

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

1.2 PURPOSE

2 LITERATURE SURVEY

2.1 EXISTING PROBLEM

2.2 REFERENCES

2.3 PROBLEM STATEMENT DEFINITION

3 IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

3.2 IDEATION & BRAINSTORMING

3.3 PROPOSED SOLUTION

3.4 PROBLEM SOLUTION FIT

4 REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

4.2 NON FUNCTIONAL REQUIREMENTS

5 PROJECT DESIGN

5.1 DATA FLOW DIAGRAM

5.2 SOLUTION & TECHNICAL ARCHITECTURE

5.3 USER STORIES

6 PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

6.2 SPRINT DELIVERY SCHEDULE

7 CODING & SOLUTIONING

8 TESTING

8.1 TEST CASES

8.2 USER ACCEPTANCE TESTING

8.2.1 DEFECT ANALYSIS

8.2.2 TEST CASE ANALYSIS

9 RESULTS

9.1 PERFORMANCE METRICS

10 ADVANTAGES & DISADVANTAGES

10.1 ADVANTAGES

10.2 DISADVANTAGES

11 CONCLUSION

12 FUTURE SCOPE

APPENDIX

SOURCE CODE

GITHUB

PROJECT DEMO

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 came 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 be 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.

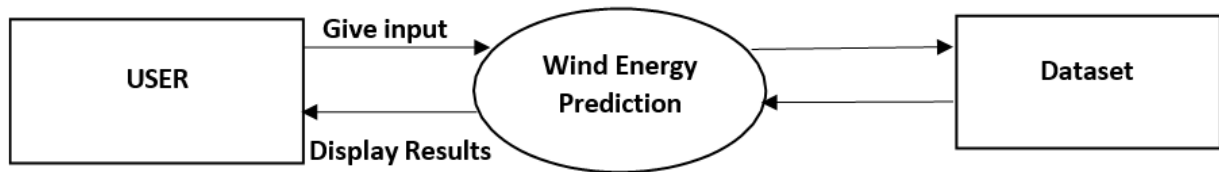
4.2 Non-Functional requirements

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.

