In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import joblib
```

In [2]:

```
path = "T1.csv"
df = pd.read_csv(path)
df.rename(columns={"Date/Time":"Time",
                   "LV ActivePower (kW)":"ActivePower(KW)",
                   "Wind Speed (m/s)": "WindSpeed(m/s)",
                   "Wind Direction(°)":"Wind_Direction"},
                   inplace=True)
```

In [4]:

```
df
```

Out[4]:

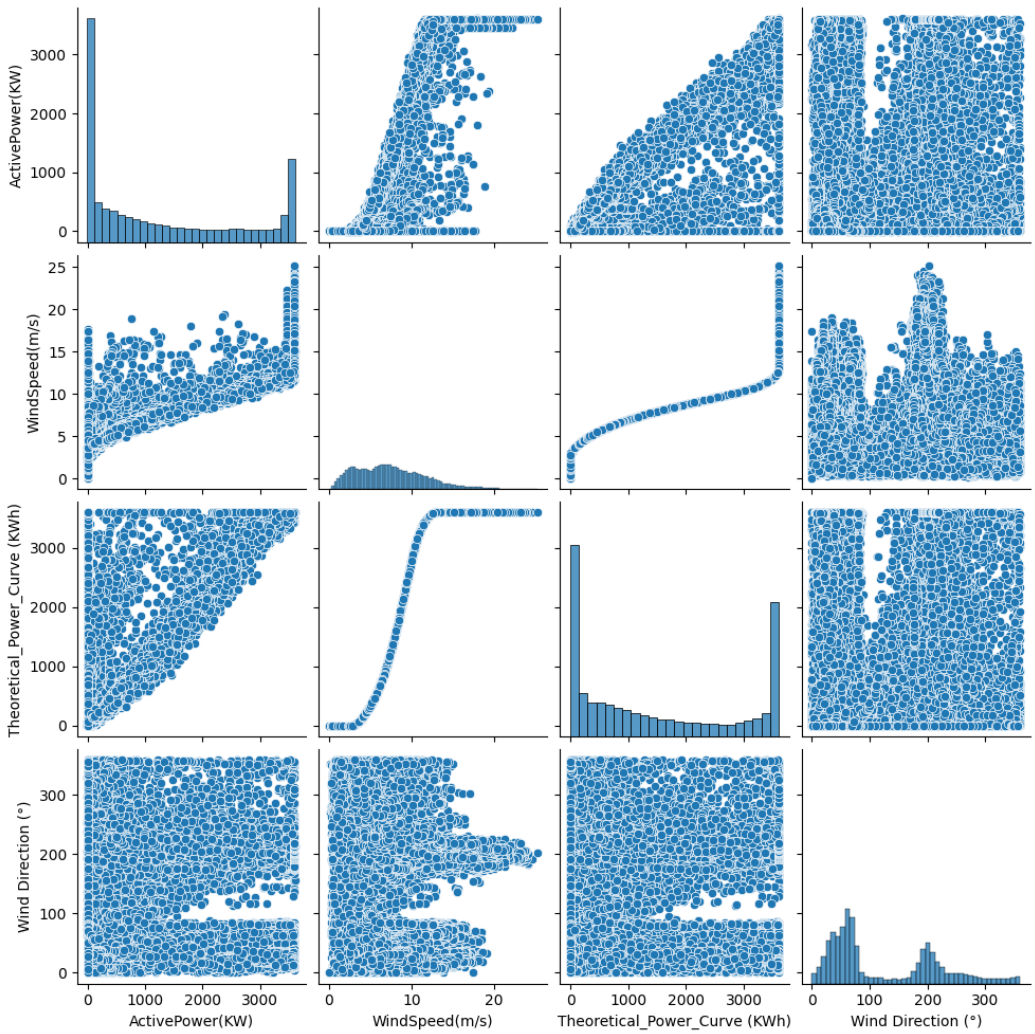
	Time	ActivePower(KW)	WindSpeed(m/s)	Theoretical_Power_Curve (KWh)	Wind Direction (°)
0	01 01 2018 00:00	380.047791	5.311336	416.328908	259.994904
1	01 01 2018 00:10	453.769196	5.672167	519.917511	268.641113
2	01 01 2018 00:20	306.376587	5.216037	390.900016	272.564789
3	01 01 2018 00:30	419.645905	5.659674	516.127569	271.258087
4	01 01 2018 00:40	380.650696	5.577941	491.702972	265.674286
...	...	...	...	...	...
50525	31 12 2018 23:10	2963.980957	11.404030	3397.190793	80.502724
50526	31 12 2018 23:20	1684.353027	7.332648	1173.055771	84.062599
50527	31 12 2018 23:30	2201.106934	8.435358	1788.284755	84.742500

50528	31 12 2018 23:40	2515.694092	9.421366	2418.382503	84.297913
50529	31 12 2018 23:50	2820.466064	9.979332	2779.184096	82.274620

50530 rows × 5 columns

In [5]:
```
sns.pairplot(df)
```

Out[5]:



In [6]:
```
plt.figure(figsize=(10, 8))
corr = df.corr()
ax = sns.heatmap(corr, vmin = -1,vmax = 1,annot = True)
bottom, top = ax.get_ylim()
ax.set_ylim(bottom + 0.5, top - 0.5)

print(corr)
```

```
                    ActivePower(KW)  WindSpeed(m/s)  \
ActivePower(KW)            1.000000        0.912774
WindSpeed(m/s)             0.912774        1.000000
```

```
Theoretical_Power_Curve (KWh)         0.949918        0.944209
Wind Direction (°)                   -0.062702       -0.077188

                                  Theoretical_Power_Curve (KWh)  \
ActivePower(KW)                                        0.949918
WindSpeed(m/s)                                         0.944209
Theoretical_Power_Curve (KWh)                          1.000000
Wind Direction (°)                                    -0.099076

                                  Wind Direction (°)
ActivePower(KW)                           -0.062702
WindSpeed(m/s)                            -0.077188
Theoretical_Power_Curve (KWh)             -0.099076
Wind Direction (°)                         1.000000
```

In [7]:
```
df["Time"] = pd.to_datetime(df["Time"], format = "%d %m %Y %H %M", errors =
```

In [8]:
```
y = df["ActivePower(KW)"]
X = df[["Theoretical_Power_Curve (KWh)", "WindSpeed(m/s)"]]

from sklearn.model_selection import train_test_split
train_X, val_X, train_y, val_y = train_test_split(X, y, random_state=0)
```

Model building

In [9]:
```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error,r2_score
from xgboost import XGBRegressor

forest_model = RandomForestRegressor(n_estimators = 750, max_depth = 4, max
```

```
forest_model = RandomForestRegressor(n_estimators = 750, max_depth = 4, max_

forest_model.fit(train_X, train_y)
```

Out[9]:
```
RandomForestRegressor(max_depth=4, max_leaf_nodes=500, n_estimators=7
50,
                      random_state=1)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [10]:
```
RandomForestRegressor(max_depth=4, max_leaf_nodes=500, n_estimators=750,
                      random_state=1)
```

Out[10]:
```
RandomForestRegressor(max_depth=4, max_leaf_nodes=500, n_estimators=7
50,
                      random_state=1)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [13]:
```
power_preds = forest_model.predict(val_X)

print(mean_absolute_error(val_y, power_preds))
print(r2_score(val_y, power_preds))
```

```
164.58015525861344
0.9113496428907649
```

In [14]:
```
joblib.dump(forest_model, "power_prediction.sav")
```

Out[14]:
```
['power_prediction.sav']
```

In [15]:
```
df
```

Out[15]:

	Time	ActivePower(KW)	WindSpeed(m/s)	Theoretical_Power_Curve (KWh)	Wind Direction (°)
0	NaT	380.047791	5.311336	416.328908	259.994904
1	NaT	453.769196	5.672167	519.917511	268.641113
2	NaT	306.376587	5.216037	390.900016	272.564789
3	NaT	419.645905	5.659674	516.127569	271.258087
4	NaT	380.650696	5.577941	491.702972	265.674286
...	...	...	...	...	...
50525	NaT	2963.980957	11.404030	3397.190793	80.502724
50526	NaT	1684.353027	7.332648	1173.055771	84.062599
50527	NaT	2201.106934	8.435358	1788.284755	84.742500

50528	NaT	2515.694092	9.421366	2418.382503	84.297913
50529	NaT	2820.466064	9.979332	2779.184096	82.274620

50530 rows × 5 columns

In [16]:
```
import pickle
pickle.dump(forest_model,open("model.pkl","wb"))
```

In [ ]: