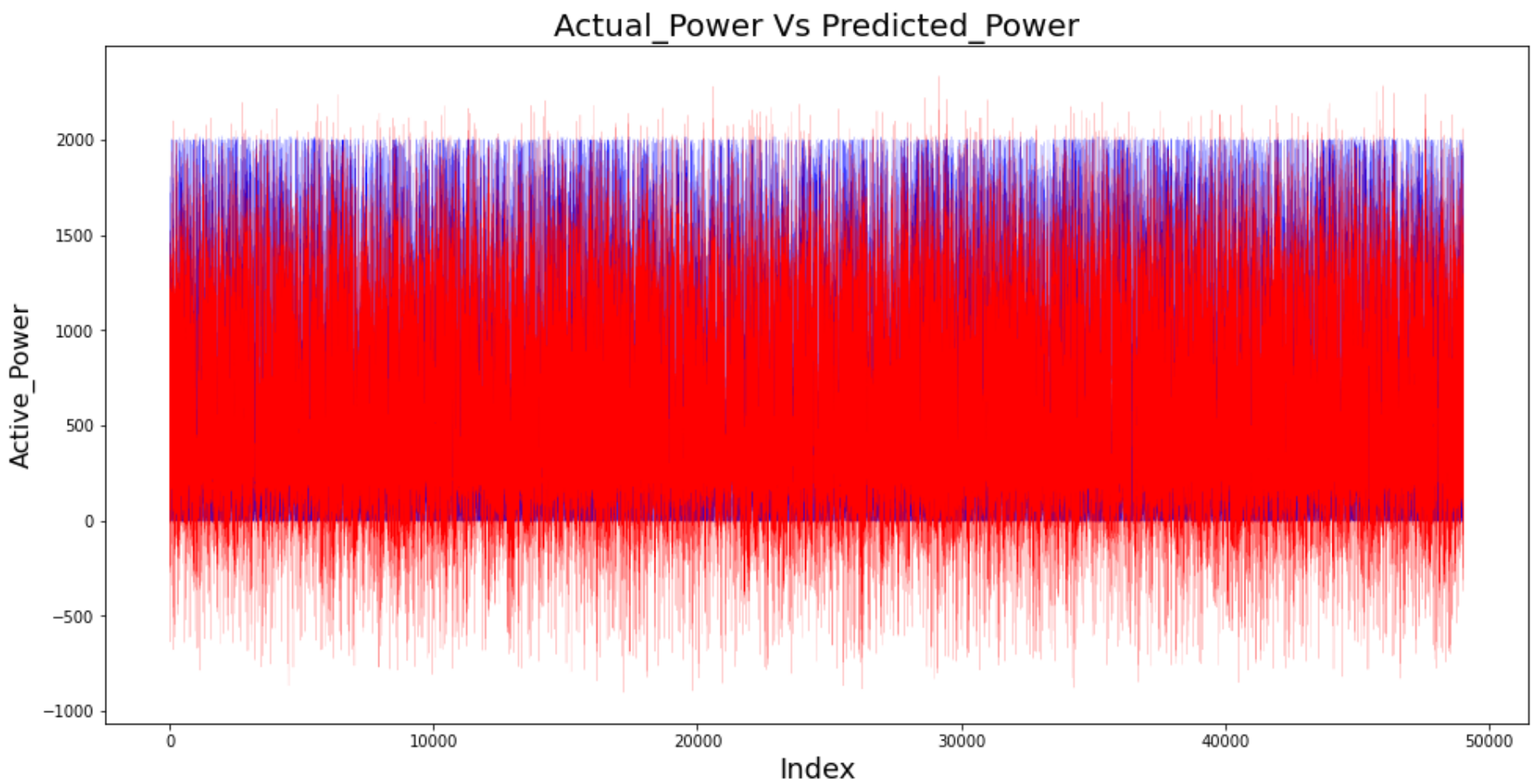
In [87]:
```
lr9 = lr.fit(x9_train,y_train)
```

In [88]:
```
y_pred9 = lr9.predict(x9_test)
```

In [89]:
```
import statsmodels.api as sm
x9_train_sm = x9_train

x9_train_sm = sm.add_constant(x9_train_sm)

mlm9 = sm.OLS(y_train,x9_train_sm).fit()

mlm9.params
print(mlm9.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:       Avg_Active_Power   R-squared:                       0.929
Model:                            OLS   Adj. R-squared:                  0.929
Method:                 Least Squares   F-statistic:                 3.206e+05
Date:                Mon, 10 May 2021   Prob (F-statistic):               0.00
Time:                        00:41:54   Log-Likelihood:            -1.2574e+06
No. Observations:              196159   AIC:                         2.515e+06
Df Residuals:                  196150   BIC:                         2.515e+06
Df Model:                           8
Covariance Type:            nonrobust
================================================================================================
                                   coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
const                        -3055.6816      9.234   -330.900      0.000   -3073.781   -3037.582
Avg_Ambient_Temp                -6.3651      0.096    -66.020      0.000      -6.554      -6.176
Avg_Pitch_Angle                 14.3732      0.040    359.233      0.000      14.295      14.452
Avg_Rotor_Speed                120.9132      0.328    368.437      0.000     120.270     121.556
Avg_Wind_Speed                  35.6474      0.326    109.257      0.000      35.008      36.287
Bearing_DE_Temp                -13.6919      0.131   -104.309      0.000     -13.949     -13.435
Generator_wind_Temp_1            6.7556      0.085     79.484      0.000       6.589       6.922
Generators_sliprings_Temp       16.0050      0.151    106.030      0.000      15.709      16.301
Gearbox_oil_Temp                27.1594      0.216    125.792      0.000      26.736      27.583
==============================================================================
Omnibus:                    46906.283   Durbin-Watson:                   1.999
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           338443.212
Skew:                           0.963   Prob(JB):                         0.00
Kurtosis:                       9.140   Cond. No.                     3.33e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.33e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

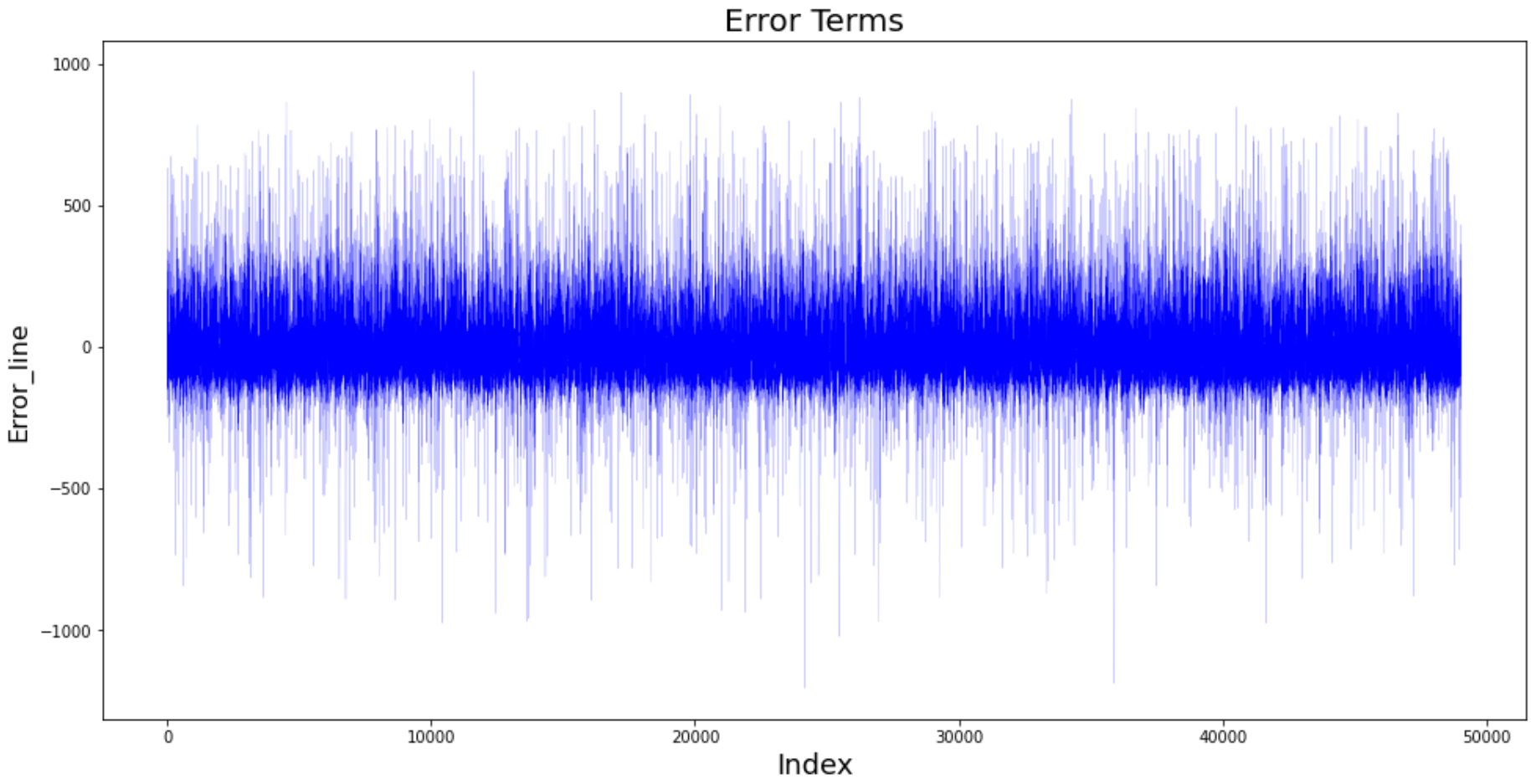
In [90]:
```
#Actual vs Predicted
c = [i for i in range(1,49041,1)]
plt.plot(c,y_test, color="blue", linewidth=0.1, linestyle="-")
plt.plot(c,y_pred9, color="red",  linewidth=0.1, linestyle="-")
plt.title('Actual_Power Vs Predicted_Power', fontsize=20)
plt.xlabel('Index', fontsize=18)
plt.ylabel('Active_Power', fontsize=16)
plt.subplots_adjust(left=0.4, bottom=0.1, right=2.5, top=1.6)
```



In [91]:
```
from statsmodels.stats.outliers_influence import variance_inflation_factor
# VIF dataframe
vif_data = pd.DataFrame()
vif_data["feature"] = x9_train.columns
# calculating VIF for each feature
vif_data["VIF"] = [variance_inflation_factor(x9_train.values, i)
                          for i in range(len(x9_train.columns))]

