In [1]:

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

In [2]:

```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It in
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='0j9ybHNI1hQ6b2L9ZiJlLBRjwWf3tbtFoNAoTtZLrz9z',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud

bucket = 'predictingenergyoutputofwindturbi-donotdelete-pr-vvuc1mxciqx0o2'
object_key = 'T1.csv'

body = cos_client.get_object(Bucket=bucket,Key=object_key)['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__

df = pd.read_csv(body)
df.head()
```

Out[2]:

	Date/Time	LV ActivePower (kW)	Wind Speed (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

In [3]:

```
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
```

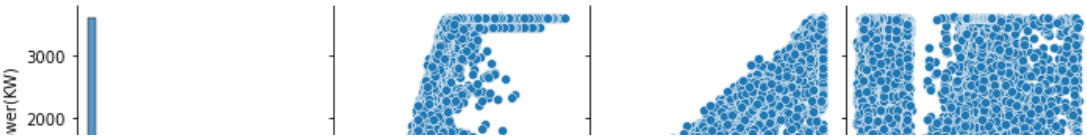
Out[4]:

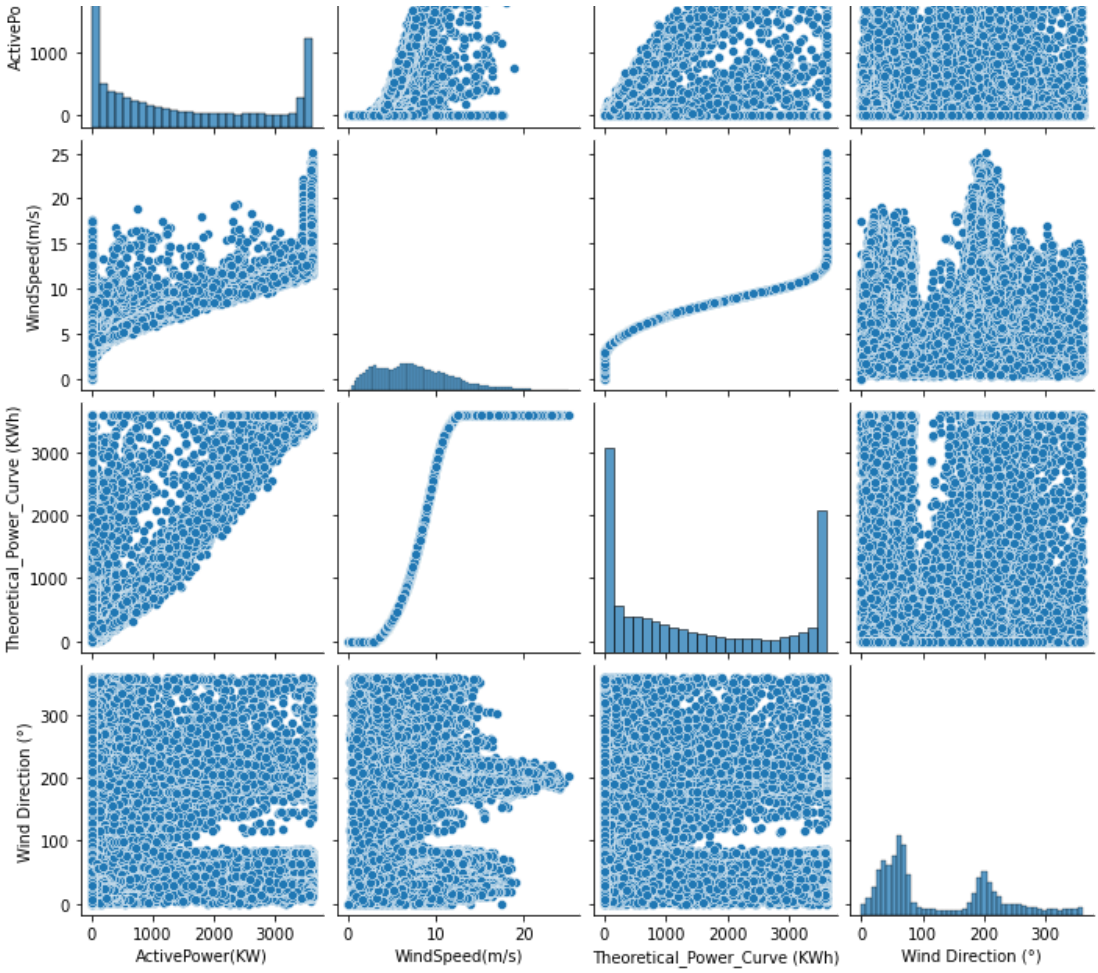
	Time	ActivePower(KW)	WindSpeed(m/s)	Theoretical_Power_Curve (KWh)	Wind Direction (°)
0	01 01 2018 00:00	380.047791	5.311336	416.328908	259.994904
1	01 01 2018 00:10	453.769196	5.672167	519.917511	268.641113
2	01 01 2018 00:20	306.376587	5.216037	390.900016	272.564789
3	01 01 2018 00:30	419.645905	5.659674	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]:

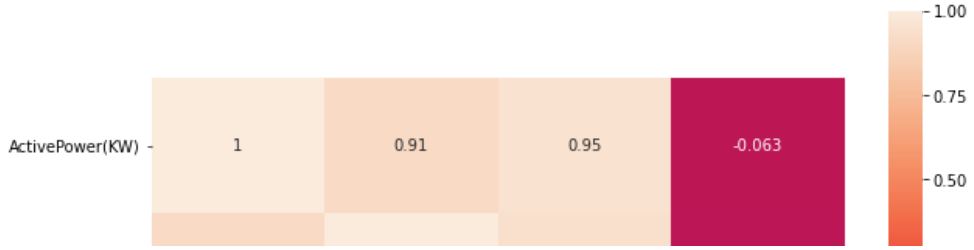


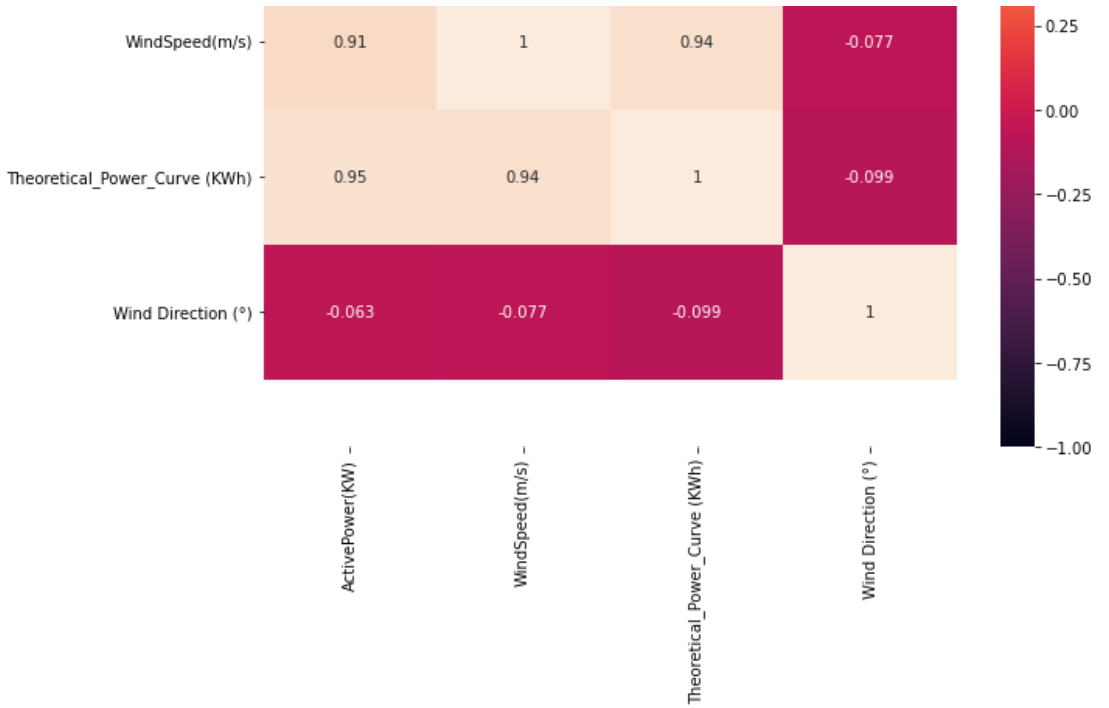


In [6]:

```
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.fit(train_X, train_y)
```

