Project Report

Predicting the energy output of wind turbine based on weather condition

Submitted by,

K.R.Harshithaa - 917719D031

K.Karunyah - 917719D037

M.Vaishnavi - 917719D106

V.Varun Kumar - 917719D107

Date	19 November 2022
Team ID	PNT2022TMID21439
Project Name	Predicting the Energy Output Of Wind Turbine Based On Weather Condition
Maximum Marks	8 Marks

TABLE OF CONTENTS

1.INTRODUCTION

1.1 Project Overview

Wind power/energy is one of the sources of generating electricity. The wind turns the propeller blades of a turbine around a rotor,which spins a generator which in turn generated electricity. But the generation of electricity solely depends on the stochastic nature of the wind in a particular region. Hence, forecasting of weather conditions plays a major role in balancing the supply and demand of electricity given any uncertainty in weather which might be a huge problem. Predicting the wind energy beforehand by accurately predicting the weather conditions will help in reducing wastage of energy as well as the cost spent on conservation. In this project, we propose a prediction system which combines both statistical and physical models to predict the wind energy output based on the weather conditions using machine learning.

1.2 Purpose

The main objective of this project is:

- Conserve energy and make use of available resources.
- Use of technologies i.e machine learning in an effective manner
- To reduce the cost of conservation of energy

Other objectives includes improving skill requirement of students by making use of Python,Cloud,Data processing Techniques,etc.

2.LITERATURE SURVEY

2.1 Existing problem

The climatic conditions present in the site decide the power output of a wind farm. Wind power availability is not known in advance, this causes problems for wind farm operators in terms of system and energy planning so this is considered as a random variable. Hence we need a accurate method to find this random variable which will also consider the other factors like topography, weather condition and wind features which also play a significant factor in the prediction.

2.2 References

- **NAME OF THE PAPER:** Energy Modeling Output of Wind System based on Wind Speed

  NAME OF THE AUTHOR: Abdelkader Harrouz, Ilhami Colak, Korhan Kayisli

  JOURNAL PUBLISHED: 2019 8th International Conference on Renewable Energy Research and Applications (ICRERA)

  MONTH AND YEAR PUBLISHED: November 2019

  OBJECTIVE OF THE PROJECT: There are many renewable energy sources that can be used to obtain electrical energy from natural sources in the world. Especially, wind energy plays an increasing role thanks to its feasibility and efficiency. Due to the source of wind energy, efficiency of wind farm is highly depending on the weather conditions. The main issue to obtain maximum performance is to predict the output. This situation provides collaborative production of different energy sources more efficiently with avoiding over-cost and overproduction. In this paper, there are three different wind models are modelled and simulated with choosing the complete and correct models .

- **NAME OF THE PAPER :** Short term wind and energy prediction for offshore wind farms using neural networks.

  NAME OF THE AUTHOR: Stefan Balluff, Jörg Bendfeld, Stefan Krauter

  JOURNAL PUBLISHED: 2015 International Conference on Renewable Energy Research and Applications (ICRERA)

  MONTH AND YEAR PUBLISHED: November 2015

OBJECTIVE OF THE PROJECT: Forecasting short term wind speed is of high importance for wind farm managers. The knowledge of the expected winds helps taking decisions (decision support) as the likes of maintenance and repair jobs or finishing works as health and safety is not guaranteed anymore. There are a number of methods and computations currently being used for forecasts: fuzzy logic, linear prediction or neural networks. For the latter there are also various algorithms and methods, from feed forward up to recurrent neural networks (RNN) and long short-term memory (LSTM). Recurrent neural networks belong to the group of machine learning algorithms and are part of artificial intelligence research. This paper is about forecasting wind speed and pressure using RNN.

- **NAME OF THE PAPER** : Pattern-Based Wind Speed Prediction Based on Generalized Principal Component Analysis

  NAME OF THE AUTHOR : Qinghua Hu, Pengyu Su, Daren Yu, Jinfu Liu

  JOURNAL PUBLISHED : IEEE Transactions on Sustainable Energy

  MONTH AND YEAR PUBLISHED : April 2014

OBJECTIVE OF THE PROJECT : Short-term wind speed prediction plays an important role in large-scale wind power penetration. However, there is still a large gap between the requirement of prediction performance and current techniques. In this paper, we propose a pattern-based approach to short-term wind speed prediction. It is well accepted that wind varies in different patterns in different weather conditions. Thus, we should use different models to describe these patterns, whereas most current works conduct wind speed prediction with a single model. Based on this observation, we introduce generalized principal component analysis to automatically discover the patterns hidden in the historical data of wind speed. Then we train a predicting function for each pattern and combine their outputs for the final prediction. Experimental results show that the proposed approach performs better than the clustering-based approach, a single model, and persistence forecasting.

- **NAME OF THE PAPER:** Wind power prediction using wavelet transform and chaotic characteristics

  NAME OF THE AUTHOR : Lijie Wang, Lei Dong,Ying Hao, Xiaozhong Liao

JOURNAL PUBLISHED : 2009 World Non-Grid-Connected Wind Power and Energy Conference

MONTH AND YEAR PUBLISHED : September 2009

OBJECTIVE OF THE PROJECT : In the electricity system, supply and demand must be equal at all times. Wind power generation is fluctuating due to the variation of wind. As more and more wind power generation is integrated into the power system, it is very important to predict the wind power production to contribute the system reserve reduction and the operational costs of the power plants. This paper brings wavelet transform into the time series of wind power and verifies that the decomposed series all have chaotic characteristic, so a new method of wind power prediction in short-term with Artificial Neural Network (ANN) model based on wavelet transform is presented. To test the approach, the wind power data from the Fujin wind farm and Saihanba wind farm of China are used for this study. The prediction results are presented and compared to the no wavelet transform method and ARMA method. The results show that the new method based on wavelet transform neural networks will be a useful tool in wind power prediction.

- **NAME OF THE PAPER**: The Use of Machine Learning and Performance Concept to Monitor and Predict Wind Power Output

NAME OF THE AUTHOR: Kelvin Palhares Bastos Sathler, Athanasios Kolios

JOURNAL PUBLISHED:2022 International Conference on Electrical, Computer and Energy Technologies (ICECET)

MONTH AND YEAR PUBLISHED: July 2022

OBJECTIVE OF THE PROJECT: Monitoring and predicting wind power output more precisely can be very beneficial for an increasingly competitive Wind Power industry. Although many advances have been made throughout the last decades, the production forecast is still based mainly on the manufacturing power curve and wind speed. Even though this approach is very useful, especially during the design phase, it does not consider other factors that affect production, such as topography, weather conditions, and wind features. A more precise prediction model that is able to recognize production fluctuation and is tailored using current operational data is proposed in this paper. The model analyzes the performance through Meteorological Mast Data (Met Mast Data) and then uses it as an input to monitor and predict power output.

2.3 Problem Statement Definition

Extracting electricity from renewable resources has been widely investigated in the past decades to decrease the worldwide crisis in the electrical energy and environmental pollution. The climatic conditions present in the site decide the power output of a wind farm. Wind power availability is not known in advance, this causes problems for wind farm operators in terms of system and energy planning so this is considered as a random variable. In our project, we propose an intelligent technique for forecasting wind speed and power output of wind turbine from several hours up to specific hours ahead. If the output is forecasted accurately, energy providers can keep away from costly overproduction. We will carry out this problem on publicly available weather and energy data sets correlating and considering different features in our project. This will enable us to cut down on the production cost and collaborate on different energy source more efficiently. Thus, we will develop an application which can forecast the wind power of the future leveraging AI tools and powerful visualization. The advantage of using ours over others is, it provides an Application with interactive UI with optimized model with higher accuracy to predict the energy output of the wind energy's accurate parameters.

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Manufacturer	Produce good and high quality of wind energy	I don't know where to place the windmills and how to use new technologies	I don't have a proper analysis of weather condition and latest technology	Concerned
PS-2	User	Find a reliable energy resource	I don't know how to use it effectively	I don't know the importance and uses of it	Worried
PS-3	Organization	Produce wind energy	I face overproduction and high-cost issues	I can't accurately predict the wind energy	Sad

3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

Building an application to recommend the power grid to suggest the best time to utilize the energy from wind farm

3.2 Ideation & Brainstorming

The brainstorming started with the definition of problem statement,what kind of problem we are trying to solve and focus on the brainstorm,by following the key rules of brainstorming.Every one in the team started to research and cam up with ideas like Tracking the weather, Analyzing ML algorithms, Efficiency of our system, System performance is evaluated using necessary metrics,Cost efficiency, Reliable under extreme conditions, Eco friendliness, Integration with existing technologies,System security features, Ideal platform for implementation, Expandable to other renewable sources of energy,Promoting the product,Analyzing existing technologies, Monitoring the weather, Performance metrics used are examined,Average power generated.

Ideal Platforms were Studied. Studying weather conditions Analyzing existing Technologies.

Then the ideas are prioritized based on the levels.Mural are shared and exported.The swot analysis is performed and the stakeholders are identified then we developed a plan.

3.3 Proposed Solution

The problem statement is, It is necessary to find a way to predict the energy output of a wind turbine in different weather conditions. The obtained wind energy must be used to give a steady supply of electricity.Now we have developed and came up with the idea that it is necessary to analyze and to store the data of the wind turbine in different weather conditions. With the past data stored in the database, we can predict the output of a wind turbine.

