Sprint 4:

|  |  |
| --- | --- |
| Date | 15 November 2022 |
| Team ID | PNT2022TMID45948 |
| Project Name | Predicting the energy output of wind turbine based on weather condition |

In [ ]:

'''sprint-4

Train thd model in IBM cloud

prediction based on scoring response of Trained model at IBM CLOUD '''

In [2]:



**import** numpy **as** np

**from** flask **import** Flask, request, jsonify, render\_template

**import** joblib

**import** requests

*# NOTE: you must manually set API\_KEY below using information retrieved from*

API\_KEY **=** "eq\_uB9S4u70ys6t-jeWs2uoNIscWVl0Gh0C9GTKlDloW"

token\_response **=** requests**.**post('https://iam.cloud.ibm.com/identity/token', da API\_KEY, "grant\_type": 'urn:ibm:params:oauth:grant-type:apikey'})

mltoken **=** token\_response**.**json()["access\_token"]

header **=** {'Content-Type': 'application/json', 'Authorization': 'Bearer ' **+** ml

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="](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 **=**[]

month\_dic **=** {"jan":1,"feb":2,"mar":3,"apr":4,"may":5,"jun":6,"jul":7,"aug m **=** request**.**form['month']

mon **=** month\_dic[m]

print(m)

*#x\_test.append(month\_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]*

x\_test **=**[[mon,d,t,direc,speed]]

payload\_scoring **=** {"input\_data": [{"field": [['m','d','t','direc','speed'

response\_scoring **=** requests**.**post('https://us-south.ml.cloud.ibm.com/ml/v4 print("Scoring response")

print(response\_scoring**.**json())

predictions **=** response\_scoring**.**json()

output **=** predictions['predictions'][0]['values'][0][0] print("final prediction",output)

print(x\_test)

**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 - - [07/Nov/2022 16:56:36] "GET / HTTP/1.1" 200 -

127.0.0.1 - - [07/Nov/2022 16:56:36] "GET /static/images/m123.gif HTTP/1.1" 4

04 -

127.0.0.1 - - [07/Nov/2022 16:56:40] "GET /predict HTTP/1.1" 200 -

127.0.0.1 - - [07/Nov/2022 16:56:40] "GET /static/css/main.css HTTP/1.1" 404

-

127.0.0.1 - - [07/Nov/2022 16:56:40] "GET /static/css/media.css HTTP/1.1" 404

-

127.0.0.1 - - [07/Nov/2022 16:56:40] "GET /static/css/items\_grid.css HTTP/1. 1" 404 -

127.0.0.1 - - [07/Nov/2022 16:56:46] "POST /windapi HTTP/1.1" 200 -

127.0.0.1 - - [07/Nov/2022 16:56:47] "GET /static/css/main.css HTTP/1.1" 404

-

127.0.0.1 - - [07/Nov/2022 16:56:47] "GET /static/css/items\_grid.css HTTP/1. 1" 404 -

127.0.0.1 - - [07/Nov/2022 16:56:47] "GET /static/css/media.css HTTP/1.1" 404

-

jun

127.0.0.1 - - [07/Nov/2022 16:57:10] "POST /y\_predict HTTP/1.1" 200 -

127.0.0.1 - - [07/Nov/2022 16:57:10] "GET /static/css/media.css HTTP/1.1" 404

-

127.0.0.1 - - [07/Nov/2022 16:57:10] "GET /static/css/items\_grid.css HTTP/1. 1" 404 -