Out[9]:
```
RandomForestRegressor(max_depth=4, max_leaf_nodes=500, n_estimators=750,
                      random_state=1)
```

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=750,
                      random_state=1)
```

In [11]:
```
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 [12]:
```
joblib.dump(forest_model, "power_prediction.sav")
```

Out[12]:
```
['power_prediction.sav']
```

In [13]:
```
df
```

Out[13]:

	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 [14]:
```
import pickle
pickle.dump(forest_model,open("model.pkl","wb"))
```

In [15]:
```
!pip install -U ibm-watson-machine-learning
```

```
Requirement already satisfied: ibm-watson-machine-learning in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.0.257)
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (0.8.9)
Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (0.3.3)
Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (1.26.7)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (2.26.0)
Requirement already satisfied: packaging in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (21.3)
Requirement already satisfied: pandas<1.5.0,>=0.24.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (1.3.4)
Requirement already satisfied: ibm-cos-sdk==2.11.* in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (2.11.0)
Requirement already satisfied: importlib-metadata in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (4.8.2)
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (2022.9.24)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Pyt
```

```
hon-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm-watson-ma
chine-learning) (0.10.0)
Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/P
ython-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm-watson-
machine-learning) (2.11.0)
Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda/
envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm-w
atson-machine-learning) (2.11.0)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/env
s/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk-core==2.11.0->ibm
-cos-sdk==2.11.*->ibm-watson-machine-learning) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/li
b/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm-watson-machine-le
arning) (2021.3)
Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/l
ib/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm-watson-machine-l
earning) (1.20.3)
Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/py
thon3.9/site-packages (from python-dateutil<3.0.0,>=2.1->ibm-cos-sdk-core==
2.11.0->ibm-cos-sdk==2.11.*->ibm-watson-machine-learning) (1.15.0)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/li
b/python3.9/site-packages (from requests->ibm-watson-machine-learning) (3.3)
Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/
Python-3.9/lib/python3.9/site-packages (from requests->ibm-watson-machine-le
arning) (2.0.4)
Requirement already satisfied: zipp>=0.5 in /opt/conda/envs/Python-3.9/lib/p
ython3.9/site-packages (from importlib-metadata->ibm-watson-machine-learnin
g) (3.6.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /opt/conda/envs/P
ython-3.9/lib/python3.9/site-packages (from packaging->ibm-watson-machine-le
arning) (3.0.4)
```

In [16]:
```
from ibm_watson_machine_learning import APIClient
import json
import numpy as np
```

In [17]:
```
wml_credentials = {
    "apikey":"Ilik-kKvZ4Lwtruh-l7Bl2FS5IEFz0Ujhq8533qqw09Z",
    "url":"https://us-south.ml.cloud.ibm.com"
}
```

In [18]:
```
wml_client = APIClient(wml_credentials)
```

In [19]:
```
wml_client.spaces.list()
```

```
Note: 'limit' is not provided. Only first 50 records will be displayed if th
e number of records exceed 50
------------------------------------  ------------  ------------------------
ID                                    NAME          CREATED
f1f8ff94-56cd-4e00-a0c9-f8ae5161c050  wind turbine  2022-11-15T04:21:43.273Z
------------------------------------  ------------  ------------------------
```