And a prediction system is developed with a method of combining statistical models and physical models. Hence the output energy can be forecasted by the auto regressive model.The novelty and uniqueness present in our idea is, present wind farms don't have any methods to predict the output energy based on the changing weather conditions. By implementing this model, it can be useful to predict the output energy before and the efficiency of the wind farms can also been improved.

It is having a huge impact on society, currently wind energy is not the primary source of electricity, but by implementing our solution we can produce more energy. So the utilization of non renewable resources can also be minimized. A wind farm with prediction mode would be more efficient than the present one. Switching to a clean source of energy is good for both human health and the environment.

Improvement of life standard, local employment, social bonds creation, income development, better health, consumer choice, demographic impacts, and community development can be achieved by the proper usage of renewable energy systems, it proposes a revenue model.

The scalability of the solution is it can be applied on the large scale in the existing wind farm. So the performance can also be improved.

3.4 Problem Solution fit

The problem Solution fit starts with the customer who is a wind energy producers.For addressing the customer we have to analyze the output energy of the wind turbine in changing weather conditions and to store the data in a dataset.The customer finds it as an efficient solution.that will automatically trigger all the other customer to do it.

The emotion of the customer changes before and after the product usage.In the available solution the estimation is calculated based on the past year energy outputs.The customer constraint is lack of budget,they are not clear on how to utilize the wind turbine effectively to produce a steady electricity.The customer do collect the data from the potential wind farms and make comparison to get the job done.

The channel behavior is to upload previous data and predict the output energy. This is done online,and offline we constantly maintain the inlet condition of the wind turbine.The root cause of the problem is high initial cost setup and unpredictable changes in weather conditions.So for this the best solution is,the inlet condition of the wind turbine is forecasted by an auto regressive model.which reduce the need for balancing energy and reserved power output energy.

4.REQUIREMENT ANALYSIS

4.1 Functional requirement

FR NO.	Functional Requirement(Epic)	Sub Requirement(Story/Sub-Task)
FR-1	User Registration and logging in by entering their username and password.	Registration through Form.
FR-2	User Confirmation by validating the username with respect to the password	Confirmation via pop-up Message.
FR-3	Displaying further information about the application.	By selecting the about button the details of the application will be displayed.
FR-4	Validating the city name.	System checks whether the city entered by the user is present or not. If present it will collect the further details else it will display the pop-up message as error in the city.
FR-5	Checking the data type of the value.	System checks for the data type of the value entered by the user.
FR-6	Validating all required fields.	Before predicting the output the system checks whether all the values are entered by the user and checks whether all values are correct.
FR-7	Displaying weather conditions for a given city.	It displays the weather of the city which has been selected.
FR-8	Displaying predicted energy output power.	The predicted output will be displayed as the amount of wind energy power generated.

4.2 Non-Functional requirements

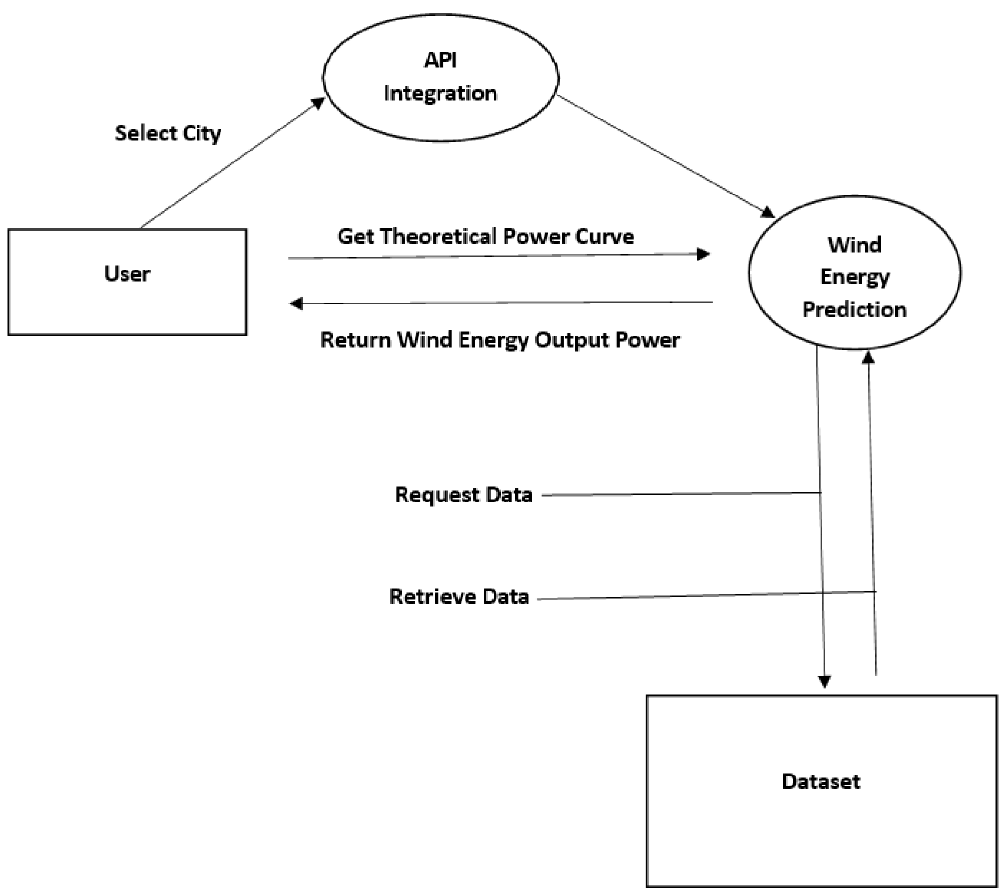
FR NO.	Non-Functional requirement	Description
NFR-1	Usability	The system satisfies the user goals and the application is easy to use.
NFR-2	Security	The data provided to system will be protected from attacks and unauthorized access
NFR-3	Reliability	The system will provide consistency in output without producing an error.
NFR-4	Performance	The performance will never degrade even if the workload is increased.
NFR-5	Availability	The application is available for 24*7
NFR-6	Scalability	The system can be used as web application as well as mobile application with sufficient internet availability.

5. PROJECT DESIGN

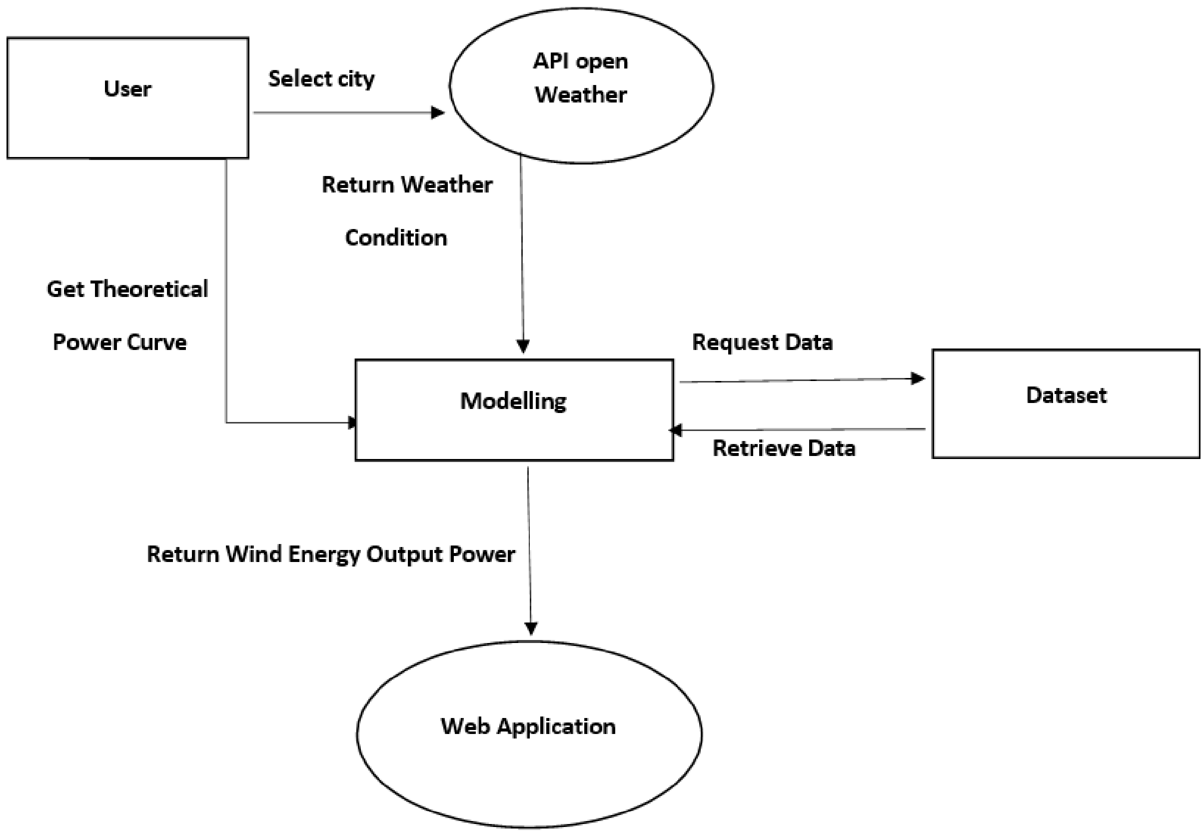
5.1 Data Flow Diagrams

LEVEL 0:

LEVEL 1:

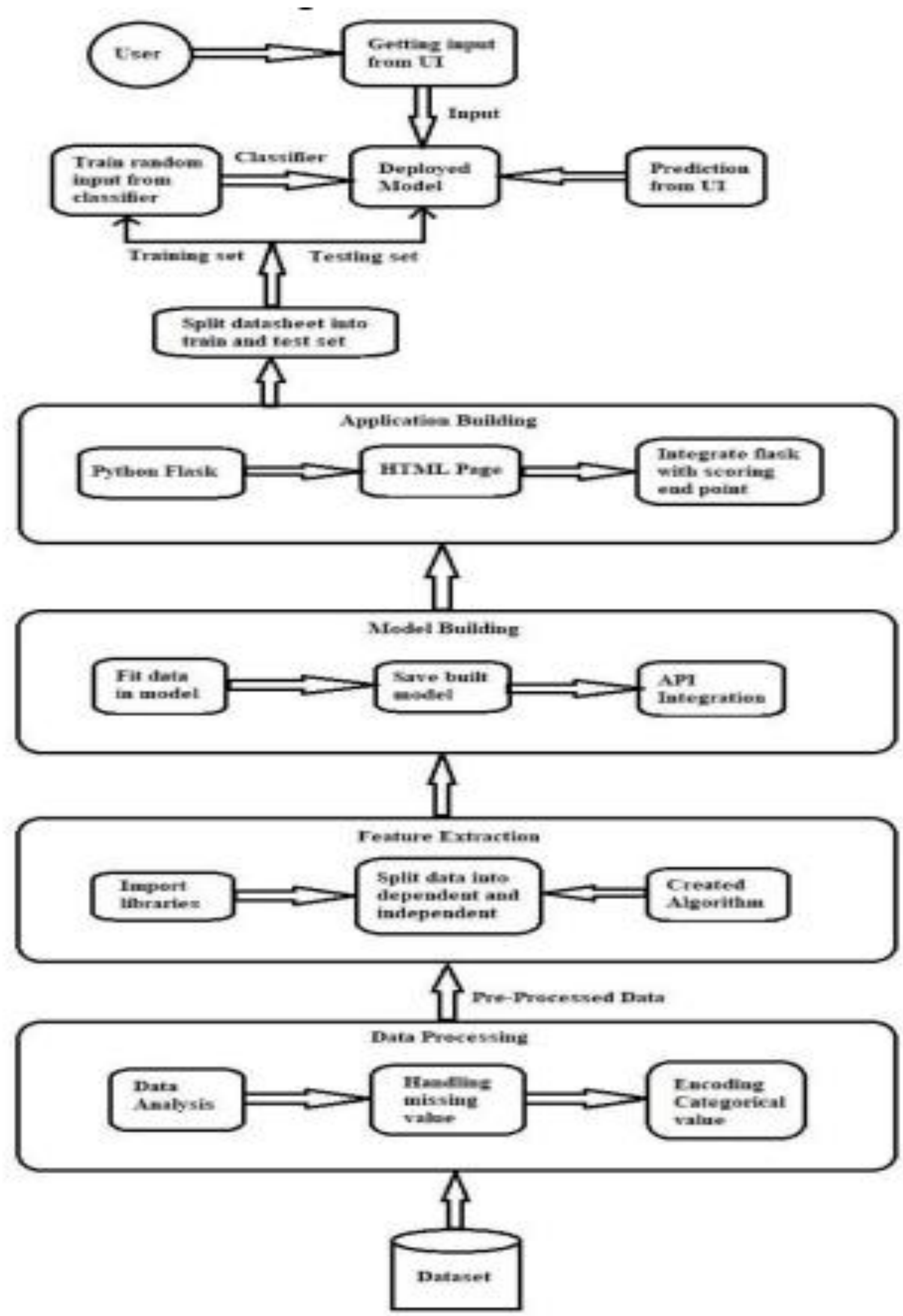


LEVEL 2:


User
Select city
API open Weather
Return Weather Condition
Get Theoretical Power Curve
Request Data
Modelling
Dataset
Retrieve Data
Return Wind Energy Output Power
Web Application

5.2 Solution & Technical Architecture

Solution architecture:



Technology architecture:

The user has to select the city and this is done through API integration, and Random forest prediction is also done through the Average all prediction model, the (random forest) algorithm establishes the outcome based on the predictions of the decision trees. It predicts by taking the average or mean of the output from various trees. Increasing the number of trees increases the precision of the outcome. Finally they are put into a flask where the datasets feed are referred and the wind energy prediction is done.

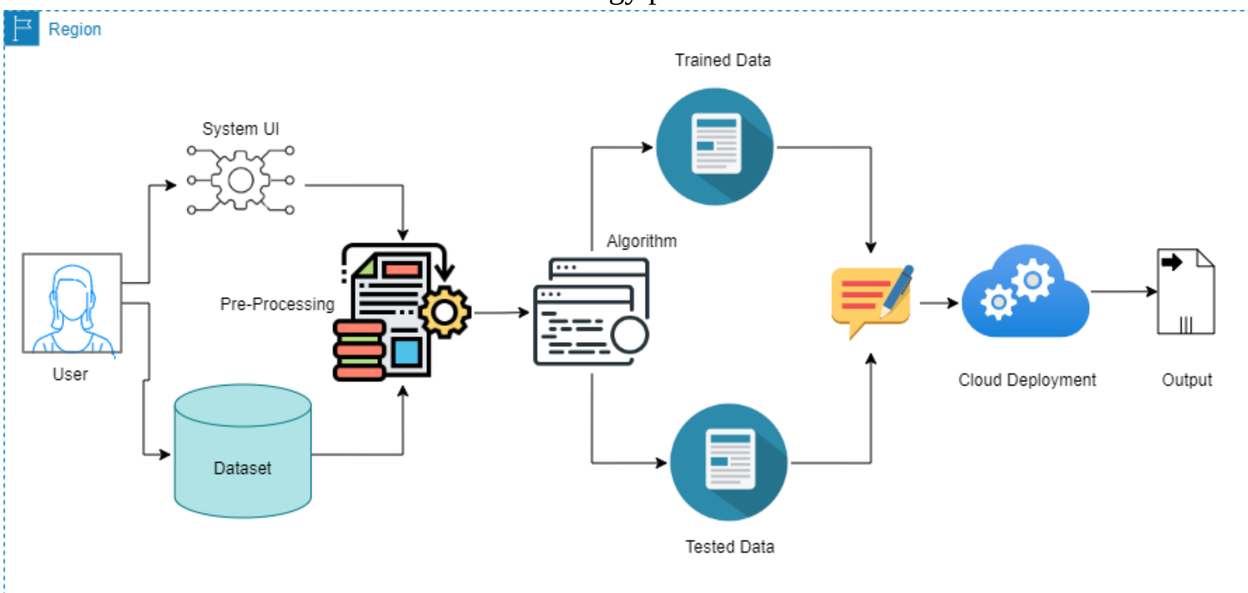


Table-1 : Components & Technologies:

z	Component	Description	Technology
1.	User Interface	User can interface with the app through login.	HTML, CSS, JavaScript.
2.	Machine Learning	Model building	Jupyter Notebook-Python
3	Deployment	Train The model on the cloud frameworks Following:	The main storage classes and resource list used.
3a.	Watson Studio	Train The model on power machine learning tool named Watson studio	IBM Watson Studio 1.0+

3b.	Kubernetes	Train The model on node red tool named Kubernetes	Kubernetes 1.8.2 (RBAC)
4.	Watson Assistant	If any Queries about Watson studio we post and get clear about queries about Watson studio.	IBM Watson Assistant
5.	Database	Google cloud SQL	MySQL, NoSQL.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant
7.	File Storage	We need to save our projects and deployments by storage	IBM Block Storage or Other Storage Service or Local Filesystem
8.	Open Weather map	To predict weather with the help of the api key	Openweathermap Api keys :(36c0639c05e38cd003ddea74d69b8822, a802b0f626c637d04185e582b5ad0d58,)
9.	IBM api	The purpose of generating and deploying models we need IBM api	IBM API keys (cROnoxEkdZHxElMwijOi2h7Q7kvTjRtpvAMzUCkCazXD , GwBW4bQSUaH4tROhskcQMbbB9uqNvz8OkKOF9l5q9R4 , HzrV2Q9Ywg3EyxMO14u62Meo2RhdZjp6np6AeCoq8QlR)
10.	Machine Learning Model	Using machine learning we can improve accuracy, efficiency, System development technologies	Random forest, Linear Regression, Kmeans, Naive Bayes Classifier and Decision tree , SVR.
11.	Infrastructure (Server / Cloud)	Application Deployment on IBM Watson studio and IBM Cloud	IBM Cloud

Project Design Phase-II Technology Stack (Technology Architecture & Stack)

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	1. Tensorflow 2. RNN 3. Theano 4. PyTorch 5. Caffe2 6. Keras 7. OpenCV 8. Scikit Learn	Scikit Learn 1.0+
2.	Security Implementations	The Data provided by the user will be surely kept safe with encryption.	AES-256, RSA, SHA-256, Hash Functions.
3.	Scalable Architecture	Three-tier architecture is a well-established software application architecture that organizes applications into three logical and physical computing tiers: the presentation tier, or user interface; the application tier, Database layer.	3-tier
4.	Availability	The model can be trained using IBM Watson Studio. IBM Cloud, API key for both Openweathermap and IBM cloud.	IBM cloud, Watson studio, API Keys.

5.3 User Stories

User Type	Functional Requirements	User Number Story	User Story/User Task	Acceptance Criteria	Priority	Release
Customer	Home (Application)	USN-1	As a user, I can view the guideline as well as the detailed information about the application	I can gain knowledge by practical method to use this application.	Low	Sprint-1
		USN-2	As a User, I can use this application by reading the instructions	I can use this in user friendly method by reading the instruction.	Low	Sprint-1
		USN-3	As a User, I can login and by entering the correct username and password	If login is correctly entered ,I can navigate to the next page.	Low	Sprint-2
		USN-4	As a user ,I am allowed to select the city and can get the weather of the city.	I can select the city ,If the city is correct I can further enter the details.	Medium	Sprint-3
		USN-5	As a user I am allowed to view the weather of the selected city.	If correct city is selected ,then the weather of the particular city will be displayed.	Medium	Sprint-4

		USN-6	As a User ,I can view the Power generated by the wind	If all values are entered correctly I can view the power generated by the	High	Sprint-5
		USN-7	As a User, I can use the web application virtually anywhere	I can use the application portably	High	Sprint-2
		USN-8	As it is open source ,I can use it cost freely.	I can use it without any payment to access	Medium	Sprint-2

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	5	High	Varun Kumar Vaishnavi Harshithaa Karunyah
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	5	High	Varun Kumar Vaishnavi Harshithaa Karunyah
Sprint-1		USN-3	As a user, I can register for the application through google	5	Low	Varun Kumar Vaishnavi Harshithaa Karunyah
Sprint-1		USN-4	As a user, I can register for the application through Gmail	5	Medium	Varun Kumar Vaishnavi Harshithaa Karunyah

Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	5	High	Varun Kumar Vaishnavi Harshithaa Karunyah
Sprint-2	Dashboard	USN-6	Once logged in, I can access my dashboard	6	Medium	Varun Kumar Vaishnavi Harshithaa Karunyah
Sprint-2	Web Access	USN-7	As a user, I can access the website to predict the turbine power	7	High	Varun Kumar Vaishnavi Harshithaa Karunyah
Sprint-2	Prediction	USN-8	As a customer, when I enter the detail the website should predict the approximate turbine power	7	High	Varun Kumar Vaishnavi Harshithaa Karunyah
Sprint-3	Analysis	USN-9	As a customer, I wish to store my predictions and make analysis	7	Medium	Varun Kumar Vaishnavi Harshithaa Karunyah
Sprint-3	Security	USN-10	As a customer I expect my data to be secured	7	Medium	Varun Kumar Vaishnavi Harshithaa Karunyah

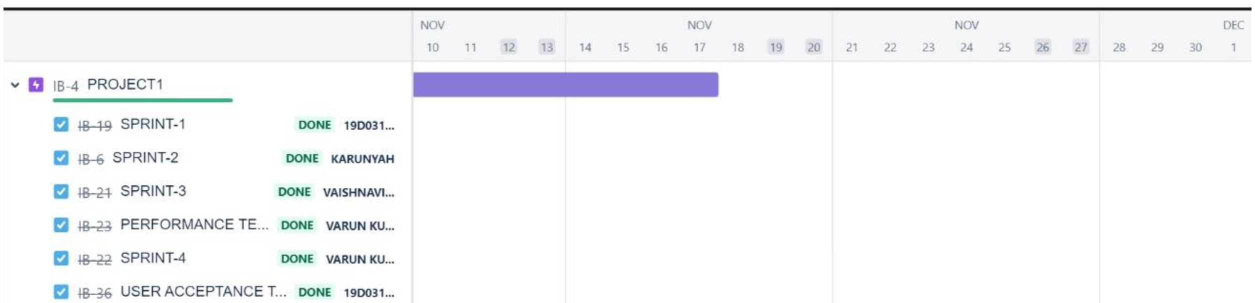
Sprint-4	Database Access	USN-11	As an administrator , I should maintain the website and keep updating it regularly	20	Medium	Varun Kumar Vaishnavi Harshithaa Karunyah

6.2 Sprint Delivery Schedule

SPRINTS	SUB-SPRINTS	CONTENT
Sprint-1	Sprint-1	1. Application Building • Download or Create Dataset
	Sprint-2	2. Data Pre-processing • Import Required Libraries •Analyze the datasets •Splitting the data into Independent and dependent variables
Sprint-2	Sprint-3	3. Application Building • Flask App
	Sprint-4	4. Model Building • Choose the appropriate model • Check the metrics of the model •Save the model
Sprint-3	Sprint-5	5. API Integration • Generate Weather API key

	Sprint-6	6. Application Building • Html pages • Testing
Sprint-4	Sprint-7	7. Train The Model On IBM • Register For IBM Cloud
	Sprint -8	8. Train The Model On IBM • Train The ML Model On IBM • Integrate Flask With Scoring End Point

6.3 Reports from JIRA



7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

HTML:

The most fundamental component of the Web is HTML (HyperText Markup Language). It is a text-based approach which describes the purpose and organization of web content. The markup language tells a web browser how to display text images and other forms of multimedia on a webpage. Links that join online pages together, either inside a single website or between websites, are referred to as "hypertext." An essential component of the Web are links. It is easy to learn and use, faster approach and allows storage of big files.

Web Framework:

Web application developers can write applications without having to be concerned about low-level details like protocol, thread management, and other issues thanks to a web framework, also known as a web application framework.

Flask Python:

Flask is a web application framework written in Python. It was developed by Armin Ronacher, who led a team of international Python enthusiasts called Poocco. Flask is based on the Werkzeg WSGI toolkit and the Jinja2 template engine.Both are Pocco projects. This provides useful tools and features for creating web applications in the Python Language. It gives developers flexibility and is an accessible framework for new developers because you can build a web application quickly using only a single Python file.

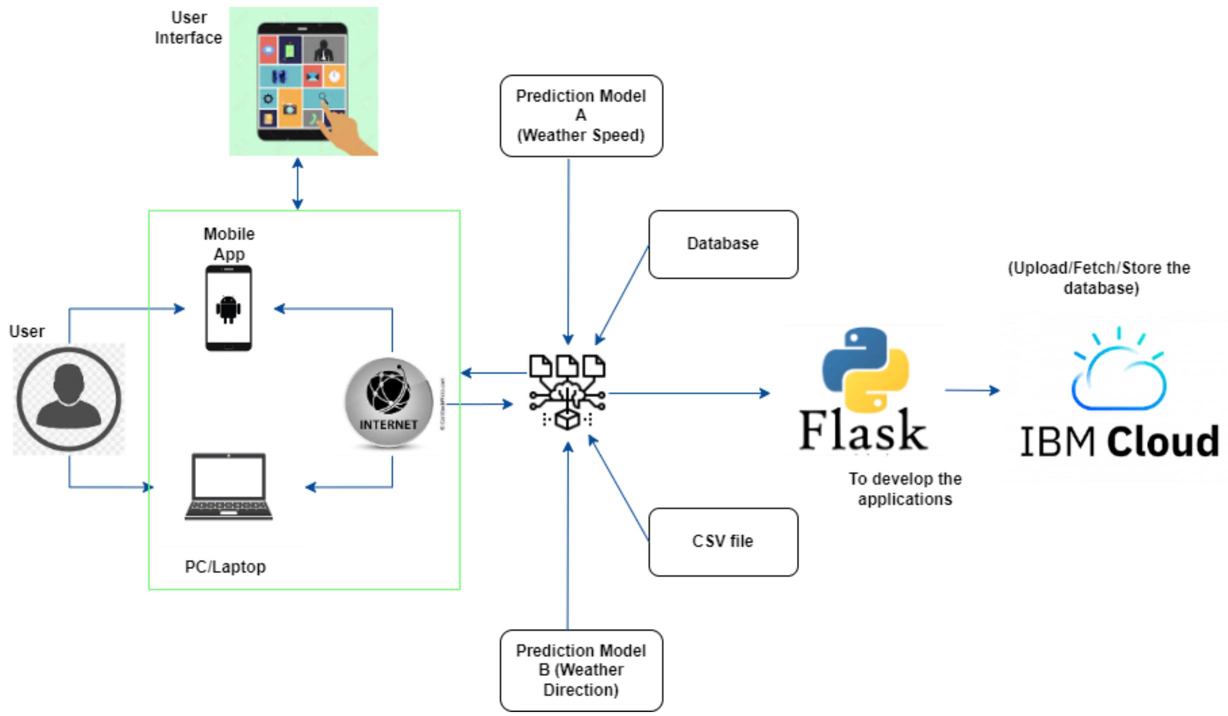
62 lines (54 sloc) | 2.44 KB

Raw | Blame

```
import numpy as np
from flask import Flask, request, jsonify, render_template
import joblib
import requests

# IBM Cloud account Credentials.
API_KEY = "58ahhsqevpUY0Eu1YKv5Kyl38OMCy3haa5WCXw8am_wL"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey": API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'}
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
app = Flask(__name__)
# model = joblib.load('Power_Prediction.sav')
@app.route('/')
def home():
    return render_template('index.html')
@app.route('/predict')
def predict():
    return render_template('predict.html')
@app.route('/windapi',methods=['POST'])
def windapi():
    city=request.form.get('city')
    apikey="a882b0f626c637d04185e582b5ad0d58"
    url="http://api.openweathermap.org/data/2.5/weather?q="+city+"&appid="+apikey
    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/s"
    return render_template('predict.html', temp=temp, humid=humid, pressure=pressure,speed=speed)
@app.route('/y_predict',methods=['POST'])
def y_predict():
    '''
    For rendering results on HTML GUI
    '''
    x_test = [[float(x) for x in request.form.values()]]
    print(x_test)
    # Deployed Link
    payload_scoring = {"input_data":
                       [{"field": [["Theoretical_Power", "Wind_Speed"]],
                       "values": x_test}]}

    response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/373bfd45-d75f-4f2a-9c9c-a94decf0ec1f/predictions?version=2022-11-1', json
    print("Scoring response")
    predictions =response_scoring.json()
    print(predictions)
    print('Final Prediction Result',predictions['predictions'][0]['values'][0][0])


    pred =response_scoring.json()
    print(pred)
    #print('Final Prediction Result',predictions['predictions'][0]['values'][0][0])

   # prediction = model.predict(x_test)
    print(pred)
    output = pred['predictions'][0]['values'][0][0]
    return render_template('predict.html', prediction_text='The energy predicted is {:.2f} KWh'.format(output))


if __name__ == "__main__":
    app.run(debug=False, port=8080)
```

