PROJECT REPORT

Predicting The Energy Output Of Wind Turbine Based On Weather Conditions

NAME: Prabhanjan Kumar

PROJECT NAME: Predicting the energy output of wind turbine based on weather conditions

PROJECT DOMAIN:Machine Learning

CONTENTS:

1.INTRODUCTION

1.1 Overview

2.LITERATURE SURVEY

2.1 Existing problem

2.2 Proposed solution

3.THEORETICAL ANALYSIS

3.1 Block diagram

3.2 Hardware/Software designing

4.EXPERIMENTAL INVESTIGATIONS

5.RESULTS

6.ADVANTAGES

7.DISADVANTAGES

8.APPLICATIONS

9.CONCLUSION

10.FUTURE SCOPE

11.BIBLIOGRAPHY

1.INTRODUCTION

1.1 OVERVIEW

Wind power has received increasing attention around the earth due to its renewable nature as well as environmental friendliness. With the global installed wind power capacity rapidly increasing , the wind industry is growing into a large-scale business. over the last decade there has been rapid growth in wind generation of electricity, with the installed wind power capacity worldwide has increased almost fourfold from circa 24.3 GW(Giga Watts) to an expected 203.5 GW(Giga Watts). In power systems, balance is maintained by continuously adjusting capacity and by controlling demand. Reliable short-term wind speed forecasts play a practical and crucial role in wind energy conversion systems , such as the dynamic control of wind turbines and power system scheduling. A precise forecast needs to overcome problems of variable energy production caused by fluctuating weather conditions.Powergenerated by wind is highly dependent on the wind speed.Though it is highly non-linear , wind speed follows a certain pattern over a certain period of time.

2.LITERATURE SURVEY:

2.1EXISTING PROBLEM:

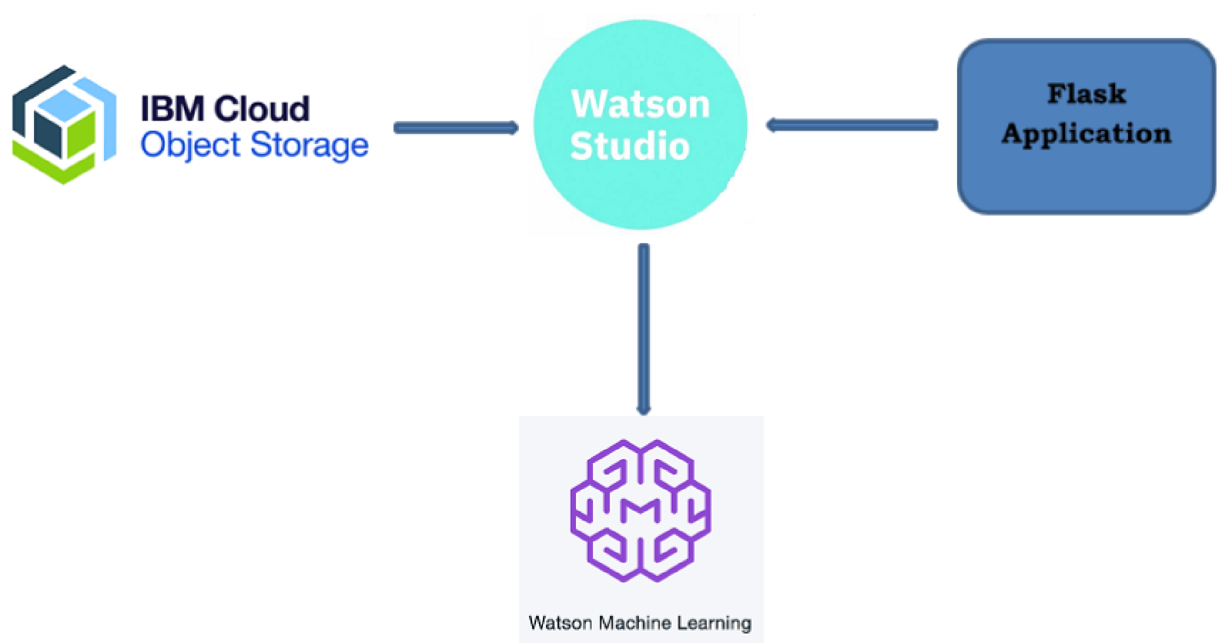
Wind energy plays an increasing role in the supply of energy worldwide. The energy output of a wind farm is highly dependent on the weather conditions present at its site. If the output can be predicted more accurately, energy suppliers can coordinate the collaborative production of different energy sources more efficiently to avoid costly overproduction. In this paper, we predict energy prediction based on weather data and analyze the important parameters as well as their correlation on the energy output

2.2PROPOSED SOLUTION:

Our aim is to map weather data for energy production. We wish to show that even data that is publicly available for weather stations close to wind farms can be used to give a good prediction of the energy output. Furthermore, we examine the impact of different weather conditions on the energy output of wind farms.

3.THEORITICAL ANALYSIS:

3.1. BLOCK DIAGRAM:



3.2. SOFTWARE DESIGNING:

- Flask
- IBM Watson Studio
- IBM Machine Learning
- IBM Cloud Object Storage

4.EXPERIMENTAL INVESTIGATIONS:

Exploring IBM Cloud Services:
IBM provides many services some of them are

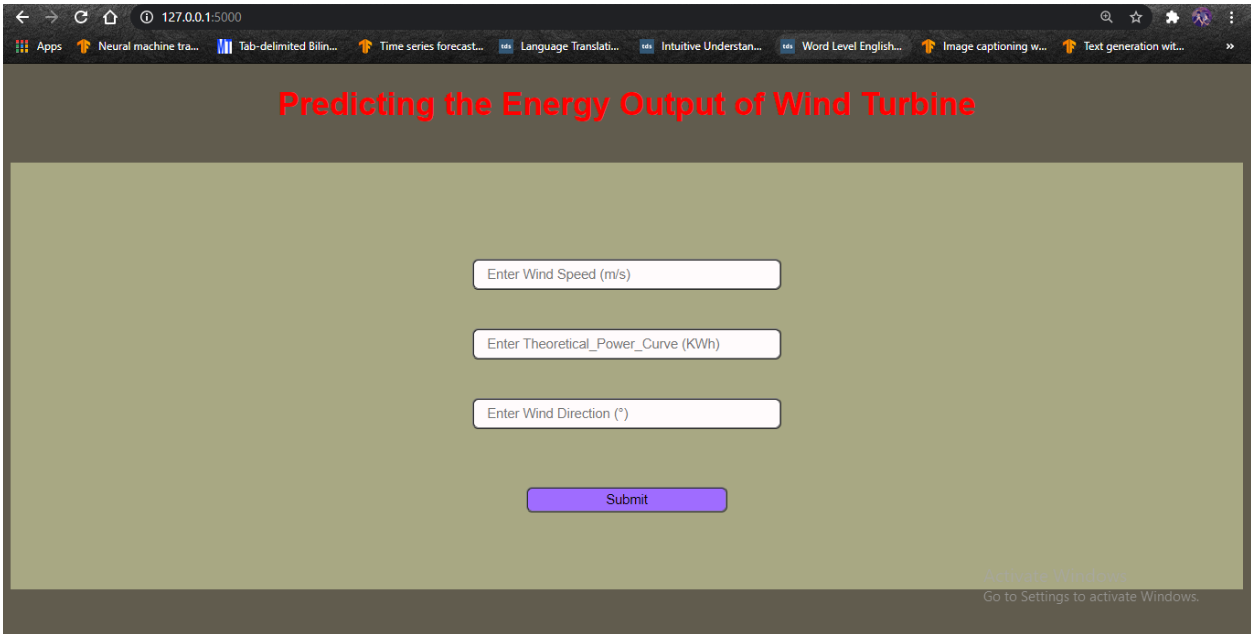
Machine Learning: IBM Watson Machine Learning make smarter decisions,slove through problems, and improve user outcomes
Watson Studio: Embed AI and Machine learning into Business,create custom models using our own data

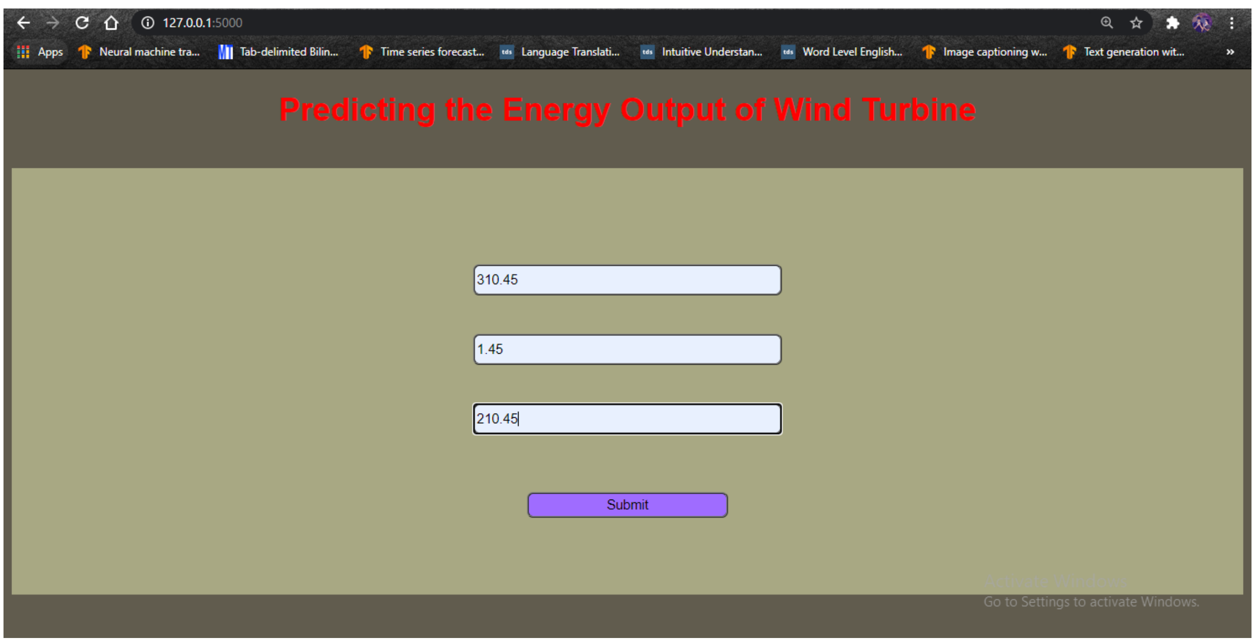
Exploring IBM Watson Services:
Watson Studio provides you with the environment and tools to solve your business problems by collaboratively working with data. You can choose the tools you need to analyze and visualize data, to cleanse and shape data, to ingest streaming data, or to create and train machine learning models.

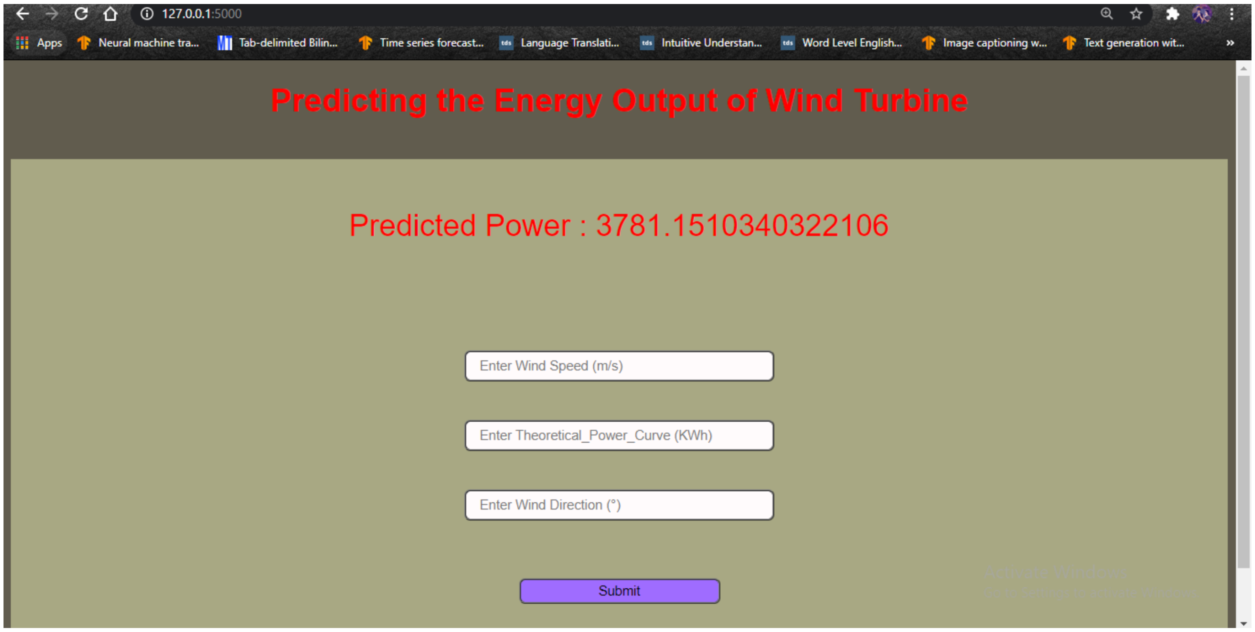
Watson Machine Learning: Using IBM Watson Machine Learning, you can build analytical models and neural networks, trained with your own data, that you can deploy for use in applications.Watson Machine Learning provides a full range of tools and services so you can build, train, and deploy Machine Learning models. Choose from tools that fully automate the training process for rapid prototyping to tools that give you complete control to create a model that matches your needs.

Watson Notebook: Watson Notebook provide an interactive programming environment for working with data, testing models, and rapid prototyping.

5. RESULTS:


127.0.0.1:5000
Apps
Neural machine tra...
Tab-delimited Bilin...
Time series forecast...
Language Translati...
Intuitive Understan...
Word Level English...
Image captioning w...
Text generation wit...
Predicting the Energy Output of Wind Turbine
Enter Wind Speed (m/s)
Enter Theoretical_Power_Curve (KWh)
Enter Wind Direction (°)
Submit
Go to Settings to activate Windows.


127.0.0.1:5000
Apps
Neural machine tra...
Tab-delimited Bilin...
Time series forecast...
Language Translati...
Intuitive Understan...
Word Level English...
Image captioning w...
Text generation wit...
Predicting the Energy Output of Wind Turbine
310.45
1.45
210.45
Submit
Go to Settings to activate Windows.



6.ADVANTAGES:

- Accurate wind power forecasts are also important in reducing the occurrence or length of curtailments(which translate to cost savings), improved worker safety, and mitigating the physical impacts of extreme weather on wind power systems
- Wind speed forecasting naturally has greater value where balancing markets are part of a competitive trading system of electricity,because the balancing market provides financial incentives to the generators and retailers for accurate output predictions

7.DISADVANTAGES:

- The challenges to face when wind generation is injected in a power system depend on the share of that renewable energy.
- For Denmark, which is a country which one of the highest shares of wind power in the electricity mix, the average wind power penetration on over the year is of 16-0%(meaning that 16-20% of the electricity consumption is met wind energy). While the instantaneous penetration (that is, the instantaneous wind power production

compared to the consumption to be met at a given time) may be above 100%

8.APPLICATIONS:

- **BETTER POWER OUTPUT:**
  Wind power forecasts are important in efficiently using wind turbines for generating power output.
- **EFFICIENT:**
  Predicting features like wind speed and wind direction can greatly help one to make decisions on when to switch on the wind turbine and when to switch it off (when it is assumed to not get the suitable conditions for generating power)
- **ENVIRONMENT FRIENDLY:**
  If we are able to achieve predicting the wind power output , then it will open up mor avenues for efficient power production in the field. This will lower the dependance on conventional sources of energy like cool which can cause harm to our environment.

9.CONCLUSION:

A machine learning model has been developed to forecast the power output of the wind turbine. A simple,efficient and a versatile model is built keeping in mind the diversity of data, computational complexity and overhead involved in making API calls to the model for prediction. The product can increase the accuracy of the forecasting the output from the wind turbine. Overall accurate wind power prediction reduces the financial and technical risk of uncertainty of wind power production for all electricity market participants

10.FUTURE SCOPE:

- A more generalized model can be developed to suit forecasting for different locations

- We can improve the predictions using the ARIMA model and other models that are powerful imparting more features to our training set will enhance the predictions and will open up new perspective an every front of wind prediction
- The model can be developed to make predictions on other sources of data such as solar power, tidal power etc,
- On device model can be developed to make much more faster predictions

11.BIBILOGRAPHY:

- https://www.hindawi.com/journals/mpe/2015/939305/
- https://machinelearningmastery.com/arima-for-time-series-forecasting-python/

Code:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

```
import 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 includes your credentials.
# You might want to remove those credentials before you share the notebook.
client_df36ec52cc714b44bfae6c60d711c038 = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='s5R8OWBXdEjb2FtwifA0xUaejPAefGQFIn2OD3noPnuJ',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.eu-geo.objectstorage.service.networklayer.com')

body = client_df36ec52cc714b44bfae6c60d711c038.get_object(Bucket='buildthon-donotdelete-pr-6eycorih6lnzcg',Key='T1.csv')['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )

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

	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

```
dataset=dataset.set_index(dataset['Date/Time']).drop('Date/Time',axis=1)
dataset.head()
```

	LV ActivePower (kW)	Wind Speed (m/s)	Theoretical_Power_Curve (KWh)	Wind Direction (°)
Date/Time				
01 01 2018 00:00	380.047791	5.311336	416.328908	259.994904
01 01 2018 00:10	453.769196	5.672167	519.917511	268.641113
01 01 2018 00:20	306.376587	5.216037	390.900016	272.564789
01 01 2018 00:30	419.645905	5.659674	516.127569	271.258087
01 01 2018 00:40	380.650696	5.577941	491.702972	265.674286

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 50530 entries, 01 01 2018 00:00 to 31 12 2018 23:50
Data columns (total 4 columns):
LV ActivePower (kW)              50530 non-null float64
Wind Speed (m/s)                 50530 non-null float64
Theoretical_Power_Curve (KWh)    50530 non-null float64
Wind Direction (°)               50530 non-null float64
dtypes: float64(4)
memory usage: 1.9+ MB
```

```
dataset.describe()
```

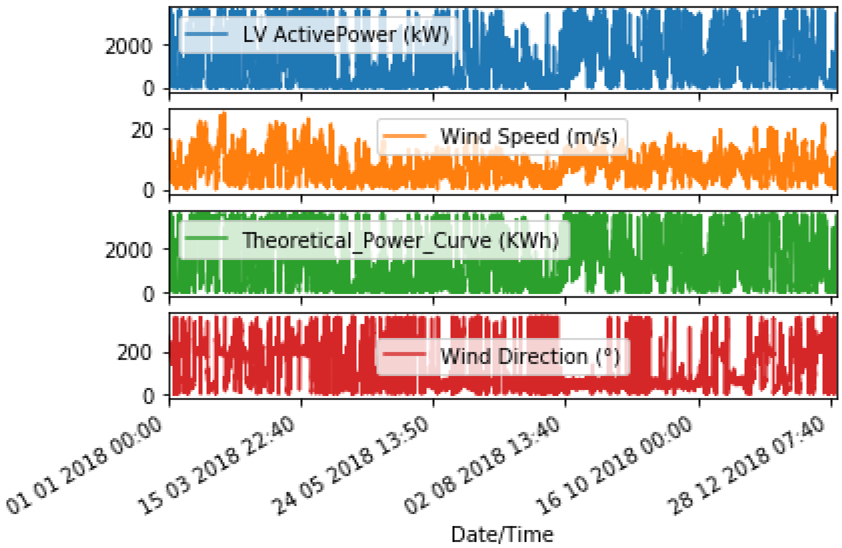
	LV ActivePower (kW)	Wind Speed (m/s)	Theoretical_Power_Curve (KWh)	Wind Direction (°)
count	50530.000000	50530.000000	50530.000000	50530.000000
mean	1307.684332	7.557952	1492.175463	123.687559
std	1312.459242	4.227166	1368.018238	93.443736
min	-2.471405	0.000000	0.000000	0.000000
25%	50.677890	4.201395	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

```
dataset.isnull().sum()
```

```
LV ActivePower (kW)              0
Wind Speed (m/s)                 0
Theoretical_Power_Curve (KWh)    0
Wind Direction (°)               0
dtype: int64
```

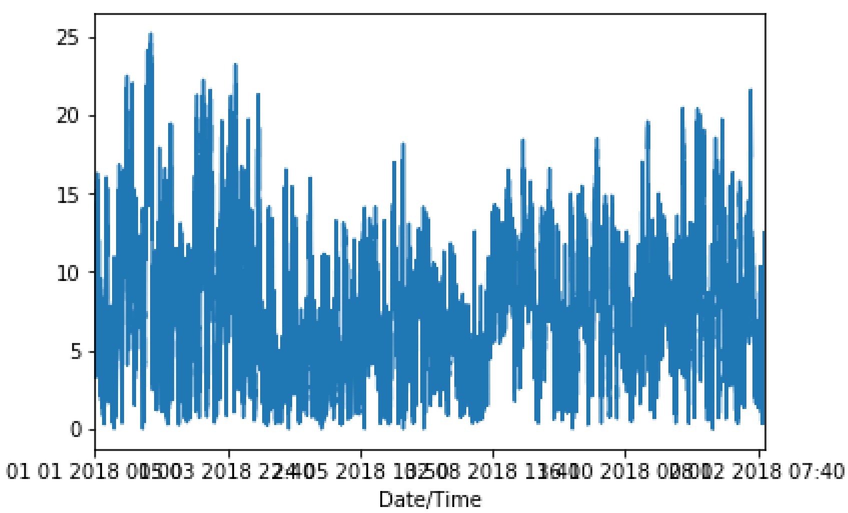
```
dataset.plot(subplots=True)
```

```
array([<matplotlib.axes._subplots.AxesSubplot object at 0x7f1f6e282d68>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f6e229fd0>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f6e1d6f98>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x7f1f6e1860f0>],
      dtype=object)
```



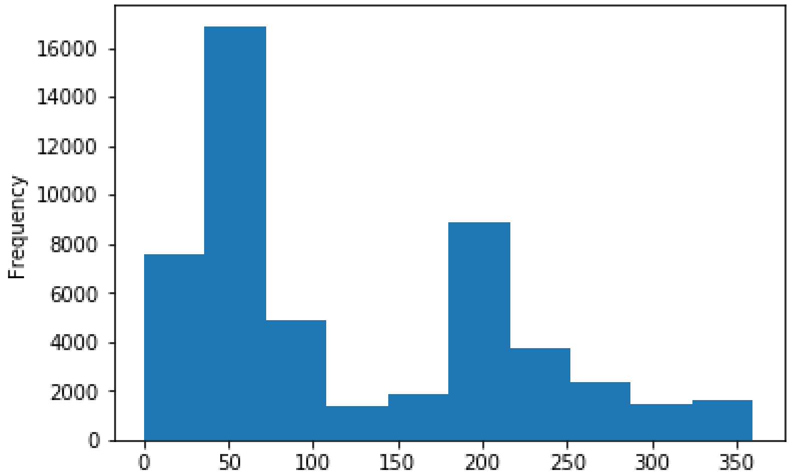

```
1  dataset['Wind Speed (m/s)'].plot(kind='line')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1f9d8f9cc0>
```

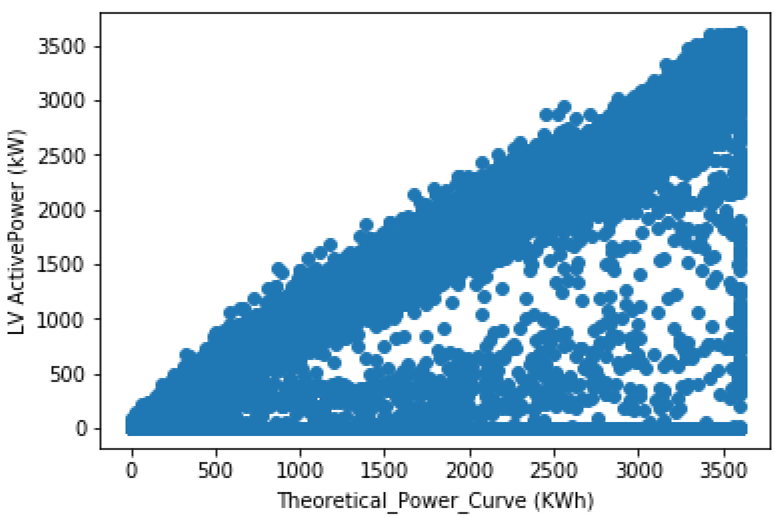



```
1  dataset['Wind Direction (°)'].plot(kind='hist')
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f1f6daa9710>

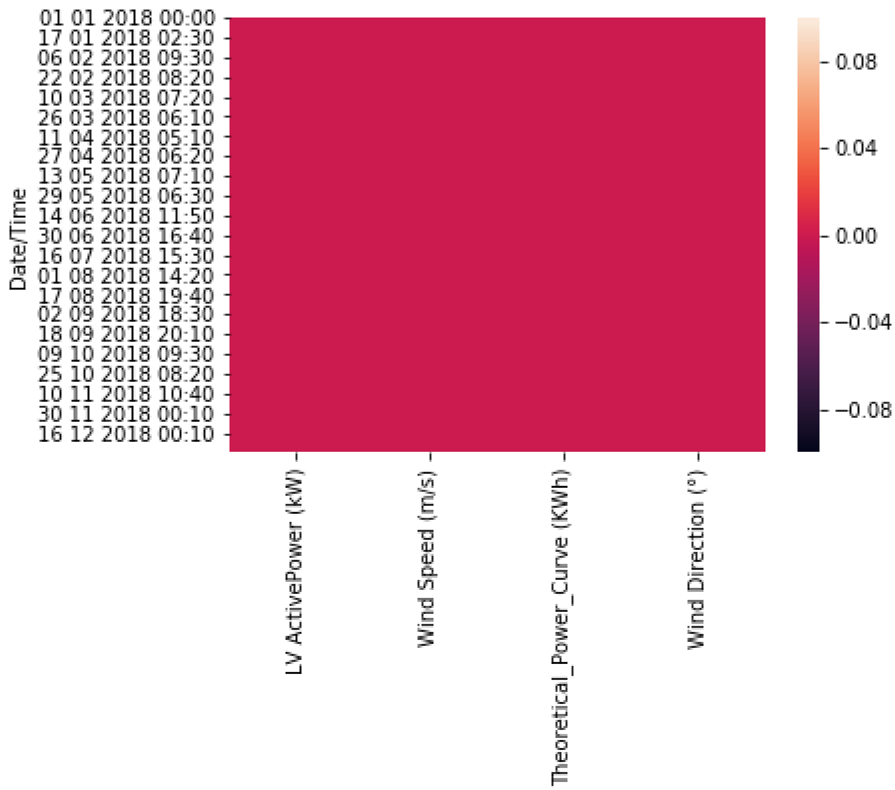



```
plt.scatter(dataset['Theoretical_Power_Curve (KWh)'],dataset['LV ActivePower (kW)'])
plt.xlabel('Theoretical_Power_Curve (KWh)')
plt.ylabel('LV ActivePower (kW)')
plt.show()
```




```
sns.heatmap(dataset.isnull())
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1f6b6e3860>
```




```
data_in=dataset.iloc[:,1:]
data_out=dataset.iloc[:,0]
data_in=np.array(data_in)
data_out=np.array(data_out)
sc=MinMaxScaler(feature_range=(0,1))
data_in=sc.fit_transform(data_in)
x_train,x_test,y_train,y_test=train_test_split(data_in,data_out,test_size=0.2)
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(40424, 3)
(10106, 3)
(40424,)
(10106,)
```

```
model=LinearRegression()
model.fit(x_train,y_train)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
         normalize=False)
```

```
predictions=model.predict(x_test)
model.score(x_test,y_test)
```

```
0.9117439268018639
```

```
from ibm_watson_machine_learning import APIClient
wml_credentials={
    "apikey": "UcwmxIfzEJr5hlZ7dPgNftCxzG6X2820RNroBHZakzus",
    "url": "https://eu-gb.ml.cloud.ibm.com",
}
client=APIClient(wml_credentials)
space_id='482c0fc8-6148-4912-b83e-4aff7315f9b9' #from deploy spaces i create MyFirstSpace
client.set.default_space(space_id)
```

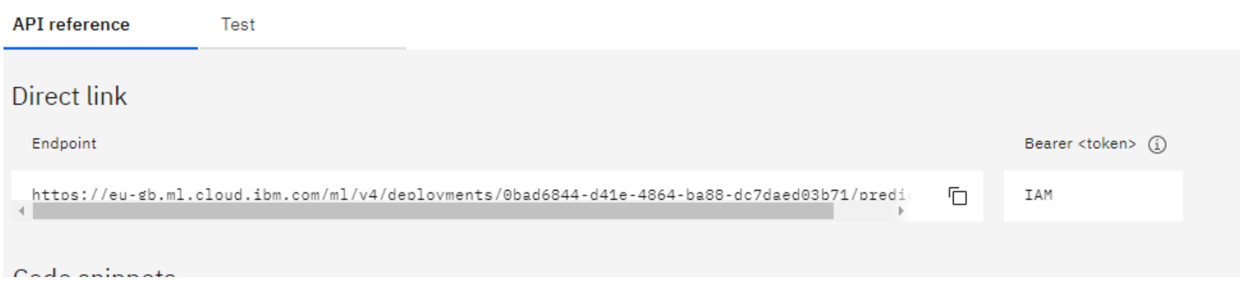
```
'SUCCESS'
```

```
sw_spec_id = client.software_specifications.get_id_by_name('scikit-learn_0.22-py3.6')
meta_props={
    client.repository.ModelMetaNames.NAME:"MyFinalPowerModel",
```

```
    client.repository.ModelMetaNames.TYPE: 'scikit-learn_0.22',
    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: sw_spec_id
}
model_details=client.repository.store_model(model,meta_props=meta_props)
model_uid=client.repository.get_model_uid(model_details)
meta_props = {
    client.deployments.ConfigurationMetaNames.NAME: "Power Predector",
    client.deployments.ConfigurationMetaNames.DESCRIPTION: "Predict Power",
    client.deployments.ConfigurationMetaNames.ONLINE: {},
    client.deployments.ConfigurationMetaNames.HARDWARE_SPEC: {'name': 'S', 'nodes': 1}
}

deployment_details = client.deployments.create(model_uid, meta_props=meta_props)

deployment_uid = client.deployments.get_uid(deployment_details)

print( deployment_uid )
```

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

Synchronous deployment creation for uid: '197fb037-474d-4614-bc02-ab2a52844fb0' started

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


initializing
ready


------------------------------------------------------------------------------------------------
Successfully finished deployment creation, deployment_uid='0bad6844-d41e-4864-ba88-dc7daed03b71'
------------------------------------------------------------------------------------------------


0bad6844-d41e-4864-ba88-dc7daed03b71
```

Endpoint :



UserInterface:

hello.py

```python
from flask import Flask,render_template,request
import requests
import urllib3, json
app=Flask(__name__)
@app.route('/',methods=['POST','GET'])
def hello():
    if request.method=='POST':
        ws=request.form['a']
        tpc=request.form['b']
        wd=request.form['c']
        try:
            ws=float(ws)
            tpc=float(tpc)
```

```
            wd=float(wd)
        except:
            return render_template('data.html',err_msg='Enter Valid Data')
        url = "https://iam.cloud.ibm.com/identity/token"
        headers = {"Content-Type": "application/x-www-form-urlencoded"}
        data = "apikey=" + 'UcwmxIfzEJr5hlZ7dPgNftCxzG6X2820RNroBHZakzus' + "&grant_type=urn:ibm:params:oauth:grant-type:apikey"
        IBM_cloud_IAM_uid = "bx"
        IBM_cloud_IAM_pwd = "bx"
        response = requests.post(url, headers=headers, data=data, auth=(IBM_cloud_IAM_uid, IBM_cloud_IAM_pwd))
        print(response)
        iam_token = response.json()["access_token"]
        header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + iam_token,
                  'ML-Instance-ID': '6ca9583e-3e40-4bb0-8f96-bf033afc56df'}
        payload_scoring = {"input_data": [
            {"fields": ["Wind Speed (m/s)", "Theoretical_Power_Curve (KWh)", "Wind Direction (°)"],
             "values": [[ws,tpc,wd]]}]}
        response_scoring = requests.post(
            'https://eu-gb.ml.cloud.ibm.com/ml/v4/deployments/0bad6844-d41e-4864-ba88-dc7daed03b71/predictions?version=2020-09-01',
            json=payload_scoring, headers=header)
        print(response_scoring)
        a = json.loads(response_scoring.text)
        print(a)
        pred = a['predictions'][0]['values'][0][0]
        return render_template('data.html',result=pred)
    else:
```

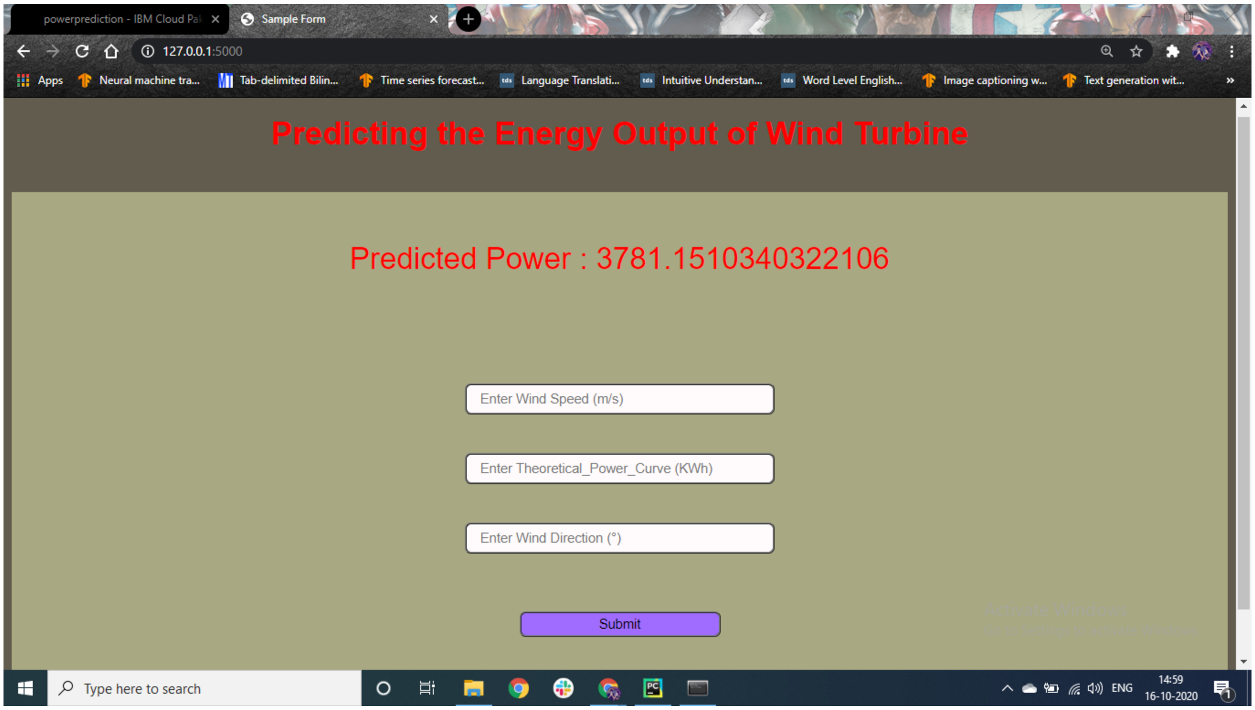
```
        return render_template('data.html')

if __name__ == '__main__':
    app.run(debug=True)

```

data.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <title>Sample Form</title>
    <style>
        .power
        {
            width:300px;
            height:25px;
            background-color:#fffbfc;
            border-style: ridge;
            border-color:gray;
            border-radius:6px;
        }
        body
        {
            font-family:sans-serif;
        }
        #sub
        {
            width:200px;
            height:25px;
            background-color:#9f6cff;
            border-style: ridge;
            border-color:gray;
            border-radius:6px;
        }
```

```
    </style>
</head>
<body style="background-color:#615c4e">
    <center>
        <h1 style="color:red">Predicting the Energy Output of Wind Turbine</h1>
        <br/>
        <div style="background-color:#a8a883;">
        <br/>
        {% if result %}
        <p style="color:red;font-size:30px;"> Predicted Power : {{result}}</p>
        {% endif %}
        {% if err_msg %}
        <p style="color:red;font-size:15px;">{{err_msg}}</p>
        {% endif %}
        <br/>
            <br/><br/><br/>
        <form method="post" action="/">
            <input type="text" name="a" class="power" placeholder="   Enter Wind Speed (m/s)" required><br/><br/><br/>
            <input type="text" name="b" class="power" placeholder="   Enter Theoretical_Power_Curve (KWh)" required><br/><br/><br/>
            <input type="text" name="c" class="power" placeholder="   Enter Wind Direction (°)" required><br/><br/><br/><br/>
            <input type="submit" value="Submit" id="sub"><br/>
            <br/><br/><br/><br/>
        </form>
        </div>
    </center>
</body>
```

```
54 </html>
```



powerprediction - IBM Cloud Pal
Sample Form
127.0.0.1:5000
Predicting the Energy Output of Wind Turbine
Predicted Power : 3781.1510340322106
Enter Wind Speed (m/s)
Enter Theoretical_Power_Curve (KWh)
Enter Wind Direction (°)
Submit