In [163]:

```
import pandas as pd
import numpy as np
import seaborn as sns
```

In [164]:

```
dataset=pd.read_excel('C:/Users/Furkan/Desktop/Project.xlsx')  #The Main Data
df=dataset.copy()
```

In [166]:

```
dataset_weather=pd.read_excel('C:/Users/Furkan/Desktop/YalovaWeather.xlsx') # The weather Data we want to add to our data
df_w=dataset_weather.copy()
```

In [167]:

```
df.head()
```

Out[167]:

	Date	Time	LV ActivePower (kW)	Wind Speed (m/s)	Theoretical_Power_Curve (KWh)	Wind Direction (°)	Month	Day/Night
0	01 01 2018	00:00:00	380.047791	5.311336	416.328908	259.994904	1	0
1	01 01 2018	00:10:00	453.769196	5.672167	519.917511	268.641113	1	0
2	01 01 2018	00:20:00	306.376587	5.216037	390.900016	272.564789	1	0
3	01 01 2018	00:30:00	419.645904	5.659674	516.127569	271.258087	1	0
4	01 01 2018	00:40:00	380.650696	5.577941	491.702972	265.674286	1	0

In [168]:

```
df_w.head()
```

Out[168]:

	Date	Day/Night	Temp	Sun Hour	Moon Illumination	Moonrise	Moonset	Sunrise	Sunset	DewPoint Temp	WindChillC	WindGustKmph
0	01 01 2018	0	4	8.7	97	17:23:00	07:20:00	08:27:00	17:46:00	3	5	6
1	01 01 2018	1	10	8.7	97	17:23:00	07:20:00	08:27:00	17:46:00	2	11	6
2	02 01 2018	0	7	7.0	100	18:27:00	08:25:00	08:27:00	17:47:00	3	7	15
3	02 01 2018	1	13	7.0	100	18:27:00	08:25:00	08:27:00	17:47:00	5	13	14
	03											

	Date	Day/Night	Temp	Sun Hour	Moon Illumination	Moonrise	Moonset	Sunrise	Sunset	DewPoint Temp	WindChillC	WindGustKmph
4	01 2018	0	9	7.0	89	19:38:00	09:24:00	08:27:00	17:48:00	6	8	18

In [169]:

```
df3=pd.merge(df,df_w) # Merging these 2 datasets to become our final dataset
```

In [170]:

```
df3.head()
```

Out[170]:

	Date	Time	LV ActivePower (kW)	Wind Speed (m/s)	Theoretical_Power_Curve (KWh)	Wind Direction (°)	Month	Day/Night	Temp	Sun Hour	...	Moonse
0	01 01 2018	00:00:00	380.047791	5.311336	416.328908	259.994904	1	0	4	8.7	...	07:20:0
1	01 01 2018	00:10:00	453.769196	5.672167	519.917511	268.641113	1	0	4	8.7	...	07:20:0
2	01 01 2018	00:20:00	306.376587	5.216037	390.900016	272.564789	1	0	4	8.7	...	07:20:0
3	01 01 2018	00:30:00	419.645904	5.659674	516.127569	271.258087	1	0	4	8.7	...	07:20:0
4	01 01 2018	00:40:00	380.650696	5.577941	491.702972	265.674286	1	0	4	8.7	...	07:20:0

5 rows × 22 columns

In [76]:

```
#Recording Merged data to an excel file
from openpyxl import Workbook, load_workbook
from openpyxl.drawing.image import Image
from openpyxl.utils.dataframe import dataframe_to_rows

# Create new workbook
wb = Workbook()
wb.save("C:/Users/Furkan/Desktop/Data.xlsx")

sheet1 = wb.create_sheet('sheet1',0)
```

In [77]:

```
# Activate worksheet to write dataframe
active = wb['sheet1']

# Write dataframe to active worksheet
for x in dataframe_to_rows(df3):
    active.append(x)

# Save workbook to write
wb.save("C:/Users/Furkan/Desktop/Data.xlsx")
```

In [171]:

```
df3.info() #Summary of the dataset
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 50530 entries, 0 to 50529
Data columns (total 22 columns):
Date                             50530 non-null object
Time                             50530 non-null object
LV ActivePower (kW)              50530 non-null float64
Wind Speed (m/s)                 50530 non-null float64
Theoretical_Power_Curve (KWh)    50530 non-null float64
Wind Direction (°)               50530 non-null float64
Month                            50530 non-null int64
Day/Night                        50530 non-null int64
Temp                             50530 non-null int64
Sun Hour                         50530 non-null float64
Moon Illimunation                50530 non-null int64
Moonrise                         50530 non-null object
Moonset                          50530 non-null object
Sunrise                          50530 non-null object
Sunset                           50530 non-null object
DewPoint Temp                    50530 non-null int64
WindChillC                       50530 non-null int64
WindGustKmph                     50530 non-null int64
Humidity                         50530 non-null int64
RainMM                           50530 non-null float64
Pressure                         50530 non-null int64
Visibility                       50530 non-null int64
dtypes: float64(6), int64(10), object(6)
memory usage: 8.9+ MB
```

In [103]:

```
df3.Month=pd.Categorical(df3.Month)                #Before dividing the dataset numeric and categor
ical I define categoric variables
df3["Day/Night"]=pd.Categorical(df3["Day/Night"])
df3.Date=pd.Categorical(df3.Date)
df3.Time=pd.Categorical(df3.Time)
df3.Moonrise=pd.Categorical(df3.Moonrise)
df3.Moonset=pd.Categorical(df3.Moonset)
df3.Sunrise=pd.Categorical(df3.Sunrise)
df3.Sunset=pd.Categorical(df3.Sunset)
```

In [104]:

```
df_cat=df3.select_dtypes(["category"]) #Dividing data into 2 parts as numeric and non-numeric
```

In [105]:

```
df_cat.head() #Non-numeric data
```

Out[105]:

	Date	Time	Month	Day/Night	Moonrise	Moonset	Sunrise	Sunset
0	01 01 2018	00:00:00	1	0	17:23:00	07:20:00	08:27:00	17:46:00
1	01 01 2018	00:10:00	1	0	17:23:00	07:20:00	08:27:00	17:46:00
2	01 01 2018	00:20:00	1	0	17:23:00	07:20:00	08:27:00	17:46:00
3	01 01 2018	00:30:00	1	0	17:23:00	07:20:00	08:27:00	17:46:00
4	01 01 2018	00:40:00	1	0	17:23:00	07:20:00	08:27:00	17:46:00

In [109]:

```
df_numeric=df3.drop(df_cat.columns,axis=1)
```

In [110]:

```
df_numeric.head()   #Numeric Data
```

Out[110]:

	LV ActivePower (kW)	Wind Speed (m/s)	Theoretical_Power_Curve (KWh)	Wind Direction (°)	Temp	Sun Hour	Moon Illumination	DewPoint Temp	WindChillC	WindGu
0	380.047791	5.311336	416.328908	259.994904	4	8.7	97	3	5	6
1	453.769196	5.672167	519.917511	268.641113	4	8.7	97	3	5	6
2	306.376587	5.216037	390.900016	272.564789	4	8.7	97	3	5	6
3	419.645904	5.659674	516.127569	271.258087	4	8.7	97	3	5	6
4	380.650696	5.577941	491.702972	265.674286	4	8.7	97	3	5	6

In [182]:

```
df_numeric.aggregate(["mean","std","count","min","max","median"]).T #Descriptive statatistics of n
umeric variables
```

Out[182]:

	mean	std	count	min	max	median
LV ActivePower (kW)	1307.684332	1312.459242	50530.0	-2.471405	3618.732910	825.838074
Wind Speed (m/s)	7.557952	4.227166	50530.0	0.000000	25.206011	7.104594
Theoretical_Power_Curve (KWh)	1492.175463	1368.018238	50530.0	0.000000	3600.000000	1063.776282
Wind Direction (°)	123.687559	93.443736	50530.0	0.000000	359.997589	73.712978
Temp	15.956818	7.478934	50530.0	-1.000000	32.000000	16.000000
Sun Hour	10.394415	3.198427	50530.0	3.400000	14.500000	11.600000
Moon Illumination	46.463131	31.548401	50530.0	0.000000	100.000000	46.000000
DewPoint Temp	10.533089	5.966731	50530.0	-5.000000	22.000000	11.000000
WindChillC	15.989234	8.341774	50530.0	-5.000000	32.000000	16.000000
WindGustKmph	14.597645	8.020789	50530.0	0.000000	52.000000	13.000000
CloudCover	35.742668	34.079858	50530.0	0.000000	100.000000	22.000000
Humidity	69.887592	16.452907	50530.0	31.000000	97.000000	72.000000
RainMM	0.004987	0.132024	50530.0	0.000000	3.500000	0.000000
Pressure	1014.676727	6.112598	50530.0	993.000000	1032.000000	1014.000000
Visibility	9.860400	0.987456	50530.0	0.000000	10.000000	10.000000

In [116]:

```
#Yorumla bunları raporda std sapma mean ve max mine göre basınç ve visibility çok az değişiyo mese
la
```

In [117]:

```
df_cat.head()
```

Out[117]:

	Date	Time	Month	Day/Night	Moonrise	Moonset	Sunrise	Sunset
0	01 01 2018	00:00:00	1	0	17:23:00	07:20:00	08:27:00	17:46:00
1	01 01 2018	00:10:00	1	0	17:23:00	07:20:00	08:27:00	17:46:00
2	01 01 2018	00:20:00	1	0	17:23:00	07:20:00	08:27:00	17:46:00
3	01 01 2018	00:30:00	1	0	17:23:00	07:20:00	08:27:00	17:46:00
4	01 01 2018	00:40:00	1	0	17:23:00	07:20:00	08:27:00	17:46:00

In [114]:

```
df_cat.Date.value_counts().head() # Her gün yapılan gözlem sayısı eşit değil aslında her gün 144 g
özlem olmalıydı
```

Out[114]:

```
31 12 2018    144
10 12 2018    144
11 02 2018    144
11 03 2018    144
11 04 2018    144
Name: Date, dtype: int64
```

In [115]:

```
print("Min Date is:",df_cat.Date.value_counts().idxmin(),"Min Freq:",df_cat.Date.value_counts().mi
n())
print("Max Date is:",df_cat.Date.value_counts().idxmax(),"Max Freq:",df_cat.Date.value_counts().ma
x())
```

```
Min Date is: 02 10 2018 Min Freq: 11
Max Date is: 31 12 2018 Max Freq: 144
```

In [118]:

```
df_cat.Time.value_counts().head() # Her saatten 365tane yok demek ki her gün her saatte ölçüm yapı
lamamış veya datasette yok.
                                  #max değerimiz 355 hiçbir saat için 365 yok
```

Out[118]:

```
16:50:00    355
17:00:00    355
18:10:00    355
16:30:00    355
16:40:00    355
Name: Time, dtype: int64
```

In [119]:

```
print("Min Time is:",df_cat.Time.value_counts().idxmin(),"Min Freq:",df_cat.Time.value_counts().mi
n())
print("Max Time is:",df_cat.Time.value_counts().idxmax(),"Max Freq:",df_cat.Time.value_counts().ma
x())
```

```
Min Time is: 11:50:00 Min Freq: 344
Max Time is: 16:50:00 Max Freq: 355
```

In [120]:

```
df_cat.Month.value_counts().head() #Her aydaki gün sayısı farklı olduğu için ve bazı günlerdeki ek
sik gözlemlerden dolayı eşit sayıda
                                   # gözlem yine yok
```

Out[120]:

```
7     4464
3     4463
5     4449
12    4447
8     4425
Name: Month, dtype: int64
```

In [121]:

```
print("Min Month is:",df_cat.Month.value_counts().idxmin(),"Min Freq:",df_cat.Month.value_counts()
.min())
print("Max Month is:",df_cat.Month.value_counts().idxmax(),"Max Freq:",df_cat.Month.value_counts()
.max())
```

```
Min Month is: 11 Min Freq: 3800
Max Month is: 7 Max Freq: 4464
```

In [122]:

```
df_cat["Day/Night"].value_counts().head() # Yine muhtemelen eksik ölçümlerden dolayı gündüz ve gec
e sayısı eşit değil
                                    #Günü tam ortadan böldüğü için aslında eşit olmalı fakat oldukça
fazla  fark var
                                  #Zaten 2 değer olduğu için min: 0 yani gece 25172 max 1: yani günd
z 25358
```

Out[122]:

```
1    25358
0    25172
Name: Day/Night, dtype: int64
```

In [125]:

```
df_cat["Moonrise"].value_counts().head() # Even it is not a real categoric variable but just hours
of moonrise there are some highly
                                   # often hours for moon to rise.But the most often case is no moor
ise
```

Out[125]:

```
No moonrise    1726
04:20:00        288
13:58:00        288
14:01:00        288
21:26:00        288
Name: Moonrise, dtype: int64
```

In [127]:

```
print("Min Moonrise is:",df_cat.Moonrise.value_counts().idxmin(),"Min
Freq:",df_cat.Moonrise.value_counts().min())
print("Max Moonrise is:",df_cat.Moonrise.value_counts().idxmax(),"Max
Freq:",df_cat.Moonrise.value_counts().max())
```

```
Min Moonrise is: 13:30:00 Min Freq: 39
Max Moonrise is: No moonrise Max Freq: 1726
```

In [128]:

```
df_cat["Moonset"].value_counts().head() # Even it is not a real categoric variable but just hours
of moonset there are some highly
                                   # often hours for moon to set. Similarly most frequent is no moor
et.
```

Out[128]:

```
No moonset    1728
00:27:00       432
12:07:00       430
15:09:00       288
13:24:00       288
Name: Moonset, dtype: int64
```

In [129]:

```
print("Min Moonset is:",df_cat.Moonset.value_counts().idxmin(),"Min
Freq:",df_cat.Moonset.value_counts().min())
print("Max Moonset is:",df_cat.Moonset.value_counts().idxmax(),"Max
Freq:",df_cat.Moonset.value_counts().max())
```

```
Min Moonset is: 14:17:00 Min Freq: 11
Max Moonset is: No moonset Max Freq: 1728
Max Moonset is: No Moonset Max Freq: 1720
```

In [130]:

```
df_cat["Sunrise"].value_counts().head() # Even it is not a real categoric variable but just hours
of sunrise there are some highly
                                  # often hours for sun to rise.
```

Out[130]:

```
17:36:00    1865
20:37:00    1439
20:38:00    1118
17:38:00     864
20:36:00     863
Name: Sunset, dtype: int64
```

In [131]:

```
print("Min Sunrise is:",df_cat.Sunrise.value_counts().idxmin(),"Min
Freq:",df_cat.Sunrise.value_counts().min())
print("Max Sunrise is:",df_cat.Sunrise.value_counts().idxmax(),"Max
Freq:",df_cat.Sunrise.value_counts().max())
```

```
Min Sunrise is: 07:01:00 Min Freq: 61
Max Sunrise is: 05:32:00 Max Freq: 2302
```

In [132]:

```
df_cat["Sunset"].value_counts().head() # Even it is not a real categoric variable but just hours o
f sunset there are some highly
                                  # often hours for sun to set.
```

Out[132]:

```
17:36:00    1865
20:37:00    1439
20:38:00    1118
17:38:00     864
20:36:00     863
Name: Sunset, dtype: int64
```

In [133]:

```
print("Min Sunset is:",df_cat.Sunset.value_counts().idxmin(),"Min
Freq:",df_cat.Sunset.value_counts().min())
print("Max Sunset is:",df_cat.Sunset.value_counts().idxmax(),"Max
Freq:",df_cat.Sunset.value_counts().max())
```

```
Min Sunset is: 17:49:00 Min Freq: 127
Max Sunset is: 17:36:00 Max Freq: 1865
```

In [175]:

```
#Entropy
import scipy.stats as sc

print(sc.entropy(df_cat["Date"].value_counts(),base=2))
print(sc.entropy(df_cat["Time"].value_counts(),base=2))
print(sc.entropy(df_cat["Month"].value_counts(),base=2))
print(sc.entropy(df_cat["Day/Night"].value_counts(),base=2))
print(sc.entropy(df_cat["Moonrise"].value_counts(),base=2))
print(sc.entropy(df_cat["Moonset"].value_counts(),base=2))
print(sc.entropy(df_cat["Sunrise"].value_counts(),base=2))
print(sc.entropy(df_cat["Sunset"].value_counts(),base=2))
```

```
8.468314035083559
7.169880651466229
3.582522824956813
0.9999902259891678
8.133445800130504
```

```
8.139591288587013
7.170046141691182
7.206536831702157
```

In [181]:

```
print(df3.Date.value_counts().count())
print(df3.Time.value_counts().count())
print(df3.Month.value_counts().count())
print(df3["Day/Night"].value_counts().count())
print(df3.Moonrise.value_counts().count())
print(df3.Moonset.value_counts().count())
print(df3.Sunrise.value_counts().count())
print(df3.Sunset.value_counts().count())
```

```
356
144
12
2
306
309
171
174
```

In [184]:

```
df.head()
```

Out[184]:

	Date	Time	LV ActivePower (kW)	Wind Speed (m/s)	Theoretical_Power_Curve (KWh)	Wind Direction (°)	Month	Day/Night
0	01 01 2018	00:00:00	380.047791	5.311336	416.328908	259.994904	1	0
1	01 01 2018	00:10:00	453.769196	5.672167	519.917511	268.641113	1	0
2	01 01 2018	00:20:00	306.376587	5.216037	390.900016	272.564789	1	0
3	01 01 2018	00:30:00	419.645904	5.659674	516.127569	271.258087	1	0
4	01 01 2018	00:40:00	380.650696	5.577941	491.702972	265.674286	1	0

In [183]:

```
count=0
for items in df["LV ActivePower (kW)"]:
    if items==0:
        count=count+1


count
```

Out[183]:

```
10781
```

In [185]:

```
count=0
for items in df["Wind Speed (m/s)"]:
    if items==0:
        count=count+1
```

```
count
```

Out[185]:

```
10
```

In [189]:

```
count=0
for items in df3["Theoretical_Power_Curve (KWh)"]:
    if items==0:
        count=count+1
        df4=df3.drop(items)

count
```

Out[189]:

```
7749
```

In [187]:

```
count=0
for items in df["Wind Direction (°)"]:
    if items==0:
        count=count+1


count
```

Out[187]:

```
75
```