In [20]:
```
SPACE_ID = "f1f8ff94-56cd-4e00-a0c9-f8ae5161c050"
```

In [21]:
```
wml_client.set.default_space(SPACE_ID)
```

Out[21]:
```
'SUCCESS'
```

In [22]:
```
wml_client.software_specifications.list()
```

```
------------------------------  ------------------------------------  ----
NAME                            ASSET_ID                              TYPE
default_py3.6                   0062b8c9-8b7d-44a0-a9b9-46c416adcbd9  base
kernel-spark3.2-scala2.12       020d69ce-7ac1-5e68-ac1a-31189867356a  base
pytorch-onnx_1.3-py3.7-edt      069ea134-3346-5748-b513-49120e15d288  base
scikit-learn_0.20-py3.6         09c5a1d0-9c1e-4473-a344-eb7b665ff687  base
spark-mllib_3.0-scala_2.12      09f4cff0-90a7-5899-b9ed-1ef348aebdee  base
pytorch-onnx_rt22.1-py3.9       0b848dd4-e681-5599-be41-b5f6fccc6471  base
ai-function_0.1-py3.6           0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda  base
shiny-r3.6                      0e6e79df-875e-4f24-8ae9-62dcc2148306  base
tensorflow_2.4-py3.7-horovod    1092590a-307d-563d-9b62-4eb7d64b3f22  base
pytorch_1.1-py3.6               10ac12d6-6b30-4ccd-8392-3e922c096a92  base
tensorflow_1.15-py3.6-ddl       111e41b3-de2d-5422-a4d6-bf776828c4b7  base
autoai-kb_rt22.2-py3.10         125b6d9a-5b1f-5e8d-972a-b251688ccf40  base
runtime-22.1-py3.9              12b83a17-24d8-5082-900f-0ab31fbfd3cb  base
scikit-learn_0.22-py3.6         154010fa-5b3b-4ac1-82af-4d5ee5abbc85  base
default_r3.6                    1b70aec3-ab34-4b87-8aa0-a4a3c8296a36  base
pytorch-onnx_1.3-py3.6          1bc6029a-cc97-56da-b8e0-39c3880dbbe7  base
kernel-spark3.3-r3.6            1c9e5454-f216-59dd-a20e-474a5cdf5988  base
pytorch-onnx_rt22.1-py3.9-edt   1d362186-7ad5-5b59-8b6c-9d0880bde37f  base
tensorflow_2.1-py3.6            1eb25b84-d6ed-5dde-b6a5-3fbdf1665666  base
spark-mllib_3.2                 20047f72-0a98-58c7-9ff5-a77b012eb8f5  base
tensorflow_2.4-py3.8-horovod    217c16f6-178f-56bf-824a-b19f20564c49  base
runtime-22.1-py3.9-cuda         26215f05-08c3-5a41-a1b0-da66306ce658  base
do_py3.8                        295addb5-9ef9-547e-9bf4-92ae3563e720  base
autoai-ts_3.8-py3.8             2aa0c932-798f-5ae9-abd6-15e0c2402fb5  base
tensorflow_1.15-py3.6           2b73a275-7cbf-420b-a912-eae7f436e0bc  base
kernel-spark3.3-py3.9           2b7961e2-e3b1-5a8c-a491-482c8368839a  base
pytorch_1.2-py3.6               2c8ef57d-2687-4b7d-acce-01f94976dac1  base
spark-mllib_2.3                 2e51f700-bca0-4b0d-88dc-5c6791338875  base
pytorch-onnx_1.1-py3.6-edt      32983cea-3f32-4400-8965-dde874a8d67e  base
spark-mllib_3.0-py37            36507ebe-8770-55ba-ab2a-eafe787600e9  base
spark-mllib_2.4                 390d21f8-e58b-4fac-9c55-d7ceda621326  base
autoai-ts_rt22.2-py3.10         396b2e83-0953-5b86-9a55-7ce1628a406f  base
xgboost_0.82-py3.6              39e31acd-5f30-41dc-ae44-60233c80306e  base
pytorch-onnx_1.2-py3.6-edt      40589d0e-7019-4e28-8daa-fb03b6f4fe12  base
pytorch-onnx_rt22.2-py3.10      40e73f55-783a-5535-b3fa-0c8b94291431  base
default_r36py38                 41c247d3-45f8-5a71-b065-8580229facf0  base
autoai-ts_rt22.1-py3.9          4269d26e-07ba-5d40-8f66-2d495b0c71f7  base
autoai-obm_3.0                  42b92e18-d9ab-567f-988a-4240ba1ed5f7  base
pmml-3.0_4.3                    493bcb95-16f1-5bc5-bee8-81b8af80e9c7  base
spark-mllib_2.4-r_3.6           49403dff-92e9-4c87-a3d7-a42d0021c095  base
xgboost_0.90-py3.6              4ff8d6c2-1343-4c18-85e1-689c965304d3  base
pytorch-onnx_1.1-py3.6          50f95b2a-bc16-43bb-bc94-b0bed208c60b  base
autoai-ts_3.9-py3.8             52c57136-80fa-572e-8728-a5e7cbb42cde  base
spark-mllib_2.4-scala_2.11      55a70f99-7320-4be5-9fb9-9edb5a443af5  base
spark-mllib_3.0                 5c1b0ca2-4977-5c2e-9439-ffd44ea8ffe9  base
autoai-obm_2.0                  5c2e37fa-80b8-5e77-840f-d912469614ee  base
spss-modeler_18.1               5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b  base
cuda-py3.8                      5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e  base
autoai-kb_3.1-py3.7             632d4b22-10aa-5180-88f0-f52dfb6444d7  base
pytorch-onnx_1.7-py3.8          634d3cdc-b562-5bf9-a2d4-ea90a478456b  base
------------------------------  ------------------------------------  ----
Note: Only first 50 records were displayed. To display more use 'limit' para
meter.
```