IBM Cloud:

IBM Cloud is a suite of cloud computing services from IBM that offers both platform as a service (PaaS) and infrastructure as a service (IaaS). With IBM Cloud IaaS, organizations can deploy and access virtualized IT resources such as compute power, storage and networking over the internet



7.2 Feature 2

```
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('https://raw.githubusercontent.com/IBM-EPBL/IBM-Project-26990-1660042752/main/Data%20collection/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()
```

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	day	Hour	minute
0	2018-01-01 00:00:00	380.047791	5.311336	416.328908	259.994904	2018	1	1	0	0
1	2018-01-01 00:10:00	453.769196	5.672167	519.917511	268.641113	2018	1	1	0	10
2	2018-01-01 00:20:00	306.376587	5.216037	390.900016	272.564789	2018	1	1	0	20
3	2018-01-01 00:30:00	419.645905	5.659674	516.127569	271.258087	2018	1	1	0	30
4	2018-01-01 00:40:00	380.650696	5.577941	491.702972	265.674286	2018	1	1	0	40

```
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	month	day	Hour	minute
0	2018-01-01 00:00:00	380.047791	5.311336	416.328908	259.994904	2018	1	1	0	0
1	2018-01-01 00:10:00	453.769196	5.672167	519.917511	268.641113	2018	1	1	0	10
2	2018-01-01 00:20:00	306.376587	5.216037	390.900016	272.564789	2018	1	1	0	20
3	2018-01-01 00:30:00	419.645905	5.659674	516.127569	271.258087	2018	1	1	0	30
4	2018-01-01 00:40:00	380.650696	5.577941	491.702972	265.674286	2018	1	1	0	40
...	...	...	...	...	...	...	...	...	...	...
50525	2018-12-31 23:10:00	2963.980957	11.404030	3397.190793	80.502724	2018	12	31	23	10
50526	2018-12-31 23:20:00	1684.353027	7.332648	1173.055771	84.062599	2018	12	31	23	20
50527	2018-12-31 23:30:00	2201.106934	8.435358	1788.284755	84.742500	2018	12	31	23	30
50528	2018-12-31 23:40:00	2515.694092	9.421366	2418.382503	84.297913	2018	12	31	23	40
50529	2018-12-31 23:50:00	2820.466064	9.979332	2779.184096	82.274620	2018	12	31	23	50

50530 rows × 10 columns

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
2	5.216037	272.564789
3	5.659674	271.258087
4	5.577941	265.674286

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

```
0    380.047791
1    453.769196
2    306.376587
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)
```

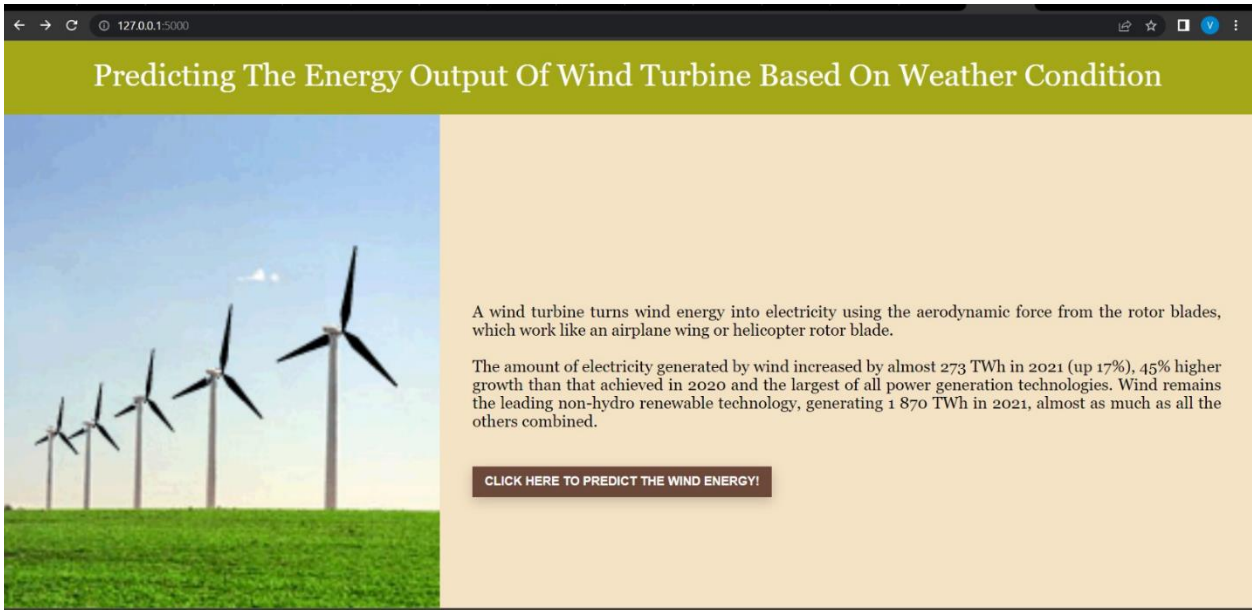
```
[18:30:25] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated in favor of reg:squarederror.
```

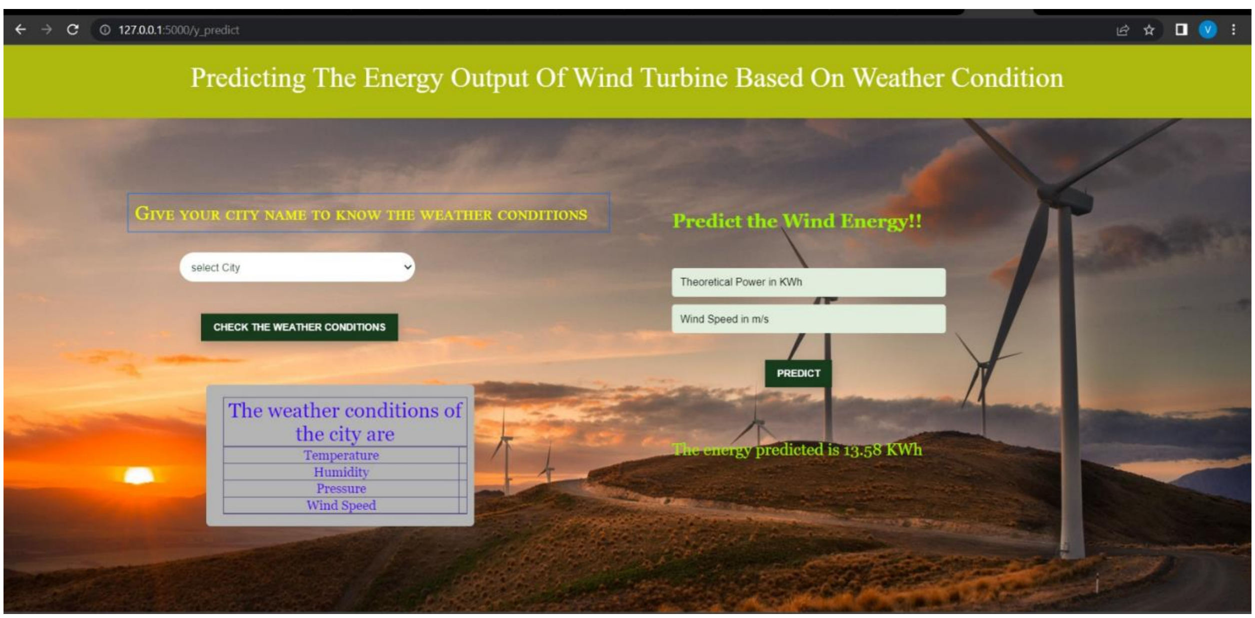
Checking the metrics

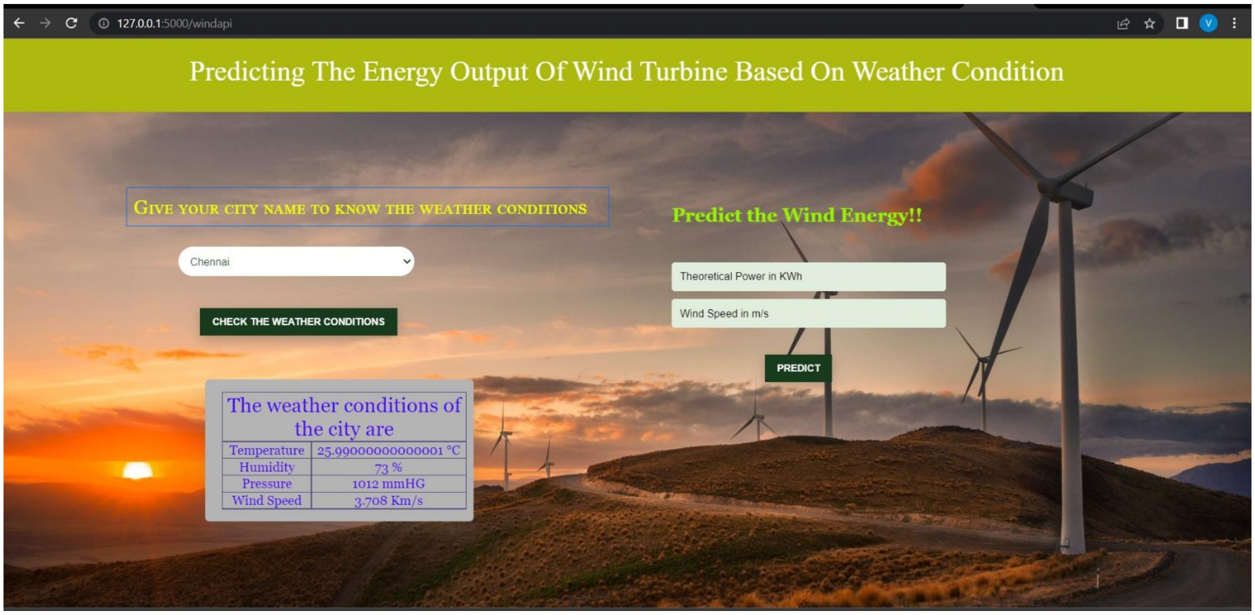
```
print('R2-xgb',r2_score(y_test,y_xg))
print('RMSE-xgb',np.sqrt(mean_squared_error(y_test,y_xg)))

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)))
```

8.TESTING

8.1 Test Cases







8.2 User Acceptance Testing

Defect Analysis:

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	2	5
Client Application	51	0	0	51
Security	2	0	0	2

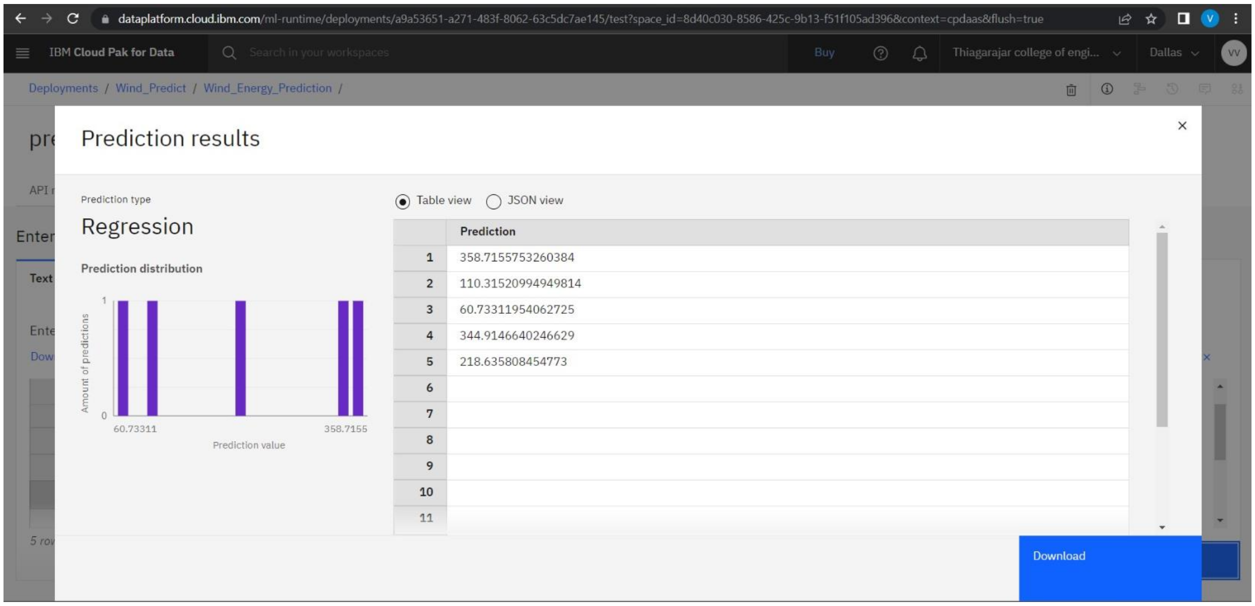
they were resolved

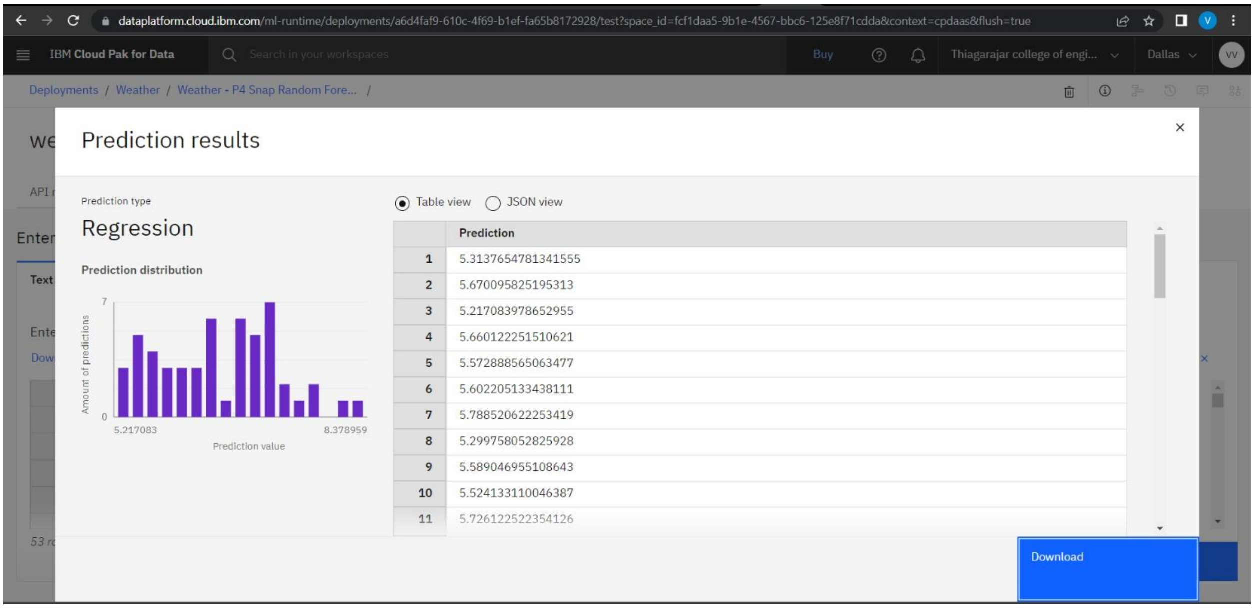
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	6	4	2	7	19
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	20	10	5	26	61
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	1	0	0	1
Totals	29	18	12	35	94

Test Case Analysis:

Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	6	0	0	6
Version Control	2	0	0	2

9. RESULTS


Prediction results
Prediction type
Regression
Prediction distribution
Table view
JSON view
Prediction
358.7155753260384
110.31520994949814
60.73311954062725
344.9146640246629
218.635808454773
Download


Prediction results
Prediction type
Regression
Prediction distribution
Table view
JSON view
Prediction
5.3137654781341555
5.670095825195313
5.217083978652955
5.660122251510621
5.572888565063477
5.602205133438111
5.788520622253419
5.299758052825928
5.589046955108643
5.524133110046387
5.726122522354126
Download

Statistical analysis:

```
data.mean()
```

```
/tmp/wsuser/ipykernel_164/531903386.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select only valid columns before calling the reduction.
  data.mean()
```

```
Active_Power         1307.684332
Wind_Speed              7.557952
Theoretical_Power    1492.175463
Wind_Direction        123.687559
dtype: float64
```

```
data.median()
```

```
/tmp/wsuser/ipykernel_164/4184645713.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select only valid columns before calling the reduction.
  data.median()
```

```
Active_Power          825.838074
Wind_Speed              7.104594
Theoretical_Power    1063.776283
Wind_Direction         73.712978
dtype: float64
```

```
data.mode()
```

	Date	Active_Power	Wind_Speed	Theoretical_Power	Wind_Direction
0	01 01 2018 00:00	0.0	0.0	0.0	0.0
1	01 01 2018 00:10	NaN	NaN	NaN	NaN
2	01 01 2018 00:20	NaN	NaN	NaN	NaN
3	01 01 2018 00:30	NaN	NaN	NaN	NaN
4	01 01 2018 00:40	NaN	NaN	NaN	NaN
...	...	...	...	...	...
50525	31 12 2018 23:10	NaN	NaN	NaN	NaN
50526	31 12 2018 23:20	NaN	NaN	NaN	NaN
50527	31 12 2018 23:30	NaN	NaN	NaN	NaN
50528	31 12 2018 23:40	NaN	NaN	NaN	NaN
50529	31 12 2018 23:50	NaN	NaN	NaN	NaN

50530 rows × 5 columns

```
data.var()
```

```
/tmp/wsuser/ipykernel_164/445316826.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select only valid columns before calling the reduction.
  data.var()
```

```
Active_Power         1.722549e+06
Wind_Speed           1.786893e+01
Theoretical_Power    1.871474e+06
Wind_Direction       8.731732e+03
dtype: float64
```

10. ADVANTAGES & DISADVANTAGES

Advantages:

- The energy output of a wind farm is highly dependent on the weather conditions present in the area. If the output can be predicted more accurately, energy suppliers can coordinate the collaborative production of different energy sources more efficiently to avoid wastage and to reduce the cost spent on conserving the energy.
- It is a very clean energy source, which does not release any pollution or produce any waste during operation
- Since wind turbines themselves run strictly on the power generated by wind, there is no use of extra fuel.
- Generating energy from the wind does not release any carbon emissions. By replacing electricity generated from other sources such as fossil fuel power stations, wind energy can lead to an overall reduction in carbon emissions.
- Use of latest technologies with which early prediction of wind energy becomes faster and more effective.
- Reduction in overall production cost and easy to find alternative in bad weather conditions by predicting them earlier.

Disadvantages:

- Wind turbines in general may affect wildlife especially birds
- Cause of noise pollution
- Expensive to construct wind farms in places where wind energy is high throughout the years.
- Due to global warming, climatic changes have become very common in every places. Hence, predictions can be unreliable sometimes.

11. CONCLUSION

With the growing population, it is difficult to reduce the consumption of energy resources. It becomes a necessity to use the available renewable energy in every way possible without wasting them and also to conserve them so as to use those energies when needed. Wind energy is available in every part of the world. But the important constraint is to find those regions which have a higher wind energy range and also those regions where wind turbines can be placed as it requires a large area. Nowadays Climatic changes has become a common problem in all regions, hence, it is important to find the weather conditions in all regions where the turbines are placed. In this project, we have proposed a prediction model which analysis the weather condition in a region and with the information provided, we can conserve wind energy which is further used for generating electricity. We made use of technologies like python flask, HTML, IBM cloud (Watson studio), machine learning and cloud storage.

12. FUTURE SCOPE

We developed an application which can forecast the wind power of the future leveraging AI tools and powerful visualization. The advantage of using ours over others is, it provides an Application with interactive UI with optimized model with higher accuracy to predict the energy output of the wind energy's accurate parameters. It is important and easy for a user to browse with the help of AI and analyse the Wind Speed, Wind Direction and Power Output of the flow of the wind. Helpful for people who want to browse the energy outputs of the wind turbine.

13. APPENDIX

Source Code

HTML Introduction code:

github.com/IBM-EPBL/IBM-Project-26990-1660042752/blob/main/Project%20development%20phase/Sprint-4/Train%20model%20on%20ibm%20cloud/Flask%20App%20Integration%2...

84 lines (79 sloc) | 2.19 KB — Raw | Blame

```
<html>
        <head>
        <title>Wind Energy Prediction</title>
        <style>

                .header {
                        top:0px;
                        margin:0px;
                        left: 0px;
                        right: 0px;
                        position: fixed;
                        background: #a4a717;
                        color: rgb(255, 255, 255);
                        overflow: hidden;
                        padding-bottom: 30px;
                        font-family:Georgia, 'Times New Roman', Times, serif, serif;
                        font-size: 2.5vw;
                        width: 100%;
                        padding-left:0px;
                        text-align: center;
                        padding-top:20px;
                }
                .second{
                        top:90px;
                        bottom:0px;
                        margin:0px;
                        left: 0px;
                        right: 0px;
                        position: fixed;
                        padding: 0px;
                        width: 100%;
                        background-image:url(https://i.pinimg.com/originals/c4/d2/f9/c4d2f98e88a85b702f8ff257d74714d8.gif);
                        background-repeat:no-repeat;
                        background-size: contain;
                }
                .inside{
                        top:90px;
                        bottom:0px;
```

github.com/IBM-EPBL/IBM-Project-26990-1660042752/blob/main/Project%20development%20phase/Sprint-4/Train%20model%20on%20ibm%20cloud/Flask%20App%20Integration%2...

```
                .inside{
                        top:90px;
                        bottom:0px;
                        margin:0px;
                        left: 35%;
                        right: 0%;
                        position: fixed;
                        padding-left: 40px;
                        padding-top:15%;
                        padding-right:40px;
                        background-color:#f5e3c5;
                        opacity: 100%;
                        font-family:Georgia, serif;
                        color:black;
                        font-size:20px;
                        text-align:justify;

                }
                .myButton{
                          border: none;
                          text-align: center;
                          cursor: pointer;
                          text-transform: uppercase;
                          outline: none;
                          overflow: hidden;
                          color: #fff;
                          font-weight: 700;
                          font-size: 15px;
                          background-color: #6c493a;
                          padding: 10px 15px;
                          margin: 0 auto;
                          box-shadow: 0 5px 15px rgba(0,0,0,0.20);
                }
        </style>
        </head>
        <body>

                <div class="header">Predicting The Energy Output Of Wind Turbine Based On Weather Condition</div>
                <div class="second">
                        <div class="inside">A wind turbine turns wind energy into electricity using the aerodynamic force from the rotor blades, which work like an airplane wi
                                The amount of electricity generated by wind increased by almost 273 TWh in 2021 (up 17%), 45% higher growth than that achieved in 2020 and the
                <br><br><br>
                <a href="{{url_for('predict')}}"><button type="button" class="myButton" >Click Here To Predict The wind Energy!</button></a>
                </div>


                </div>
        </body>
</html>
```

Prediction Code:

github.com/IBM-EPBL/IBM-Project-26990-1660042752/blob/main/Project%20development%20phase/Sprint-4/Train%20model%20on%20ibm%20cloud/Flask%20App%20Integration%2...

290 lines (252 sloc) | 7.66 KB

Raw Blame

```
<html>
        <head>
                <meta charset="UTF-8" />
                <meta name="viewport" content="width=device-width, initial-scale=1.0" />
                <meta http-equiv="X-UA-Compatible" content="ie=edge" />
                <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.2/css/all.css"
                        integrity="sha384-fnmOCqbTlWIlj8LyTjo7mOUStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr" crossorigin="anonymous" />
                <link href="https://fonts.googleapis.com/css?family=Dosis" rel="stylesheet" />
                <link rel="stylesheet" href="static/css/main.css"/>
                <link rel="stylesheet" href="static/css/media.css"/>
                <link rel="stylesheet" href="static/css/items_grid.css"/>



        <title>Wind Energy Prediction</title>
        <style>
#page {
  max-width: 80%;
  margin: auto;
}
body {
        background-image: url(https://images2.alphacoders.com/753/753985.jpg);
        width: 100%;
        height: 100%;
        background-repeat: no-repeat;
        background-attachment: fixed;
        background-size: cover;
        overflow: hidden;

}
table {
                width: 100%;
                border-collapse: collapse;
}

.card {
            margin-right: auto;
            margin-left: 15%;
            width: 300px;
            box-shadow: 0 15px 25px rgba(129, 124, 124, 0.2);

            border-radius: 5px;
            backdrop-filter: blur(14px);
            background-color: rgb(180, 180, 180);
            padding: 15px;
            text-align: center;
        }
```

github.com/IBM-EPBL/IBM-Project-26990-1660042752/blob/main/Project%20development%20phase/Sprint-4/Train%20model%20on%20ibm%20cloud/Flask%20App%20Integration%2...

```
        }
                .head {
                        top:0px;
                        margin:0px;
                        left: 0px;
                        right: 0px;
                        position: fixed;
                        background: #aeb90f;
                        color: white;
                        overflow: hidden;
                        padding-bottom: 30px;
                        font-size: 2.25vw;
                        width: 100%;
                        padding-left:0px;
                        text-align: center;
                        padding-top:20px;
                }
                .second{
                        top:80px;
                        bottom:0px;
                        margin:0px;
                        left: 0px;
                        right: 0px;
                        position: fixed;
                        padding: 0px;
                        width: 100%;

                        font-family:Georgia, serif;
                        color:black;
                        font-size:20px;
                }
                .inside{
                        top:80px;
                        bottom:0px;
                        margin:0px;
                        left: 51%;
                        right: 0%;
                        position: fixed;
                        padding-left: 40px;
                        padding-top:8%;
                        padding-right:40px;

                        font-family:Georgia, serif;
                        color:#96f400;
                        font-size:20px;
                        text-align:justify;

                }
                .myButton{
```

github.com/IBM-EPBL/IBM-Project-26990-1660042752/blob/main/Project%20development%20phase/Sprint-4/Train%20model%20on%20ibm%20cloud/Flask%20App%20Integration%2...

```
                .myButton{
                        border: none;
                        text-align: center;
                        cursor: pointer;
                        text-transform: uppercase;
                        outline: none;
                        overflow: hidden;
                        color: #fff;
                        font-weight: 700;
                        font-size: 12px;
                        background-color: #183a1d;
                        padding: 10px 15px;
                        margin: 0 auto;
                        box-shadow: 0 5px 15px rgba(0,0,0,0.20);
                        margin-left:17%;
                }
        input {
        width:50%;
        margin-bottom: 10px;
        background: #e1eedd;
        border: none;
        outline: none;
        padding: 10px;
        font-size: 13px;
        color: #6c493a;
        text-shadow: white;
        border: #6c493a;
        border-radius: 4px;
        box-shadow: white;
}
::placeholder {
  color: black;
  opacity: 1;
}

.left{
                top:80px;
                        bottom:0px;
                        margin:0px;
                        left: 0%;
                        right: 45.5%;
                        position: fixed;
                        padding-left: 10%;
                        padding-top:5%;
                        padding-right:40px;

                        font-family:bold,Georgia, serif;
                        color:rgb(238, 255, 0);;
                        font-size:25px;
```

github.com/IBM-EPBL/IBM-Project-26990-1660042752/blob/main/Project%20development%20phase/Sprint-4/Train%20model%20on%20ibm%20cloud/Flask%20App%20Integration%2...

```
                        font-size:25px;

}
        select {
        width:50%;
        margin-bottom: 10px;
        background: white;
        border: none;
        outline: none;
        padding: 10px;
        font-size: 13px;
        color: #183a1d;
        text-shadow: white;
        border: #6c493a;
        border-radius: 40px;
        box-shadow: white;
}

input:focus { box-shadow: inset 0 -5px 45px rgba(100,100,100,0.4), 0 1px 1px rgba(255,255,255,0.2); }



                table, th, td {
  border: 1px solid rgb(86, 72, 128);
  border-collapse: collapse;
  color: #3f00ff;
}

@media screen and (max-width: 500px) {

  .left,
  .second,
  .third {
          width: 70%;
  }
}



        </style>
        </head>
        <body>


                <header id="head">
                <div class="head">Predicting The Energy Output Of Wind Turbine Based On Weather Condition</div>
                </header>


```

github.com/IBM-EPBL/IBM-Project-26990-1660042752/blob/main/Project%20development%20phase/Sprint-4/Train%20model%20on%20ibm%20cloud/Flask%20App%20Integration%2...

```
            <div class="second">
            <div class="left">

                <p style="padding: 8px; border: 1px solid rgb(0, 110, 255); width: 575px;"> GIVE YOUR CITY NAME TO KNOW THE WEATHER CONDITIONS</p>

            <div style="margin-left:10%">
            <form action="{{ url_for('windapi')}}"method="post" >
                    <select name="city"  required >
                        <option value="" selected>select City</option>
                        <option value ="Ariyalur"       >       Ariyalur        </option>
                        <option value ="Andimadam"      >       Andimadam       </option>
                        <option value ="Coimbatore"     >       Coimbatore      </option>
                        <option value ="Chengalpattu"   >       Chengalpattu    </option>
                        <option value ="Cuddalore"      >       Cuddalore       </option>
                        <option value ="Chennai"        >       Chennai </option>
                        <option value ="Dindigul"       >       Dindigul        </option>
                        <option value ="Dharmapuri"     >       Dharmapuri      </option>
                        <option value ="Erode"  >       Erode   </option>
                        <option value ="Karur"  >       Karur   </option>
                        <option value ="Kancheepuram"   >       Kancheepuram    </option>
                        <option value ="Krishnagiri"    >       Krishnagiri     </option>
                        <option value ="Kallakurichi"   >       Kallakurichi    </option>
                        <option value ="Madurai"        >       Madurai </option>
                        <option value ="Mayiladuthurai" >       Mayiladuthurai  </option>
                        <option value ="Nagapattinam"   >       Nagapattinam    </option>
                        <option value ="Kanyakumari"    >       Kanyakumari     </option>
                        <option value ="Namakkal"       >       Namakkal        </option>
                        <option value ="Perambalur"     >       Perambalur      </option>
                        <option value ="Pudukottai"     >       Pudukottai      </option>
                        <option value ="Ramanathapuram" >       Ramanathapuram  </option>
                        <option value ="Ranipet"        >       Ranipet </option>
                        <option value ="Salem"  >       Salem   </option>
                        <option value ="Sivagangai"     >       Sivagangai      </option>
                        <option value ="Tenkasi"        >       Tenkasi </option>
                        <option value ="Thanjavur"      >       Thanjavur       </option>
                        <option value ="Theni"  >       Theni   </option>
                        <option value ="Thiruvallur"    >       Thiruvallur     </option>
                        <option value ="Thiruvarur"     >       Thiruvarur      </option>
                        <option value ="Tuticorin"      >       Tuticorin       </option>
                        <option value ="Trichirapalli"  >       Trichirapalli   </option>
                        <option value ="Thirunelveli"   >       Thirunelveli    </option>
                        <option value ="Tirupathur"     >       Tirupathur      </option>
                        <option value ="Tiruppur"       >       Tiruppur        </option>
                        <option value ="Tiruvannamalai" >       Tiruvannamalai  </option>
                        <option value ="The Nilgiris"   >       The Nilgiris    </option>
                        <option value ="Vellore"        >       Vellore </option>
                        <option value ="Viluppuram"     >       Viluppuram      </option>
                        <option value ="Virudhunagar"   >       Virudhunagar    </option>

```

github.com/IBM-EPBL/IBM-Project-26990-1660042752/blob/main/Project%20development%20phase/Sprint-4/Train%20model%20on%20ibm%20cloud/Flask%20App%20Integration%2...

```

                    </select><br><br>
                    <div style="margin-left:-15%"><button type="submit" class="myButton" >Check the Weather Conditions</button></div>

                </form>
                </div>
                <br>

                <div class="card">
                <table style="margin-left:2%; text-align:center; border-spacing:20px;">
                    <tr>
                        <td colspan="2" style="font-size:25px;">The weather conditions of the city are</td>
                    </tr>
                    <tr>
                        <td>Temperature</td><td>{{temp}}</td>
                    </tr>
                    <tr>
                        <td>Humidity</td><td>{{humid}}</td>
                    </tr>
                    <tr>
                        <td>Pressure</td><td>{{pressure}}</td>
                    </tr>
                    <tr>
                        <td>Wind Speed</td><td>{{speed}}</td>
                    </tr>
                </table>
                </div>
            </div>
                <div class="inside">
                <div style="font-size:23px;font-weight:bold;">Predict the Wind Energy!!</div>
                <br><br>
                <form action="{{ url_for('y_predict')}}"method="post">
                    <input type="text" name="theo" placeholder="Theoretical Power in KWh" required="required"  />
                    <input type="text" name="wind" placeholder="Wind Speed in m/s" required="required"  /><br><br>
                    <button type="submit" class="myButton" >Predict</button>

                </form>

              <br>
              <br>
              {{ prediction_text }}

            </div>


            </div>

        </body>

</html>
```

Python- Application:

github.com/IBM-EPBL/IBM-Project-26990-1660042752/blob/main/Project%20development%20phase/Sprint-4/Train%20model%20on%20ibm%20cloud/Flask%20App%20Integration%2...

62 lines (53 sloc) | 2.41 KB

Raw Blame

```python
import numpy as np
from flask import Flask, request, jsonify, render_template
import joblib
import requests


# IBM Cloud Integration with Credentials
API_KEY = "ae56da0f-f4b2-4aa5-9480-6baa88746ac3"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey": API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
app = Flask(__name__)
# model = joblib.load('Power_Prediction.sav')
@app.route('/')
def home():
    return render_template('intro.html')
@app.route('/predict')
def predict():
    return render_template('predict.html')
@app.route('/windapi',methods=['POST'])
def windapi():
    city=request.form.get('city')
    apikey="a802b0f626c637d04185e582b5ad0d58"
    url="http://api.openweathermap.org/data/2.5/weather?q="+city+"&appid="+apikey
    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/s"
    return render_template('predict.html', temp=temp, humid=humid, pressure=pressure,speed=speed)
@app.route('/y_predict',methods=['POST'])
def y_predict():
    '''
    For rendering results on HTML GUI
    '''
    x_test = [[float(x) for x in request.form.values()]]
    print(x test)
```

github.com/IBM-EPBL/IBM-Project-26990-1660042752/blob/main/Project%20development%20phase/Sprint-4/Train%20model%20on%20ibm%20cloud/Flask%20App%20Integration%2...

```python
    pressure = str(resp["main"]["pressure"])+" mmHG"
    speed = str((resp["wind"]["speed"])*3.6)+" Km/s"
    return render_template('predict.html', temp=temp, humid=humid, pressure=pressure,speed=speed)
@app.route('/y_predict',methods=['POST'])
def y_predict():
    '''
    For rendering results on HTML GUI
    '''
    x_test = [[float(x) for x in request.form.values()]]
    print(x_test)
    payload_scoring = {"input_data":
                      [{"field": [["Theoretical_Power", "Wind_Speed"]],
                      "values": x_test}]}

    response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/1a772765-e95e-4920-a46b-25ede8ef1b44/predictions?version=2022-11-06', json=payload_sc
    print("Scoring response")
    predictions =response_scoring.json()
    print(predictions)
    print('Final Prediction Result',predictions['predictions'][0]['values'][0][0])


    pred =response_scoring.json()
    print(pred)
    #print('Final Prediction Result',predictions['predictions'][0]['values'][0][0])

   # prediction = model.predict(x_test)
    print(pred)
    output = pred['predictions'][0]['values'][0][0]
    return render_template('predict.html', prediction_text='The energy predicted is {:.2f} KWh'.format(output))


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

IBM API KEY:

{
    "name": "Wind_pred",
    "description": "",
    "createdAt": "2022-11-18T11:33+0000",
    "apikey": "HzrV2Q9Ywg3EyxMO14u62Meo2RhdZjp6np6AeCoq8QlR"
}

Weather API KEY:

a802b0f626c637d04185e582b5ad0d58

GitHub & Project Demo Link

GitHub Link : https://github.com/IBM-EPBL/IBM-Project-26990-1660042752.git

Youtube Link : https://youtu.be/DFsfgR3Z90s