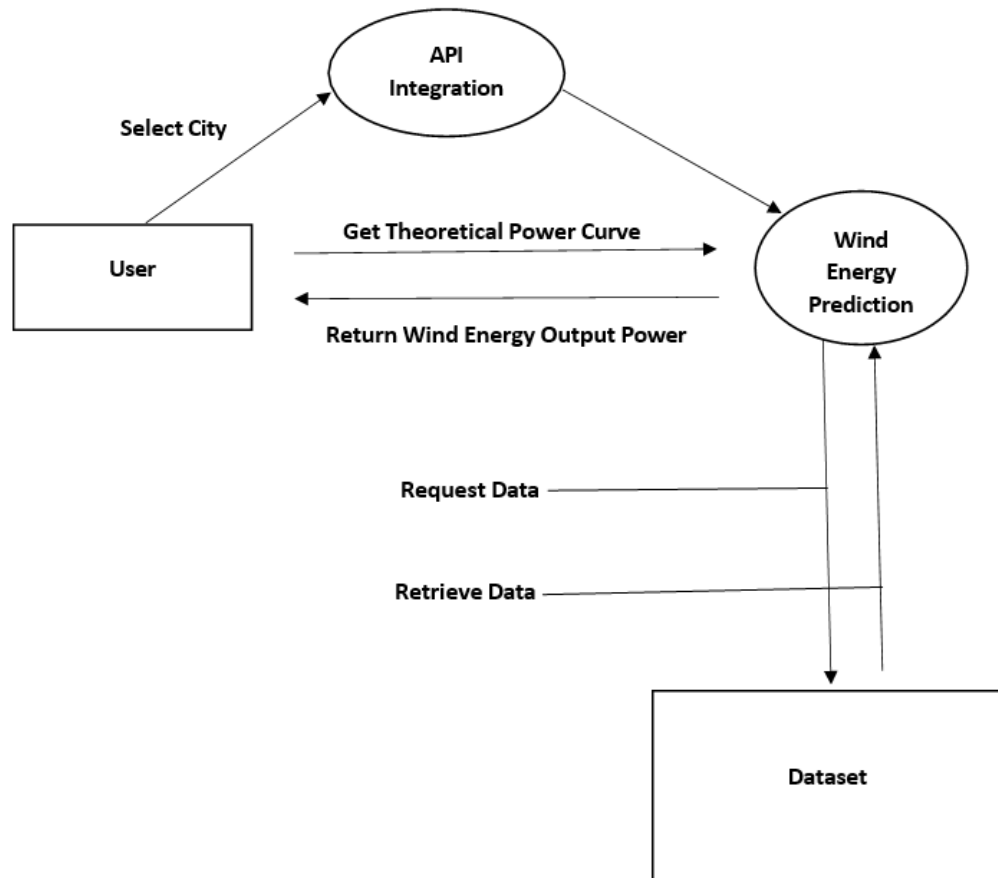
5. PROJECT DESIGN

5.1 Data Flow Diagrams

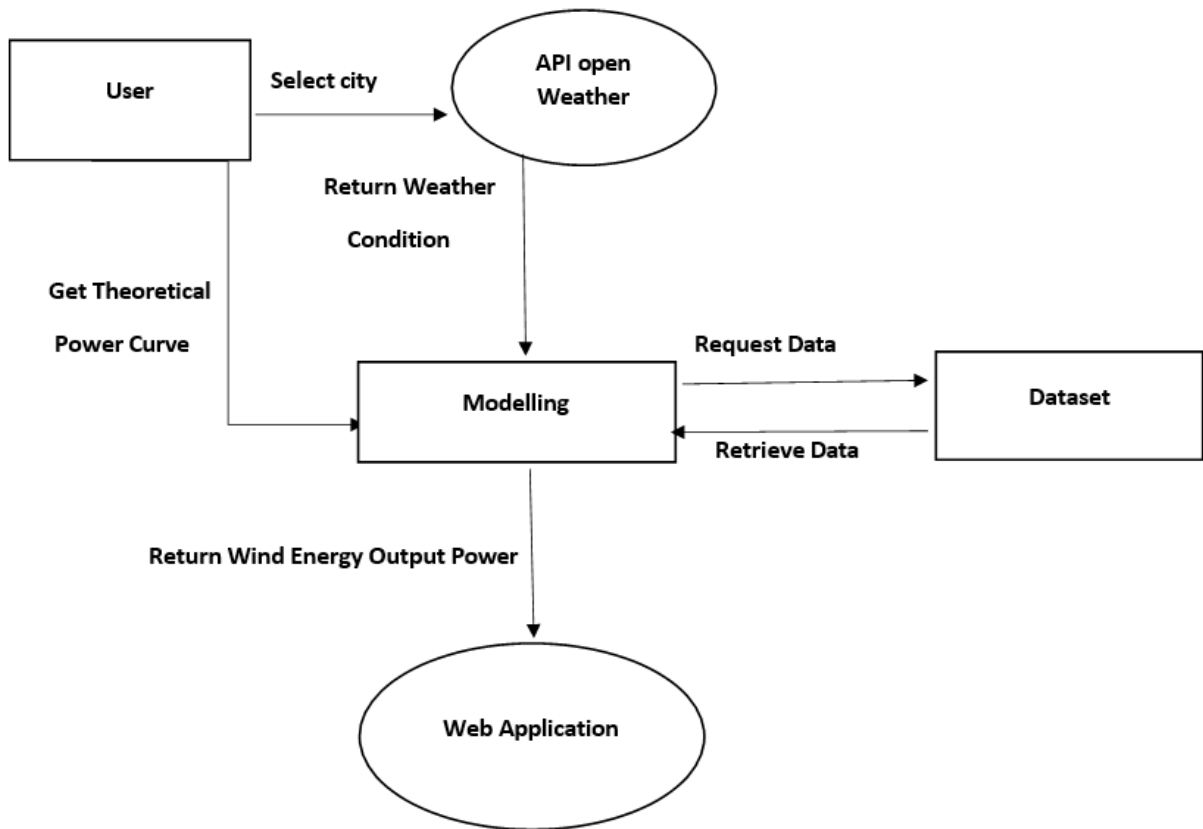
LEVEL 0:



LEVEL 1:

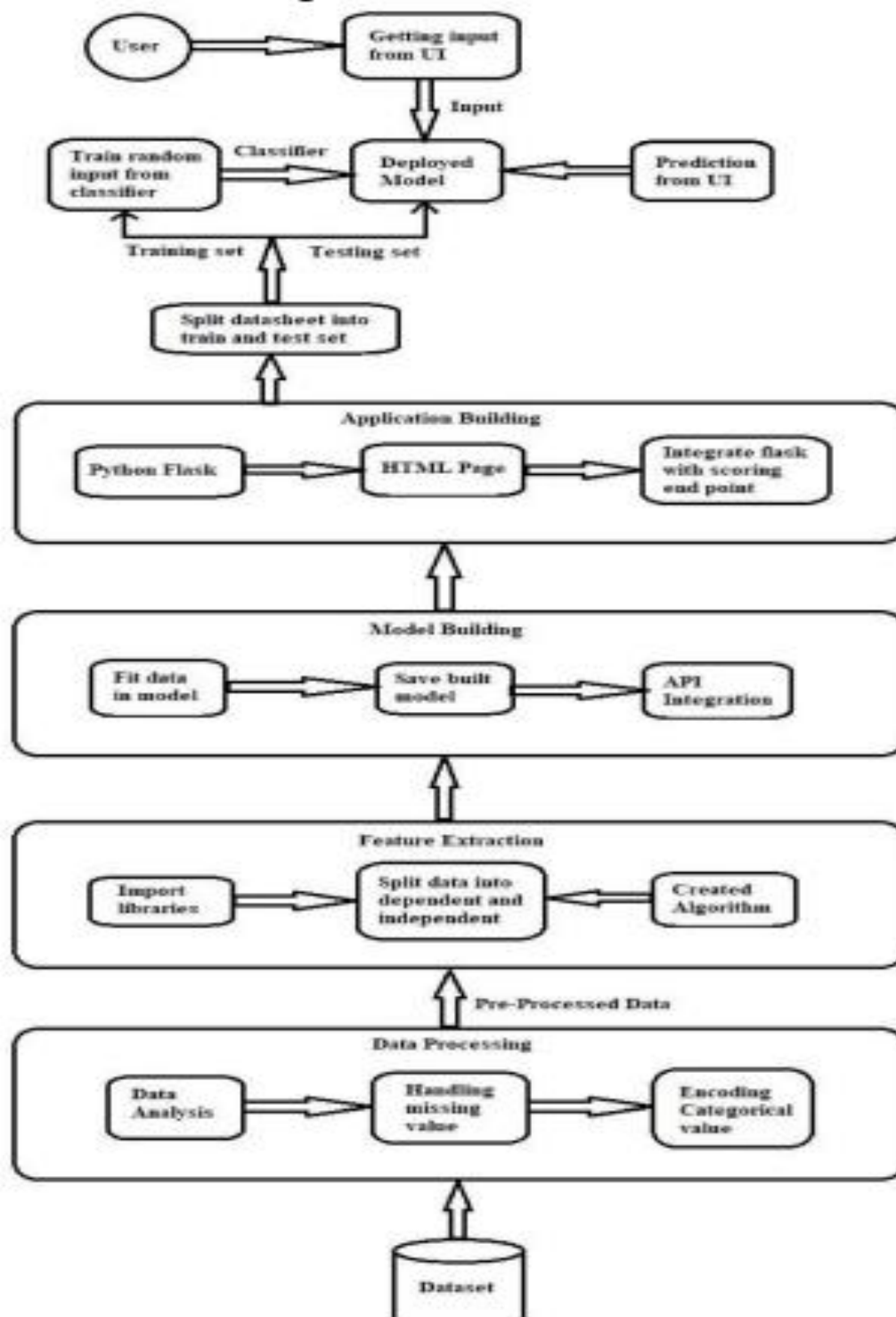


LEVEL 2:



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.

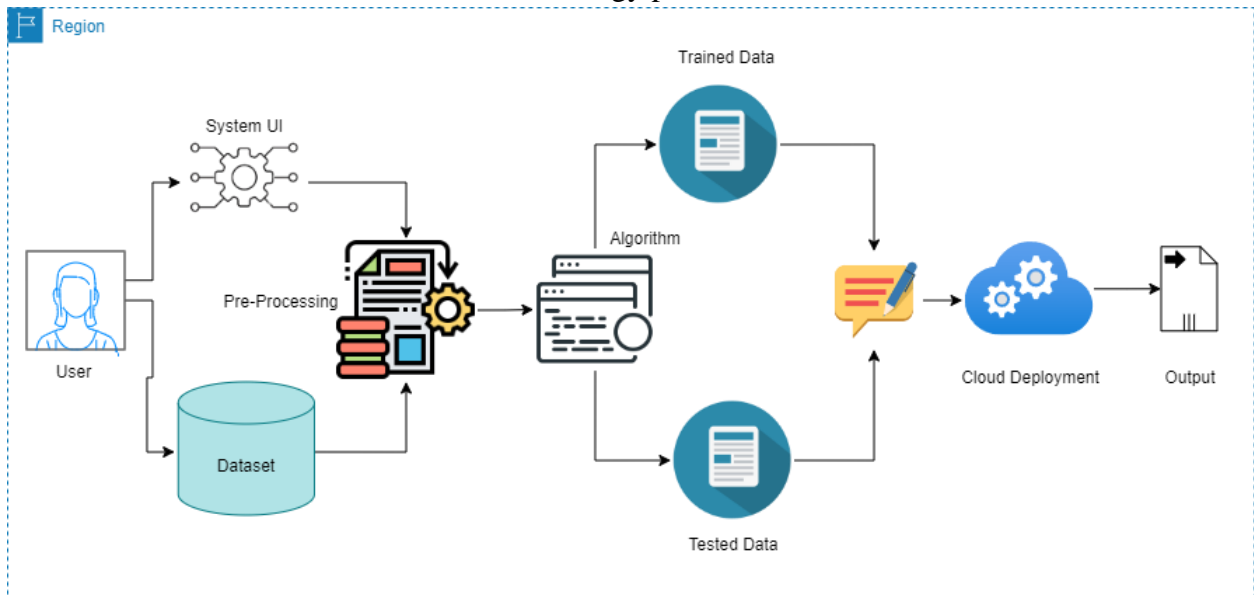


Table-1 : Components & Technologies:

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 (cR0noxEkdZHxE1Mwij0i2h7Q7kvTjRtpvAMzUCKCazXD , GwBW4bQSUah4tR0hskcQMbbB9uqNvz80kK0F915q9R4 , HzrV2Q9Ywg3EyxM014u62Meo2RhdZjp6np6AeCoq8Q1R)
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	<ol style="list-style-type: none"> 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

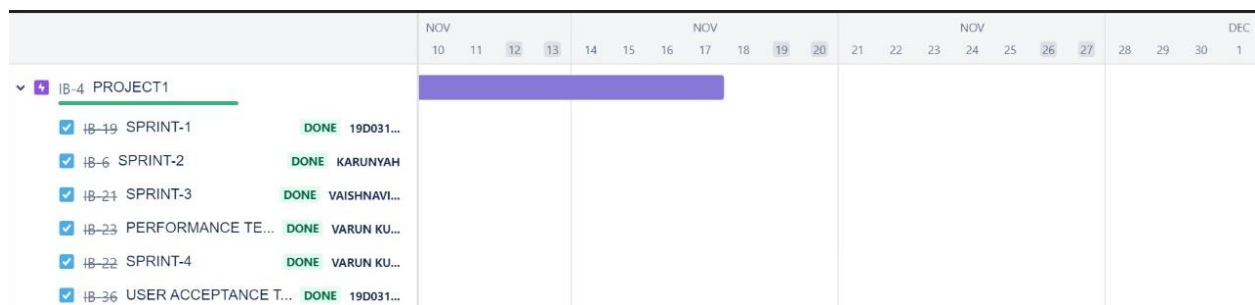
Sprint-4	Database Access	USN-11	As an administrator , I should maintain the website and keep updating it regularly	20	Medium	Varun Kumar Vaishnavi Harshithaa Karunyah
----------	-----------------	--------	--	----	--------	--

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 <ul style="list-style-type: none"> • Html pages • Testing
Sprint-4	Sprint-7	7. Train The Model On IBM <ul style="list-style-type: none"> • Register For IBM Cloud
	Sprint -8	8. Train The Model On IBM <ul style="list-style-type: none"> • 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 Pocco. Flask is based on the Werkzeug 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

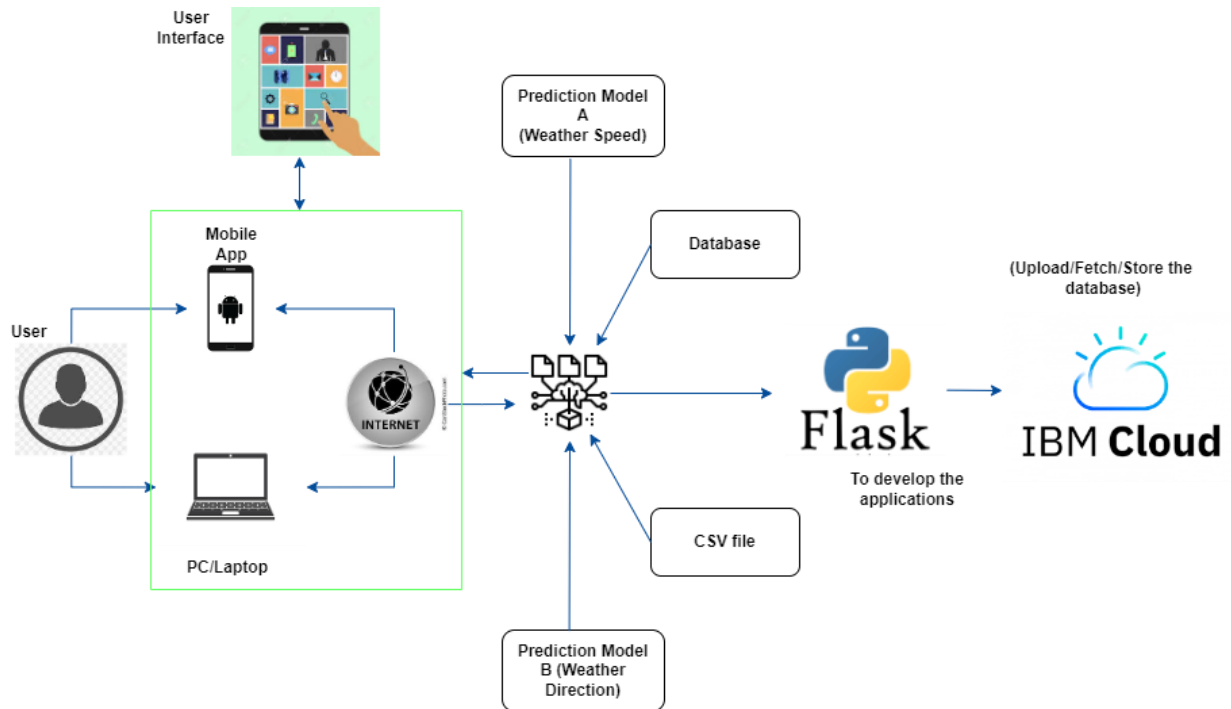
```

1  import numpy as np
2  from flask import Flask, request, jsonify, render_template
3  import joblib
4  import requests
5
6  # IBM Cloud account Credentials.
7  API_KEY = "SBahhsqevpUY8Eu1YKv5Kyl38DMCy3haa5WC0w8am_wL"
8  token_response = requests.post("https://iam.cloud.ibm.com/identity/token", data={"apikey": API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
9  mltoken = token_response.json()["access_token"]
10
11  header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
12  app = Flask(__name__)
13  # model = joblib.load('Power_Prediction.sav')
14  @app.route('/')
15  def home():
16      return render_template('index.html')
17  @app.route('/predict')
18  def predict():
19      return render_template('predict.html')
20  @app.route('/windapi', methods=['POST'])
21  def windapi():
22      city=request.form.get('city')
23      apikey="a802b0f626c637d84185e582b5ad8d58"
24      url="http://api.openweathermap.org/data/2.5/weather?q="+city+"&appid="+apikey
25      resp = requests.get(url)
26      resp=resp.json()
27      temp = str((resp["main"]["temp"])-273.15) + " °C"
28      humid = str(resp["main"]["humidity"])+ " %"
29      pressure = str(resp["main"]["pressure"])+ " mmHG"
30      speed = str((resp["wind"]["speed"])*3.6)+ " Km/s"
31      return render_template('predict.html', temp=temp, humid=humid, pressure=pressure,speed=speed)
32  @app.route('/y_predict', methods=['POST'])
33  def y_predict():
34      '''
35      For rendering results on HTML GUI
36      '''
37      x_test = [[float(x) for x in request.form.values()]]
38      print(x_test)
39      # Deployed Link
40      payload_scoring = {"input_data":
41                          [{"field": ["Theoretical_Power", "Wind_Speed"]},
42                           "values": x_test]}
43
44      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=payload_scoring)
45      print("Scoring response")
46      predictions =response_scoring.json()
47      print(predictions)
48      print('Final Prediction Result',predictions['predictions'][0]['values'][0][0])
49
50
51      pred =response_scoring.json()
52      print(pred)
53      #print('Final Prediction Result',predictions['predictions'][0]['values'][0][0])
54
55      # prediction = model.predict(x_test)
56      print(pred)
57      output = pred['predictions'][0]['values'][0][0]
58      return render_template('predict.html', prediction_text='The energy predicted is {:.2f} KWh'.format(output))
59
60
61  if __name__ == "__main__":
62      app.run(debug=False, port=8888)

```

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)))

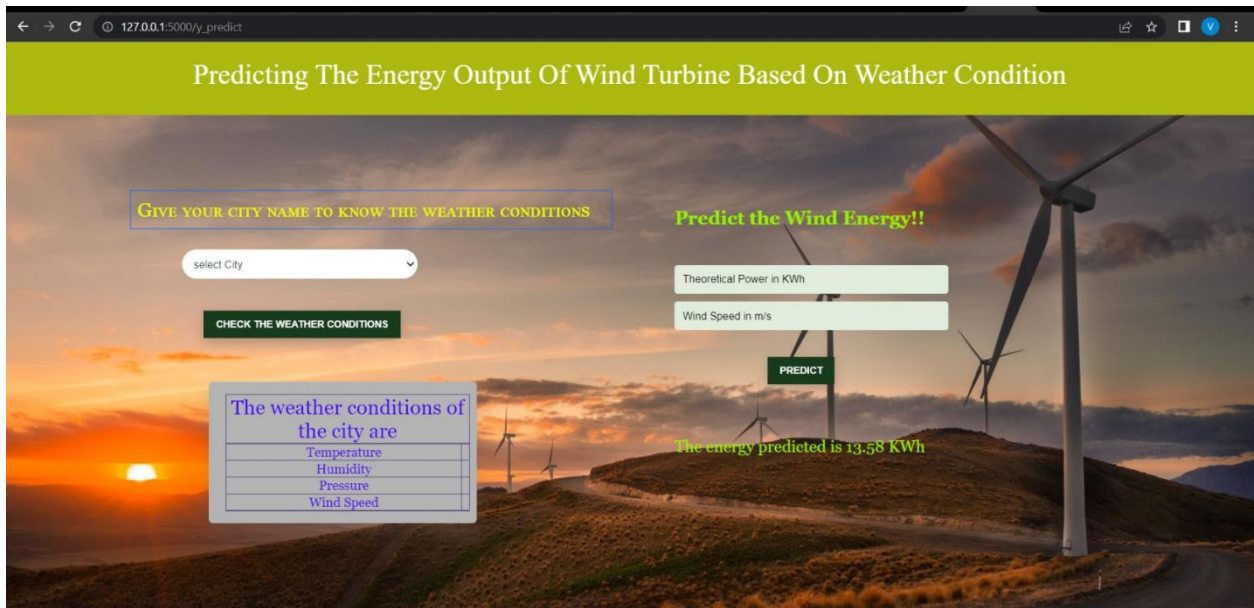
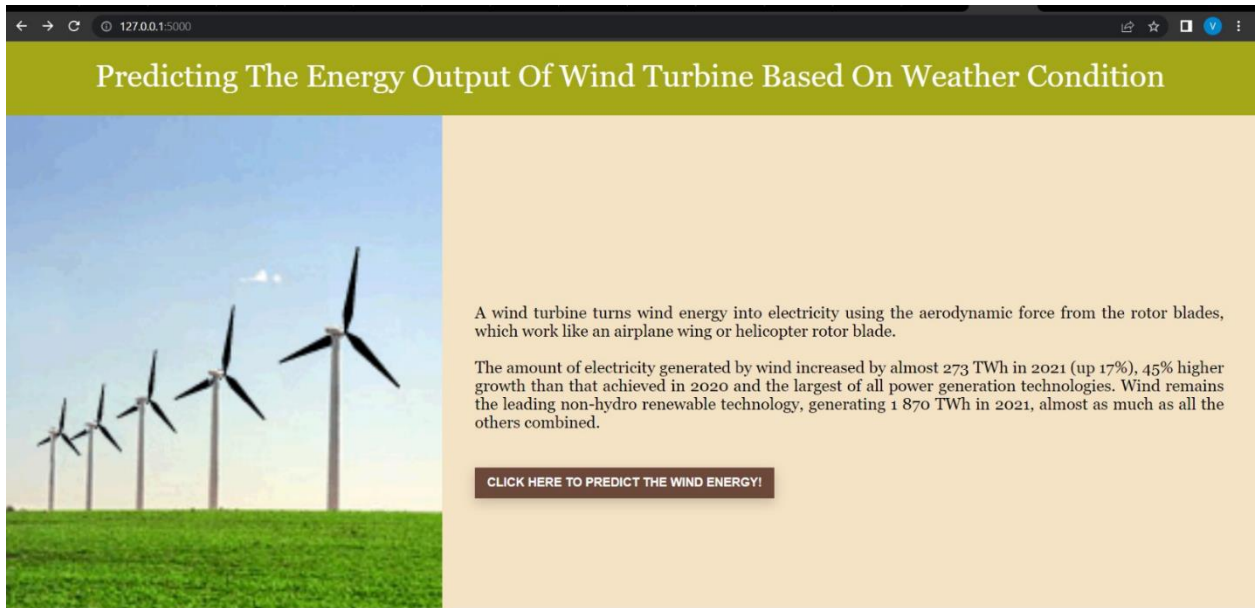
print('R2-lr', r2_score(y_test, y_lr))
print('RMSE-lr', np.sqrt(mean_squared_error(y_test, y_lr)))

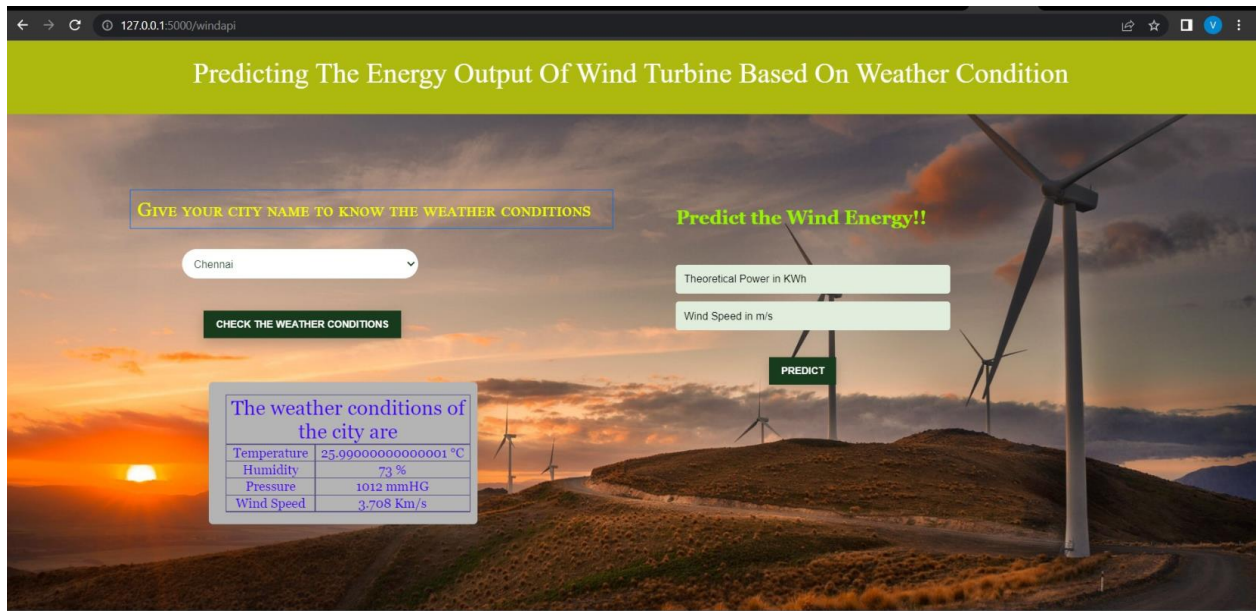
print('R2-dt', r2_score(y_test, y_dt))
print('RMSE-dt', np.sqrt(mean_squared_error(y_test, y_dt)))

print('R2-sm', r2_score(y_test, y_sm))
print('RMSE-sm', 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

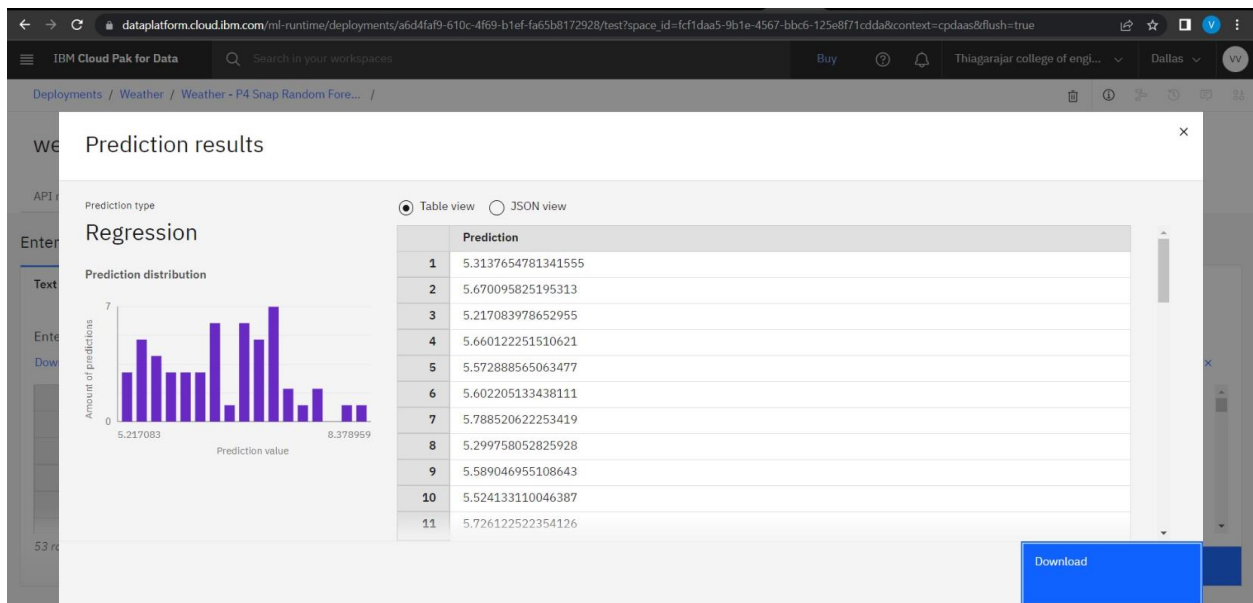
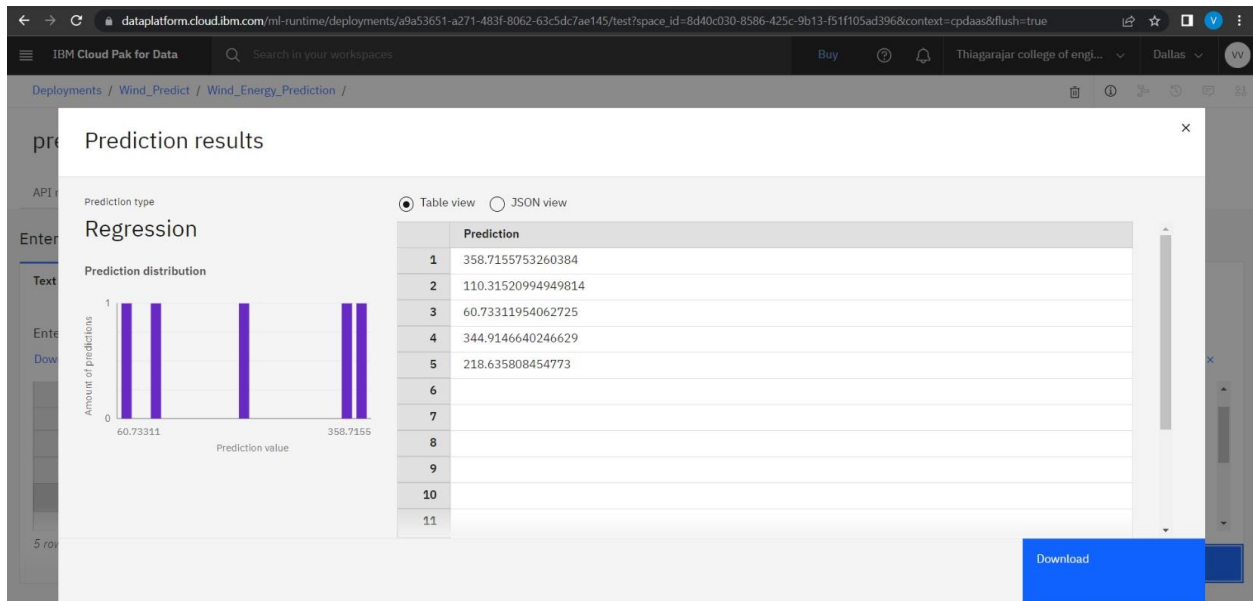
they were resolved

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



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%20mode%20on%20bm%20cloud/Flask%20App%20Integration%2...
84 lines (79 sloc) | 2.19 KB
Raw Blame
1 <html>
2 <head>
3 <title>Wind Energy Prediction</title>
4 <style>
5
6     .header {
7         top:0px;
8         margin:0px;
9         left: 0px;
10        right: 0px;
11        position: fixed;
12        background: #a4a717;
13        color: rgb(255, 255, 255);
14        overflow: hidden;
15        padding-bottom: 30px;
16        font-family:Georgia, 'Times New Roman', Times, serif, serif;
17        font-size: 2.5vw;
18        width: 100%;
19        padding-left:0px;
20        text-align: center;
21        padding-top:20px;
22    }
23
24    .second{
25        top:90px;
26        bottom:0px;
27        margin:0px;
28        left: 0px;
29        right: 0px;
30        position: fixed;
31        padding: 0px;
32        width: 100%;
33        background-image:url(https://i.pinimg.com/originals/c4/d2/f9/c4d2f98e88a85b702f8ff257d74714d8.gif);
34        background-repeat:no-repeat;
35        background-size: contain;
36    }
37
38    .inside{
39        top:90px;
40        bottom:0px;
41        margin:0px;
42        left: 35%;
43        right: 0%;
44        position: fixed;
45        padding-left: 40px;
46        padding-top:15%;
47        padding-right:40px;
48        background-color:#f5e3c5;
49        opacity: 100%;
50        font-family:Georgia, serif;
51        color:black;
52        font-size:20px;
53        text-align:justify;
54    }
55
56    .mybutton{
57        border: none;
58        text-align: center;
59        cursor: pointer;
60        text-transform: uppercase;
61        outline: none;
62        overflow: hidden;
63        color: #fff;
64        font-weight: 700;
65        font-size: 15px;
66        background-color: #c4493a;
67        padding: 10px 15px;
68        margin: 0 auto;
69        box-shadow: 0 5px 15px rgba(0,0,0,0.20);
70    }
71
72 </style>
73 </head>
74 <body>
75
76 <div class="header">Predicting The Energy Output Of Wind Turbine Based On Weather Condition</div>
77
78 <div class="second">
79     <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
80     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
81     <br><br><br>
82     <a href="{url_for('predict')}">button type="button" class="mybutton" >Click Here To Predict The Wind Energy!</button></a>
83 </div>
84 </body>
85 </html>
```

Prediction Code:

github.com/IBM-EPBL/IBM-Project-26990-1660042752/blob/main/Project%20development%20phase/Sprint-4/Train%20model%20on%20a%20cloud/Flask%20App%20Integration%22... 7.66 KB

```

47     }
48     .head {
49         top: 0px;
50         margin: 0px;
51         left: 0px;
52         right: 0px;
53         position: fixed;
54         background: #eee3e3;
55         color: white;
56         overflow: hidden;
57         padding-bottom: 30px;
58         font-size: 2.25vw;
59         width: 100%;
60         padding-left: 40px;
61         text-align: center;
62         padding-top: 20px;
63     }
64     .second {
65         top: 80px;
66         bottom: 0px;
67         margin: 0px;
68         left: 0px;
69         right: 0px;
70         position: fixed;
71         padding: 0px;
72         width: 100%;
73
74         font-family: Georgia, serif;
75         color: black;
76         font-size: 20px;
77     }
78     .inside {
79         top: 80px;
80         bottom: 0px;
81         margin: 0px;
82         left: 5%;
83         right: 0%;
84         position: fixed;
85         padding-left: 40px;
86         padding-top: 15%;
87         padding-right: 40px;
88
89         font-family: Georgia, serif;
90         color: #99f400;
91         font-size: 20px;
92         text-align: justify;
93
94     }
95     .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...
95     .myButton{
96         border: none;
97         text-align: center;
98         cursor: pointer;
99         text-transform: uppercase;
100         outline: none;
101         overflow: hidden;
102         color: #fff;
103         font-weight: 700;
104         font-size: 12px;
105         background-color: #183a1d;
106         padding: 10px 15px;
107         margin: 0 auto;
108         box-shadow: 0 5px 15px rgba(0,0,0,0.20);
109         margin-left: 17%;
110     }
111     input {
112         width: 50%;
113         margin-bottom: 10px;
114         background: #e1e2e3;
115         border: none;
116         outline: none;
117         padding: 10px;
118         font-size: 13px;
119         color: #6c493a;
120         text-shadow: white;
121         border: #6c493a;
122         border-radius: 4px;
123         box-shadow: white;
124     }
125     ::placeholder {
126         color: black;
127         opacity: 1;
128     }
129     .left{
130         top: 80px;
131         bottom: 80px;
132         margin: 0px;
133         left: 0%;
134         right: 45.5%;
135         position: fixed;
136         padding-left: 10%;
137         padding-top: 1%;
138         padding-right: 40px;
139         font-family: bold, georgia, serif;
140         color: rgb(238, 255, 0);
141         font-size: 25px;
142     }
143     ...
144     font-size: 13px;
145 }
146 select {
147     width: 50%;
148     margin-bottom: 10px;
149     background: white;
150     border: none;
151     outline: none;
152     padding: 10px;
153     font-size: 13px;
154     color: #183a1d;
155     text-shadow: white;
156     border: #6c493a;
157     border-radius: 40px;
158     box-shadow: white;
159 }
160 input:focus { box-shadow: inset 0 -5px 4px rgba(100,100,100,0.4), 0 1px 1px rgba(255,255,0.2); }
161
162
163
164
165     table, th, td {
166     border: 1px solid rgb(86, 72, 128);
167     border-collapse: collapse;
168     color: #3f00ff;
169     }
170
171 @media screen and (max-width: 500px) {
172
173     .left,
174     .second,
175     .third {
176         width: 70%;
177     }
178 }
179
180
181
182 </style>
183 </head>
184 <body>
185
186
187 <header id="head">
188 <div class="head">Predicting The Energy Output Of Wind Turbine Based On Weather Condition</div>
189 </header>
190
191
```

```
github.com/IBM-EPBL/IBM-Project-26990-1660042752/blob/main/Project%20development%20phase/Sprint-4/Train%20model%20on%20ibm%20cloud/Flask%20App%20Integration%2...
143     font-size: 13px;
144
145 }
146     select {
147         width: 50%;
148         margin-bottom: 10px;
149         background: white;
150         border: none;
151         outline: none;
152         padding: 10px;
153         font-size: 13px;
154         color: #183a1d;
155         text-shadow: white;
156         border: #6c493a;
157         border-radius: 40px;
158         box-shadow: white;
159     }
160     input:focus { box-shadow: inset 0 -5px 4px rgba(100,100,100,0.4), 0 1px 1px rgba(255,255,0.2); }
161
162
163
164
165     table, th, td {
166     border: 1px solid rgb(86, 72, 128);
167     border-collapse: collapse;
168     color: #3f00ff;
169     }
170
171 @media screen and (max-width: 500px) {
172
173     .left,
174     .second,
175     .third {
176         width: 70%;
177     }
178 }
179
180
181
182 </style>
183 </head>
184 <body>
185
186
187 <header id="head">
188 <div class="head">Predicting The Energy Output Of Wind Turbine Based On Weather Condition</div>
189 </header>
190
191
```

```
192         <div class="second">
193         <div class="left">
194
195             <p style="padding: 8px; border: 1px solid rgb(0, 118, 255); width: 575px;"> Give your CITY NAME to KNOW THE WEATHER CONDITIONS</p>
196
197         <div style="margin-left:105">
198             <form action="{ { url_for('wanda1') }}" method="post" >
199                 <select name="city" required >
200                     <option value="" selected><select><select City</option>
201                     <option value ="Ariyalur" > Ariyalur </option>
202                     <option value ="Andinadan" > Andinadan </option>
203                     <option value ="Coimbatore" > Coimbatore </option>
204                     <option value ="Chengalpattu" > Chengalpattu </option>
205                     <option value ="Cuddalore" > Cuddalore </option>
206                     <option value ="Chennai" > Chennai </option>
207                     <option value ="Dindigul" > Dindigul </option>
208                     <option value ="Dharmapuri" > Dharmapuri </option>
209                     <option value ="Erode" > Erode </option>
210                     <option value ="Karur" > Karur </option>
211                     <option value ="Kancheepuram" > Kancheepuram </option>
212                     <option value ="Krininagiri" > Krishnagiri </option>
213                     <option value ="Kallakurichi" > Kallakurichi </option>
214                     <option value ="Madurai" > Madurai </option>
215                     <option value ="Mayiladuthurai" > Mayiladuthurai </option>
216                     <option value ="Nagapattinam" > Nagapattinam </option>
217                     <option value ="Kanyakumari" > Kanyakumari </option>
218                     <option value ="Namakkal" > Namakkal </option>
219                     <option value ="Perambalur" > Perambalur </option>
220                     <option value ="Pudukottai" > Pudukottai </option>
221                     <option value ="Ramanathapuram" > Ramanathapuram </option>
222                     <option value ="Ranipet" > Ranipet </option>
223                     <option value ="Salem" > Salem </option>
224                     <option value ="Sivagangai" > Sivagangai </option>
225                     <option value ="Tenkasi" > Tenkasi </option>
226                     <option value ="Thanjavur" > Thanjavur </option>
227                     <option value ="Theni" > Theni </option>
228                     <option value ="Thiruvallur" > Thiruvallur </option>
229                     <option value ="Thiruvavur" > Thiruvavur </option>
230                     <option value ="Tuticorin" > Tuticorin </option>
231                     <option value ="Trichirappalli" > Trichirappalli </option>
232                     <option value ="Thiruvallur" > Thiruvallur </option>
233                     <option value ="Tirupathur" > Tirupathur </option>
234                     <option value ="Tiruppur" > Tiruppur </option>
235                     <option value ="Tiruvannamalai" > Tiruvannamalai </option>
236                     <option value ="The Nilgiris" > The Nilgiris </option>
237                     <option value ="Vellore" > Vellore </option>
238                     <option value ="Viluppuram" > Viluppuram </option>
239                     <option value ="Virudhunagar" > Virudhunagar </option>
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
```

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
```

```
1 import numpy as np
2 from flask import Flask, request, jsonify, render_template
3 import joblib
4 import requests
5
6
7 # IBM Cloud Integration with Credentials
8 API_KEY = "ae56da0f-f4b2-4aa5-9480-6baa8b746ac3"
9 token_response = requests.post("https://iam.cloud.ibm.com/identity/token", data={"apikey": API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
10 mltoken = token_response.json()["access_token"]
11
12 header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
13 app = Flask(__name__)
14 # model = joblib.load('Power_Prediction.sav')
15 @app.route('/')
16 def home():
17     return render_template('intro.html')
18 @app.route('/predict')
19 def predict():
20     return render_template('predict.html')
21 @app.route('/windapi', methods=['POST'])
22 def windapi():
23     city=request.form.get('city')
24     apiKey="a802b0f626c637004185e582b5d0d058"
25     url="http://api.openweathermap.org/data/2.5/weather?q="+city+"&appid="+apiKey
26     resp = requests.get(url)
27     resp=resp.json()
28     temp = str((resp["main"]["temp"])-273.15) + " °C"
29     humid = str((resp["main"]["humidity"])+ "%")
30     pressure = str((resp["main"]["pressure"])+ " mmHg")
31     speed = str((resp["wind"]["speed"])*3.6)+ " Km/s"
32     return render_template('predict.html', temp=temp, humid=humid, pressure=pressure,speed=speed)
33 @app.route('/y_predict',methods=['POST'])
34 def y_predict():
35     ...
36     For rendering results on HTML GUI
37     ...
38     x_test = [[float(x) for x in request.form.values()]]
39     print(x_test)
40     payload_scoring = {"input_data":
41         [{"field": [{"Theoretical_Power", "Wind_Speed"}],
42          "values": x_test}]}
43
44     response_scoring = requests.post("https://us-south.ml.cloud.ibm.com/ml/v4/deployments/1a772765-e95e-4928-a46b-25ede8ef1b44/predictions?version=2022-11-06", json=payload_scoring)
45     print("Scoring response")
46     predictions =response_scoring.json()
47     print(predictions)
48     print('Final Prediction Result',predictions['predictions'][0]['values'][0][0])
49
50
51     pred =response_scoring.json()
52     print(pred)
53     #print('Final Prediction Result',predictions['predictions'][0]['values'][0][0])
54
55     # prediction = model.predict(x_test)
56     print(pred)
57     output = pred['predictions'][0]['values'][0][0]
58     return render_template('predict.html', prediction_text='The energy predicted is {:.2f} KWh'.format(output))
59
60
61 if __name__ == "__main__":
62     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

<https://youtu.be/DFsfgR3Z90s>