print(vif_data)
```

```
                    feature         VIF
0           Avg_Ambient_Temp   63.729292
1            Avg_Pitch_Angle    9.520462
2            Avg_Rotor_Speed  112.348573
3             Avg_Wind_Speed   39.174582
4            Bearing_DE_Temp  400.707358
5     Generator_wind_Temp_1   264.268772
6  Generators_sliprings_Temp  383.082762
7           Gearbox_oil_Temp  230.627366
```

In [92]:
```
c = [i for i in range(1,49041,1)]

plt.plot(c,y_test-y_pred9, color="blue", linewidth=0.1, linestyle="-")
plt.title('Error Terms', fontsize=20)              # Plot heading
plt.xlabel('Index', fontsize=18)                      # X-label
plt.ylabel('Error_line', fontsize=16)
plt.subplots_adjust(left=0.4, bottom=0.1, right=2.5, top=1.6)
```



In [93]:
```
from sklearn.metrics import mean_squared_error, r2_score
mse = mean_squared_error(y_test, y_pred9)
r_squared = r2_score(y_test, y_pred9)
print('Mean_Squared_Error :' ,mse)
print('r_square_value :', round(r_squared,3),"% Variance of the Active Power is Explained by the Wind Speed, Pitch Angle
```

```
Mean_Squared_Error : 21443.519337970352
r_square_value : 0.929 % Variance of the Active Power is Explained by the Wind Speed, Pitch Angle and Rotor Speed
```

MODEL-10

In [94]:
```
x10_train = x_train[['Avg_Pitch_Angle', 'Avg_Rotor_Speed', 'Avg_Wind_Speed']]
x10_test = x_test[['Avg_Pitch_Angle', 'Avg_Rotor_Speed', 'Avg_Wind_Speed']]
```

Removed all temperature variables.

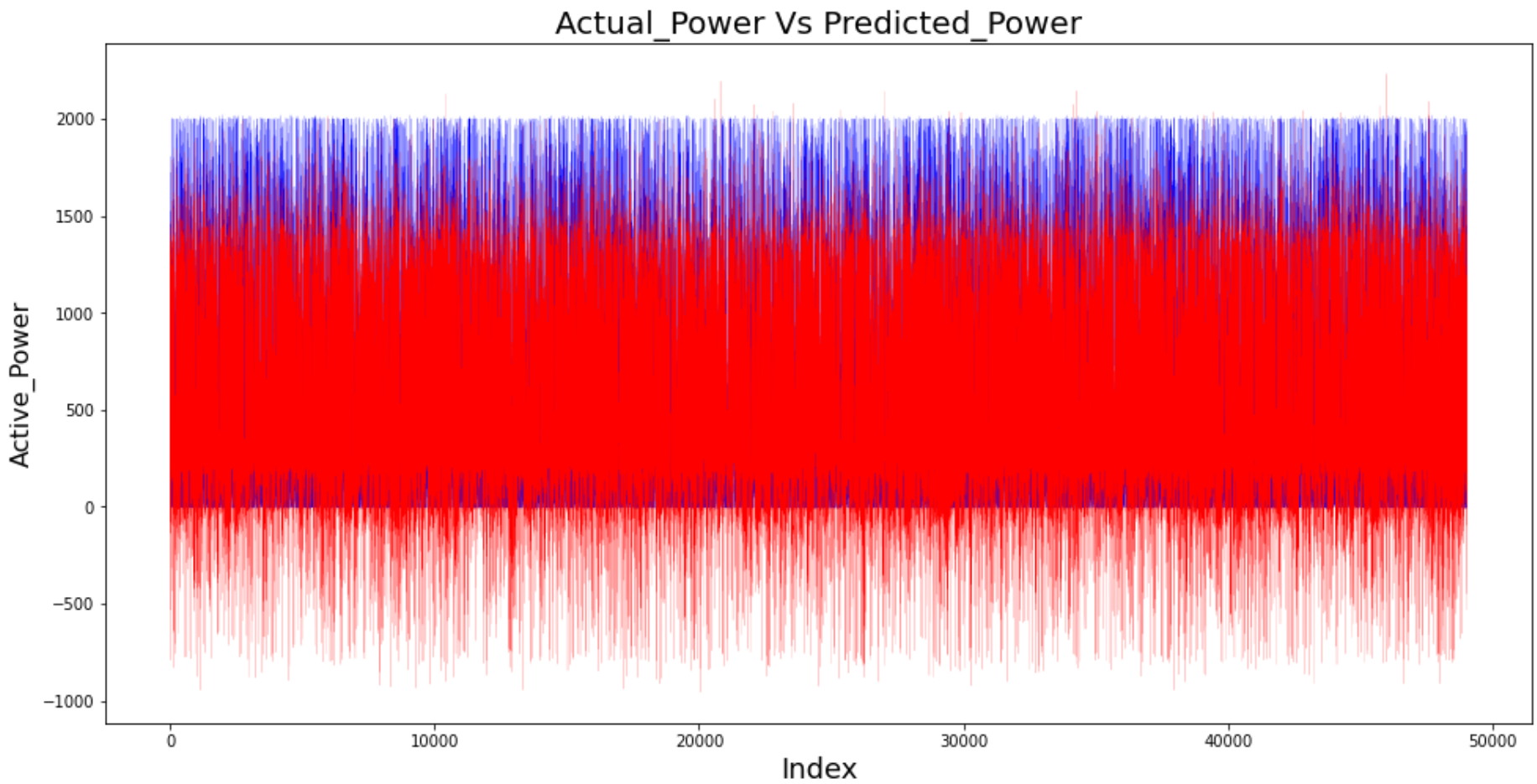
In [95]:
```
lr10 = lr.fit(x10_train,y_train)
```

In [96]:
```
y_pred10 = lr10.predict(x10_test)
```

In [97]:
```
import statsmodels.api as sm
x10_train_sm = x10_train

x10_train_sm = sm.add_constant(x10_train_sm)

mlm10 = sm.OLS(y_train,x10_train_sm).fit()

mlm10.params
print(mlm10.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:       Avg_Active_Power   R-squared:                       0.901
Model:                            OLS   Adj. R-squared:                  0.901
Method:                 Least Squares   F-statistic:                 5.930e+05
Date:                Mon, 10 May 2021   Prob (F-statistic):               0.00
Time:                        00:41:58   Log-Likelihood:            -1.2903e+06
No. Observations:              196159   AIC:                         2.581e+06
Df Residuals:                  196155   BIC:                         2.581e+06
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          -1667.8072      2.493   -668.920      0.000   -1672.694   -1662.920
Avg_Pitch_Angle    17.4895      0.045    391.660      0.000      17.402      17.577
Avg_Rotor_Speed   163.9809      0.332    494.522      0.000     163.331     164.631
Avg_Wind_Speed     54.4955      0.308    176.775      0.000      53.891      55.100
==============================================================================
Omnibus:                    54262.901   Durbin-Watson:                   1.997
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           227683.002
Skew:                           1.313   Prob(JB):                         0.00
Kurtosis:                       7.578   Cond. No.                         165.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

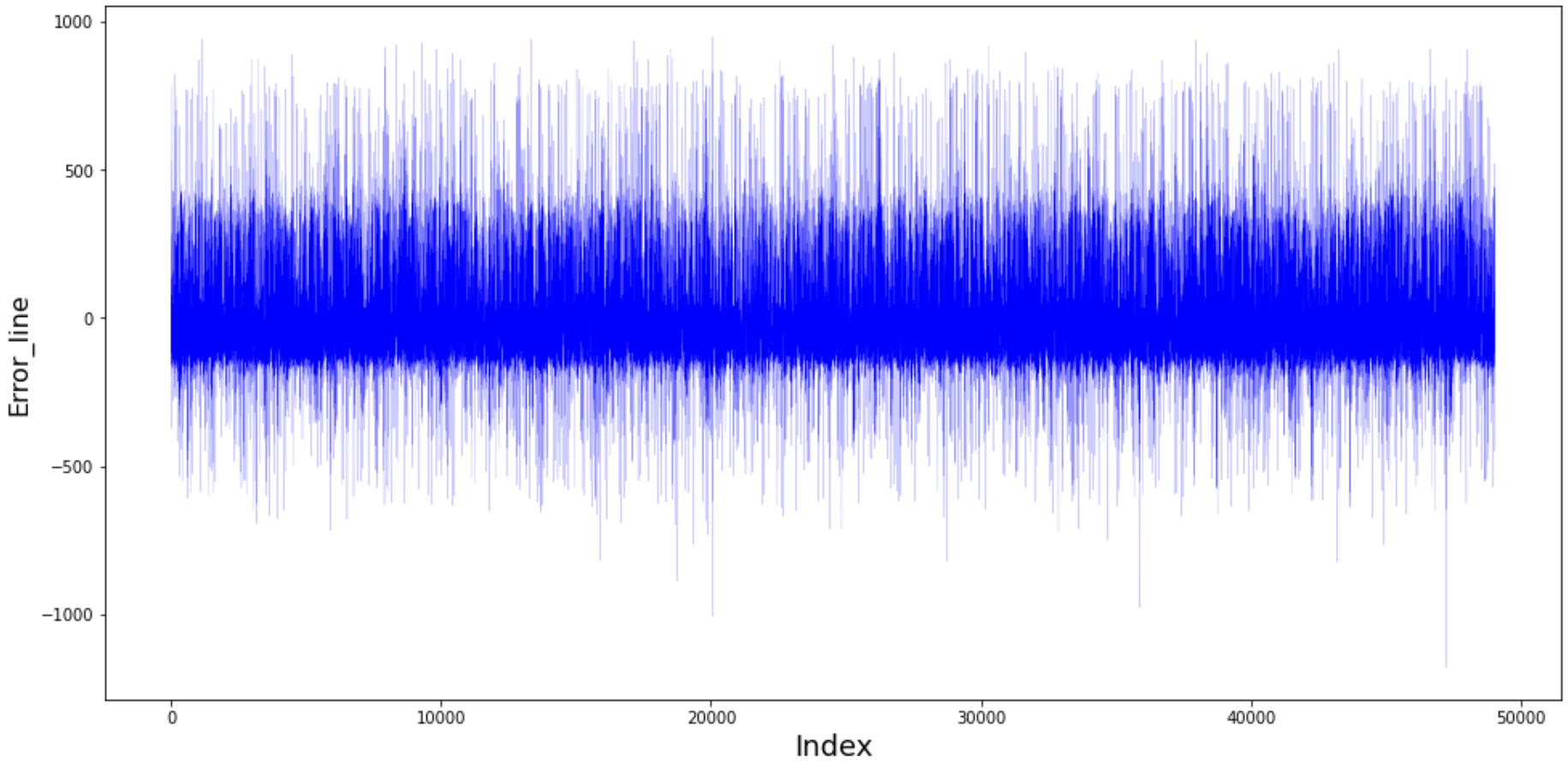
In [98]:
```
#Actual vs Predicted
c = [i for i in range(1,49041,1)]
plt.plot(c,y_test, color="blue", linewidth=0.1, linestyle="-")
plt.plot(c,y_pred10, color="red",  linewidth=0.1, linestyle="-")
plt.title('Actual_Power Vs Predicted_Power', fontsize=20)
plt.xlabel('Index', fontsize=18)
plt.ylabel('Active_Power', fontsize=16)
plt.subplots_adjust(left=0.4, bottom=0.1, right=2.5, top=1.6)
```



In [99]:
```
from statsmodels.stats.outliers_influence import variance_inflation_factor
# VIF dataframe
vif_data = pd.DataFrame()
vif_data["feature"] = x10_train.columns
# calculating VIF for each feature
vif_data["VIF"] = [variance_inflation_factor(x10_train.values, i)
                          for i in range(len(x10_train.columns))]

print(vif_data)
```

```
           feature        VIF
0  Avg_Pitch_Angle   1.699851
1  Avg_Rotor_Speed  15.872537
2   Avg_Wind_Speed  16.956543
```

In [100]:
```
c = [i for i in range(1,49041,1)]

plt.plot(c,y_test-y_pred10, color="blue", linewidth=0.1, linestyle="-")
plt.title('Error Terms', fontsize=20)              # Plot heading
plt.xlabel('Index', fontsize=18)                      # X-label
plt.ylabel('Error_line', fontsize=16)
plt.subplots_adjust(left=0.4, bottom=0.1, right=2.5, top=1.6)
```



In [101]:
```
from sklearn.metrics import mean_squared_error, r2_score
mse = mean_squared_error(y_test, y_pred10)
r_squared = r2_score(y_test, y_pred10)
print('Mean_Squared_Error :' ,mse)
print('r_square_value :', round(r_squared,3),"% Variance of the Active Power is Explained by the Wind Speed, Pitch Angle
```

```
Mean_Squared_Error : 30233.20183664925
r_square_value : 0.9 % Variance of the Active Power is Explained by the Wind Speed, Pitch Angle and Rotor Speed


 REGRESSION EQUATION
```

In [102]:
```
@widgets.interact(Wind_Speed = range(1, 25, 1),
                  Pitch_Angle = range(1, 95, 1), Rotor_Speed = range(1, 20, 1))
def Regression_Equation(Wind_Speed, Pitch_Angle, Rotor_Speed):
    Intercept = mlm10.params[0]
    Coefficient = mlm10.params[1]*Pitch_Angle + mlm10.params[2]*Rotor_Speed + mlm10.params[3]*Wind_Speed

    y_Predicted = Intercept + Coefficient  # y = a + bx  Equation.
    if y_Predicted < 0:
        print("Please Choose Other Values: Active Power is '0' at these values")
    else:
        print('Active Power Predicted: ',y_Predicted)  # Basic Starting Values are 3, 77, 1
```

```
interactive(children=(Dropdown(description='Wind_Speed', options=(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 1…
```

```
Thank You...
```