Sprint 3

Train the model on IBM :

Team ID : PNT2022TMID03363

Project Name : Predicting the energy output of wind turbine based on weather condition

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import Lasso
from sklearn.linear_model import Ridge
from sklearn.metrics import mean_squared_error , r2_score
```

```
import joblib
%matplotlib inline
```

```
data = pd.read_csv('wind_dataset.csv')
data.rename(columns = {'LV ActivePower (kW)':'ActivePower(kW)',
                       "Wind Speed (m/s)":"WindSpeed(m/s)",
                        "Wind Direction (°)":"WindDirection","Theoretical_Power_Curve (KWh)":"TheoreticalPowerCurve(KWh)"},
            inplace = True)
data.head()
```

	Date/Time	ActivePower(kW)	WindSpeed(m/s)	TheoreticalPowerCurve(KWh)	WindDirection
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

```
data.shape
```

```
(50530, 5)
```

```
data.describe()
```

	ActivePower(kW)	WindSpeed(m/s)	TheoreticalPowerCurve(KWh)	WindDirection
count	50530.000000	50530.000000	50530.000000	50530.000000
mean	1307.684332	7.557952	1492.175463	123.687559
std	1312.459242	4.227166	1368.018238	93.443736
min	-2.471405	0.000000	0.000000	0.000000
25%	50.677890	4.201395	161.328167	49.315437
50%	825.838074	7.104594	1063.776283	73.712978
75%	2482.507568	10.300020	2964.972462	201.696720
max	3618.732910	25.206011	3600.000000	359.997589

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50530 entries, 0 to 50529
Data columns (total 5 columns):
 # Column Non-Null Count Dtype

 0 Date/Time 50530 non-null object
 1 ActivePower(kW) 50530 non-null float64
```

```
 2 WindSpeed(m/s) 50530 non-null float64
 3 TheoreticalPowerCurve(KWh) 50530 non-null float64
 4 WindDirection 50530 non-null float64
dtypes: float64(4), object(1)
memory usage: 1.9+ MB
```

```
data.isnull().any()
```

```
Date/Time False
ActivePower(kW) False
WindSpeed(m/s) False
TheoreticalPowerCurve(KWh) False
WindDirection False
dtype: bool
```

Data Preprocessing

```
data['Date/Time'] = pd.to_datetime(data['Date/Time'],format='%d %m %Y %H:%M')
data['year'] = data['Date/Time'].dt.year
data['month'] = data['Date/Time'].dt.month
data['day'] = data['Date/Time'].dt.day
data['Hour'] = data['Date/Time'].dt.hour
data['minute'] = data['Date/Time'].dt.minute
data.head()
```

	Date/Time	ActivePower(kW)	WindSpeed(m/s)	TheoreticalPowerCurve(KWh)	WindDirection	year	month
0	2018-01-01 00:00:00	453.769196	5.672167	519.917511	268.641113	2018	1
1	2018-01-01 00:10:00	380.047791	5.311336	416.328908	259.994904	2018	1
2	01 306 376587 5 216037 390 900016 272 564789 2018 1						

```
data["Date/Time"] = pd.to_datetime(data["Date/Time"], format = "%d %m %Y %H:%M", errors = "coerce")
data
```

	Date/Time	ActivePower(kW)	WindSpeed(m/s)	TheoreticalPowerCurve(KWh)	WindDirection	year	mon
0	2018-01-01 00:00:00	453.769196	5.672167	519.917511	268.641113	2018	
1	2018-01-01 00:10:00	306.376587	5.216037	390.900016	272.564789	2018	
2	2018-01-01 00:20:00	419.645905	5.659674	516.127569	271.258087	2018	
3	2018-01-01 00:30:00	380.650696	5.577941	491.702972	265.674286	2018	
4	2018-01-01 00:40:00 380.047791 5.311336 416.328908 259.994904 2018						
...	 2018-12-						

Splitting the dataset

```
X=data[['WindSpeed(m/s)','WindDirection']]
X.head()
```

	WindSpeed(m/s)	WindDirection
0	5.311336	259.994904
1	5.672167	268.641113

```
y = data['ActivePower(kW)']
y.head()
```

```
 2 5.216037 272.564789
0 380.047791
1 453.769196
 3 5.659674 271.258087
2 306.376587
 4 5.577941 265.674286
3 419.645905
4 380.650696
Name: ActivePower(kW), dtype: float64
```

```
X_train, X_test,y_train, y_test = train_test_split(X,y ,
                                   random_state=6,
                                   test_size=0.25)
```

Importing the regression Models

```
from sklearn.tree import DecisionTreeRegressor
from sklearn.svm import SVR
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
from sklearn.metrics import accuracy_score,r2_score,mean_squared_error
xgr=XGBRegressor()
rf=RandomForestRegressor()
lr=LinearRegression()
dt=DecisionTreeRegressor()
sm=SVR()
```

Fitting the models with the dataset

```
model_xg=xgr.fit(X_train,y_train)
y_xg=model_xg.predict(X_test)
# model_rf=rf.fit(X_train,y_train)
# y_rf=model_rf.predict(X_test)
# model_lr=lr.fit(X_train,y_train)
# y_lr=model_lr.predict(X_test)
# model_dt=dt.fit(X_train,y_train)
# y_dt=model_dt.predict(X_test)
# model_sm=sm.fit(X_train,y_train)
# y_sm=model_sm.predict(X_test)
```

Checking the metrics

```
print('R2-xgb',r2_score(y_test,y_xg))
print('RMSE-xgb',np.sqrt(mean_squared_error(y_test,y_xg)))

# print('R2-rf',r2_score(y_test,y_rf))
# print('RMSE-rf',np.sqrt(mean_squared_error(y_test,y_rf)))

# print('R2-lr',r2_score(y_test,y_lr))
# print('RMSE-lr',np.sqrt(mean_squared_error(y_test,y_lr)))

# print('R2-dt',r2_score(y_test,y_dt))
# print('RMSE-dt',np.sqrt(mean_squared_error(y_test,y_dt)))

# print('R2-svm',r2_score(y_test,y_sm))
# print('RMSE-svm',np.sqrt(mean_squared_error(y_test,y_sm)))
```

```
R2-xgb 0.9197743106205652
RMSE-xgb 370.6768884049128
```

```
# import pickle
# file_name = "xgb_reg.pkl"
# # save
# pickle.dump(xgb_model, open(file_name, "wb"))

# model_xg.save_model('test_model.bin')
```

```
# data=[[5.311336,259.994904]]
# df = pd.DataFrame(data, columns=[ 'WindSpeed(m/s)','WindDirection'])
# xgr.predict(df)
```

IBM Deployment

```
!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 (1.0.255)
Collecting ibm-watson-machine-learning
  Downloading ibm_watson_machine_learning-1.0.256-py3-none-any.whl (1.8 MB)
     |████████████████████████████████| 1.8 MB 13.7 MB/s eta 0:00:01
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (20
Requirement already satisfied: importlib-metadata in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-le
Requirement already satisfied: ibm-cos-sdk==2.11.* in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-l
Requirement already satisfied: packaging in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (
Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (1.
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (2
Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (0.3
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (0
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
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-s
Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk
Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm
Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm
Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dateutil<3.0.0,>=2.1->ib
Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->ibm
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->ibm-watson-machin
Requirement already satisfied: zipp>=0.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from importlib-metadata->ibm-watson
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from packaging->ibm
Installing collected packages: ibm-watson-machine-learning
  Attempting uninstall: ibm-watson-machine-learning
    Found existing installation: ibm-watson-machine-learning 1.0.255
    Uninstalling ibm-watson-machine-learning-1.0.255:
      Successfully uninstalled ibm-watson-machine-learning-1.0.255
Successfully installed ibm-watson-machine-learning-1.0.256
```

```
from ibm_watson_machine_learning import APIClient
import json
```

Authenticate and set Space

t1xJwH_pNvesyStso2tawTlpypHX0HEQJVMev99cmAtK

```
wml_credentials = {
    "apikey":"t1xJwH_pNvesyStso2tawTlpypHX0HEQJVMev99cmAtK",
    "url":"https://us-south.ml.cloud.ibm.com"
}
```

```
wml_client = APIClient(wml_credentials)
```

```
wml_client.spaces.list()
#9ebdfdcd-9254-4c80-bfaf-df2ab971a807
```

```
Note: 'limit' is not provided. Only first 50 records will be displayed if the number of records exceed 50

ID NAME CREATED
9ebdfdcd-9254-4c80-bfaf-df2ab971a807 XGB_1 2022-10-24T16:22:07.291Z
```

```
SPACE_ID= "9ebdfdcd-9254-4c80-bfaf-df2ab971a807"
```

```
wml_client.set.default_space(SPACE_ID)
    'SUCCESS'
```

```
wml_client.software_specifications.list(100)
```

```
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
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
xgboost_0.82-py3.6 39e31acd-5f30-41dc-ae44-60233c80306e base
pytorch-onnx_1.2-py3.6-edt 40589d0e-7019-4e28-8daa-fb03b6f4fe12 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
spark-mllib_2.3-r_3.6 6586b9e3-ccd6-4f92-900f-0f8cb2bd6f0c base
tensorflow_2.4-py3.7 65e171d7-72d1-55d9-8ebb-f813d620c9bb base
spss-modeler_18.2 687eddc9-028a-4117-b9dd-e57b36f1efa5 base
pytorch-onnx_1.2-py3.6 692a6a4d-2c4d-45ff-a1ed-b167ee55469a base
spark-mllib_2.3-scala_2.11 7963efe5-bbec-417e-92cf-0574e21b4e8d base spark
mllib_2.4-py37 7abc992b-b685-532b-a122-a396a3cdbaab base
caffe_1.0-py3.6 7bb3dbe2-da6e-4145-918d-b6d84aa93b6b base
pytorch-onnx_1.7-py3.7 812c6631-42b7-5613-982b-02098e6c909c base
cuda-py3.6 82c79ece-4d12-40e6-8787-a7b9e0f62770 base
```

```
import sklearn
sklearn. version
```

```
'1.0.2'
```

```
MODEL_NAME = 'XGB_1'
DEPLOYMENT_NAME = 'XGB_1'
DEMO_MODEL = model_xg


# Set Python Version
software_spec_uid = wml_client.software_specifications.get_id_by_name('runtime-22.1-py3.9')


# 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_uid
}

#Save model

model_details = wml_client.repository.store_model(
    model=DEMO_MODEL,
    meta_props=model_props,
    training_data=X_train,
    training_target=y_train
)


model_details
```

```
     {'entity': {'hybrid_pipeline_software_specs': [],
       'label_column': 'ActivePower(kW)',
       'schemas': {'input': [{'fields': [{'name': 'WindSpeed(m/s)',
            'type': 'float64'},
           {'name': 'WindDirection', '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': '2022-10-24T16:27:10.568Z',
       'id': '4e6c5b96-fab5-44db-ac39-3f744f3cb469',
       'modified_at': '2022-10-24T16:27:14.343Z',
       'name': 'XGB_1',
       'owner': 'IBMid-666002LP2L',
       'resource_key': '8f2e5b48-b33a-4255-89d9-33adb5729b2d',
       'space_id': '9ebdfdcd-9254-4c80-bfaf-df2ab971a807'},
      'system': {'warnings': []}}
```

```
model_id = wml_client.repository.get_model_id(model_details)
model_id
```

```
     '4e6c5b96-fab5-44db-ac39-3f744f3cb469'
```

```
# Set meta
deployment_props = {
    wml_client.deployments.ConfigurationMetaNames.NAME:DEPLOYMENT_NAME,
    wml_client.deployments.ConfigurationMetaNames.ONLINE: {}
}


# Deploy
deployment = wml_client.deployments.create(
    artifact_uid=model_id,
    meta_props=deployment_props
)
```

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

     Synchronous deployment creation for uid: '4e6c5b96-fab5-44db-ac39-3f744f3cb469' 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='0644c680-478f-475f-bc23-2a64fc6490a5'
```