save and Deploy the model

In [23]:

In [23]:
```
import sklearn
sklearn.__version__
```

Out[23]:
```
'1.0.2'
```

In [24]:
```
MODEL_NAME = 'DemoModel_MLR'
DEPLOYMENT_NAME = 'Wind turbine'
DEMO_MODEL = forest_model
```

In [25]:
```
software_spec_uid = wml_client.software_specifications.get_id_by_name('runti
```

In [26]:
```
# setup model meta
model_props ={
    wml_client.repository.ModelMetaNames.NAME: MODEL_NAME,
    wml_client.repository.ModelMetaNames.TYPE:'scikit-learn_1.0',
    wml_client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_spec_ui

}
```

In [27]:
```
#save model
model_details = wml_client.repository.store_model(
    model=DEMO_MODEL,
    meta_props=model_props,
    training_data= train_X,
    training_target= train_y
)
```

In [28]:
```
model_details
```

Out[28]:

```
{'entity': {'hybrid_pipeline_software_specs': [],
  'label_column': 'ActivePower(KW)',
  'schemas': {'input': [{'fields': [{'name': 'Theoretical_Power_Curve (KW
h)',
       'type': 'float64'},
      {'name': 'WindSpeed(m/s)', 'type': 'float64'}],
     'id': '1',
     'type': 'struct'}],
   'output': []},
  'software_spec': {'id': '12b83a17-24d8-5082-900f-0ab31fbfd3cb',
   'name': 'runtime-22.1-py3.9'},
  'type': 'scikit-learn_1.0'},
 'metadata': {'created_at': '2022-11-16T16:01:18.347Z',
  'id': '59ecfa2f-181a-47a9-930d-648f237de348',
  'modified_at': '2022-11-16T16:01:22.204Z',
  'name': 'DemoModel_MLR',
  'owner': 'IBMid-668000E1AW',
  'resource_key': '4c7ea175-3c08-4b77-92d3-b966c6ac9624',
  'space_id': 'f1f8ff94-56cd-4e00-a0c9-f8ae5161c050'},
 'system': {'warnings': []}}
```


In [29]:
```
model_id = wml_client.repository.get_model_id(model_details)
```

In [30]:
```
model_id
```

Out[30]:
```
'59ecfa2f-181a-47a9-930d-648f237de348'
```

In [31]:
```
#deploy in purpose
deployment_props = {
     wml_client.deployments.ConfigurationMetaNames.NAME:DEPLOYMENT_NAME,
     wml_client.deployments.ConfigurationMetaNames.ONLINE:{}
}
```

In [32]:
```
deployment = wml_client.deployments.create(
    artifact_uid=model_id,
    meta_props=deployment_props
)
```

```
#######################################################################################

Synchronous deployment creation for uid: '59ecfa2f-181a-47a9-930d-648f237de348' started

#######################################################################################


initializing
Note: online_url is deprecated and will be removed in a future release. Use serving_urls instead.

ready


------------------------------------------------------------------------------------------
Successfully finished deployment creation, deployment_uid='8e0aa5c6-3b8f-46d6-a314-bf5b6a73d66d'
------------------------------------------------------------------------------------------
```

In [ ]:

In [ ]:

In [ ]: