Predicting The Energy Output Of Wind Turbine Based On Weather Condition

Project Report

Kings Engineering College

B.E. Electronics and Communication Engineering
(2019-2023)
Team ID : PNT2022TMID25465

Renuka Devi M(210819106061)
Princy mol Joseph(210819106054)
Preethi S(210819106053)
Siva Priya S(210819106302)
Sri Gayathri Devi A(210819106073)

INDEX

CHAPTER-1

INTRODUCTION

1.1 Project Overview

Wind turbine technology has reached a mature status during the past 15 years as a result of international commercial competition, mass production and continuing technical success in research and development (R&D). The earlier concerns that wind turbines were expensive and unreliable have largely been allayed. Wind energy project costs have declined and wind turbine technical availability is now consistently above 97%. Wind energy project plant capacity factors have also improved from 15% to over 30% today, for sites with a good wind regime.

Wind power generation differs from conventional thermal generation due to the stochastic nature of wind. Thus wind power forecasting plays a key role in dealing with the challenges of balancing supply and demand in any electricity system, given the uncertainty associated with the wind farm power output. Accurate wind power forecasting reduces the need for additional balancing energy and reserve power to integrate wind power. For a wind farm that converts wind energy into electricity power, a real-time prediction system of the output power is significant. In this guided project , a prediction system is developed with a method of combining statistical models and physical models. In this system, the inlet condition of the wind farm is forecasted by the auto regressive model.

1.2 Purpose

We are building an IBM Watson AutoAI Machine Learning technique to predict the energy output of wind turbine. The model is deployed on IBM cloud to get scoring end point which can be used as API in mobile app or web app building. We are developing a web application which is built using node red service.

CHAPTER-2

LITERATURE SURVEY

Here, we will take a look at all the previous solutions, attempts and implementations to Predict the energy output of wind turbine based on weather condition.

2.1 Existing problem

1. 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.
2. 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. As a result, the model proposed achieves high accuracy and can be key to understanding the wind turbine asset's behavior throughout its lifespan, assisting operators in decision making to increase overall power production.
3. 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. For a wind farm which converts the wind power to electrical energy, a big challenge is to predict the wind power precisely in spite of the instabilities. The climatic conditions present in the site decides the power output of a wind farm. As the schedule of wind power availability is not known in advance, this causes problems for wind farm operators in terms of system and energy planning. A precise forecast is required to overcome the difficulties initiated by the fluctuating weather conditions. If the output is forecasted

accurately, energy providers can keep away from costly overproduction. In this paper, an end-to-end web application has been developed to predict and forecast the wind turbine's power generation based on the weather conditions. The prediction model has been developed using Bidirectional Long Short-Term Memory which is a unique kind of RNN (Recurrent Neural Network). It performs admirably in terms of capturing long-term dependencies along with the time steps and is hence ideal for wind power forecasting.

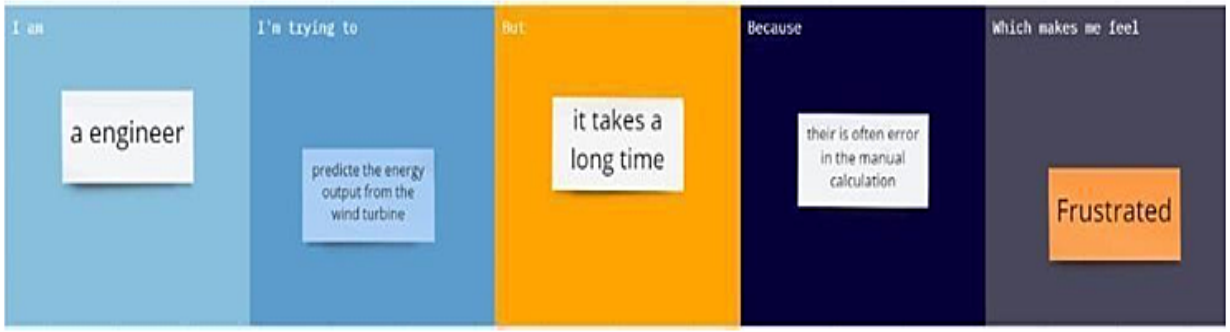
2.2 References

1. Abdelkader Harrouz, Ilhami Colak, Korhan Kayisli, "Energy Modeling Output Of Wind System based on Wind Speed", 2019 8th International Conference on Renewable Energy Research and Applications (ICRERA). https://ieeexplore.ieee.org/document/8996525/
2. Kelvin Palhares Bastos Sathler, Athanasios Kolios, "The Use of Machine Learning and Performance Concept to Monitor and Predict Wind Power Output", 2022 International Conference on Electrical, Computer and Energy Technologies (ICECET). https://ieeexplore.ieee.org/document/9873076/
3. S Preethi, H Prithika, M Pramila, S Birundha, "Predicting the Wind Turbine Power Generation based on Weather Conditions", 2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA). https://ieeexplore.ieee.org/document/9676051/

2.3 Problem Statement Definition

To comprehend the viewpoint of your customer,write an issue statement. In order to build experiences that customers will appreciate, the Customer ProblemStatement template can help you concentrate on what is important.

You and your team can identify the appropriate solution to the problems your clients are facing with the help of a clearlystated customer problemstatement. You'll also develop an empathy for your clientsduring the procedure, which will enable you to comprehend how they view your goods or service.



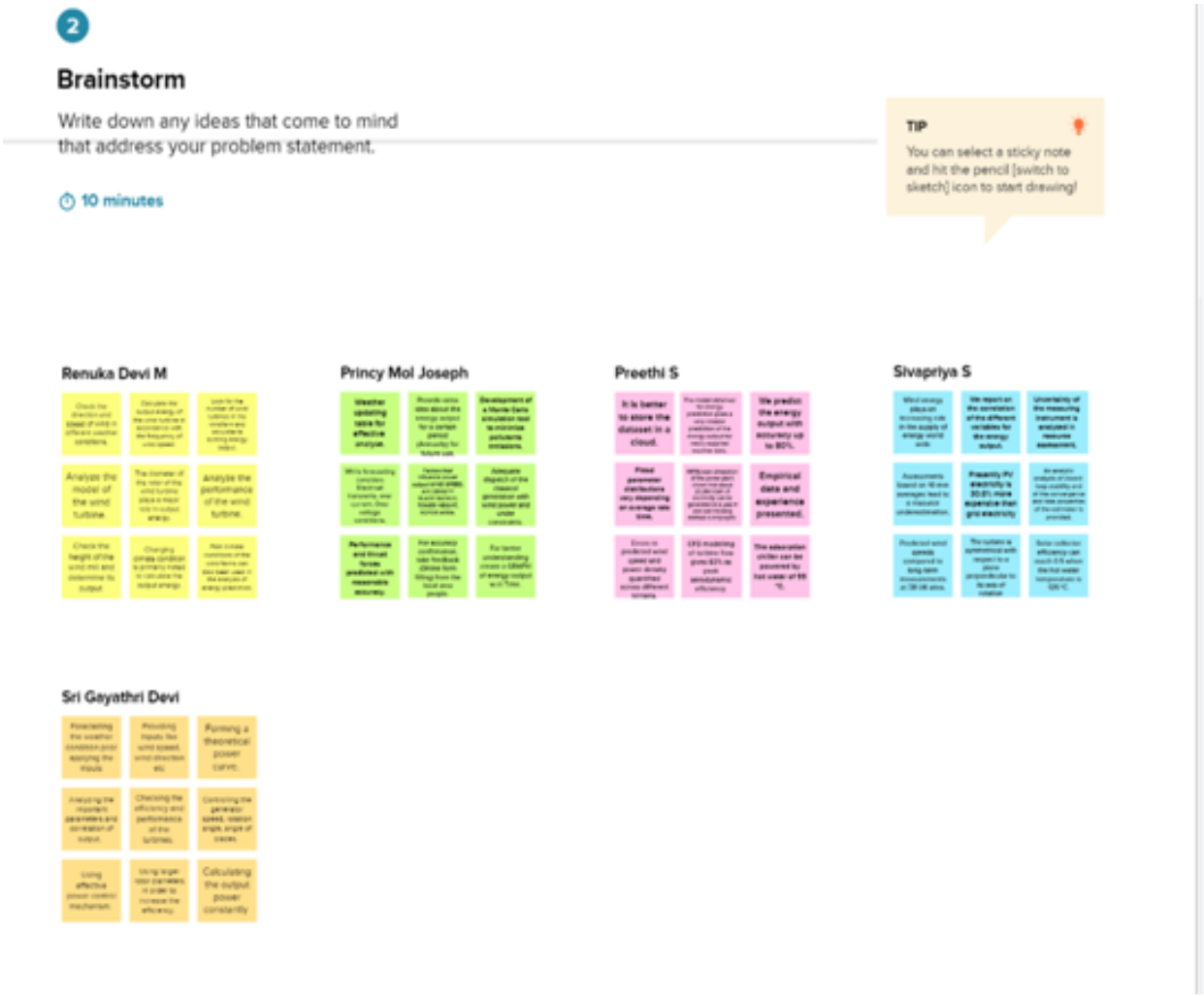
Problem Statement (PS)	**I am (Customer)**	**I'm tryingto**	**But**	**Because**	**Which makesme feel**
PS-1	A engineer	Predict the energy output fromthe wind turbine	It takes too long	There is often errorin the manual calculation	frustrated
PS-2	A Teenager	Try this new application	I don't understa nd anything in the application	It is very hard formeto understand	useless

CHAPTER-3

IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

3.2 Ideation & Brainstorming



Weather Condition

Check the direction and speed of wind in different weather conditions.

Calculate the output energy of the wind turbine in accordance with the frequency of wind speed.

The model obtained for energy prediction gives a very reliable prediction of the energy output for newly supplied weather data.

The adsorption chiller can be powered by hot water of 55 °C.

Forecasting the weather condition prior applying the inputs

Output Energy

Look for the number of wind turbines in the windfarm and calculate its existing energy output.

The diameter of the rotor of the wind turbine plays a major role in output energy.

Adequate dispatch of the classical generation with wind power and under constraints.

We report on the correlation of the different variables for the energy output.

Calculating the output power constantly

Hassle Free

Development of a Monte Carlo simulation tool to minimize pollutants emissions.

An hourly simulation of the power plant shows that about 35,360 GWh of electricity can be generated in a year if one axis tracking method is employed.

Errors in predicted wind speed and power density quantified across different terrains.

Wind energy plays an increasing role in the supply of energy world wide

Solar collector efficiency can reach 0.5 when the hot water temperature is 125 °C.

Using effective power control mechanism.

Wind Turbine

Analyze the model of the wind turbine.

Analyze the performance of the wind turbine.

Check the height of the wind mill and determine its output.

Predicted wind speeds compared to long term measurements at 38 UK sites.

The turbine is symmetrical with respect to a plane perpendicular to its axis of rotation

Checking the efficiency and performance of the turbines.

Controlling the generator speed, rotation angle, angle of blades.

Using larger rotor diameters in order to increase the efficiency.

Dataset Analysis

Changing climate scenarios is primarily noted to calculate the output energy

Past climate conditions of the wind farms can also been used in the analysis of energy prediction.

Weather updating table for effective analyse.

Provide extra idea about the energy output for a certain period (Annually) for future use.

For accurately confirmation, take feedback (Online form fillings) from the local area people.

An analytic analysis of closed-loop stability and of the convergence and bias properties of the estimator is provided.

Influencing Factors

While forecasting considers Electrical transients, over current, Over voltage conditions.

Factors that influence power outputrelmb SPEED, AIR DENSITY BLADE RADIUS, POWER COEFFICIENT, ROTOR AREA.

Assessments based on 10 min averages lead to a resource underestimation.

Providing inputs like wind speed, wind direction etc

Uncertainty of the measuring instrument is analyzed in resource assessment.

Performance Analysis

Performance and thrust forces predicted with reasonable accuracy.

For better understanding create a GRAPH of energy output w.r.t Time.

Empirical data and experience presented.

Presently PV electricity is 30.8% more expensive than grid electricity

Forming a theoretical power curve.

Cloud

It is better to store the dataset in a cloud.

Accuracy

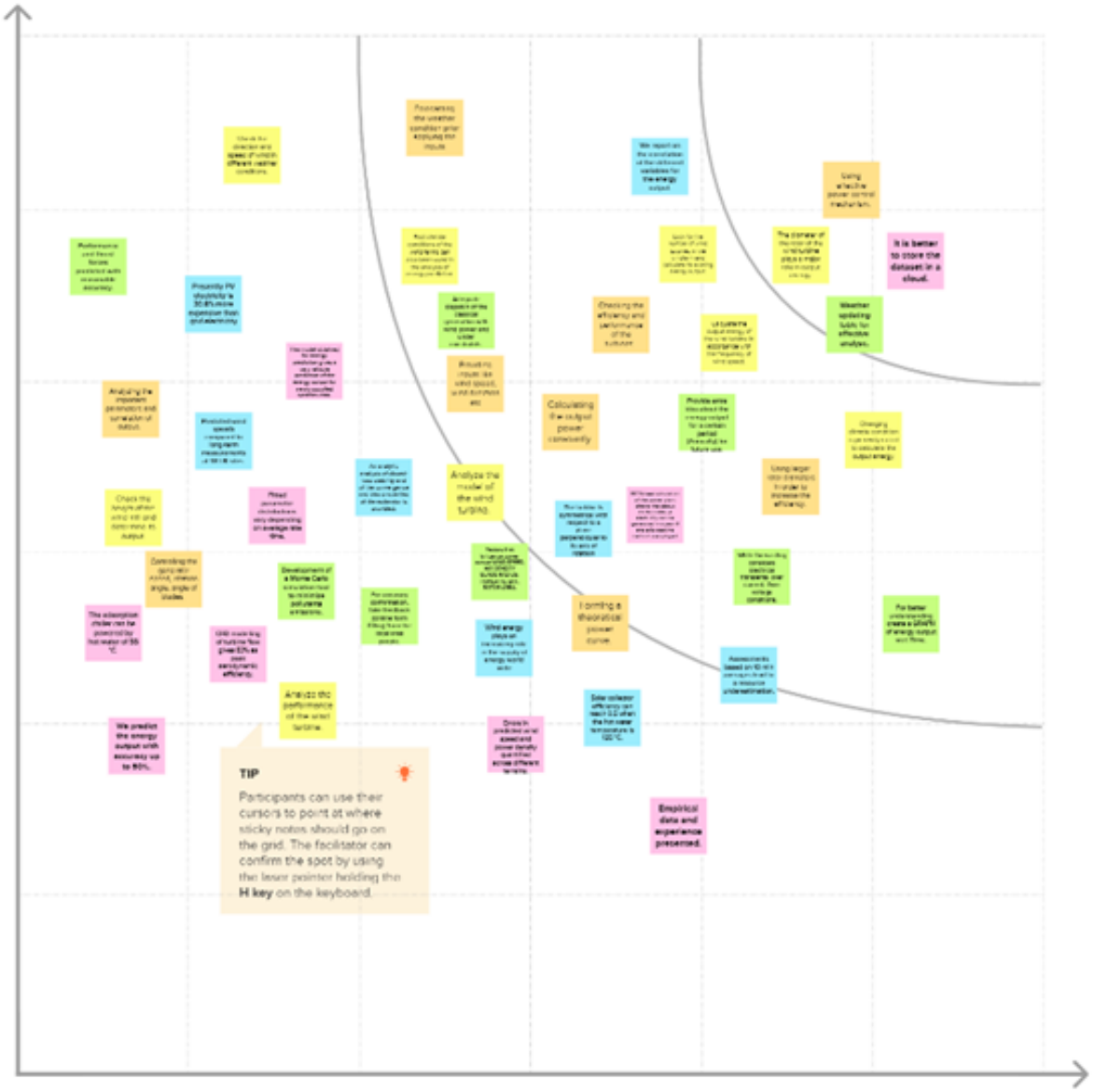
We predict the energy output with accuracy up to 80%.

Fitted parameter distributions vary depending on average rate time.

CFD modelling of turbine flow gives 82% as peak aerodynamic efficiency.

Analysing the important parameters and correlation of output.

Checking the efficiency and performance of the turbines.


Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)
TIP
Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the H key on the keyboard.
It is better to store the dataset in a cloud.
Calculating the output power conveniently
Analyze the model of the wind turbine.
Empirical data and experience preserved.

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to besolved)	It is necessary to find a way to predict the energy output of a wind turbine in different weather conditions. The obtained windenergymust be used to give a steady supply of electricity.
2.	Idea / Solution description	It is necessary to analyse andto store thedata 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.
3.	Novelty / Uniqueness	Present windfarms don't haveany methods to predict the output energy based on the changing weather conditions. By implementingthis model, it can be useful to predict the output energy before and the efficiency of the wind farms can alsobeen improved.
4.	Social Impact / Customer Satisfaction	Currently wind energy is not the primary source of electricity, but by implementing our solution we can produce more energy. So the utilisation of non renewable resources can also be minimised. A wind farm with prediction mode would be more efficient than the present one. Switching to a cleansource of energyis good forboth human healthand the environment.
5.	Business Model (Revenue Model)	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 energysystems.
6.	Scalability of the Solution	It can be appliedon the largescale in the existing wind farm.So the performance can alsobe improved.

3.4 Problem Solution fit

Project Design Phase-I - Solution Fit Template

Project Title: Predicting the energy output of a wind turbine based on weather conditions.
Team ID: PNT2022TMID25465

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) — CS

Wind energy producers.

6. CUSTOMER CONSTRAINTS — CC

Lack of Budget, They are not clear on how to utilize the wind turbine effectively to produce a steady electricity.

5. AVAILABLE SOLUTIONS — AS

Estimation is calculated based in past year energy output.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS — J&P

To analyse the output energy of wind turbine in changing weather conditions. And to store the data in a dataset.

9. PROBLEM ROOT CAUSE — RC

High initial cost setup and unpredictable changes in weather condition.

7. BEHAVIOUR — BE

Calculates the usage and benefits. Collects the data from the potential wind farms and makes a comparison.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS — TR

If the customer finds it as an efficient solution. It will automatically trigger all other customers to do it.

4. EMOTIONS: BEFORE / AFTER — EM

Before: Confused with improper energy flow.
After: Happy with the efficient technique.

10. YOUR SOLUTION — SL

The inlet condition of the wind turbine is forecasted by an auto regressive model. Hence it reduces the need for balancing energy and reserved power output energy.

8. CHANNELS of BEHAVIOUR — CH

8.1 ONLINE

It will analyse the data which are previously uploaded and predict the output energy.

8.2 OFFLINE

The inlet condition of the wind turbine is maintained constantly.

Identify strong TR & EM

CHAPTER-4

REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story/ Sub-Task)
FR-1	User Registration and logging in by entering their username andpassword.	Registration through Form.
FR-2	User Confirmation by validating the username with respect to the password	Confirmation via pop-up Message.
FR-3	Displaying furtherinformation about theapplication.	By selecting the aboutbutton the detailsof theapplication 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 detailselse it willdisplay the pop-upmessage as errorinthe city.
FR-5	Checking the data typeof the value.	System checksfor the datatype of the value entered bythe user.
FR-6	Validating all required fields.	Before predicting the output the system checks whether all the valuesare entered by the userand checks whether allvalues are correct.
FR-7	Displaying weather conditionsfor a given city.	It displays the weather of the citywhich has beenselected.
FR-8	Displaying predicted energyoutput power.	The predicted output will be displayed as the amountof wind energypower generated.

4.2 Non-Functional requirements

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The system satisfies the user goalsand the application is easy to use.
NFR-2	**Security**	The data provided to system willbe protected fromattacks and unauthorised access
NFR-3	**Reliability**	The systemwill provide consistency in output withoutproducing an error.
NFR-4	**Performance**	The performance will neverdegrade even if the workload is increased.
NFR-5	**Availability**	The application is available for 24*7
NFR-6	**Scalability**	The systemcan be used as web application as wellas a mobile application with sufficient internet availability.

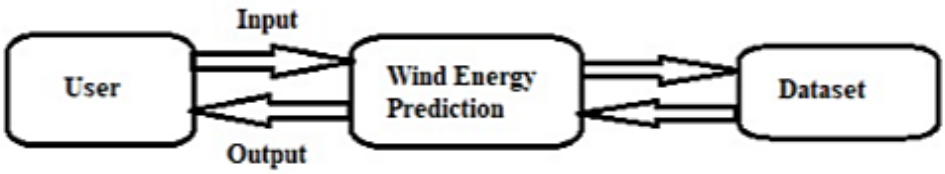
CHAPTER-5

PROJECT DESIGN

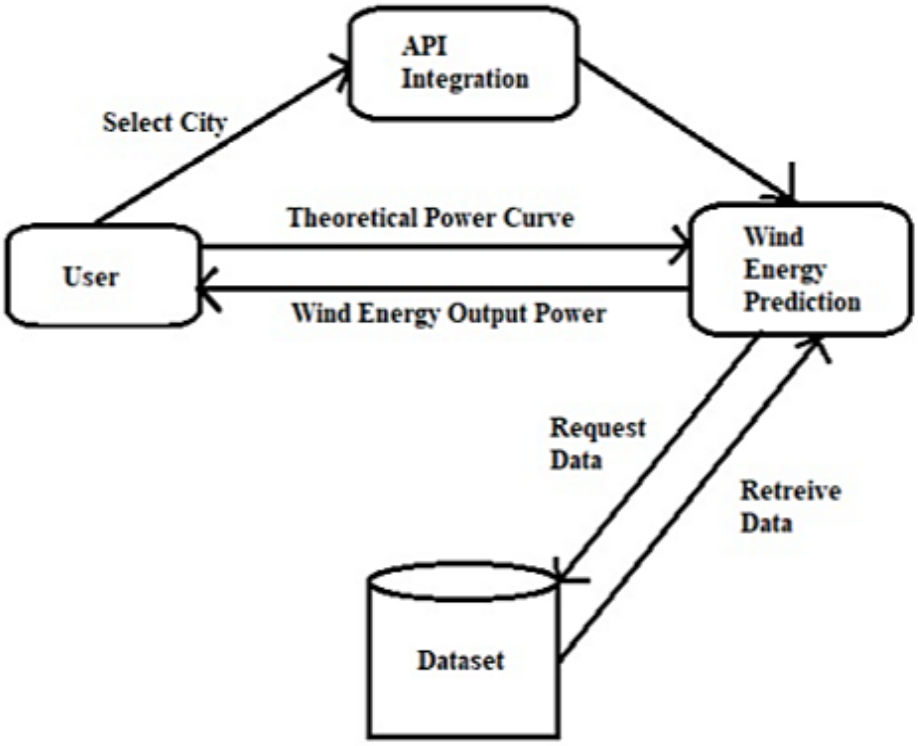
5.1 Data Flow Diagrams

Data Flow:

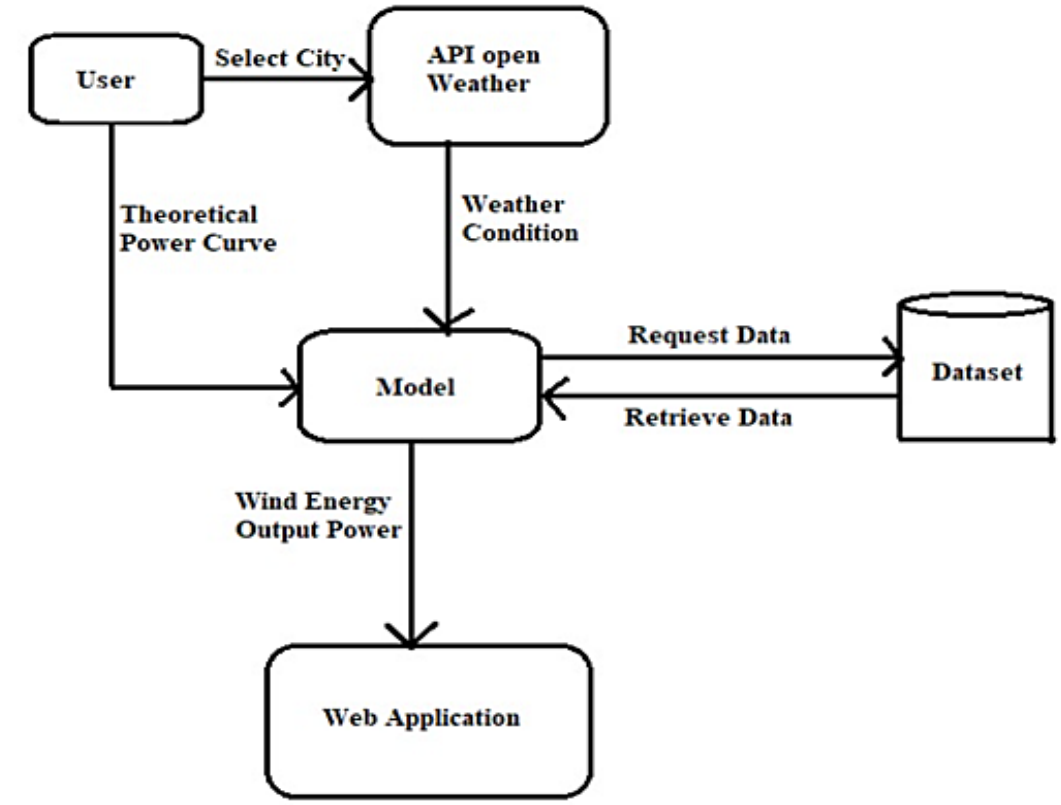
LEVEL 0:



LEVEL 1:



LEVEL 2:

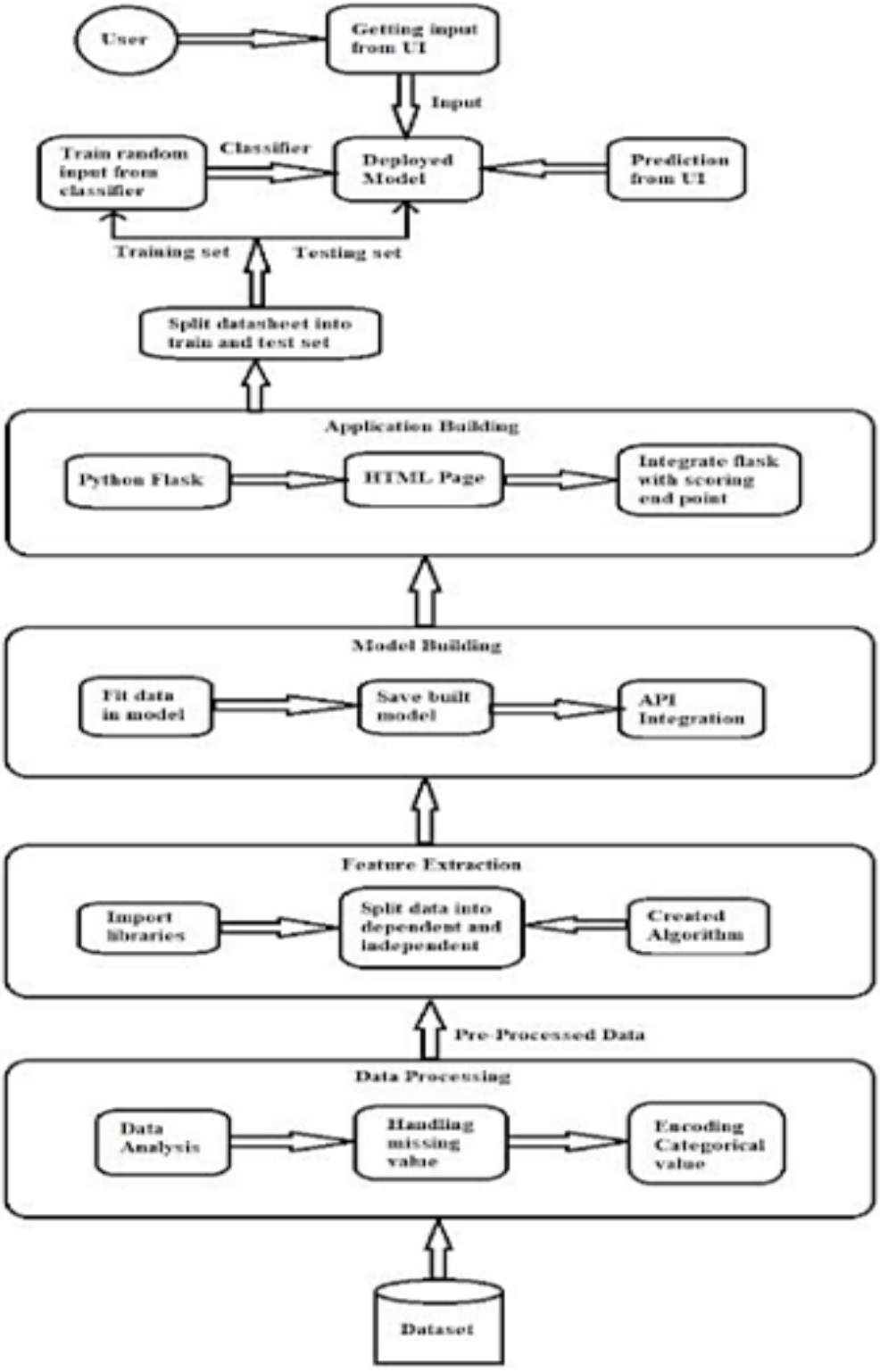

User
Select City
API open Weather
Theoretical Power Curve
Weather Condition
Model
Request Data
Retrieve Data
Dataset
Wind Energy Output Power
Web Application

5.2 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 a user-friendly method by reading the instructions.	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 the 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 freely.	I can use it without any payment to access	Medium	Sprint-2

5.3 Solution & Technical Architecture



CHAPTER-6

PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Product Backlog, Sprint Schedule, and Estimation

Sprint	Milestone
Sprint 1	1. Users register into the application through entering Email Id, Password and Re-entering Password for confirmation. 2. Users receive a confirmation mail to their registered Email. 3. Users can also register to the application through a mobile number. 4. User logs in into the website using Email Id and password or through Gmail
Sprint 2	1. Users can access the dashboard 2. Users can enter the required details on weather conditions and get the desired turbine power output based on the model's prediction.
Sprint 3	1. Application should store the predictions, and these predictions can be used for future analysis. 2. The data stored should be secure.
Sprint 4	1. Administrator should properly maintain the website and update it when required.

Product Backlog, Sprint Schedule, and 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	Renuka Princy Preethi Sivapriya Gayathri
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	5	High	Renuka Princy Preethi Sivapriya Gayathri
Sprint-1		USN-3	As a user, I can register for the application through Google	5	Low	Renuka Princy Preethi Sivapriya Gayathri

Sprint-1		USN-4	As a user, I can register for the application through Gmail	5	Medium	Renuka Princy Preethi Sivapriya Gayathri

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	5	High	Renuka Princy Preethi Sivapriya Gayathri
Sprint-2	Dashboard	USN-6	Once logged in, I can access my dashboard	6	Medium	Renuka Princy Preethi Sivapriya Gayathri
Sprint-2	Web Access	USN-7	As a user, I can access the website to predict the turbine power	7	High	Renuka Princy Preethi Sivapriya Gayathri
Sprint-2	Prediction	USN-8	As a customer, when I enter the detail the website should predict the approximate turbine power	7	High	Renuka Princy Preethi Sivapriya Gayathri
Sprint-3	Analysis	USN-9	As a customer, I wish to store my predictions and make analysis	10	Medium	Renuka Princy Preethi Sivapriya Gayathri

Sprint-3	Security	USN-10	As a customer I expect my data to be secured	10	Medium	Renuka Princy Preethi Sivapriya Gayathri
Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Story Points**	**Priority**	**Team Members**
Sprint-4	Database Access	USN-11	As an administrator, I should maintain the website and keep updating it regularly	20	Medium	Renuka Princy Preethi Sivapriya Gayathri

6.2 Reports from JIRA

Jira

Displaying **19** issues at **18/Nov/22 10:28 PM**

Project	Key	Summary	Issue Type	Status	Priority	Resolution	Assignee	Reporter	Creator	Created	Last Viewed	Updated	Resolved
IBM Project	IP-20	Integrate Flask with scoring end point	Story	Done	Medium	Done	Sri Gayathri Devi	Renuka Devi M	Renuka Devi M	15/Nov/22 9:48 PM	18/Nov/22 10:14 PM	18/Nov/22 10:14 PM	18/Nov/22 10:14 PM
IBM Project	IP-19	Train the ML model on IBM	Story	Done	Medium	Done	Princy mol Joseph	Renuka Devi M	Renuka Devi M	15/Nov/22 9:47 PM	18/Nov/22 10:13 PM	18/Nov/22 10:13 PM	18/Nov/22 10:13 PM
IBM Project	IP-18	Register for IBM cloud	Story	Done	Medium	Done	preethi072002	Renuka Devi M	Renuka Devi M	15/Nov/22 9:47 PM	18/Nov/22 10:13 PM	18/Nov/22 10:13 PM	18/Nov/22 10:13 PM
IBM Project	IP-17	Train the model on IBM	Story	Done	Medium	Done	Renuka Devi M	Renuka Devi M	Renuka Devi M	15/Nov/22 9:47 PM	18/Nov/22 10:08 PM	18/Nov/22 10:08 PM	18/Nov/22 10:08 PM
IBM Project	IP-16	Execute and test your model	Story	Done	Medium	Done	sivapriyasharon	Renuka Devi M	Renuka Devi M	15/Nov/22 9:45 PM	18/Nov/22 10:12 PM	18/Nov/22 10:13 PM	18/Nov/22 10:13 PM
IBM Project	IP-15	Build an HTML page	Story	Done	Medium	Done	Princy mol Joseph	Renuka Devi M	Renuka Devi M	15/Nov/22 9:44 PM	18/Nov/22 10:12 PM	18/Nov/22 10:12 PM	18/Nov/22 10:12 PM
IBM Project	IP-14	Build the python Flask app	Story	Done	Medium	Done	preethi072002	Renuka Devi M	Renuka Devi M	15/Nov/22 9:44 PM	18/Nov/22 10:12 PM	18/Nov/22 10:12 PM	18/Nov/22 10:12 PM
IBM Project	IP-13	Application Building	Story	Done	Medium	Done	Renuka Devi M	Renuka Devi M	Renuka Devi M	15/Nov/22 9:44 PM	18/Nov/22 10:08 PM	18/Nov/22 10:08 PM	18/Nov/22 10:08 PM
IBM Project	IP-11	API Integration	Story	Done	Medium	Done	Sri Gayathri Devi	Renuka Devi M	Renuka Devi M	15/Nov/22 9:39 PM	18/Nov/22 10:11 PM	18/Nov/22 10:11 PM	18/Nov/22 10:11 PM
IBM Project	IP-10	Save the model	Story	Done	Medium	Done	sivapriyasharon	Renuka Devi M	Renuka Devi M	15/Nov/22 9:39 PM	18/Nov/22 10:11 PM	18/Nov/22 10:11 PM	18/Nov/22 10:11 PM
IBM Project	IP-9	check the metrics of the model	Story	Done	Medium	Done	Princy mol Joseph	Renuka Devi M	Renuka Devi M	15/Nov/22 9:39 PM		18/Nov/22 9:58 PM	18/Nov/22 9:49 PM
IBM Project	IP-8	Choose the appropriate model	Story	Done	Medium	Done	preethi072002	Renuka Devi M	Renuka Devi M	15/Nov/22 9:38 PM	18/Nov/22 10:10 PM	18/Nov/22 10:11 PM	18/Nov/22 10:11 PM
IBM Project	IP-7	Split data into dependent and independent variables	Story	Done	Medium	Done	Renuka Devi M	Renuka Devi M	Renuka Devi M	15/Nov/22 9:32 PM	18/Nov/22 10:06 PM	18/Nov/22 10:06 PM	18/Nov/22 10:06 PM
IBM Project	IP-6	Analyze the dataset	Story	Done	Medium	Done	Sri Gayathri Devi	Renuka Devi M	Renuka Devi M	15/Nov/22 9:31 PM	18/Nov/22 10:10 PM	18/Nov/22 10:10 PM	18/Nov/22 10:10 PM
IBM Project	IP-5	Import required Libraries	Story	Done	Medium	Done	sivapriyasharon	Renuka Devi M	Renuka Devi M	15/Nov/22 9:31 PM	18/Nov/22 10:10 PM	18/Nov/22 10:10 PM	18/Nov/22 10:10 PM
IBM Project	IP-4	Data Preprocessing	Story	Done	Medium	Done	preethi072002	Renuka Devi M	Renuka Devi M	15/Nov/22 9:30 PM	18/Nov/22 10:09 PM	18/Nov/22 10:09 PM	18/Nov/22 10:09 PM
IBM Project	IP-3	Collect the Dataset	Story	Done	Medium	Done	Princy mol Joseph	Renuka Devi M	Renuka Devi M	15/Nov/22 9:26 PM	18/Nov/22 10:09 PM	18/Nov/22 10:09 PM	18/Nov/22 10:09 PM
IBM Project	IP-2	Model Building	Story	Done	Medium	Done	Renuka Devi M	Renuka Devi M	Renuka Devi M	15/Nov/22 9:25 PM	18/Nov/22 10:07 PM	18/Nov/22 10:07 PM	18/Nov/22 10:07 PM
IBM Project	IP-1	Data Collection	Story	Done	Medium	Done	Renuka Devi M	Renuka Devi M	Renuka Devi M	15/Nov/22 9:22 PM	15/Nov/22 9:35 PM	18/Nov/22 9:28 PM	15/Nov/22 9:35 PM

Generated at Fri Nov 18 16:58:19 UTC 2022 by Renuka Devi M using Jira 1001.0.0-SNAPSHOT#100210-sha1:152fb16687f0f0f2c8e8e54c0f41d52385d6cfdd.

6.3 Sprint Delivery Schedule

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	25	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

CHAPTER-7

CODING & SOLUTIONING

7.1 Home Page (Web page)

```
<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;
```

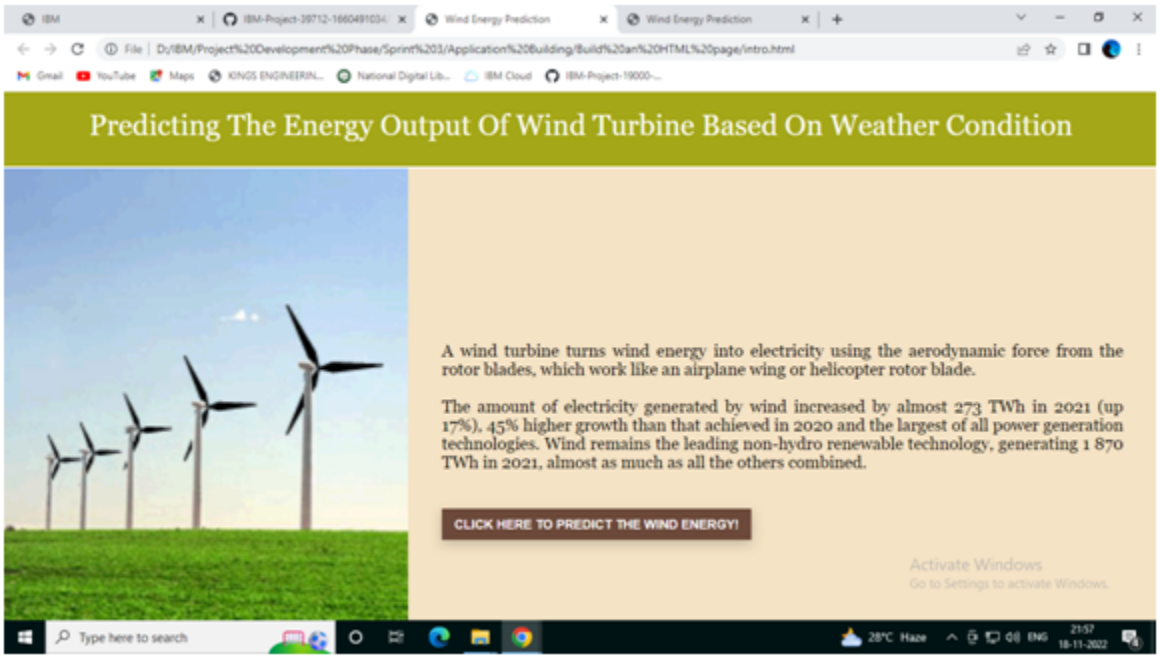
```
                bottom:0px;
                margin:0px;
                left: 35%;
                right: 0%;
                position: fixed;
                padding-left: 40px;
                padding-top:15%;
                padding-right:40px;
                background-color:#f5e3c5;
                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 wing or helicopter rotor blade. <br><br>
```

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 largest of all power generation technologies. Wind remains the leading non-hydro renewable technology, generating 1 870 TWh in 2021, almost as much as all the others combined.

<button type="button" class="myButton" >Click Here To Predict The wind Energy!</button>

</body>

</html>



7.2 Web application

```
<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;
}
            .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{
                 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;

}
        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>


                <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>


                                </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>
```

7.3 Database Schema

```
Date/Time,LV ActivePower (kW),Wind Speed (m/s),Theoretical_Power_Curve (KWh),Wind Direction (°)
01 01 2018 00:00,380.047790527343,5.3113360404968 2,416.328907824861,259.994903564453
01 01 2018 00:10,453.76919555664,5.67216682434082,519.917511061494,268.64111328125
01 01 2018 00:20,306.376586914062,5.21603679656982,390.900015810951,272.564788818359
01 01 2018 00:30,419.645904541015,5.65967416763305,516.127568975674,271.258087158203
01 01 2018 00:40,380.650695800781,5.57794094085693,491.702971953588,265.674285888671
01 01 2018 00:50,402.391998291015,5.60405206680297,499.436385024805,264.57861328125
01 01 2018 01:00,447.605712890625,5.79300785064697,557.372363290225,266.163604736328
01 01 2018 01:10,387.2421875,5.30604982376098,414.898178826186,257.949493408203
01 01 2018 01:20,463.651214599609,5.58462905883789,493.677652137077,253.480697631835
01 01 2018 01:30,439.725708007812,5.52322816848754,475.706782818068,258.72378540039
01 01 2018 01:40,498.181701660156,5.72411584854125,535.841397042263,251.850997924804
01 01 2018 01:50,526.816223144531,5.93419885635375,603.014076510633,265.504699707031
01 01 2018 02:00,710.587280273437,6.54741382598876,824.662513585882,274.23291015625
01 01 2018 02:10,655.194274902343,6.19974613189697,693.472641075637,266.733184814453
01 01 2018 02:20,754.762512207031,6.50538301467895,808.098138482693,266.76040649414
01 01 2018 02:30,790.173278808593,6.63411617279052,859.459020788565,270.493194580078
01 01 2018 02:40,742.985290527343,6.37891292572021,759.434536596592,266.593292236328
01 01 2018 02:50,748.229614257812,6.4466528892517,785.28100987646,265.571807861328
01 01 2018 03:00,736.647827148437,6.41508293151855,773.172863451736,261.15869140625
01 01 2018 03:10,787.246215820312,6.43753099441528,781.7712157188,257.56021118164
01 01 2018 03:20,722.864074707031,6.22002410888671,700.764699868076,255.926498413085
01 01 2018 03:30,935.033386230468,6.89802598953247,970.736626881787,250.012893676757
01 01 2018 03:40,1220.60900878906,7.60971117019653,1315.04892785216,255.985702514648
01 01 2018 03:50,1053.77197265625,7.28835582733154,1151.26574355584,255.444595336914
01 01 2018 04:00,1493.80798339843,7.94310188293457,1497.58372354361,256.407409667968
01 01 2018 04:10,1724.48803710937,8.37616157531738,1752.19966204818,252.41259765625
01 01 2018 04:20,1636.93505859375,8.23695755004882,1668.47070685152,247.979400634765
01 01 2018 04:30,1385.48803710937,7.87959098815917,1461.81579081391,238.609603881835
01 01 2018 04:40,1098.93200683593,7.10137605667114,1062.28503444311,245.095596313476
01 01 2018 04:50,1021.4580078125,6.95530700683593,995.995854606612,245.410202026367
01 01 2018 05:00,1164.89294433593,7.09829807281494,1060.85971215544,235.227905273437
01 01 2018 05:10,1073.33203125,6.95363092422485,995.250960801046,242.872695922851
01 01 2018 05:20,1165.30798339843,7.24957799911499,1132.4168612641,244.835693359375
01 01 2018 05:30,1177.98999023437,7.29469108581542,1154.36530469206,242.48159790039
01 01 2018 05:40,1170.53601074218,7.37636995315551,1194.8430985043,247.97720336914
01 01 2018 05:50,1145.53601074218,7.44855403900146,1231.43070603717,249.682998657226
01 01 2018 06:00,1114.02697753906,7.2392520904541,1127.43320551345,248.401000976562
01 01 2018 06:10,1153.18505859375,7.32921123504638,1171.35504358957,244.621704101562
01 01 2018 06:20,1125.3310546875,7.13970518112182,1080.13908466205,244.631805419921
01 01 2018 06:30,1228.73205566406,7.47422885894775,1244.63353439737,245.785995483398
01 01 2018 06:40,1021.79302978515,7.03317403793334,1030.99268581181,248.652206420898
01 01 2018 06:50,957.378173828125,6.88645505905151,965.683334443832,244.611694335937
01 01 2018 07:00,909.887817382812,6.88782119750976,966.279104864065,235.84829711914
01 01 2018 07:10,1000.95397949218,7.21643209457397,1116.4718990154,232.842697143554
01 01 2018 07:20,1024.47802734375,7.0685977935791,1047.17023059277,229.933197021484
01 01 2018 07:30,1009.53399658203,6.93829584121704,988.451940715539,230.13670349121
01 01 2018 07:40,899.492980957031,6.53668785095214,820.416658585943,234.933807373046
01 01 2018 07:50,725.110107421875,6.18062496185302,686.636942163399,232.837905883789
01 01 2018 08:00,585.259399414062,5.81682586669921,564.927659543473,240.328796386718
01 01 2018 08:10,443.913909912109,5.45015096664428,454.773587146918,238.12629699707
01 01 2018 08:20,565.253784179687,5.81814908981323,565.349093224668,235.80029296875
01 01 2018 08:30,644.037780761718,6.13027286529541,668.823569309414,224.958694458007
01 01 2018 08:40,712.058898925781,6.34707784652709,747.460673422601,216.803894042968
01 01 2018 08:50,737.394775390625,6.34743690490722,747.595109122642,205.785293579101
01 01 2018 09:00,725.868103027343,6.19436883926391,691.546334303948,199.848495483398
01 01 2018 09:10,408.997406005859,4.97719812393188,330.417630427964,207.997802734375
01 01 2018 09:20,628.436828613281,5.95911121368408,611.283836510667,210.954895019531
01 01 2018 09:30,716.1005859375,6.21137619018554,697.649474372052,215.69400024414
01 01 2018 09:40,711.49560546875,6.11145305633544,662.235163012206,220.84260559082
01 01 2018 09:50,838.151916503906,6.45632219314575,789.011422412419,237.065307617187
01 01 2018 10:00,881.062072753906,6.66665792465209,872.739625855708,235.667495727539
01 01 2018 10:10,663.703125,6.16287899017333,680.327891653483,229.329696655273
```

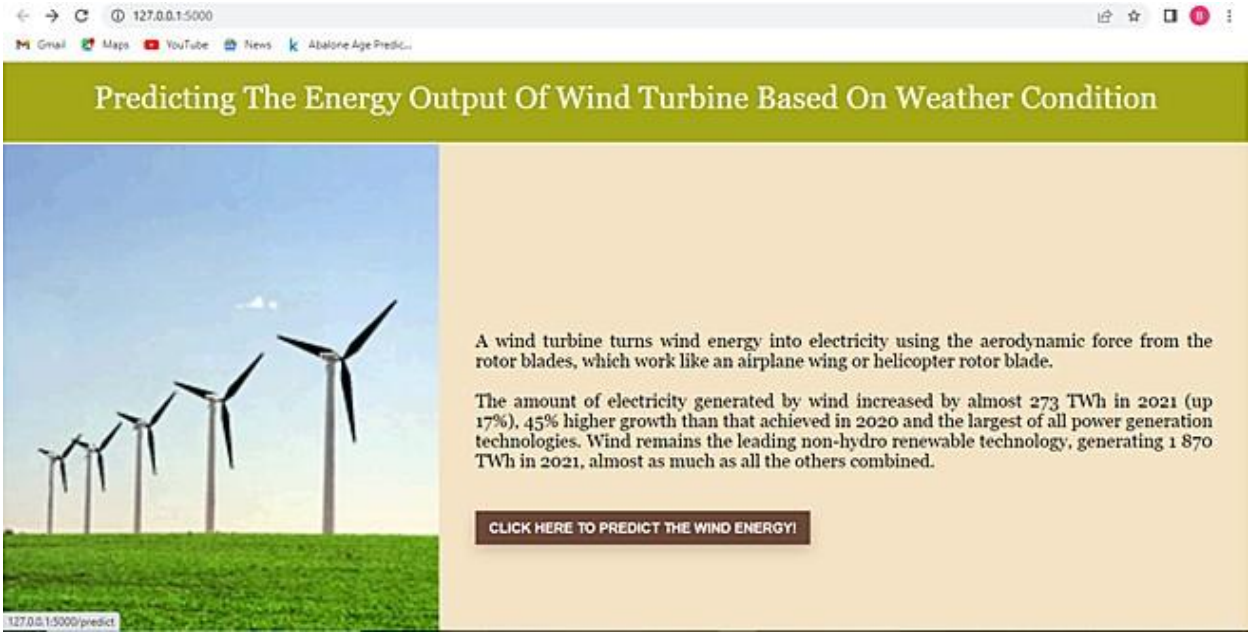
CHAPTER-8

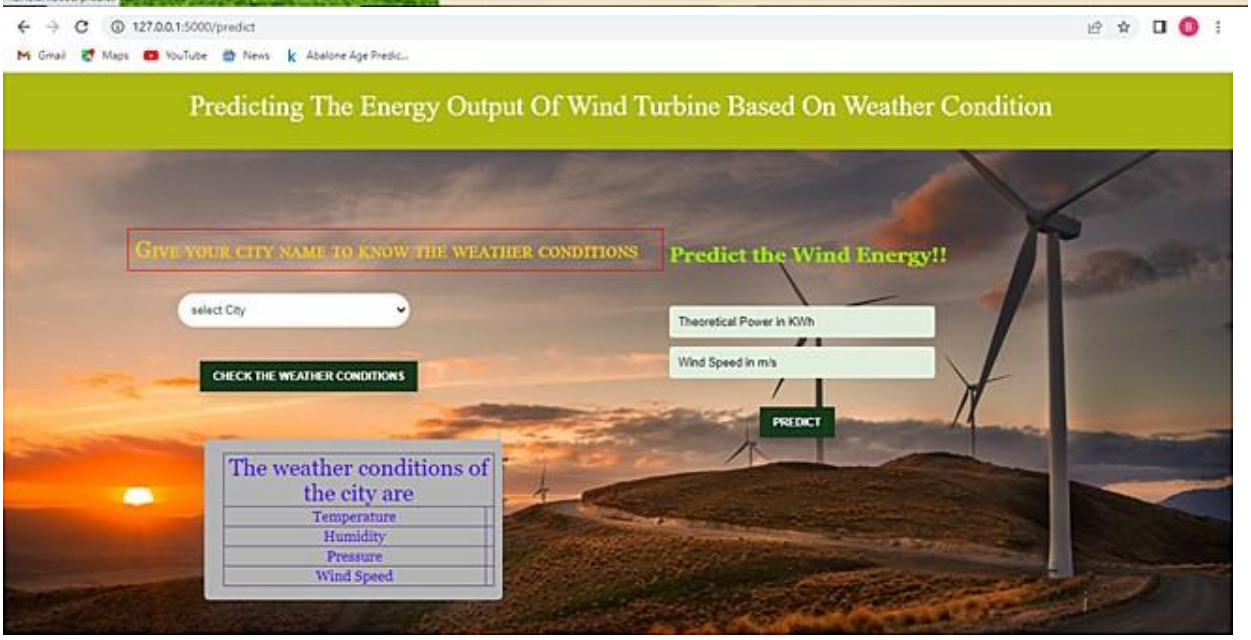
TESTING

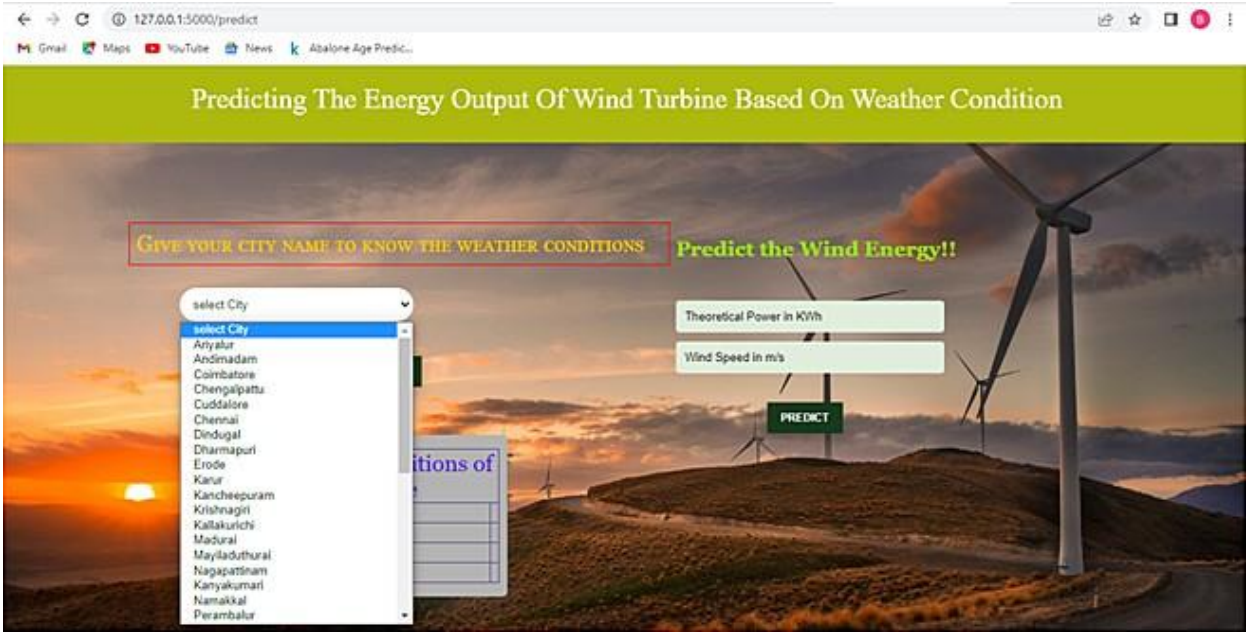
8.1 Test Cases

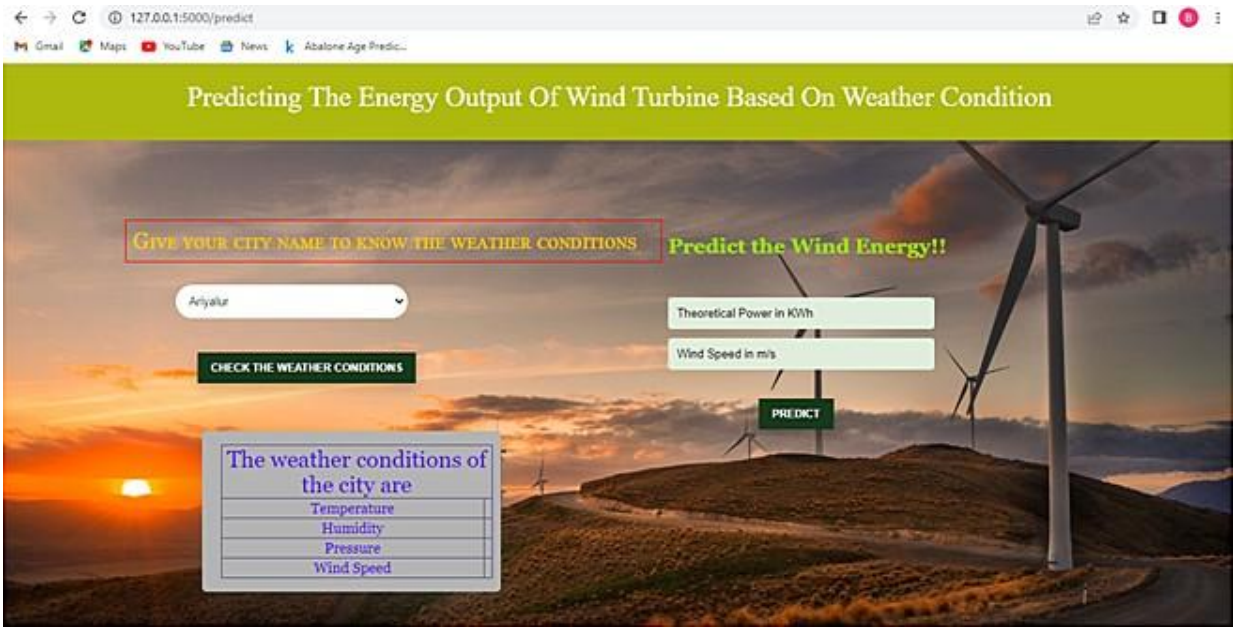
```
import numpy as np
from flask import Flask, request, jsonify, render_template
import joblib
import requests
app = Flask(__name__)
model = joblib.load('Power_Prediction.sav')
@app.route('/')
def home():
    return render_template('intro.html')
@app.route('/predict')
def predict():
    return render_template('predict.html')
@app.route('/windapi',methods=['POST'])
def windapi():
    city=request.form.get('city')
    apikey="a29ea469a6c914ddabcbb20fc4950fb1"
    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()]]
    prediction = model.ppredict(x_test)
    print(prediction)
    output = prediction[0]
    return render_template('predict.html', prediction_text='The energy predicted is {:.2f}
KWh'.format(output))
if __name__ == "__main__":
    app.run(debug=False)
```

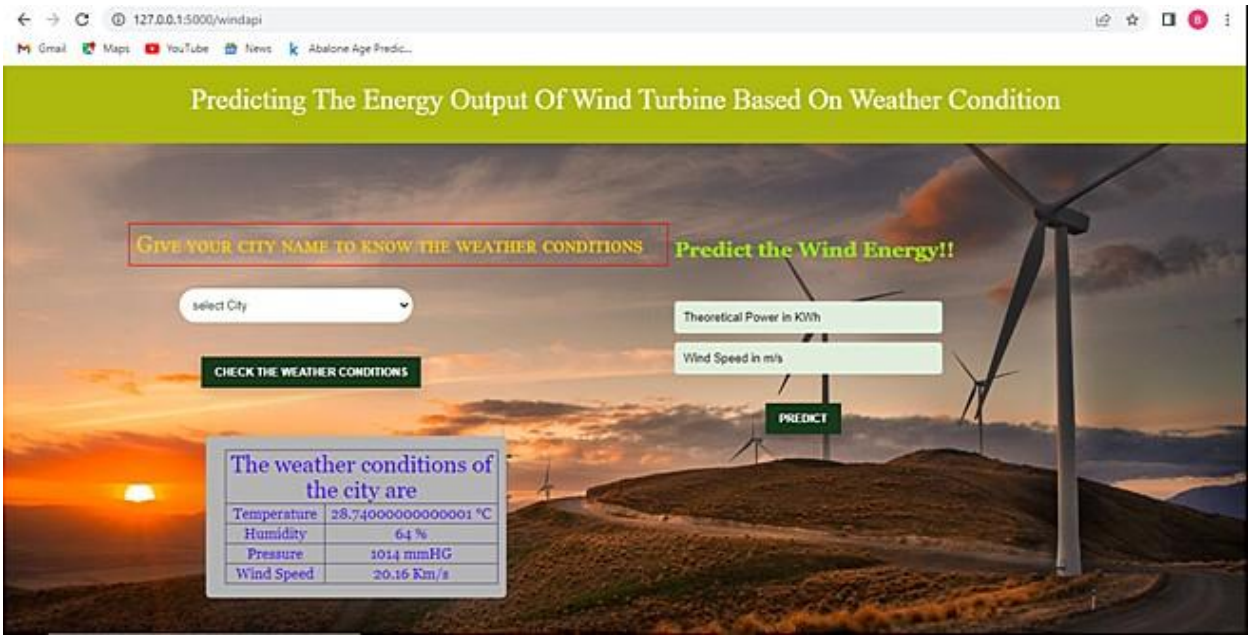
8.2 User Acceptance Testing

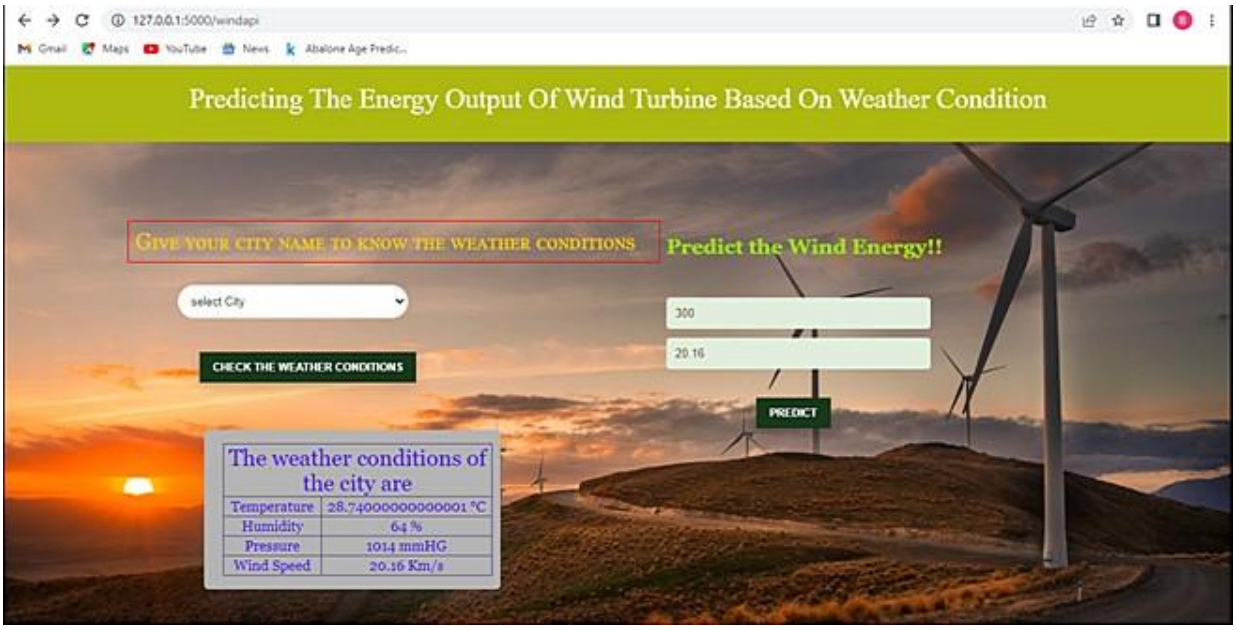


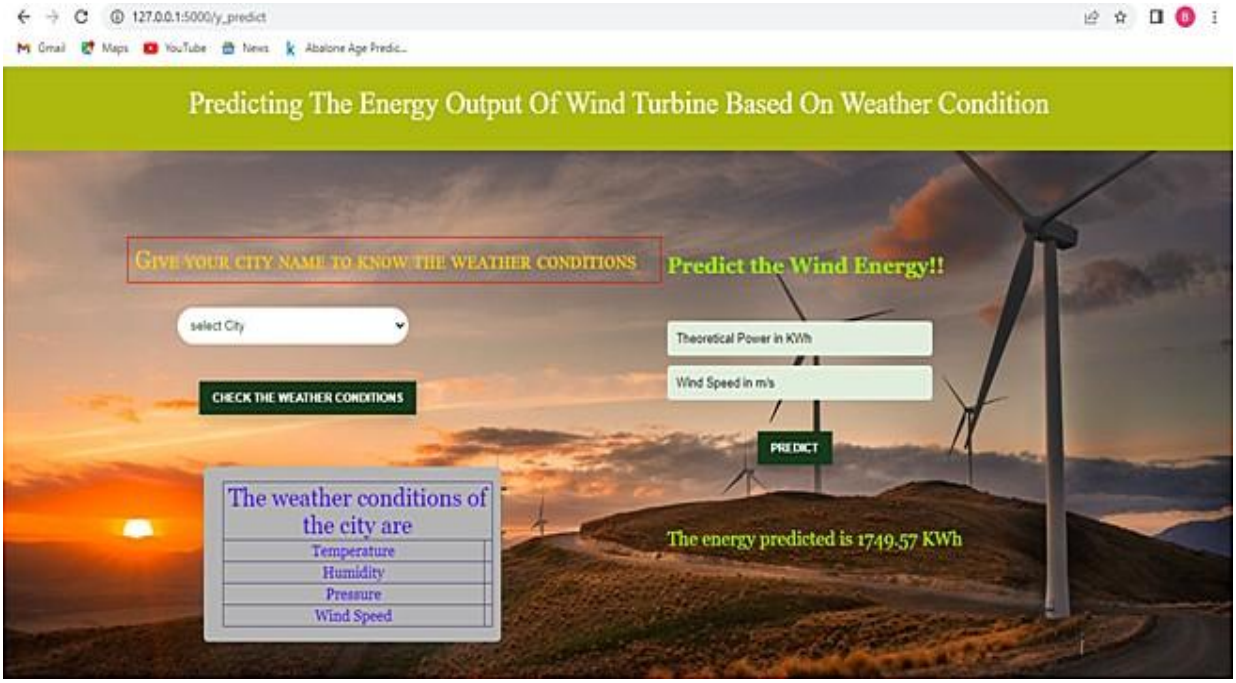



127.0.0.1:5000/predict
Gmail
Maps
YouTube
News
Abalone Age Predic...
Predicting The Energy Output Of Wind Turbine Based On Weather Condition
GIVE YOUR CITY NAME TO KNOW THE WEATHER CONDITIONS
Predict the Wind Energy!!
select City
select City
Ariyalur
Andimadam
Coimbatore
Chengalpattu
Cuddalore
Chennai
Dindugal
Dharmapuri
Erode
Karur
Kancheepuram
Krishnagiri
Kallakurichi
Madurai
Mayiladuthurai
Nagapattinam
Kanyakumari
Namakkal
Perambalur
Theoretical Power in KWh
Wind Speed in m/s
PREDICT


127.0.0.1:5000/predict
Gmail
Maps
YouTube
News
Abalone Age Predic...
Predicting The Energy Output Of Wind Turbine Based On Weather Condition
GIVE YOUR CITY NAME TO KNOW THE WEATHER CONDITIONS
Predict the Wind Energy!!
Ariyalur
CHECK THE WEATHER CONDITIONS
Theoretical Power in KWh
Wind Speed in m/s
PREDICT
The weather conditions of the city are
Temperature
Humidity
Pressure
Wind Speed


127.0.0.1:5000/windapi
Gmail
Maps
YouTube
News
Abalone Age Predic...
Predicting The Energy Output Of Wind Turbine Based On Weather Condition
GIVE YOUR CITY NAME TO KNOW THE WEATHER CONDITIONS
select City
CHECK THE WEATHER CONDITIONS
Predict the Wind Energy!!
Theoretical Power in KWh
Wind Speed in m/s
PREDICT
The weather conditions of the city are
Temperature 28.74000000000001 °C
Humidity 64 %
Pressure 1014 mmHG
Wind Speed 20.16 Km/s


127.0.0.1:5000/windapi
Gmail
Maps
YouTube
News
Abalone Age Predic...
Predicting The Energy Output Of Wind Turbine Based On Weather Condition
GIVE YOUR CITY NAME TO KNOW THE WEATHER CONDITIONS
select City
CHECK THE WEATHER CONDITIONS
Predict the Wind Energy!!
300
20.16
PREDICT
The weather conditions of the city are
Temperature 28.74000000000001 °C
Humidity 64 %
Pressure 1014 mmHG
Wind Speed 20.16 Km/s


127.0.0.1:5000/y_predict
Gmail
Maps
YouTube
News
Abalone Age Predic...
Predicting The Energy Output Of Wind Turbine Based On Weather Condition
GIVE YOUR CITY NAME TO KNOW THE WEATHER CONDITIONS
select City
CHECK THE WEATHER CONDITIONS
The weather conditions of the city are
Temperature
Humidity
Pressure
Wind Speed
Predict the Wind Energy!!
Theoretical Power in KWh
Wind Speed in m/s
PREDICT
The energy predicted is 1749.57 KWh

CHAPTER-9

RESULTS

9.1 Performance Metrics

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='eWp6fT7T7buAFR73-VCjD5d2F1JCvlD0HehuAxvNOSPa',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'deploymlcustommodelonibm-donotdelete-pr-c0wv1gunt29sal'
object_key = 'T1.csv'

body = cos_client.get_object(Bucket=bucket,Key=object_key)['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )

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

Out[2]:

	Date/Time	LV ActivePower (kW)	Wind Speed (m/s)	Theoretical_Power_Curve (KWh)	Wind Direction (°)
0	01 01 2018 00:00	380.047791	5.311336	416.328908	259.994904
1	01 01 2018 00:10	453.769196	5.672167	519.917511	268.641113
2	01 01 2018 00:20	306.376587	5.216037	390.900016	272.564789
3	01 01 2018 00:30	419.645905	5.659674	516.127569	271.258087
4	01 01 2018 00:40	380.650696	5.577941	491.702972	265.674286

In [3]:
```
data = data.rename(columns = {"Date/Time":"Date",
                              "LV ActivePower (kW)":"Active_Power",
                              "Wind Speed (m/s)":"Wind_Speed",
                              "Theoretical_Power_Curve (KWh)":"Theoretical_Power",
                              "Wind Direction (°)" :"Wind_Direction"
                             })
```

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='eWp6fT7T7buAFR73-VCjD5d2F13CvlD0HehuAxvNO5Pa',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'deploymlcustommodelonibm-donotdelete-pr-c0wv1gwnt29sal'
object_key = 'T1.csv'

body = cos_client.get_object(Bucket=bucket,Key=object_key)['Body']
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )

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

Out[2]:

	Date/Time	LV ActivePower (kW)	Wind Speed (m/s)	Theoretical_Power_Curve (KWh)	Wind Direction (°)
0	01 01 2018 00:00	380.047791	5.311336	416.328908	259.994904
1	01 01 2018 00:10	453.769196	5.672167	519.917511	268.641113
2	01 01 2018 00:20	306.376587	5.216037	390.900016	272.564789
3	01 01 2018 00:30	419.645905	5.659674	516.127569	271.258087
4	01 01 2018 00:40	380.650696	5.577941	491.702972	265.674286

In [3]:
```
data = data.rename(columns = {"Date/Time":"Date",
                              "LV ActivePower (kW)":"Active_Power",
                              "Wind Speed (m/s)":"Wind_Speed",
                              "Theoretical_Power_Curve (KWh)":"Theoretical_Power",
                              "Wind Direction (°)" :"Wind_Direction"
                             })
```

In [4]:
```
data.tail() #last 5 rows of the dataset
```

Out[4]:

	Date	Active_Power	Wind_Speed	Theoretical_Power	Wind_Direction
50525	31 12 2018 23:10	2963.980957	11.404030	3397.190793	80.502724
50526	31 12 2018 23:20	1684.353027	7.332648	1173.055771	84.062599
50527	31 12 2018 23:30	2201.106934	8.435358	1788.284755	84.742500
50528	31 12 2018 23:40	2515.694092	9.421366	2418.382503	84.297913
50529	31 12 2018 23:50	2820.466064	9.979332	2779.184096	82.274620

In [5]:
```
#shape of the dataset
data.shape
```

Out[5]:
```
(50530, 5)
```

In [6]:
```
#missing values

data.isna().sum()
```

Out[6]:
```
Date                0
Active_Power        0
Wind_Speed          0
Theoretical_Power   0
Wind_Direction      0
dtype: int64
```

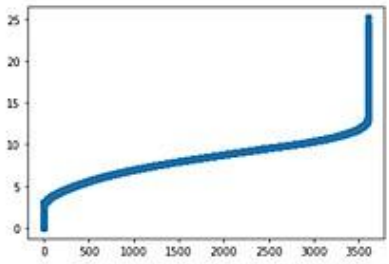
In [7]:
```
#statisticak overview of the data
data.describe().T
```

Out[7]:

	count	mean	std	min	25%	50%	75%	max
Active_Power	50530.0	1307.684332	1312.459242	-2.471405	50.677890	825.838074	2482.507568	3618.732910
Wind_Speed	50530.0	7.557952	4.227166	0.000000	4.201395	7.104594	10.300020	25.206011
Theoretical_Power	50530.0	1492.175463	1368.018238	0.000000	161.328167	1063.776283	2964.972462	3600.000000
Wind_Direction	50530.0	123.687559	93.443736	0.000000	49.315437	73.712978	201.696720	359.997589

In [8]:
```
#scatterplot
plt.scatter(data['Theoretical_Power'],data['Wind_Speed'])
```

Out[8]:



In [9]:
```
#split the data

x=x = data[["Theoretical_Power", "Wind_Speed"]]
y=data["Active_Power"]
```

In [10]:
```
x=x = data[["Theoretical_Power", "Wind_Speed"]].values
y=data["Active_Power"].values
```

In [11]:
```
x
```

Out[11]:
```
array([[ 416.32890782,    5.31133604],
       [ 519.91751106,    5.67216682],
       [ 390.90001581,    5.2160368 ],
       ...,
       [1788.28475526,    8.43535805],
       [2418.38250336,    9.42136574],
       [2779.18409628,    9.97933197]])
```

In [14]:
```
y
```

Out[14]:
```
array([ 380.04779053,  453.76919556,  306.37658691, ..., 2201.10693359,
       2515.6940918 , 2820.46606445])
```

In [15]:
```
from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
```

In [16]:
```
from sklearn.ensemble import RandomForestRegressor

RFR= RandomForestRegressor(n_estimators = 750, max_depth = 4, max_leaf_nodes = 500, random_state = 1)

RFR.fit(x_train,y_train)
```

Out[16]:
```
RandomForestRegressor(max_depth=4, max_leaf_nodes=500, n_estimators=750,
                      random_state=1)
```

In [17]:
```
x_test
```

Out[17]:
```
array([[1.79773993e+03, 8.45079613e+00],
       [3.53917285e+03, 1.21183996e+01],
       [0.00000000e+00, 1.11770403e+00],
       ...,
       [4.98472948e+02, 5.60081100e+00],
       [1.27198717e+03, 7.52718592e+00],
       [1.16382083e+03, 7.31394577e+00]])
```

In [18]:
```
#predcition on the test data
y_pred=RFR.predict(x_test)
y_pred
```

Out[18]:
```
array([1521.38207331, 3278.03226245,   10.5937957 , ...,  371.04787511,
       1014.01006202, 1000.18075254])
```

In [19]:
```
#predccition in the train data
pred=RFR.predict(x_train)
pred
```

Out[19]:
```
array([3355.89202838,   10.5937957 ,  154.75014095, ...,   10.5937957 ,
        775.62647942,  366.46538096])
```

In [20]:
```
#Finding accuracy

from sklearn.metrics import r2_score

acc=r2_score(y_test,y_pred)

acc
```

Out[20]:
```
0.9068180877803967
```

In [21]:
```
!pip install -U ibm-watson-machine-learning
```

```
Requirement already satisfied: ibm-watson-machine-learning in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.0.257)
Requirement already satisfied: pandas<1.5.0,>=0.24.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (1.
3.4)
Requirement already satisfied: ibm-cos-sdk==2.11.* in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (2.11.
0)
Requirement already satisfied: importlib-metadata in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (4.8.2)
Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (0.3.3)
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (0.8.9)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (2.26.0)
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (2022.9.24)
Requirement already satisfied: packaging in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (21.3)
Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (1.26.7)
Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm-watso
n-machine-learning) (2.11.0)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm-watson-
machine-learning) (0.10.0)
Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.*->ibm
-watson-machine-learning) (2.11.0)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk-core==2.11.0->i
bm-cos-sdk==2.11.*->ibm-watson-machine-learning) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm-watson-machine-
learning) (2021.3)
Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm-watson-machine
-learning) (1.20.3)
Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dateutil<3.0.0,>=2.1->ibm-cos-sdk-core=
=2.11.0->ibm-cos-sdk==2.11.*->ibm-watson-machine-learning) (1.15.0)
Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->ibm-watson-machine-
learning) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->ibm-watson-machine-learning) (3.
3)
Requirement already satisfied: zipp>=0.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from importlib-metadata->ibm-watson-machine-learni
ng) (3.6.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from packaging->ibm-watson-machine-
learning) (3.0.4)
```

In [22]:
```
from ibm_watson_machine_learning import APIClient
```

In [23]:
```
wml_credentials = {"url": "https://us-south.ml.cloud.ibm.com",
                   "apikey": "kjYd_9_yy5eOxypn_u5YnMwzBnOFU6owmeF1V3fiGUrh"}
```

In [24]:
```
wml_client = APIClient(wml_credentials)
wml_client.spaces.list()
```

```
Note: 'limit' is not provided. Only first 50 records will be displayed if the number of records exceed 50
------------------------------------  ------  ------------------------
ID                                    NAME    CREATED
3906d4d8-6d6d-40d6-a81e-73437a81424a  models  2022-11-18T10:01:33.456Z
------------------------------------  ------  ------------------------
```

In [26]:
```
space_id = "3906d4d8-6d6d-40d6-a81e-73437a81424a"
```

In [27]:
```
wml_client.set.default_space(space_id)
```

Out[27]:
```
'SUCCESS'
```

In [28]:
```
wml_client.software_specifications.list()
```

```
------------------------------  ------------------------------------  ----
NAME                            ASSET_ID                              TYPE
default_py3.6                   0062b8c9-8b7d-44a0-a9b9-46c416adcbd9  base
kernel-spark3.2-scala2.12       020d69ce-7ac1-5e68-ac1a-31189867356a  base
pytorch-onnx_1.3-py3.7-edt      069ea134-3346-5748-b513-49120e15d288  base
scikit-learn_0.20-py3.6         09c5a1d0-9c1e-4473-a344-eb7b665ff687  base
spark-mllib_3.0-scala_2.12      09f4cff0-90a7-5899-b9ed-1ef348aebdee  base
pytorch-onnx_rt22.1-py3.9       0b848dd4-e681-5599-be41-b5f6fccc6471  base
ai-function_0.1-py3.6           0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda  base
shiny-r3.6                      0e6e79df-875e-4f24-8ae9-62dcc2148306  base
tensorflow_2.4-py3.7-horovod    1092590a-307d-563d-9b62-4eb7d64b3f22  base
pytorch_1.1-py3.6               10ac12d6-6b30-4ccd-8392-3e922c096a92  base
tensorflow_1.15-py3.6-ddl       111e41b3-de2d-5422-a4d6-bf776828c4b7  base
autoai-kb_rt22.2-py3.10         125b6d9a-5b1f-5e8d-972a-b251688ccf40  base
runtime-22.1-py3.9              12b83a17-24d8-5082-900f-0ab31fbfd3cb  base
scikit-learn_0.22-py3.6         154010fa-5b3b-4ac1-82af-4d5ee5abbc85  base
default_r3.6                    1b70aec3-ab34-4b87-8aa0-a4a3c8296a36  base
pytorch-onnx_1.3-py3.6          1bc6029a-cc97-56da-b8e0-39c3880dbbe7  base
kernel-spark3.3-r3.6            1c9e5454-f216-59dd-a20e-474a5cdf5988  base
pytorch-onnx_rt22.1-py3.9-edt   1d362186-7ad5-5b59-8b6c-9d0880bde37f  base
tensorflow_2.1-py3.6            1eb25b84-d6ed-5dde-b6a5-3fbdf1665666  base
spark-mllib_3.2                 20047f72-0a98-58c7-9ff5-a77b012eb8f5  base
tensorflow_2.4-py3.8-horovod    217c16f6-178f-56bf-824a-b19f20564c49  base
runtime-22.1-py3.9-cuda         26215f05-08c3-5a41-a1b0-da66306ce658  base
do_py3.8                        295addb5-9ef9-547e-9bf4-92ae3563e720  base
autoai-ts_3.8-py3.8             2aa0c932-798f-5ae9-abd6-15e0c2402fb5  base
tensorflow_1.15-py3.6           2b73a275-7cbf-420b-a912-eae7f436e0bc  base
kernel-spark3.3-py3.9           2b7961e2-e3b1-5a8c-a491-482c8368839a  base
pytorch_1.2-py3.6               2c8ef57d-2687-4b7d-acce-01f94976dac1  base
spark-mllib_2.3                 2e51f700-bca0-4b0d-88dc-5c6791338875  base
pytorch-onnx_1.1-py3.6-edt      32983cea-3f32-4400-8965-dde874a8d67e  base
spark-mllib_3.0-py37            36507ebe-8770-55ba-ab2a-eafe787600e9  base
spark-mllib_2.4                 390d21f8-e58b-4fac-9c55-d7ceda621326  base
autoai-ts_rt22.2-py3.10         396b2e83-0953-5b86-9a55-7ce1628a406f  base
```

In [29]:
```
MODEL_NAME = 'Power_Prediction'
DEPLOYMENT_NAME = 'Power_deploy'
POWER_MODEL = RFR
```

In [30]:
```
# Set Python Version
software_spec_uid = wml_client.software_specifications.get_id_by_name('runtime-22.1-py3.9')
```

In [31]:
```
# Setup model meta
model_props = {
    wml_client.repository.ModelMetaNames.NAME: MODEL_NAME,
    wml_client.repository.ModelMetaNames.TYPE: 'scikit-learn_1.0',
    wml_client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_spec_uid
}
```

In [32]:
```
#Save model
model_details = wml_client.repository.store_model(
    model=POWER_MODEL,
    meta_props=model_props,
    training_data=x_train,
    training_target=y_train
)
```

In [33]:
```
model_details
```

Out[33]:

```
{'entity': {'hybrid_pipeline_software_specs': [],
  'label_column': 'l1',
  'schemas': {'input': [{'fields': [{'name': 'f0', 'type': 'float'},
     {'name': 'f1', 'type': 'float'}],
    'id': '1',
    'type': 'struct'}],
   'output': []},
  'software_spec': {'id': '12b83a17-24d8-5082-900f-0ab31fbfd3cb',
   'name': 'runtime-22.1-py3.9'},
  'type': 'scikit-learn_1.0'},
 'metadata': {'created_at': '2022-11-18T13:41:54.396Z',
  'id': 'e229181a-d72e-4b99-b60b-d7b6cdfad412',
  'modified_at': '2022-11-18T13:41:57.168Z',
  'name': 'Power_Prediction',
  'owner': 'IBMid-6610042OCF',
  'resource_key': 'd564440d-648b-4dac-8f1d-67300f7eb235',
  'space_id': '3906d4d8-6d6d-40d6-a81e-73437a81424a'},
 'system': {'warnings': []}}
```

CHAPTER-10

ADVANTAGES & DISADVANTAGES

Advantages of wind energy

1. It is free - As the fuel used in the working of the wind turbine is just wind and wind alone, there is no expense involved on this account.
2. It is the cleanest energy source - Unlike fossil fuels, there is no burning involved in energy generation using wind power. As wind rotates the turbines, the wind installation produces electricity.
3. It is renewable and sustainable - The wind blows all the time whether we are harnessing it to generate electricity or not. As the sun heats various components on the earth's surface differently, it results in wind.
4. It has the highest conversion rate - Conversion rate or energy efficiency refers to the percentage of electricity generated using the input energy, here it is wind energy. Among all the renewable and non-renewable energy sources, wind power tops the table
5. It can share the landscape - Wind turbines can coexist with crops on farmland or ranches. As they need to be set up at some distance apart, the remaining landscape can be used for farming or grazing purposes. This is one clear advantage wind turbines have over solar panels
6. It has made big strides in technology - We have come a long way since the use of wind energy in sailboats and grinding flours centuries ago. Now, with the added emphasis on renewable energy, wind power is getting a lot of attention.
7. It reduces our dependence on fossil fuels - As we all know, fossil fuels are wreaking havoc on our planet, with more carbon emissions and atmospheric pollution.

Disadvantages of wind energy

1. It has a high upfront cost - The wind turbines and other equipment required to harness wind energy and generate electricity don’t come cheap. These massive structures are expensive. Setting up a wind energy installation involves a substantial initial investment.
2. It poses a threat to wildlife - As wind turbines have blades moving all the time, unlike solar power, they may cause harm to animals, birds, and marine life. The positioning of wind turbines may disrupt the habitats and migratory paths of various members of the animal kingdom. Birds may fly into the rotating turbines unawares, injuring or killing them
3. It creates noise pollution - Unlike solar panels, wind turbines have moving blades. This means they make a whirring sound whenever the blade is turning. The whirring sound of the turbines can be highly upsetting for all life in the vicinity, affecting their mental as well as physical health.
4. It is not reliable or predictable - It is hard to predict when the wind will blow. Unless the wind blows, the installation cannot generate electricity. This means the wind installation is at the complete
5. It is a blot on the landscape - Or seascape for an offshore wind installation. Wind turbines are massive structures, hundreds of feet tall so that they can capture the blowing wind effectively.

CHAPTER-11

CONCLUSION

We started with the aim of improving the predictions of power generated using wind energy and we have achieved that using Watson studio as a machine learning model and performing model optimization on it. We have also observed that if the wind speed is less than 4 m/s the power generated by the system is zero. LSTM is not able to learn this pattern as this is not the part which it can understand in time series analysis. So, if a hybrid new model is created which can work as the combination of Decision Tree/Random Forest and LSTM we can improve upon these results as well.

Wind energy is cost-effective, has low operating costs, creates jobs, and helps in making countries self-reliant on the energy front. On the other hand, the environmental impact of wind power cannot be ignored. Setting up wind installations may not be the most profitable way to use the land space. The need of the hour is the development of newer technologies with the focus on lowering the cost, improving reliability, and raising energy efficiency. Regional deployment issues need to be resolved, the resource area needs to be expanded, and infrastructure and manufacturing facilities need to be developed further. Minimizing the environmental impact is to be afforded high priority.

Large-scale deployment of wind energy such as wind farms may be postponed so that technological advancements have a chance to catch up with the demands.

CHAPTER-12

FUTURE SCOPE

The power request in India is persistently expanding at a high rate and India has restricted power creation for satisfying the interest in this manner. Research, improvement, creation and show have been completed eagerly in India to search out a potential response to the perpetual issue of intensity lack for as long as three decades. India has gotten the use of an assortment of sustainable power source advances to be utilized in various divisions as well. There are sufficient open doors with great topography and geology with the gigantic client base and augmenting hole among request and flexibly. Mechanical progression, appropriate administrative approaches, charge discounts, effectiveness improvement in result to R&D endeavors are the couple of pathways to vitality and condition preservation and it will ensure that these huge, clean asset bases are abused as fast and cost-viably as could be expected under the circumstances with inexhaustible assets. Therefore wind energy can be the future scope for energy sources.

CHAPTER-13

APPENDIX

SOURCE CODE

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

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud
account.
API_KEY = "kjYd_9_yy5eOxypn_u5YnMwzBnOFU6owmeF1V3fiGUrh"
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="a29ea469a6c914ddabcbb20fc4950fb1"
    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)
    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/a587ce3b-4b42-4afd-9826-
a1c8bcfed060/predictions?version=2022-11-18', json=payload_scoring,
headers={'Authorization': 'Bearer ' + mltoken})
    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)
```

GITHUB & PROJECT DEMO LINK

Demo Video Link -
https://drive.google.com/file/d/1XMQKqK0B8ib3IhTijfOR6BEIDRzjZnz3/view?usp=share_link

Git Hub Link - https://github.com/IBM-EPBL/IBM-Project-39712-1660491034