SPRINT3:

Date	15 November 2022
Team ID	PNT2022TMID51231
Project Name	Predicting the energy output of wind turbine based on weather condition

In [ ]:
```python
'''Application Building:
    After the model is built, integrating it to a web application
    using Flask framework so that normal users can also use it
    to predict the energy in a no-code manner. In the application,
    the user provides the required values and get the prediction.
    user provides input as city and get the weather conditions for that city
    such as temperature, wind direction, wind speed from openweather API and
    give the particular month,day to predict the output energy power for a gi
```

In [ ]:
```python
import numpy as np
from flask import Flask, request, jsonify, render_template
import joblib
import requests

app = Flask(__name__)
#model = joblib.load('xg_RFR_forecast_model.sav')
model = joblib.load('dec_model.sav')

@app.route('/')
def home():
    return  render_template('intro.html')
@app.route('/predict')
def predict():
    return render_template('predict_page.html')
@app.route('/windapi',methods=['POST'])
def windapi():
    city=request.form.get('city')
    apikey="e26bb531d3393dec23475ee08ea9559b"
    url="http://api.openweathermap.org/data/2.5/weather?q="+city+"&appid="+ap
    resp = requests.get(url)
    resp=resp.json()
    temp = str((resp["main"]["temp"])-273.15) +" °C"
    humid = str(resp["main"]["humidity"])+" %"
    pressure = str(resp["main"]["pressure"])+" mmHG"
    speed =  str((resp["wind"]["speed"])*3.6)+"  Km/hr"
    direc = str((resp["wind"]["deg"]))+" deg"
    return  render_template('predict_page.html',  temp=temp,  humid=humid,  press
@app.route('/y_predict',methods=['POST'])
def y_predict():
    x_test =[]
```

```python
month_dic = {"jan":1,"feb":2,"mar":3,"apr":4,"may":5,"jun":6,"jul":7,"aug
m

= request.form['month']
print(m)
x_test.append(mont
h_dic[m])
d = int(request.form['day'])
x_test.append(d)
t = float(request.form['temp'])
x_test.append(t)
direc = float(request.form['direc'])
x_test.append(direc)
speed= float(request.form['wind'])
x_test.append(speed)
x_test = [x_test]
print(x_test)
prediction = model.predict(x_test)
print(prediction)
output = prediction[0]
```

```python
        return render_template('predict_page.html', prediction_text='The energy p


if __name__ == "__main__":
    app.run(debug=False)
```

```
 * Serving Flask app '__main__' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deplo
yment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [04/Nov/2022 11:26:00] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [04/Nov/2022 11:26:00] "GET /static/images/m123.gif HTTP/1.1" 4
04 -
127.0.0.1 - - [04/Nov/2022 11:26:00] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [04/Nov/2022 11:26:02] "GET /predict HTTP/1.1" 200 -
127.0.0.1 - - [04/Nov/2022 11:26:02] "GET /static/css/main.css HTTP/1.1" 404
-
127.0.0.1 - - [04/Nov/2022 11:26:02] "GET /static/css/items_grid.css HTTP/1.
1" 404 -
127.0.0.1 - - [04/Nov/2022 11:26:02] "GET /static/css/media.css HTTP/1.1" 404
-
127.0.0.1 - - [04/Nov/2022 11:27:39] "POST /windapi HTTP/1.1" 200 -
127.0.0.1 - - [04/Nov/2022 11:27:40] "GET /static/css/main.css HTTP/1.1" 404
-
127.0.0.1 - - [04/Nov/2022 11:27:40] "GET /static/css/media.css HTTP/1.1" 404
-
127.0.0.1 - - [04/Nov/2022 11:27:40] "GET /static/css/items_grid.css HTTP/1.
1" 404 -
127.0.0.1 - - [04/Nov/2022 11:28:06] "POST /y_predict HTTP/1.1" 200 -
127.0.0.1 - - [04/Nov/2022 11:28:06] "GET /static/css/main.css HTTP/1.1" 404
-
127.0.0.1 - - [04/Nov/2022 11:28:06] "GET /static/css/media.css HTTP/1.1" 404
```