In [4]:

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

In [5]:

```
data=pd.read_excel("C:/Users/Furkan/Desktop/FinalData.xlsx")
```

In [6]:

```
df=data.copy()
```

In [7]:

```
for items in df.index:
    if df.loc[items,"LV ActivePower (kW)"]==0 and df.loc[items,"Wind Speed (m/s)"]>=3.5:
        df=df.drop(items)
```

In [8]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 48313 entries, 0 to 50529
Data columns (total 23 columns):
 #   Column                         Non-Null Count  Dtype  
---  ------                         --------------  -----  
 0   Date                           48313 non-null  object 
 1   Time                           48313 non-null  object 
 2   LV ActivePower (kW)            48313 non-null  float64
 3   Wind Speed (m/s)               48313 non-null  float64
 4   Theoretical_Power_Curve (KWh)  48313 non-null  float64
 5   Wind Direction (°)             48313 non-null  float64
 6   Month                          48313 non-null  int64  
 7   Day/Night                      48313 non-null  int64  
 8   Temp                           48313 non-null  int64  
 9   Sun Hour                       48313 non-null  float64
 10  Moon Illumunation              48313 non-null  int64  
 11  Moonrise                       48313 non-null  object 
 12  Moonset                        48313 non-null  object 
 13  Sunrise                        48313 non-null  object 
 14  Sunset                         48313 non-null  object 
 15  DewPoint                       48313 non-null  int64  
 16  WindChillC                     48313 non-null  int64  
 17  WindGust                       48313 non-null  int64  
 18  Humidity                       48313 non-null  int64  
 19  RainMM                         48313 non-null  float64
 20  Pressure                       48313 non-null  int64  
 21  Visibility                     48313 non-null  int64  
 22  Density                        48313 non-null  float64
dtypes: float64(7), int64(10), object(6)
memory usage: 10.1+ MB
```

In [9]:

```
from sklearn.cluster import KMeans
```

In [10]:

```
list_of_inertia=[]
for items in range(3,21):
    kmeans_clustering=KMeans(n_clusters=items,random_state=15,max_iter=300)
    kmeans_clustering.fit(df["LV ActivePower (kW)"].values.reshape(-1,1))
    list_of_inertia.append(kmeans_clustering.inertia_)
```

In [11]:

```
list_of_inertia
```

Out[11]:

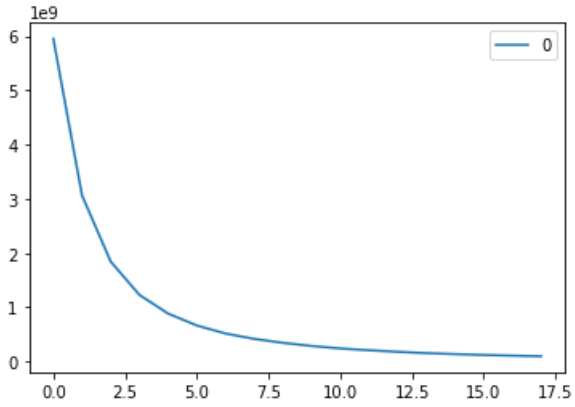
```
[5949550946.887177,
 3062634716.7541704,
 1843421959.7294416,
 1230028904.8773026,
 885934449.7084732,
 666440942.4969716,
 517857872.4725709,
 419305093.72755724,
 346678709.8857388,
 288836067.85985726,
 242668668.34289414,
 209975224.967791,
 181129917.14061648,
 156892388.05392748,
 138381687.9385177,
 123093652.02511719,
 110023872.17902423,
 99113616.82461463]
```

In [12]:

```
sns.lineplot(data=pd.DataFrame(list_of_inertia))
```

Out[12]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f2795be388>
```



In [13]:

```
kmeans_exp_1=KMeans(n_clusters=8,max_iter=300,random_state=15)
```

In [14]:

```
kmeans_exp_1.fit(df["LV ActivePower (kW)"].values.reshape(-1,1))
```

Out[14]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=8, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=15, tol=0.0001, verbose=0)
```

In [15]:

```
kmeans_exp_1.inertia_
```

Out[15]:

```
666440942.4969716
```

In [16]:

```
kmeans_exp_1.labels_
```

Out[16]:

```
array([1, 1, 1, ..., 0, 6, 2])
```

In [17]:

```
pd.DataFrame(kmeans_exp_1.labels_).T
```

Out[17]:

	0	1	2	3	4	5	6	7	8	9	...	48303	48304	48305	48306	48307	48308	48309	48310	48311	48312
0	1	1	1	1	1	1	1	1	1	1	...	2	5	5	5	5	2	0	0	6	2

1 rows × 48313 columns

In [18]:

```
kmeans_exp_1.cluster_centers_
```

Out[18]:

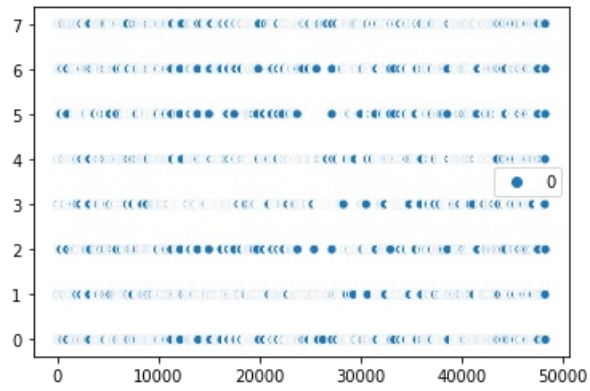
```
array([[1935.05539594],
       [ 433.21839602],
       [3025.16372689],
       [  41.04942939],
       [ 880.57582071],
       [3533.70742627],
       [2490.74450172],
       [1389.33031345]])
```

In [19]:

```
sns.scatterplot(data=pd.DataFrame(kmeans_exp_1.labels_),x_bins=500,y_bins=50)
```

Out[19]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f27a8fc708>
```



In [ ]:

```

```

In [20]:

```
a=pd.DataFrame(kmeans_exp_1.labels_)
a.columns=["Cluster"]
```

In [21]:

```
a.Cluster.value_counts()
```

Out[21]:

```
3    14462
5     7687
1     6504
4     5270
7     4239
0     3548
6     3463
2     3140
Name: Cluster, dtype: int64
```

In [22]:

```
list_of_inertia_2=[]
for items in range(3,21):
    kmeans_clustering=KMeans(n_clusters=items,random_state=15,max_iter=300)
    kmeans_clustering.fit(df["Wind Speed (m/s)"].values.reshape(-1,1))
    list_of_inertia_2.append(kmeans_clustering.inertia_)
```

In [23]:

```
list_of_inertia_2
```

Out[23]:

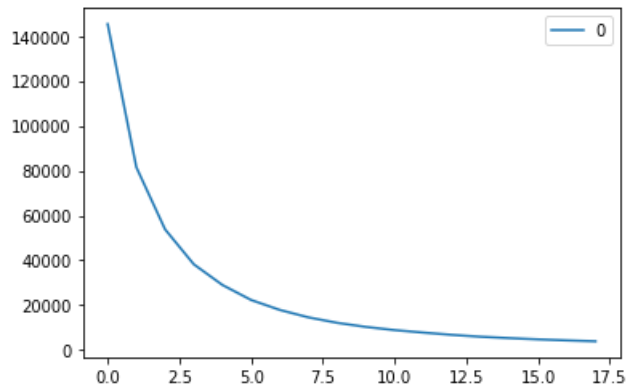
```
[145334.95261306292,
 81492.11070325745,
 53831.74157278696,
 38253.03414548813,
 29017.357650404876,
 22327.138177423283,
 17893.591427936903,
 14571.942860809306,
 12145.259835077311,
 10330.604953440565,
 8902.550900205048,
 7803.625455928432,
 6787.12293813029,
 5949.031223443224,
 5329.895123532263,
 4777.6103202084105,
 4321.373463335205,
 3919.086080769841]
```

In [24]:

```
sns.lineplot(data=pd.DataFrame(list_of_inertia_2))
```

Out[24]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f279551748>
```



In [25]:

```
kmeans_exp_2=KMeans(n_clusters=8,max_iter=300,random_state=15)
```

In [26]:

```
kmeans_exp_2.fit(df["Wind Speed (m/s)"].values.reshape(-1,1))
```

Out[26]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=8, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=15, tol=0.0001, verbose=0)
```

In [27]:

```
kmeans_exp_2.labels_
```

Out[27]:

```
array([3, 3, 3, ..., 0, 7, 7])
```

In [28]:

```
kmeans_exp_2.cluster_centers_
```

Out[28]:

```
array([[ 8.00176898],
       [ 3.93157946],
       [15.02390396],
       [ 6.07843663],
       [12.30419596],
       [18.87039919],
       [ 1.93647288],
       [10.09920461]])
```

In [29]:

```
kmeans_exp_2.inertia_
```

Out[29]:

```
22327.138177423283
```

In [30]:

```
b=pd.DataFrame(kmeans_exp_2.labels_)
b.columns=["Cluster"]
```

In [31]:

```
b.Cluster.value_counts()
```

Out[31]:

```
3    8703
0    8302
1    7518
6    7473
7    6650
4    5445
2    2786
5    1436
Name: Cluster, dtype: int64
```

In [32]:

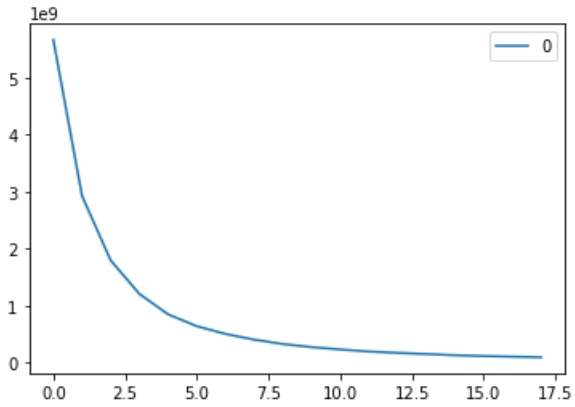
```
list_of_inertia_3=[]
for items in range(3,21):
    kmeans_clustering=KMeans(n_clusters=items,random_state=15,max_iter=300)
    kmeans_clustering.fit(df["Theoretical_Power_Curve (KWh)"].values.reshape(-1,1))
    list_of_inertia_3.append(kmeans_clustering.inertia_)
```

In [33]:

```
sns.lineplot(data=pd.DataFrame(list_of_inertia_3))
```

Out[33]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f279a80bc8>
```



In [34]:

```
kmeans_exp_3=KMeans(n_clusters=8,max_iter=300,random_state=15)
```

In [35]:

```
kmeans_exp_3.fit(df["Theoretical_Power_Curve (KWh)"].values.reshape(-1,1))
```

Out[35]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=8, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=15, tol=0.0001, verbose=0)
```

In [36]:

```
kmeans_exp_3.labels_
```

Out[36]:

```
array([1, 1, 1, ..., 2, 4, 7])
```

In [37]:

```
kmeans_exp_3.cluster_centers_
```

Out[37]:

```
array([[3553.06177379],
       [ 414.67225006],
       [1861.90060217],
       [ 835.02853479],
       [2456.24476182],
       [  38.65452645],
       [1318.37116904],
       [3067.27232653]])
```

In [38]:

```
kmeans_exp_3.inertia_
```

Out[38]:

```
639416324.5652404
```

In [39]:

```
c=pd.DataFrame(kmeans_exp_3.labels_)
c.columns=["Cluster"]
```

In [40]:

```
c.Cluster.value_counts()
```

Out[40]:

```
5    13280
0     9893
1     5642
3     5192
6     4304
2     3490
7     3481
4     3031
Name: Cluster, dtype: int64
```

In [41]:

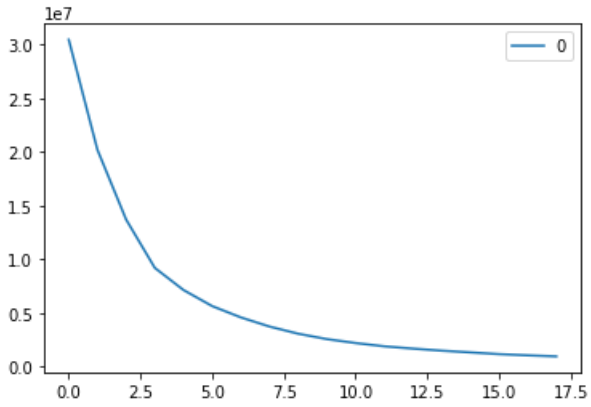
```
list_of_inertia_4=[]
for items in range(3,21):
    kmeans_clustering=KMeans(n_clusters=items,random_state=15,max_iter=300)
    kmeans_clustering.fit(df["Wind Direction (°)"].values.reshape(-1,1))
    list_of_inertia_4.append(kmeans_clustering.inertia_)
```

In [42]:

```
sns.lineplot(data=pd.DataFrame(list_of_inertia_4))
```

Out[42]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f27c2750c8>
```



In [43]:

```
kmeans_exp_4=KMeans(n_clusters=8,max_iter=300,random_state=15)
```

In [44]:

```
kmeans_exp_4.fit(df["Wind Direction (°)"].values.reshape(-1,1))
```

Out[44]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=8, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=15, tol=0.0001, verbose=0)
```

In [45]:

```
kmeans_exp_4.labels_
```

Out[45]:

```
array([4, 4, 4, ..., 5, 5, 5])
```

In [46]:

```
kmeans_exp_4.inertia_
```

Out[46]:

```
5638038.056385048
```

In [47]:

```
kmeans_exp_4.cluster_centers_
```

Out[47]:

```
array([[191.70151862],
       [ 51.57363633],
       [332.69337911],
       [ 25.11108028],
       [270.95486598],
       [ 72.75491057],
       [128.00789744],
       [222.52916315]])
```

In [48]:

```
d=pd.DataFrame(kmeans_exp_4.labels_)
d.columns=["Cluster"]
d.Cluster.value_counts()
```

Out[48]:

```
1    10214
5     9485
0     8060
3     7838
7     4944
4     3222
2     2413
6     2137
Name: Cluster, dtype: int64
```

In [49]:

```
#Experiments
```

In [50]:

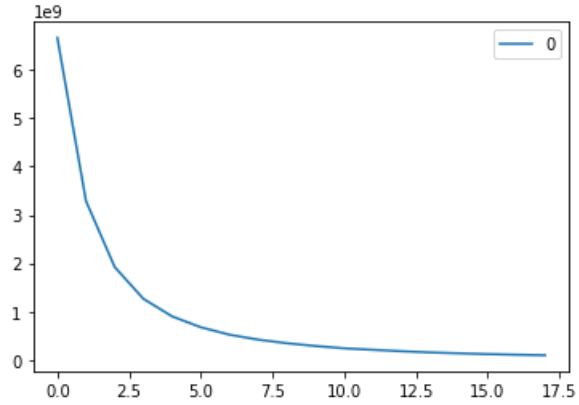
```
list_of_inertia_5=[]
for items in range(3,21):
    kmeans_clustering=KMeans(n_clusters=items,random_state=15,max_iter=300)
    kmeans_clustering.fit(df[["LV ActivePower (kW)","Wind Speed (m/s)"]].values.reshape(-1,1))
    list_of_inertia_5.append(kmeans_clustering.inertia_)
```

In [51]:

```
sns.lineplot(data=pd.DataFrame(list_of_inertia_5))
```

Out[51]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f27c7514c8>
```



In [52]:

```
kmeans_exp_5=KMeans(random_state=15,n_clusters=8)
```

In [53]:

```
kmeans_exp_5.fit(df[["LV ActivePower (kW)","Wind Speed (m/s)"]])
```

Out[53]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=8, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=15, tol=0.0001, verbose=0)
```

In [54]:

```
kmeans_exp_5.labels_
```

Out[54]:

```
array([6, 6, 6, ..., 2, 7, 4])
```

In [55]:

```
kmeans_exp_5.cluster_centers_
```

Out[55]:

```
array([[4.40490354e+01, 2.94735640e+00],
       [3.52632525e+03, 1.45580498e+01],
       [1.93302957e+03, 9.07341861e+00],
       [8.95755764e+02, 7.03511708e+00],
       [2.98950651e+03, 1.12346637e+01],
       [1.39752407e+03, 8.04796232e+00],
       [4.48160588e+02, 5.85808982e+00],
       [2.47007454e+03, 1.01720227e+01]])
```

In [56]:

```
kmeans_exp_5.inertia_
```

Out[56]:

```
665909780.2417341
```

In [57]:

```
e=pd.DataFrame(kmeans_exp_5.labels_)
```

```
e.columns=["Cluster"]
e.Cluster.value_counts()
```

Out[57]:

```
0    14616
1     7849
6     6535
3     5184
5     4156
2     3478
7     3340
4     3155
Name: Cluster, dtype: int64
```

In [58]:

```
df_cluster_1=df[["LV ActivePower (kW)","Wind Speed (m/s)"]]
```

In [59]:

```
df_cluster_1=df_cluster_1.assign(Cluster=kmeans_exp_5.labels_)
```

In [60]:

```
df_cluster_1
```

Out[60]:

	LV ActivePower (kW)	Wind Speed (m/s)	Cluster
0	380.047791	5.311336	6
1	453.769196	5.672167	6
2	306.376587	5.216037	6
3	419.645904	5.659674	6
4	380.650696	5.577941	6
...	...	...	...
50525	2963.980957	11.404030	4
50526	1684.353027	7.332648	2
50527	2201.106934	8.435358	2
50528	2515.694092	9.421366	7
50529	2820.466064	9.979332	4

48313 rows × 3 columns

In [61]:

```
from matplotlib import pyplot as plt
```

In [62]:

```
np.unique(kmeans_exp_5.labels_)
```

Out[62]:

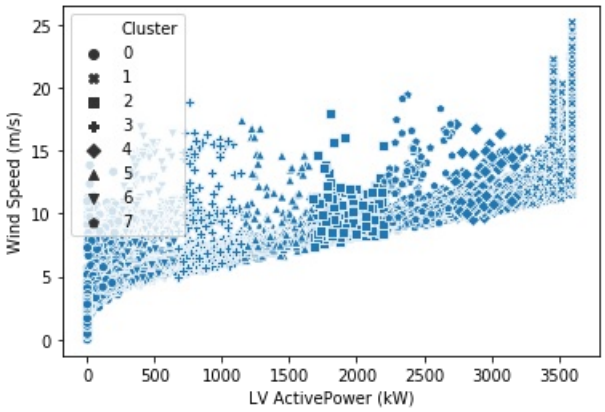
```
array([0, 1, 2, 3, 4, 5, 6, 7])
```

In [63]:

```
sns.scatterplot(x="LV ActivePower (kW)",y="Wind Speed
(m/s)",style="Cluster",data=df_cluster_1,x_bins=50,y_bins=50)
```

Out[63]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f27cf98d88>
```



In [64]:

```
from sklearn.preprocessing import scale
```

In [65]:

```
kmeans_exp_6=KMeans(random_state=15,n_clusters=8)
```

In [66]:

```
kmeans_exp_6.fit(scale(df[["LV ActivePower (kW)","Wind Speed (m/s)"]]))
```

Out[66]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=8, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=15, tol=0.0001, verbose=0)
```

In [67]:

```
df_cluster_2=pd.DataFrame(scale(df[["LV ActivePower (kW)","Wind Speed (m/s)"]]))
```

In [68]:

```
df_cluster_2=df_cluster_2.assign(Cluster=kmeans_exp_6.labels_)
```

In [69]:

```
df_cluster_2.columns=[["LV ActivePower (kW)","Wind Speed (m/s)","Cluster"]]
```

In [70]:

```
df_cluster_2
```

Out[70]:

	LV ActivePower (kW)	Wind Speed (m/s)	Cluster
0	-0.753184	-0.531873	4
1	-0.696963	-0.447458	7
2	-0.809366	-0.554168	4
3	-0.722986	-0.450381	7
4	-0.752724	-0.469502	4
...	...	...	...
48308	1.217341	0.893498	1
48309	0.241488	-0.058992	2

	LV ActivePower (kW)	Wind Speed (m/s)	Cluster
48310	0.635568	0.198984	5
48311	0.875474	0.429658	5
48312	1.107895	0.560193	1

48313 rows × 3 columns

In [71]:

```
#Experiment 2
```

In [72]:

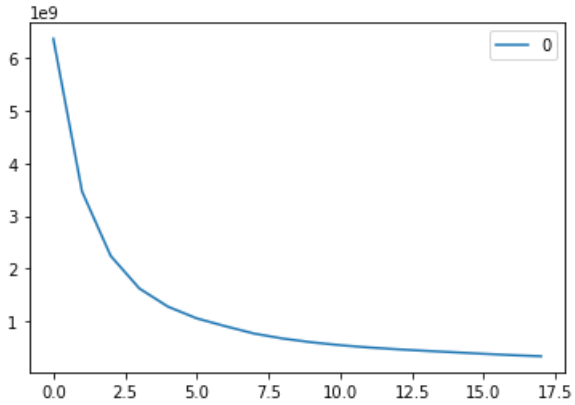
```
list_of_inertia_6=[]
for items in range(3,21):
    kmeans_clustering=KMeans(n_clusters=items,random_state=15,max_iter=300)
    kmeans_clustering.fit(df[["LV ActivePower (kW)","Wind Speed (m/s)","Wind Direction (°)"]])
    list_of_inertia_6.append(kmeans_clustering.inertia_)
```

In [73]:

```
sns.lineplot(data=pd.DataFrame(list_of_inertia_6))
```

Out[73]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f27c09ecc8>
```



In [74]:

```
kmeans_exp_7=KMeans(random_state=15,n_clusters=8)
```

In [75]:

```
kmeans_exp_7.fit(df[["LV ActivePower (kW)","Wind Speed (m/s)","Wind Direction (°)"]])
```

Out[75]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=8, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=15, tol=0.0001, verbose=0)
```

In [76]:

```
kmeans_exp_7.cluster_centers_
```

Out[76]:

```
array([[8.71679274e+02, 6.97645576e+00, 1.02251719e+02],
       [2.47270823e+03, 1.01750805e+01, 9.65651085e+01],
       [3.52870266e+03, 1.45829497e+01, 1.40932588e+02],
       [3.96666706e+01, 2.90153597e+00, 1.54360598e+02],
       [1.38089091e+03, 8.01542537e+00, 9.93574330e+01],
       [2.99892153e+03, 1.12559996e+01, 1.00508500e+02],
```

```
       [4.25853699e+02, 5.78146154e+00, 1.10517297e+02],
       [1.92598839e+03, 9.06017077e+00, 1.02406715e+02]])
```

In [77]:

```
kmeans_exp_7.labels_
```

Out[77]:

```
array([6, 6, 6, ..., 1, 1, 5])
```

In [78]:

```
kmeans_exp_7.inertia_
```

Out[78]:

```
1059685098.6015022
```

In [79]:

```
df_cluster_3=df[["LV ActivePower (kW)","Wind Speed (m/s)","Wind Direction (°)"]]
```

In [80]:

```
df_cluster_3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 48313 entries, 0 to 50529
Data columns (total 3 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   LV ActivePower (kW)  48313 non-null  float64
 1   Wind Speed (m/s)     48313 non-null  float64
 2   Wind Direction (°)   48313 non-null  float64
dtypes: float64(3)
memory usage: 2.7 MB
```

In [81]:

```
len(kmeans_exp_7.labels_)
```

Out[81]:

```
48313
```

In [82]:

```
df_cluster_3=df_cluster_3.assign(Cluster=kmeans_exp_7.labels_)
```

In [83]:

```
df_cluster_3
```

Out[83]:

	LV ActivePower (kW)	Wind Speed (m/s)	Wind Direction (°)	Cluster
0	380.047791	5.311336	259.994904	6
1	453.769196	5.672167	268.641113	6
2	306.376587	5.216037	272.564789	6
3	419.645904	5.659674	271.258087	6
4	380.650696	5.577941	265.674286	6
...	...	...	...	...
50525	2963.980957	11.404030	80.502724	5

	LV ActivePower (kW)	Wind Speed (m/s)	Wind Direction (°)	Cluster
50526	1684.353027	7.332648	84.062599	7
50527	2201.106934	8.435358	84.742500	1
50528	2515.694092	9.421366	84.297913	1
50529	2820.466064	9.979332	82.274620	5

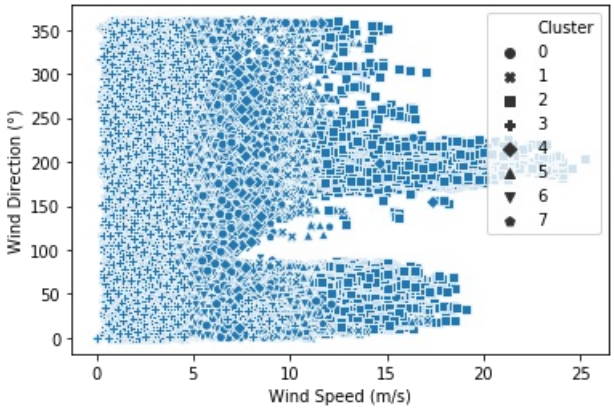
48313 rows × 4 columns

In [84]:

```
sns.scatterplot(x="Wind Speed (m/s)",y="Wind Direction (°)",data=df_cluster_3,style="Cluster")
```

Out[84]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f27c3ffe08>
```



In [85]:

```
#Experiment3
```

In [86]:

```
kmeans_exp_8=KMeans(n_clusters=17,random_state=15)
```

In [87]:

```
kmeans_exp_8.fit(df[["LV ActivePower (kW)","Wind Speed (m/s)","Theoretical_Power_Curve (KWh)","Wind Direction (°)"]])
```

Out[87]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=17, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=15, tol=0.0001, verbose=0)
```

In [88]:

```
kmeans_exp_8.cluster_centers_
```

Out[88]:

```
array([[3.18892271e+03, 1.17305781e+01, 3.45068529e+03, 1.04410666e+02],
       [6.07250965e+02, 6.24421476e+00, 7.13340567e+02, 1.08308492e+02],
       [1.91484139e+03, 8.97638876e+00, 2.12956778e+03, 1.01882472e+02],
       [1.36480088e+01, 2.44771293e+00, 2.18992563e+01, 2.68583960e+02],
       [2.21937305e+03, 9.57235500e+00, 2.50994626e+03, 9.85757588e+01],
       [2.81541025e+03, 1.09917608e+01, 3.23128845e+03, 9.69887333e+01],
       [1.33197716e+03, 7.87675999e+00, 1.46232510e+03, 1.02315276e+02],
       [2.05947349e+02, 4.62689235e+00, 2.52556431e+02, 1.24044097e+02],
       [3.57231380e+02, 8.73115664e+00, 1.98011573e+03, 4.77371718e+01],
       [2.49712438e+03, 1.02663271e+01, 2.92557651e+03, 9.79219401e+01],
       [8.33893515e+02, 6.81883103e+00, 9.39115571e+02, 1.03827983e+02],
       [1.07728902e+03, 7.35263062e+00, 1.18552096e+03, 1.00136633e+02],
```

```
       [3.55003707e+03, 1.48542516e+01, 3.59068598e+03, 1.43880367e+02],
       [7.06541265e+02, 1.21533188e+01, 3.27561613e+03, 7.06871175e+01],
       [1.61795160e+03, 8.40412050e+00, 1.77081753e+03, 9.87370313e+01],
       [1.51305250e+01, 2.59055822e+00, 2.42820134e+01, 6.92179479e+01],
       [4.05151699e+02, 5.53906153e+00, 4.83638020e+02, 1.15626449e+02]])
```

In [89]:

```
kmeans_exp_8.labels_
```

Out[89]:

```
array([16, 16, 16, ...,  2,  4,  9])
```

In [90]:

```
df_cluster_4=df[["LV ActivePower (kW)","Wind Speed (m/s)","Theoretical_Power_Curve (KWh)","Wind Direction (°)"]]
```

In [91]:

```
df_cluster_4=df_cluster_4.assign(Cluster=kmeans_exp_8.labels_)
```

In [92]:

```
df_cluster_4
```

Out[92]:

	LV ActivePower (kW)	Wind Speed (m/s)	Theoretical_Power_Curve (KWh)	Wind Direction (°)	Cluster
0	380.047791	5.311336	416.328908	259.994904	16
1	453.769196	5.672167	519.917511	268.641113	16
2	306.376587	5.216037	390.900016	272.564789	16
3	419.645904	5.659674	516.127569	271.258087	16
4	380.650696	5.577941	491.702972	265.674286	16
...	...	...	...	...	...
50525	2963.980957	11.404030	3397.190793	80.502724	5
50526	1684.353027	7.332648	1173.055771	84.062599	6
50527	2201.106934	8.435358	1788.284755	84.742500	2
50528	2515.694092	9.421366	2418.382503	84.297913	4
50529	2820.466064	9.979332	2779.184096	82.274620	9

48313 rows × 5 columns

In [93]:

```
kmeans_exp_8.inertia_
```

Out[93]:

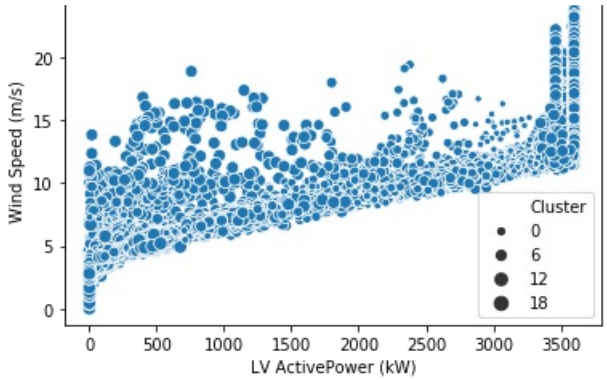
```
1165187574.7937226
```

In [94]:

```
sns.scatterplot(x="LV ActivePower (kW)",y="Wind Speed (m/s)",size="Cluster",data=df_cluster_4)
```

Out[94]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f27c405988>
```



In [95]:

```
df_cluster_4.Cluster.value_counts()
```

Out[95]:

```
12    7093
15    6482
3     5428
7     3465
16    3150
1     2890
10    2579
11    2354
0     2270
6     2185
5     2097
14    2063
2     1943
9     1919
4     1854
8      306
13     235
Name: Cluster, dtype: int64
```

In [96]:

```
df_clust=df.drop(["Sunrise","Sunset","Moonrise","Moonset","Day/Night","Month","Pressure","Visibility","Moon Illumination",
                  "Date","Time"],axis=1)
```

In [97]:

```
df_clust.describe()
```

Out[97]:

	LV ActivePower (kW)	Wind Speed (m/s)	Theoretical_Power_Curve (KWh)	Wind Direction (°)	Temp	Sun Hour	DewPoint	WindChillC
count	48313.000000	48313.000000	48313.000000	48313.000000	48313.000000	48313.000000	48313.000000	48313.000000
mean	1367.691704	7.584808	1504.010263	124.291200	16.340095	10.529758	10.836007	16.457682
std	1311.305617	4.274504	1375.027496	93.271962	7.281467	3.125715	5.794409	8.027477
min	-2.471405	0.000000	0.000000	0.000000	-1.000000	3.400000	-5.000000	-5.000000
25%	120.164101	4.192660	159.625118	50.099098	10.000000	8.700000	6.000000	10.000000
50%	924.296509	7.139705	1080.139085	73.949280	16.000000	11.600000	11.000000	17.000000
75%	2566.322998	10.381380	3007.323841	201.751099	21.000000	11.600000	16.000000	22.000000
max	3618.732910	25.206011	3600.000000	359.997589	32.000000	14.500000	22.000000	32.000000

In [98]:

```
list_of_inertia_7=[]
for items in range(3,21):
```

```
    kmeans_clustering=KMeans(n_clusters=items,random_state=15,max_iter=300)
    kmeans_clustering.fit(df_clust)
    list_of_inertia_7.append(kmeans_clustering.inertia_)
```

In [99]:

```
list_of_inertia_7
```

Out[99]:

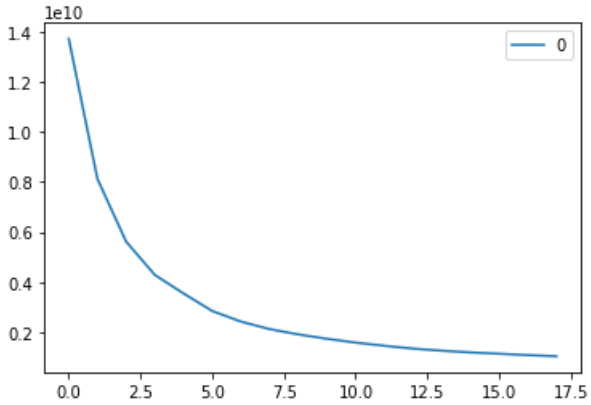
```
[13721860012.030508,
 8117318385.580713,
 5613944087.730892,
 4282542631.9201155,
 3542798450.39081,
 2841629808.8082385,
 2419008520.392374,
 2117104020.1980238,
 1905898599.8631473,
 1730404069.8885908,
 1577359796.8438053,
 1451298124.43838,
 1337392036.3970506,
 1254152778.6105337,
 1186587577.509508,
 1132244043.7206438,
 1078255697.032554,
 1032032064.070267]
```

In [100]:

```
sns.lineplot(data=pd.DataFrame(list_of_inertia_7))
```

Out[100]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1f27c0a5b88>
```



In [101]:

```
kmeans_exp_9=KMeans(n_clusters=8,random_state=15)
```

In [102]:

```
kmeans_exp_9.fit(df_clust)
```

Out[102]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=8, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=15, tol=0.0001, verbose=0)
```

In [103]:

```
kmeans_exp_9.cluster_centers_
```

Out[103]:

```
array([[ 4.69092918e+01,  2.90238656e+00,  6.20968747e+01,
         1.53422873e+02,  1.64290577e+01,  1.13074142e+01,
         1.13610430e+01,  1.70467729e+01,  1.01635349e+01,
         7.01887936e+01, -1.38083989e-15,  1.22158875e+00],
       [ 3.48250858e+03,  1.42231300e+01,  3.56516388e+03,
         1.35747473e+02,  1.48104356e+01,  9.41408577e+00,
         8.39357295e+00,  1.43059008e+01,  2.11003247e+01,
         6.62817154e+01,  1.56757362e-03,  1.22904866e+00],
       [ 1.51905346e+03,  8.22188808e+00,  1.66451583e+03,
         1.00850290e+02,  1.70736095e+01,  1.06714793e+01,
         1.17391716e+01,  1.70979882e+01,  1.45398817e+01,
         7.10731361e+01,  7.45562130e-03,  1.21972587e+00],
       [ 4.78368622e+02,  5.78863721e+00,  5.67853427e+02,
         1.12837460e+02,  1.70926119e+01,  1.09160398e+01,
         1.17914375e+01,  1.74767355e+01,  1.21715475e+01,
         6.99467816e+01, -9.41954847e-16,  1.21893702e+00],
       [ 2.70968389e+03,  1.07527045e+01,  3.12800214e+03,
         9.79235082e+01,  1.71065172e+01,  1.00725765e+01,
         1.10696099e+01,  1.69467414e+01,  1.82258818e+01,
         6.88199019e+01,  4.90539594e-03,  1.22011821e+00],
       [ 2.10233038e+03,  9.35471936e+00,  2.37109451e+03,
         9.94034930e+01,  1.67007939e+01,  1.03494114e+01,
         1.12031207e+01,  1.65767862e+01,  1.62496578e+01,
         7.05688475e+01,  2.29947988e-02,  1.22207784e+00],
       [ 9.55938306e+02,  7.13158846e+00,  1.08563788e+03,
         1.01585287e+02,  1.70920906e+01,  1.05935202e+01,
         1.17513925e+01,  1.72515782e+01,  1.33685481e+01,
         7.05575566e+01,  1.94949870e-03,  1.21948268e+00],
       [ 5.24553516e+02,  1.06499947e+01,  2.74485758e+03,
         5.78790778e+01,  5.78199052e+00,  4.88317536e+00,
         2.74881517e+00,  2.71327014e+00,  2.20165877e+01,
         8.11255924e+01,  2.15639810e-01,  1.27120343e+00]])
```

In [104]:

```
kmeans_exp_9.labels_
```

Out[104]:

```
array([3, 3, 3, ..., 5, 5, 4])
```

In [105]:

```
kmeans_exp_9.inertia_
```

Out[105]:

```
2841629808.8082385
```

In [106]:

```
from sklearn.cluster import DBSCAN
```

In [107]:

```
dbscan_exp_1=DBSCAN(eps=1000000,min_samples=1500)
```

In [108]:

```
#dbscan_exp_1.fit(df_clust)
```

In [109]:

```
#dbscan_exp_1
```

In [110]:

```
# Silhoutte Values
```

In [111]:

```
from sklearn.metrics import silhouette_samples, silhouette_score
```

In [3]:

```
?silhouette_score
```

In [122]:

```
silhouette_score(df["LV ActivePower (kW)"].values.reshape(-1,1),kmeans_exp_1.labels_)
```

Out[122]:

```
0.6478462783673685
```

In [124]:

```
silhouette_score(df["Wind Speed (m/s)"].values.reshape(-1,1),kmeans_exp_2.labels_)
```

Out[124]:

```
0.5289507261539236
```

In [125]:

```
silhouette_score(df["Theoretical_Power_Curve (KWh)"].values.reshape(-1,1),kmeans_exp_3.labels_)
```

Out[125]:

```
0.660672243563822
```

In [126]:

```
silhouette_score(df["Wind Direction (°)"].values.reshape(-1,1),kmeans_exp_4.labels_)
```

Out[126]:

```
0.5342132017381143
```

In [ ]:

```
silhouette_score(df["Wind Direction (°)"].values.reshape(-1,1),kmeans_exp_4.labels_)
```

In [127]:

```
silhouette_score(df[["LV ActivePower (kW)","Wind Speed (m/s)"]],kmeans_exp_5.labels_)
```

Out[127]:

```
0.6479522758184177
```

In [128]:

```
kmeans_exp_5.inertia_
```

Out[128]:

```
665909780.2417341
```

In [129]:

```
kmeans_exp_10=KMeans(n_clusters=6,random_state=15)
```

In [131]:

```
kmeans_exp_10.fit((df[["LV ActivePower (kW)","Wind Speed (m/s)"]]))
```

Out[131]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=6, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=15, tol=0.0001, verbose=0)
```

In [132]:

```
kmeans_exp_10.inertia_
```

Out[132]:

```
1229441473.9298115
```

In [133]:

```
silhouette_score(df[["LV ActivePower (kW)","Wind Speed (m/s)"]],kmeans_exp_10.labels_)
```

Out[133]:

```
0.6546166342498526
```

In [134]:

```
kmeans_exp_11=KMeans(n_clusters=5,random_state=15)
```

In [135]:

```
kmeans_exp_11.fit((df[["LV ActivePower (kW)","Wind Speed (m/s)"]]))
```

Out[135]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=5, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=15, tol=0.0001, verbose=0)
```

In [136]:

```
silhouette_score(df[["LV ActivePower (kW)","Wind Speed (m/s)"]],kmeans_exp_11.labels_)
```

Out[136]:

```
0.6576921304462101
```

In [137]:

```
kmeans_exp_11.inertia_
```

Out[137]:

```
1843590397.3715093
```

In [138]:

```
kmeans_exp_12=KMeans(n_clusters=7,random_state=15)
```

In [139]:

```
kmeans_exp_12.fit((df[["LV ActivePower (kW)","Wind Speed (m/s)"]]))
```

Out[139]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=7, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=15, tol=0.0001, verbose=0)
```

In [140]:

```
silhouette_score(df[["LV ActivePower (kW)","Wind Speed (m/s)"]],kmeans_exp_12.labels_)
```

Out[140]:

```
0.650806137309017
```

In [141]:

```
kmeans_exp_12.inertia_
```

Out[141]:

```
885643293.4037602
```

In [143]:

```
silhouette_score(df[["LV ActivePower (kW)","Wind Speed (m/s)","Wind Direction (°)"]],kmeans_exp_7.
labels_)
```

Out[143]:

```
0.5346192051379163
```

In [144]:

```
kmeans_exp_7.inertia_
```

Out[144]:

```
1059685098.6015022
```

In [145]:

```
kmeans_exp_13=KMeans(n_clusters=7,random_state=15)
```

In [146]:

```
kmeans_exp_13.fit(df[["LV ActivePower (kW)","Wind Speed (m/s)","Wind Direction (°)"]])
```

Out[146]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=7, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=15, tol=0.0001, verbose=0)
```

In [147]:

```
kmeans_exp_13.inertia_
```

Out[147]:

```
1280774023.9213927
```

In [148]:

```
silhouette_score(df[["LV ActivePower (kW)","Wind Speed (m/s)","Wind Direction (°)"]],kmeans_exp_13
.labels_)
```

Out[148]:

0.5561459813073801

In [149]:

```
kmeans_exp_14=KMeans(n_clusters=6,random_state=15)
```

In [150]:

```
kmeans_exp_14.fit(df[["LV ActivePower (kW)","Wind Speed (m/s)","Wind Direction (°)"]])
```

Out[150]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=6, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=15, tol=0.0001, verbose=0)
```

In [151]:

```
kmeans_exp_14.inertia_
```

Out[151]:

```
1626139832.088378
```

In [152]:

```
silhouette_score(df[["LV ActivePower (kW)","Wind Speed (m/s)","Wind Direction (°)"]],kmeans_exp_14
.labels_)
```

Out[152]:

```
0.5782076403302493
```

In [153]:

```
silhouette_score(df[["LV ActivePower (kW)","Wind Speed (m/s)","Theoretical_Power_Curve (KWh)","Win
d Direction (°)"]],kmeans_exp_8.labels_)
```

Out[153]:

```
0.4475707098056214
```

In [154]:

```
kmeans_exp_8.inertia_
```

Out[154]:

```
1165187574.7937226
```

In [155]:

```
kmeans_exp_15=KMeans(n_clusters=7,random_state=15)
```

In [156]:

```
kmeans_exp_15.fit(df[["LV ActivePower (kW)","Wind Speed (m/s)","Theoretical_Power_Curve (KWh)","Wi
nd Direction (°)"]])
```

Out[156]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=7, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=15, tol=0.0001, verbose=0)
```

In [157]:

```
kmeans_exp_15.inertia_
```

Out[157]:

```
3519873936.391018
```

In [158]:

```
silhouette_score(df[["LV ActivePower (kW)","Wind Speed (m/s)","Theoretical_Power_Curve (KWh)","Win
d Direction (°)"]],kmeans_exp_15.labels_)
```

Out[158]:

```
0.5884869663912284
```

In [159]:

```
kmeans_exp_16=KMeans(n_clusters=6,random_state=15)
```

In [160]:

```
kmeans_exp_16.fit(df[["LV ActivePower (kW)","Wind Speed (m/s)","Theoretical_Power_Curve (KWh)","Wi
nd Direction (°)"]])
```

Out[160]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=6, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=15, tol=0.0001, verbose=0)
```

In [161]:

```
silhouette_score(df[["LV ActivePower (kW)","Wind Speed (m/s)","Theoretical_Power_Curve (KWh)","Win
d Direction (°)"]],kmeans_exp_16.labels_)
```

Out[161]:

```
0.5800855002629363
```

In [162]:

```
kmeans_exp_16.inertia_
```

Out[162]:

```
4259717568.4028482
```

In [163]:

```
silhouette_score(df_clust,kmeans_exp_9.labels_)
```

Out[163]:

```
0.564922161212547
```

In [164]:

```
kmeans_exp_9.inertia_
```

Out[164]:

```
2841629808.8082385
```

In [165]:

```
kmeans_exp_17=KMeans(n_clusters=7,random_state=15)
```

In [166]:

```
kmeans_exp_17.fit(df_clust)
```

Out[166]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=7, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=15, tol=0.0001, verbose=0)
```

In [167]:

```
kmeans_exp_17.inertia_
```

Out[167]:

```
3542798450.39081
```

In [168]:

```
silhouette_score(df_clust,kmeans_exp_17.labels_)
```

Out[168]:

```
0.5858043461274995
```

In [170]:

```
kmeans_exp_18=KMeans(n_clusters=6,random_state=15)
```

In [171]:

```
kmeans_exp_18.fit(df_clust)
```

Out[171]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=6, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=15, tol=0.0001, verbose=0)
```

In [172]:

```
kmeans_exp_18.inertia_
```

Out[172]:

```
4282542631.9201155
```

In [173]:

```
silhouette_score(df_clust,kmeans_exp_18.labels_)
```

Out[173]:

```
0.5773901469511109
```

In [ ]:

```

```

In [1]:

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

In [2]:

```
data=pd.read_excel("C:/Users/Furkan/Desktop/FinalData.xlsx")
```

In [3]:

```
df=data.copy()
```

In [4]:

```
for items in df.index:
    if df.loc[items,"LV ActivePower (kW)"]==0 and df.loc[items,"Wind Speed (m/s)"]>=3.5:
        df=df.drop(items)
```

In [5]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 48313 entries, 0 to 50529
Data columns (total 23 columns):
 #   Column                         Non-Null Count  Dtype  
---  ------                         --------------  -----  
 0   Date                           48313 non-null  object 
 1   Time                           48313 non-null  object 
 2   LV ActivePower (kW)            48313 non-null  float64
 3   Wind Speed (m/s)               48313 non-null  float64
 4   Theoretical_Power_Curve (KWh)  48313 non-null  float64
 5   Wind Direction (°)             48313 non-null  float64
 6   Month                          48313 non-null  int64  
 7   Day/Night                      48313 non-null  int64  
 8   Temp                           48313 non-null  int64  
 9   Sun Hour                       48313 non-null  float64
 10  Moon Illumunation              48313 non-null  int64  
 11  Moonrise                       48313 non-null  object 
 12  Moonset                        48313 non-null  object 
 13  Sunrise                        48313 non-null  object 
 14  Sunset                         48313 non-null  object 
 15  DewPoint                       48313 non-null  int64  
 16  WindChillC                     48313 non-null  int64  
 17  WindGust                       48313 non-null  int64  
 18  Humidity                       48313 non-null  int64  
 19  RainMM                         48313 non-null  float64
 20  Pressure                       48313 non-null  int64  
 21  Visibility                     48313 non-null  int64  
 22  Density                        48313 non-null  float64
dtypes: float64(7), int64(10), object(6)
memory usage: 10.1+ MB
```

In [22]:

```
df.describe().T
```

Out[22]:

	count	mean	std	min	25%	50%	75%	max
LV ActivePower (kW)	48313.0	1367.691704	1311.305617	-2.471405	120.164101	924.296509	2566.322998	3618.732910
Wind Speed (m/s)	48313.0	7.584808	4.274504	0.000000	4.192660	7.139705	10.381380	25.206011
Theoretical_Power_Curve								

	count	mean	std	min	25%	50%	75%	max
Theoretical_Power_Curve (KWh)	48313.0	1504.010263	1375.027496	0.000000	159.625118	1080.139085	3007.323841	3600.000000
Wind Direction (°)	48313.0	124.291200	93.271962	0.000000	50.099098	73.949280	201.751099	359.997589
Month	48313.0	6.563140	3.338178	1.000000	4.000000	7.000000	9.000000	12.000000
Day/Night	48313.0	0.504150	0.499988	0.000000	0.000000	1.000000	1.000000	1.000000
Temp	48313.0	16.340095	7.281467	-1.000000	10.000000	16.000000	21.000000	32.000000
Sun Hour	48313.0	10.529758	3.125715	3.400000	8.700000	11.600000	11.600000	14.500000
Moon Illumunation	48313.0	46.147041	31.648975	0.000000	17.000000	45.000000	74.000000	100.000000
DewPoint	48313.0	10.836007	5.794409	-5.000000	6.000000	11.000000	16.000000	22.000000
WindChillC	48313.0	16.457682	8.027477	-5.000000	10.000000	17.000000	22.000000	32.000000
WindGust	48313.0	14.482996	7.908197	0.000000	9.000000	13.000000	19.000000	52.000000
Humidity	48313.0	69.554261	16.489826	31.000000	56.000000	71.000000	85.000000	97.000000
RainMM	48313.0	0.005216	0.135015	0.000000	0.000000	0.000000	0.000000	3.500000
Pressure	48313.0	1014.512181	5.962183	993.000000	1010.000000	1014.000000	1019.000000	1032.000000
Visibility	48313.0	9.881605	0.899149	0.000000	10.000000	10.000000	10.000000	10.000000
Density	48313.0	1.222541	0.033878	1.146970	1.198180	1.222526	1.248446	1.310768

In [6]:

```
from sklearn.cluster import KMeans
```

Wind Gust

In [7]:

```
list_of_inertia=[]
for items in range(3,21):
    kmeans_clustering=KMeans(n_clusters=items,random_state=15,max_iter=300)
    kmeans_clustering.fit(df["WindGust"].values.reshape(-1,1))
    list_of_inertia.append(kmeans_clustering.inertia_)
```

In [8]:

```
list_of_inertia
```

Out[8]:

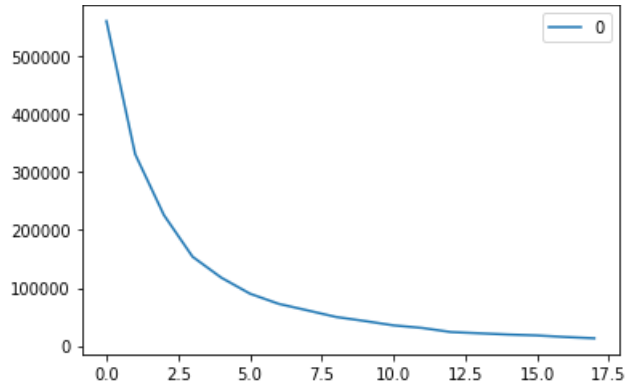
```
[560690.5094988057,
 330874.88989972015,
 225864.79518516368,
 153801.14222590494,
 117701.95160097053,
 90027.02506275156,
 72471.37050124576,
 61212.63110711296,
 49811.318345247564,
 42827.63603060018,
 35324.19737825483,
 30841.221294785984,
 23774.48097732406,
 21609.707532905617,
 19461.753499364975,
 17910.915737616986,
 15153.905581603882,
 12971.064329382765]
```

In [12]:

```
sns.lineplot(data=pd.DataFrame(list_of_inertia))
```

Out[12]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x29847b1ec08>
```



In [7]:

```
kmeans_exp_1=KMeans(n_clusters=6,max_iter=300,random_state=15)
```

In [8]:

```
kmeans_exp_1.fit(df["WindGust"].values.reshape(-1,1))
```

Out[8]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=6, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=15, tol=0.0001, verbose=0)
```

In [9]:

```
from sklearn.metrics import silhouette_samples, silhouette_score
```

In [11]:

```
silhouette_score(df["WindGust"].values.reshape(-1,1),kmeans_exp_1.labels_)
```

Out[11]:

```
0.5662882537635232
```

In [28]:

```
kmeans_exp_1.inertia_
```

Out[28]:

```
153801.14222590494
```

In [29]:

```
kmeans_exp_1.labels_
```

Out[29]:

```
array([2, 2, 2, ..., 2, 2, 2])
```

In [30]:

```
pd.DataFrame(kmeans_exp_1.labels_).T
```

Out[30]:

	0	1	2	3	4	5	6	7	8	9	...	48303	48304	48305	48306	48307	48308	48309	48310	48311	48312
0	2	2	2	2	2	2	2	2	2	2	...	2	2	2	2	2	2	2	2	2	2

1 rows × 48313 columns

In [31]:

```
kmeans_exp_1.cluster_centers_
```

Out[31]:

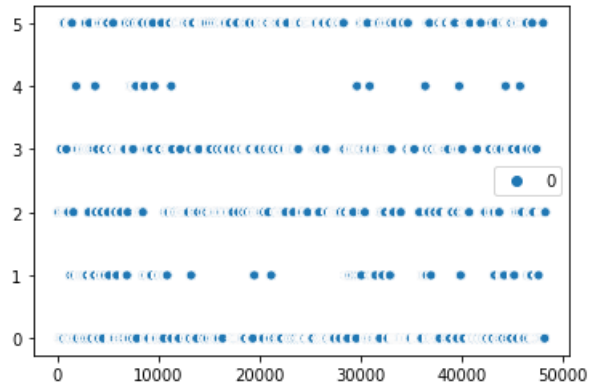
```
array([[13.91631151],
       [27.24555461],
       [ 5.17871009],
       [19.7060504 ],
       [38.74057315],
       [ 9.61526495]])
```

In [32]:

```
sns.scatterplot(data=pd.DataFrame(kmeans_exp_1.labels_),x_bins=500,y_bins=50)
```

Out[32]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2984c04c188>
```



In [33]:

```
a=pd.DataFrame(kmeans_exp_1.labels_)
a.columns=["Cluster"]
```

In [34]:

```
a.Cluster.value_counts()
```

Out[34]:

```
0    12905
5    10285
3     9801
2     9272
1     4724
4     1326
Name: Cluster, dtype: int64
```

Wind Gust (clustering with different k value)

In [40]:

```
list_of_inertia=[]
for items in range(1,31):
    kmeans_clustering=KMeans(n_clusters=items,random_state=15,max_iter=300)
    kmeans_clustering.fit(df["WindGust"].values.reshape(-1,1))
    list_of_inertia.append(kmeans_clustering.inertia_)
```

In [41]:

```
list_of_inertia
```

Out[41]:

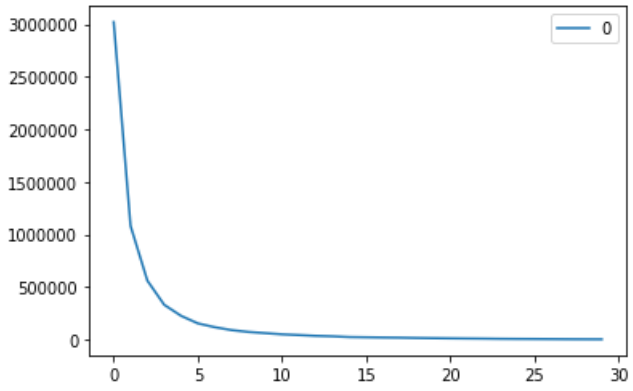
```
[3021412.281456337,
 1082273.3349676395,
 560690.5094988057,
 330874.88989972015,
 225864.79518516368,
 153801.14222590494,
 117701.95160097053,
 90027.02506275156,
 72471.37050124576,
 61212.63110711296,
 49811.318345247564,
 42827.63603060018,
 35324.19737825483,
 30841.221294785984,
 23774.48097732406,
 21609.707532905617,
 19461.753499364975,
 17910.915737616986,
 15153.905581603882,
 12971.064329382765,
 11279.900323258145,
 9955.664031567925,
 8713.02384911417,
 6979.952502423299,
 5966.990452152229,
 5316.046409518047,
 4446.022189028068,
 3513.871659366495,
 2862.9276167323114,
 2441.678952043884]
```

In [42]:

```
sns.lineplot(data=pd.DataFrame(list_of_inertia))
```

Out[42]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2984c0bd988>
```



In [45]:

```
kmeans_exp_1=KMeans(n_clusters=4,max_iter=300,random_state=15)
```

In [12]:

```
kmeans_exp_1.fit(df["WindGust"].values.reshape(-1,1))
```

Out[12]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=6, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=15, tol=0.0001, verbose=0)
```

In [13]:

```
silhouette_score(df["WindGust"].values.reshape(-1,1),kmeans_exp_1.labels_)
```

Out[13]:

```
0.5662882537635232
```

In [47]:

```
kmeans_exp_1.inertia_
```

Out[47]:

```
330874.88989972015
```

In [48]:

```
kmeans_exp_1.labels_
```

Out[48]:

```
array([2, 2, 2, ..., 2, 2, 2])
```

In [49]:

```
pd.DataFrame(kmeans_exp_1.labels_).T
```

Out[49]:

	0	1	2	3	4	5	6	7	8	9	...	48303	48304	48305	48306	48307	48308	48309	48310	48311	48312
0	2	2	2	2	2	2	2	2	2	2	...	2	2	2	2	2	2	2	2	2	2

1 rows × 48313 columns

In [50]:

```
kmeans_exp_1.cluster_centers_
```

Out[50]:

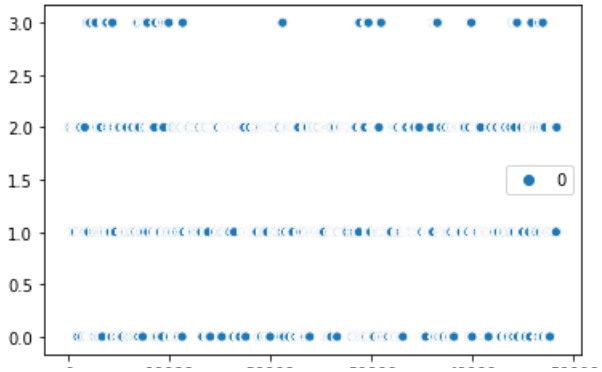
```
array([[22.28330206],
       [14.05703032],
       [ 6.95319452],
       [34.45831843]])
```

In [64]:

```
sns.scatterplot(data=pd.DataFrame(kmeans_exp_1.labels_))
```

Out[64]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2984b5f6288>
```



0 10000 20000 30000 40000 50000

In [65]:

```
a=pd.DataFrame(kmeans_exp_1.labels_)
a.columns=["Cluster"]
```

In [66]:

```
a.Cluster.value_counts()
```

Out[66]:

```
1    18534
2    16857
0    10127
3     2795
Name: Cluster, dtype: int64
```

Humidity

In [70]:

```
list_of_inertia=[]
for items in range(3,21):
    kmeans_clustering=KMeans(n_clusters=items,random_state=15,max_iter=300)
    kmeans_clustering.fit(df["Humidity"].values.reshape(-1,1))
    list_of_inertia.append(kmeans_clustering.inertia_)
```

In [71]:

```
list_of_inertia
```

Out[71]:

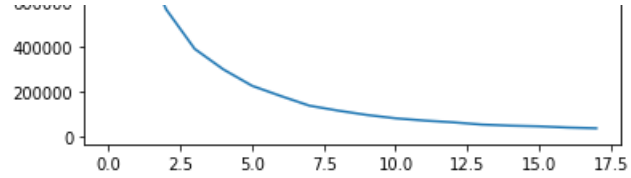
```
[1454518.7254992437,
 874136.2237488809,
 569560.506177118,
 392136.17812633904,
 300510.126817572,
 227300.26051588322,
 181731.33333743754,
 138458.7453470317,
 116460.8566828362,
 97198.92966412017,
 82160.05914513885,
 72118.51065125204,
 64312.99578001878,
 54230.39278320354,
 49294.816293515454,
 46298.492143974145,
 40838.34098810151,
 37526.728872356776]
```

In [72]:

```
sns.lineplot(data=pd.DataFrame(list_of_inertia))
```

Out[72]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2984bdb3148>
```



The plot is not readable, with different range ;

In [84]:

```
list_of_inertia=[]
for items in range(3,15):
    kmeans_clustering=KMeans(n_clusters=items,random_state=15,max_iter=300)
    kmeans_clustering.fit(df["Humidity"].values.reshape(-1,1))
    list_of_inertia.append(kmeans_clustering.inertia_)
```

In [85]:

```
list_of_inertia
```

Out[85]:

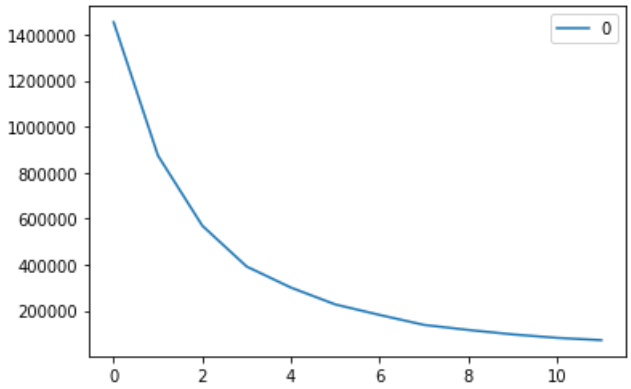
```
[1454518.7254992437,
 874136.2237488809,
 569560.506177118,
 392136.17812633904,
 300510.126817572,
 227300.26051588322,
 181731.33333743754,
 138458.7453470317,
 116460.8566828362,
 97198.92966412017,
 82160.05914513885,
 72118.51065125204]
```

In [86]:

```
sns.lineplot(data=pd.DataFrame(list_of_inertia))
```

Out[86]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2984a185988>
```



In [8]:

```
kmeans_exp_1=KMeans(n_clusters=6,max_iter=300,random_state=15)
```

In [14]:

```
kmeans_exp_1.fit(df["Humidity"].values.reshape(-1,1))
```

Out[14]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=6, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=15, tol=0.0001, verbose=0)
```

In [15]:

```
silhouette_score(df["Humidity"].values.reshape(-1,1),kmeans_exp_1.labels_)
```

Out[15]:

```
0.5591162167677304
```

In [10]:

```
kmeans_exp_1.inertia_
```

Out[10]:

```
392136.17812633904
```

In [11]:

```
kmeans_exp_1.labels_
```

Out[11]:

```
array([0, 0, 0, ..., 4, 4, 4])
```

In [12]:

```
pd.DataFrame(kmeans_exp_1.labels_).T
```

Out[12]:

	0	1	2	3	4	5	6	7	8	9	...	48303	48304	48305	48306	48307	48308	48309	48310	48311	48312
0	0	0	0	0	0	0	0	0	0	0	...	4	4	4	4	4	4	4	4	4	4

1 rows × 48313 columns

In [13]:

```
kmeans_exp_1.cluster_centers_
```

Out[13]:

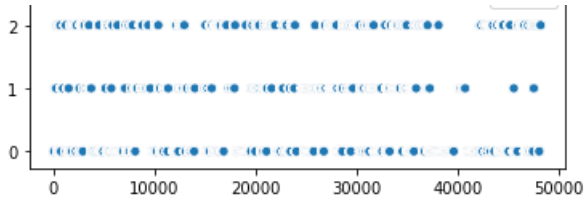
```
array([[89.11449502],
       [51.12732697],
       [80.59997668],
       [39.93187815],
       [61.50582608],
       [71.17458243]])
```

In [93]:

```
sns.scatterplot(data=pd.DataFrame(kmeans_exp_1.labels_),x_bins=500,y_bins=50)
```

Out[93]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2984aefe5c8>
```



In [14]:

```
a=pd.DataFrame(kmeans_exp_1.labels_)
a.columns=["Cluster"]
a.Cluster.value_counts()
```

Out[14]:

```
0    12149
2     8577
1     8380
4     8153
5     6885
3     4169
Name: Cluster, dtype: int64
```

"Visibility" feature is not suitable for clustering.

In [15]:

```
list_of_inertia=[]
for items in range(3,6):
    kmeans_clustering=KMeans(n_clusters=items,random_state=15,max_iter=300)
    kmeans_clustering.fit(df["Visibility"].values.reshape(-1,1))
    list_of_inertia.append(kmeans_clustering.inertia_)
```

In [16]:

```
list_of_inertia
```

Out[16]:

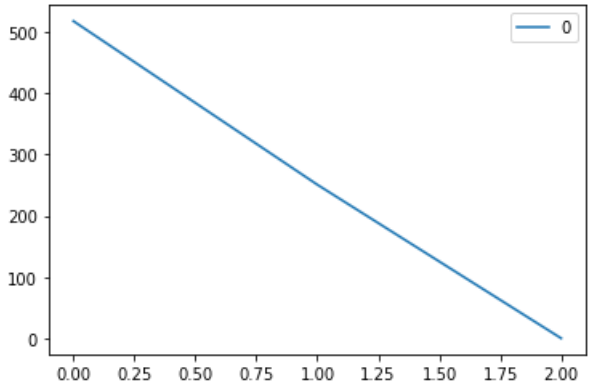
```
[517.6318652395157, 250.70503597122303, 6.348619609588698e-23]
```

In [17]:

```
sns.lineplot(data=pd.DataFrame(list_of_inertia))
```

Out[17]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1658641e648>
```



In [18]:

```
kmeans_exp_1=KMeans(n_clusters=2,max_iter=300,random_state=15)
```

In [19]:

```
kmeans_exp_1.fit(df["Visibility"].values.reshape(-1,1))
```

Out[19]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=2, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=15, tol=0.0001, verbose=0)
```

In [20]:

```
kmeans_exp_1.inertia_
```

Out[20]:

```
1784.0621563933598
```

In [108]:

```
kmeans_exp_1.labels_
```

Out[108]:

```
array([0, 0, 0, ..., 0, 0, 0])
```

In [109]:

```
pd.DataFrame(kmeans_exp_1.labels_).T
```

Out[109]:

	0	1	2	3	4	5	6	7	8	9	...	48303	48304	48305	48306	48307	48308	48309	48310	48311	48312
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

1 rows × 48313 columns

In [110]:

```
kmeans_exp_1.cluster_centers_
```

Out[110]:

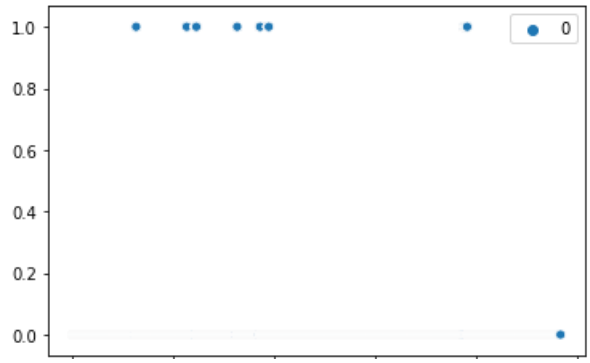
```
array([[9.97638043],
       [1.74100719]])
```

In [111]:

```
sns.scatterplot(data=pd.DataFrame(kmeans_exp_1.labels_),x_bins=500,y_bins=50)
```

Out[111]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x29847aff248>
```



In [34]:

```
a=pd.DataFrame(kmeans_exp_1.labels_)
a.columns=["Cluster"]
```

Density

In [113]:

```
list_of_inertia=[]
for items in range(3,21):
    kmeans_clustering=KMeans(n_clusters=items,random_state=15,max_iter=300)
    kmeans_clustering.fit(df["Density"].values.reshape(-1,1))
    list_of_inertia.append(kmeans_clustering.inertia_)
```

In [114]:

```
list_of_inertia
```

Out[114]:

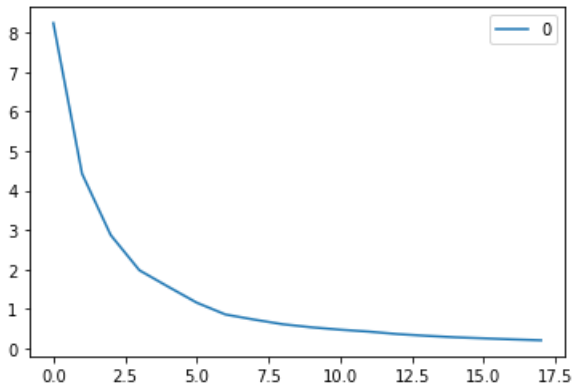
```
[8.246558114067083,
 4.427515061115515,
 2.869137558298023,
 1.9760965163653235,
 1.5605874423794044,
 1.1539972287782083,
 0.8543979490696777,
 0.7249435357148676,
 0.6068996832406915,
 0.5292445084547235,
 0.4689009956214695,
 0.4199026260148145,
 0.3576310834948045,
 0.3139064626167676,
 0.27621216384400277,
 0.2496742431580268,
 0.22287729255557584,
 0.1998165145676149]
```

In [115]:

```
sns.lineplot(data=pd.DataFrame(list_of_inertia))
```

Out[115]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x29848fc0208>
```



In [116]:

```
kmeans_exp_1=KMeans(n_clusters=7,max_iter=300,random_state=15)
```

In [16]:

```
kmeans_exp_1.fit(df["Density"].values.reshape(-1,1))
```

Out[16]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=6, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=15, tol=0.0001, verbose=0)
```

In [17]:

```
silhouette_score(df["Density"].values.reshape(-1,1),kmeans_exp_1.labels_)
```

Out[17]:

```
0.5656484186287785
```

In [118]:

```
kmeans_exp_1.inertia_
```

Out[118]:

```
1.5605874423794044
```

In [119]:

```
kmeans_exp_1.labels_
```

Out[119]:

```
array([5, 5, 5, ..., 1, 1, 1])
```

In [120]:

```
pd.DataFrame(kmeans_exp_1.labels_).T
```

Out[120]:

	0	1	2	3	4	5	6	7	8	9	...	48303	48304	48305	48306	48307	48308	48309	48310	48311	48312
0	5	5	5	5	5	5	5	5	5	5	...	1	1	1	1	1	1	1	1	1	1

1 rows × 48313 columns

In [121]:

```
kmeans_exp_1.cluster_centers_
```

Out[121]:

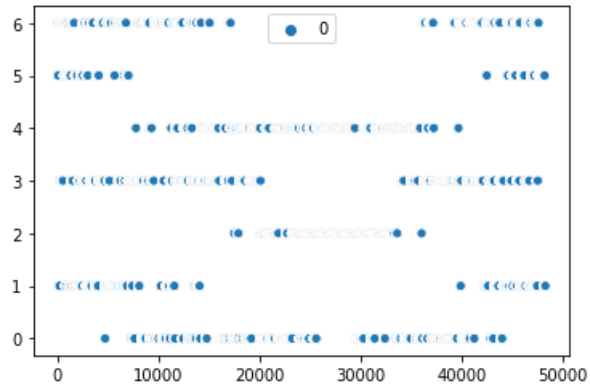
```
array([[1.21284286],
       [1.26773644],
       [1.16788297],
       [1.23059769],
       [1.19498369],
       [1.28614076],
       [1.24965204]])
```

In [122]:

```
sns.scatterplot(data=pd.DataFrame(kmeans_exp_1.labels_),x_bins=500,y_bins=50)
```

Out[122]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2984904b508>
```



In [21]:

```
a=pd.DataFrame(kmeans_exp_1.labels_)
a.columns=["Cluster"]
a.Cluster.value_counts()
```

Out[21]:

```
0    47757
1      556
Name: Cluster, dtype: int64
```

In [ ]:

```

```

Density with different random state

In [124]:

```
list_of_inertia=[]
for items in range(3,21):
    kmeans_clustering=KMeans(n_clusters=items,random_state=10,max_iter=300)
    kmeans_clustering.fit(df["Density"].values.reshape(-1,1))
    list_of_inertia.append(kmeans_clustering.inertia_)
```

In [125]:

```
list_of_inertia
```

Out[125]:

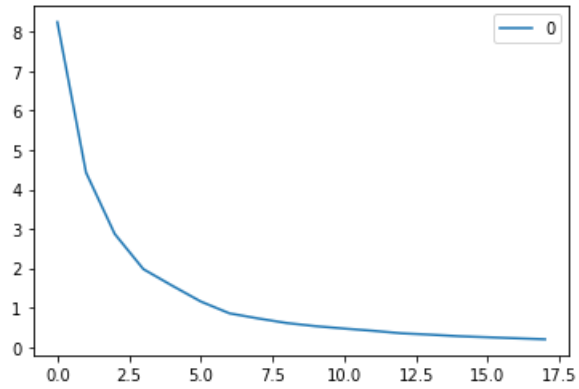
```
[8.246558114067083,
 4.427679012974612,
 2.870086287502298,
 1.9760965163653235,
 1.5583140932428412,
 1.1551164576306077,
 0.8544922796011799,
 0.7249435357148676,
 0.6085390971154986,
 0.5293742119983692,
 0.46816860711018193,
 0.4116567359221708,
 0.3479207230858452,
 0.3137822443787309,
 0.2736731747898067,
 0.24753742055827682,
 0.22136008230950516,
 0.1957749662622888]
```

In [126]:

```
sns.lineplot(data=pd.DataFrame(list_of_inertia))
```

Out[126]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2984a636288>
```



In [127]:

```
kmeans_exp_1=KMeans(n_clusters=7,max_iter=300,random_state=10)
```

In [128]:

```
kmeans_exp_1.fit(df["Density"].values.reshape(-1,1))
```

Out[128]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=7, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=10, tol=0.0001, verbose=0)
```

In [129]:

```
kmeans_exp_1.inertia_
```

Out[129]:

```
1.5583140932428412
```

In [130]:

```
kmeans_exp_1.labels_
```

Out[130]:

```
array([6, 6, 6, ..., 2, 2, 2])
```

In [131]:

```
pd.DataFrame(kmeans_exp_1.labels_).T
```

Out[131]:

	0	1	2	3	4	5	6	7	8	9	...	48303	48304	48305	48306	48307	48308	48309	48310	48311	48312
0	6	6	6	6	6	6	6	6	6	6	...	2	2	2	2	2	2	2	2	2	2

1 rows × 48313 columns

In [132]:

```
kmeans_exp_1.cluster_centers_
```

Out[132]:

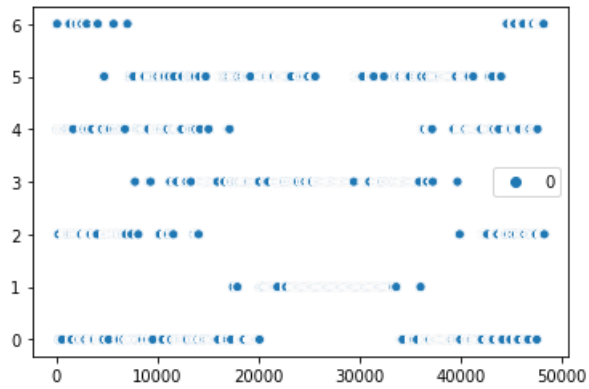
```
array([[1.23059769],
       [1.16788297],
       [1.26872308],
       [1.19498369],
       [1.24965204],
       [1.21284286],
       [1.2886508 ]])
```

In [133]:

```
sns.scatterplot(data=pd.DataFrame(kmeans_exp_1.labels_),x_bins=500,y_bins=50)
```

Out[133]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2984b018cc8>
```



In [134]:

```
a=pd.DataFrame(kmeans_exp_1.labels_)
a.columns=["Cluster"]
```

Density with different k value

In [139]:

```
list_of_inertia=[]
for items in range(3,15):
    kmeans_clustering=KMeans(n_clusters=items,random_state=10,max_iter=300)
    kmeans_clustering.fit(df["Density"].values.reshape(-1,1))
    list_of_inertia.append(kmeans_clustering.inertia_)
```

In [140]:

```
list_of_inertia
```

Out[140]:

```
[8.246558114067083,
 4.427679012974612,
 2.870086287502298,
 1.9760965163653235,
 1.5583140932428412,
 1.1551164576306077,
 0.8544922796011799,
 0.7249435357148676,
 0.6085390971154986,
 0.5293742119983692,
 0.46816860711018193,
 0.4116567359221708]
```

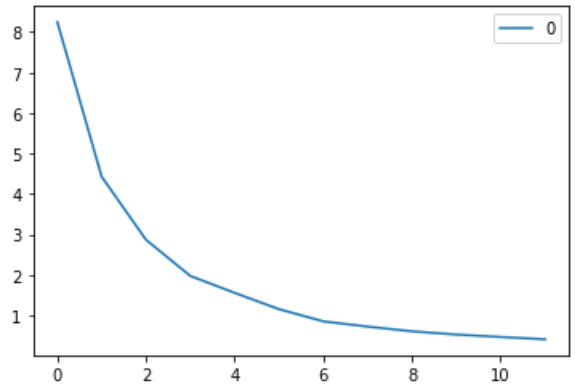
In [141]:

```
sns.lineplot(data=pd.DataFrame(list_of_inertia))
```

Out[141]:

Out[141]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x29849d77848>
```



In [142]:

```
kmeans_exp_1=KMeans(n_clusters=5,max_iter=300,random_state=10)
```

In [143]:

```
kmeans_exp_1.fit(df["Density"].values.reshape(-1,1))
```

Out[143]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=5, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=10, tol=0.0001, verbose=0)
```

In [144]:

```
kmeans_exp_1.inertia_
```

Out[144]:

```
2.870086287502298
```

In [145]:

```
kmeans_exp_1.labels_
```

Out[145]:

```
array([2, 2, 2, ..., 2, 2, 2])
```

In [146]:

```
pd.DataFrame(kmeans_exp_1.labels_).T
```

Out[146]:

	0	1	2	3	4	5	6	7	8	9	...	48303	48304	48305	48306	48307	48308	48309	48310	48311	48312
0	2	2	2	2	2	2	2	2	2	2	...	2	2	2	2	2	2	2	2	2	2

1 rows × 48313 columns

In [147]:

```
kmeans_exp_1.cluster_centers_
```

Out[147]:

```
array([[1.22208931],
       [1.17044742],
```

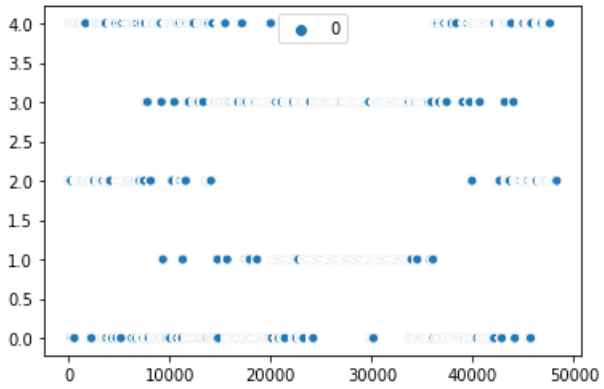
```
       [1.2739256 ],
       [1.19971744],
       [1.24602981]])
```

In [148]:

```
sns.scatterplot(data=pd.DataFrame(kmeans_exp_1.labels_),x_bins=500,y_bins=50)
```

Out[148]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x298489adc08>
```



In [134]:

```
a=pd.DataFrame(kmeans_exp_1.labels_)
a.columns=["Cluster"]
```

kmeans_exp_2=KMeans(n_clusters=6,random_state=15)

In [20]:

```
kmeans_exp_2.fit(df["Temp"].values.reshape(-1,1))
```

Out[20]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=6, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=15, tol=0.0001, verbose=0)
```

In [22]:

```
silhouette_score(df["Temp"].values.reshape(-1,1),kmeans_exp_2.labels_)
```

Out[22]:

```
0.5760048070855869
```

In [23]:

```
kmeans_exp_3=KMeans(n_clusters=5,random_state=15)
```

In [26]:

```
kmeans_exp_3.fit(df["Sun Hour"].values.reshape(-1,1))
```

Out[26]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=5, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=15, tol=0.0001, verbose=0)
```

In [27]:

In [27]:

```
silhouette_score(df["Sun Hour"].values.reshape(-1,1),kmeans_exp_3.labels_)
```

Out[27]:

```
0.7675712626413547
```

In [28]:

```
kmeans_exp_4=KMeans(n_clusters=8,random_state=15)
```

In [31]:

```
kmeans_exp_4.fit(df["DewPoint"].values.reshape(-1,1))
```

Out[31]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=8, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=15, tol=0.0001, verbose=0)
```

In [32]:

```
silhouette_score(df["DewPoint"].values.reshape(-1,1),kmeans_exp_4.labels_)
```

Out[32]:

```
0.6045743854538681
```

In [33]:

```
kmeans_exp_5=KMeans(n_clusters=8,random_state=15)
```

In [35]:

```
kmeans_exp_5.fit(df["WindChillC"].values.reshape(-1,1))
```

Out[35]:

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=8, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=15, tol=0.0001, verbose=0)
```

In [36]:

```
silhouette_score(df["WindChillC"].values.reshape(-1,1),kmeans_exp_5.labels_)
```

Out[36]:

```
0.5996954694443724
```

In [ ]:

```

```