M.KUMARASAMY
COLLEGE OF ENGINEERING
NAAC Accredited Autonomous Institution
Approved by AICTE & Affiliated to Anna University
ISO 9001:2015 & ISO 14001:2015 Certified Institution
Thalavapalayam, Karur – 639 113.

A Project Report
on

PREDICITING THE ENERGY OUTPUT OF WIND TURBINE BASED ON WEATHER CONDITION

Submitted in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

Under the Guidance of

Mr. V. RAJESHRAM M.E.,
Assistant Professor/CSE

Submitted by

TEAM ID: PNT2022TMID15396

927619BCS4029 - ELAVARASAN S
927619BCS4030 - GANDHIKUMAR S
927619BCS4036 - GOWSIK M P
927619BCS4040 - HARISH KUMAR S

NAALAIYA THIRAN – EXPERIENTIAL PROJECT BASED LEARNING INITIATIVE

18CSE040L - PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTERPRENURSHIP

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
M.KUMARASAMY COLLEGE OF ENGINEERING
(Autonomous)
Karur - 639 113

November, 2022

TABLE OF CONTENTS

CHAPTER.No.	TITLE	PAGE No.

ABSTRACT

In this paper, we take a computer science perspective on energy prediction based on weather data and analyze the important parameters as well as their correlation on the energy output. Most of the countries around the world are facing huge environmental impact, and the most promising solution to mitigate these is the use of renewable energy, especially wind power. Though, the use of offshore wind energy is rapidly increasing to meet the elevating electricity demand. The researchers and policymakers have become aware of the importance of providing near accurate prediction of output power. Wind energy is tied to variabilities of weather patterns, especially wind speed, which are irregular in climates with erratic weather conditions. In this paper, we predicted the output power of the wind turbines.

CHAPTER 1
INTRODUCTION

1.1 PROJECT OVERVIEW

Wind energy plays an increasing role in the supply of energy world-wide.The energy output of a wind farm is highly dependent on the weather conditions present at its site. If the output can be predicted more accurately, energy suppliers can coordinate the collaborative production of different energy sources more efficiently to avoid costly overproduction. In this paper, we take a computer science perspective on energy prediction based on weather data and analyze the important parameters as well as their correlation on the energy output. Most of the countries around the world are facing huge environmental impact, and the most promising solution to mitigate these is the use of renewable energy, especially wind power. Though, the use of offshore wind energy is rapidly increasing to meet the elevating electricity demand. The researchers and policymakers have become aware of the importance of providing near accurate prediction of output power. Wind energy is tied to variabilities of weather patterns, especially wind speed, which are irregular in climates with erratic weather conditions. In this paper, we predicted the output power of the wind turbines.

1.2 PURPOSE

Wind speed/power has received increasing attention around the earth due to its renewable nature as well as environmental friendliness. With the global installed wind power capacity rapidly increasing, the wind industry is growing into a large-scale business. Reliable short-term wind speed forecasts play a practical and crucial role in wind energy conversion systems, such as the dynamic control of wind turbines and power system scheduling. A precise forecast needs to overcome problems of variable energy production caused by fluctuating weather conditions. Power generated by wind is highly dependent on the wind speed. Though it is highly non-linear, wind speed follows a certain pattern over a certain period of time. We exploit this time series pattern to gain useful information and use it for power prediction. Wind energy plays an increasing role in the supply of energy world-wide. The energy output of a wind farm is highly dependent on the weather conditions present at its site. If the output can be predicted more accurately, energy suppliers can coordinate the collaborative production of different energy sources more efficiently to avoid costly overproduction. In this paper, we predict energy prediction based on weather data and analyse the important parameters as well as their correlation on the energy output.

CHAPTER 2
LITERATURE SURVEY

2.1 EXISTING PROBLEM

S.No.	Year	Author	Title	Algorithm	Disadvantages
1.	2012	A. M. Foley, P. G.Leahy ,A.Marvuglia, and E.J. McKeogh.	Current methods and advances in forecasting of wind power generation	Data-Mining	Some of the data science related properties and terms are not well defined in detail
2.	2011	O. Kramer and F. Gieseke.	Analysis of wind energy time series with kernel methods and neural networks.	Support Vector Regression	Concentrated mostly on the wind power based on neural networks and kernel-based methods rather than including any other methods of data science and machine learning like support vector regression.
3.	2011	O. Kramer and F. Gieseke.	Short-term wind energy forecasting using Support vector regression.	Support Vector Regression	Should have worked out this paper based on the advanced techniques such as Neural networks
4.	2010	M. Schmidt and H. Lipson.	Age-fitness Pareto optimization.	ALPS Algorithm, Pareto Algorithm, Deterministic Crowding Algorithm.	As it is a multi-objective approach if failure or interruption occurs in any one of the objectives then the system fails

5.	2009	A. Kusiak, H. Zheng, and Z. Song.	Short-term prediction of wind farm power: A data mining approach.	Data-Mining	Wind speed can be predicted fairly accurately based on its historical values; however, the power cannot be accurately determined given a power curve model and the predicted wind speed.

Table: 2.1 - Existing Problem

2.2 REFERENCES

1. A. M. Foley, P. G. Leahy,A.Marvuglia, and E.J. McKeogh, "Current methods and advances in forecasting of wind power generation", July 2012.

2. O. Kramer and F. Gieseke,"Analysis of wind energy time series with kernel methods and neural networks", Feb 2011.

3. O. Kramer and F. Gieseke."Short-term wind energy forecasting using Support vector regression",May 2011.

4. M. Schmidt and H. Lipson"Age-fitness Pareto optimization", Oct 2010.

5. A. Kusiak, H. Zheng, and Z. Song "Short-term prediction of wind farm power: A data mining approach ", Dec 2009.

2.3 PROBLEM STATEMENT DEFINITION

The prediction of the variation in wind speed with height, the variation in wind speed over the site area and the wake interaction between wind turbines are calculated within a bespoke suite of computer programmes, which are specifically designed to facilitate accurate predictions of wind farm energy production. The use of such tools allows the energy production of different options of layout, turbine type and hub height to be established rapidly once models are set up. The wind speed on a site has a very powerful effect on the economics of a wind farm and wind provides both the fuel to generate electricity and, potentially, loads that can destroy the turbines. This part describes how it can be quantified, harnessed and put to work in an economic and predictable manner. This area of the wind farm energy calculation is in need of the greatest level of fundamental research and development. Flow models that can be used in commercial wind farm development have to be quick to execute.

CHAPTER 3
IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

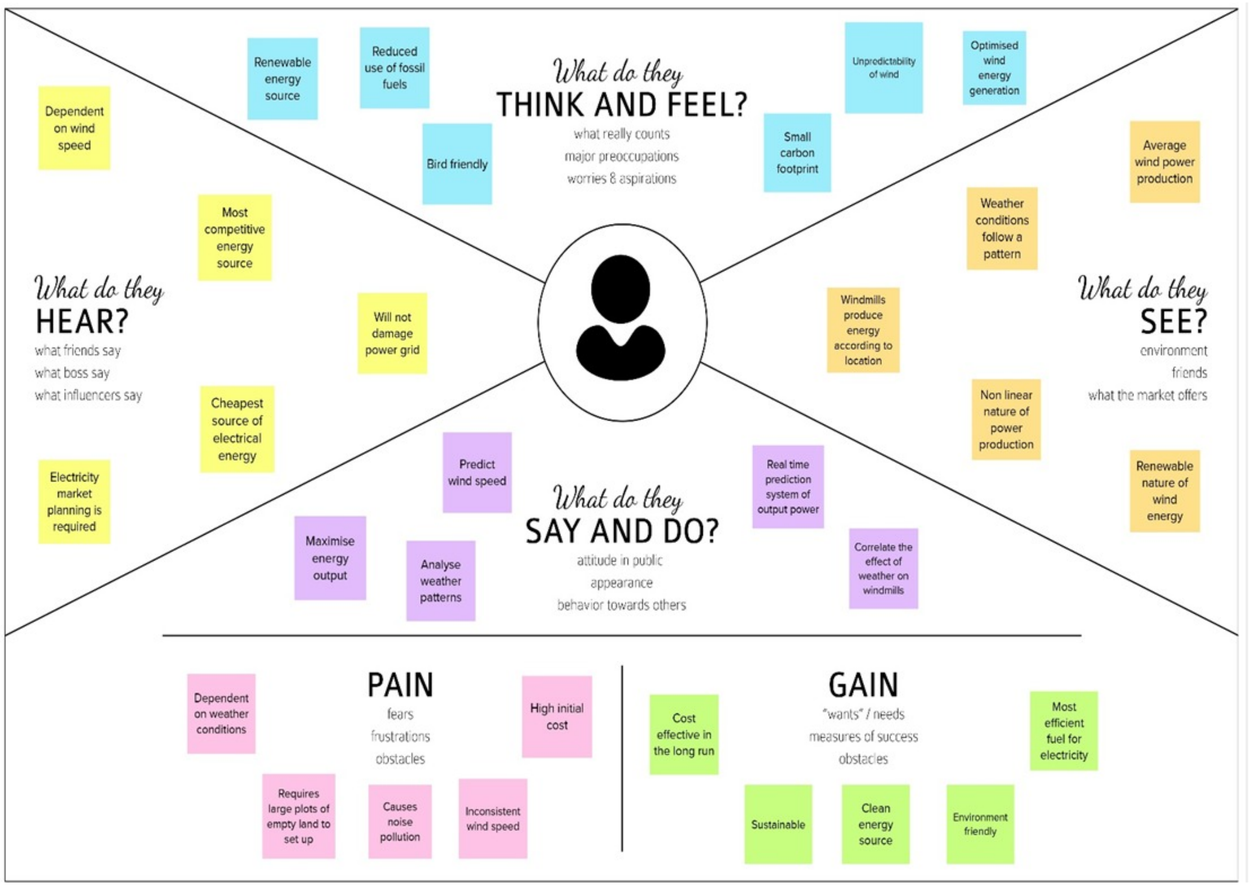
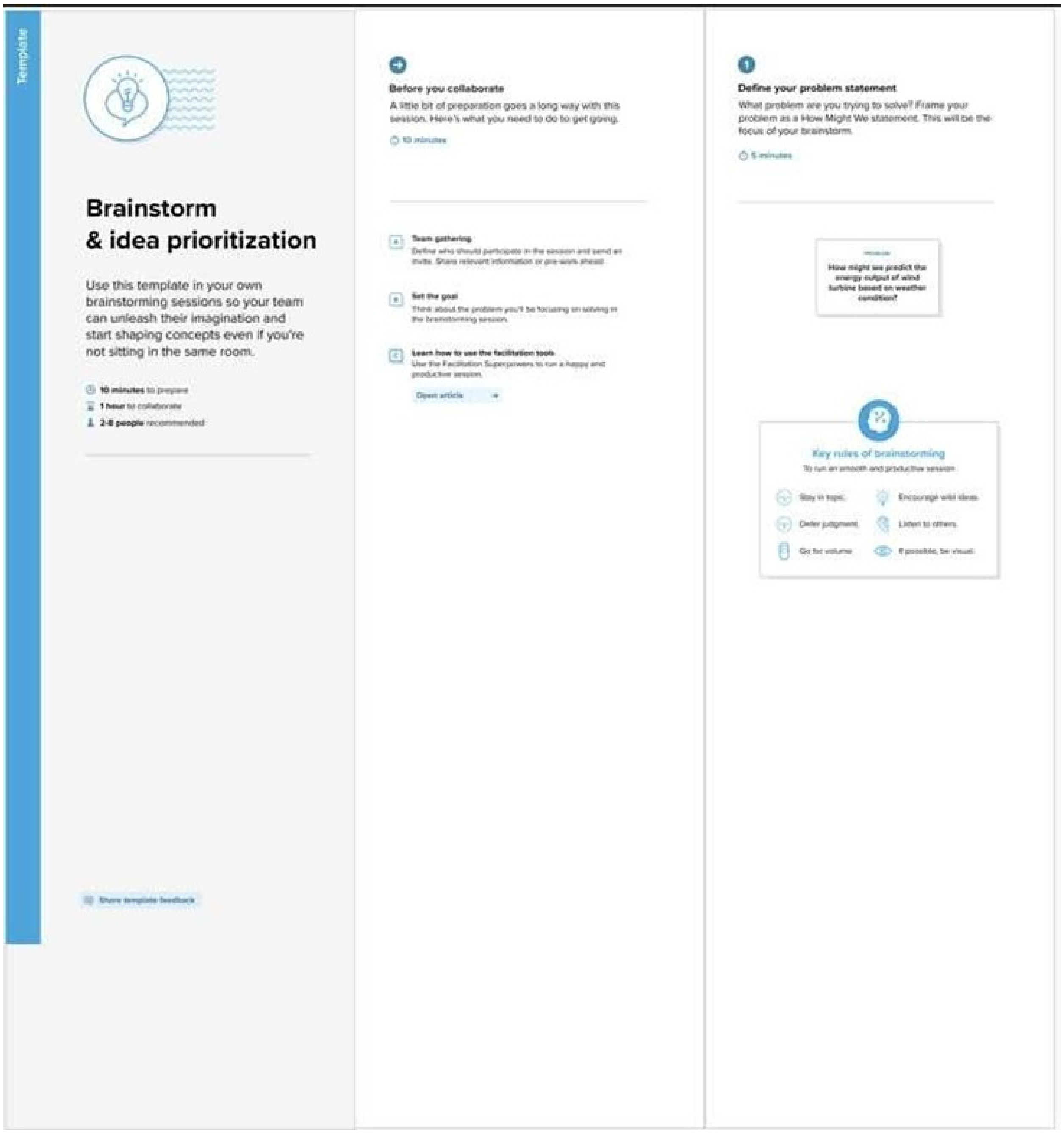

What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
Renewable energy source
Reduced use of fossil fuels
Bird friendly
Unpredictability of wind
Optimised wind energy generation
Small carbon footprint
What do they HEAR?
what friends say
what boss say
what influencers say
Dependent on wind speed
Most competitive energy source
Will not damage power grid
Cheapest source of electrical energy
Electricity market planning is required
What do they SEE?
environment
friends
what the market offers
Average wind power production
Weather conditions follow a pattern
Windmills produce energy according to location
Non linear nature of power production
Renewable nature of wind energy
What do they SAY AND DO?
attitude in public
appearance
behavior towards others
Predict wind speed
Maximise energy output
Analyse weather patterns
Real time prediction system of output power
Correlate the effect of weather on windmills
PAIN
fears
frustrations
obstacles
Dependent on weather conditions
High initial cost
Requires large plots of empty land to set up
Causes noise pollution
Inconsistent wind speed
GAIN
"wants" / needs
measures of success
obstacles
Cost effective in the long run
Most efficient fuel for electricity
Sustainable
Clean energy source
Environment friendly

Figure: 3.1 - Empathy Map

3.2 IDEATION AND BRAINSTORMING

Step - 1: Team Gathering, Collaboration and Select the Problem Statement


Template

Brainstorm
& idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Share template feedback

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article

1 **Define your problem statement**

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

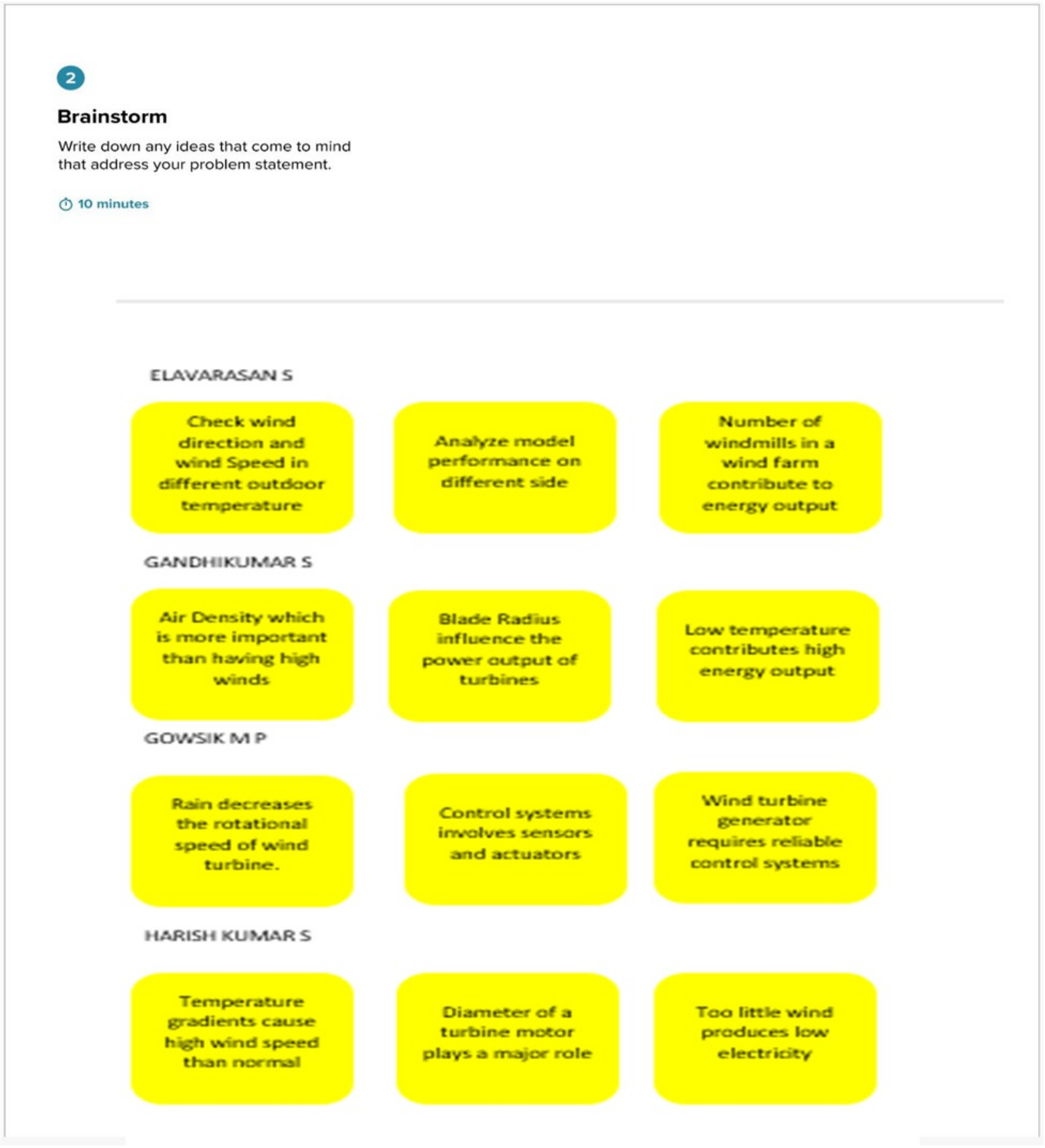
PROBLEM
How might we predict the energy output of wind turbine based on weather condition?

Key rules of brainstorming
To run an smooth and productive session

Stay in topic.
Encourage wild ideas.
Defer judgment.
Listen to others.
Go for volume.
If possible, be visual.

Figure: 3.2 - Team collaboration and problem statement

Step - 2: Brainstorm, Idea Listing and Grouping



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

ELAVARASAN S

Check wind direction and wind Speed in different outdoor temperature

Analyze model performance on different side

Number of windmills in a wind farm contribute to energy output

GANDHIKUMAR S

Air Density which is more important than having high winds

Blade Radius influence the power output of turbines

Low temperature contributes high energy output

GOWSIK M P

Rain decreases the rotational speed of wind turbine.

Control systems involves sensors and actuators

Wind turbine generator requires reliable control systems

HARISH KUMAR S

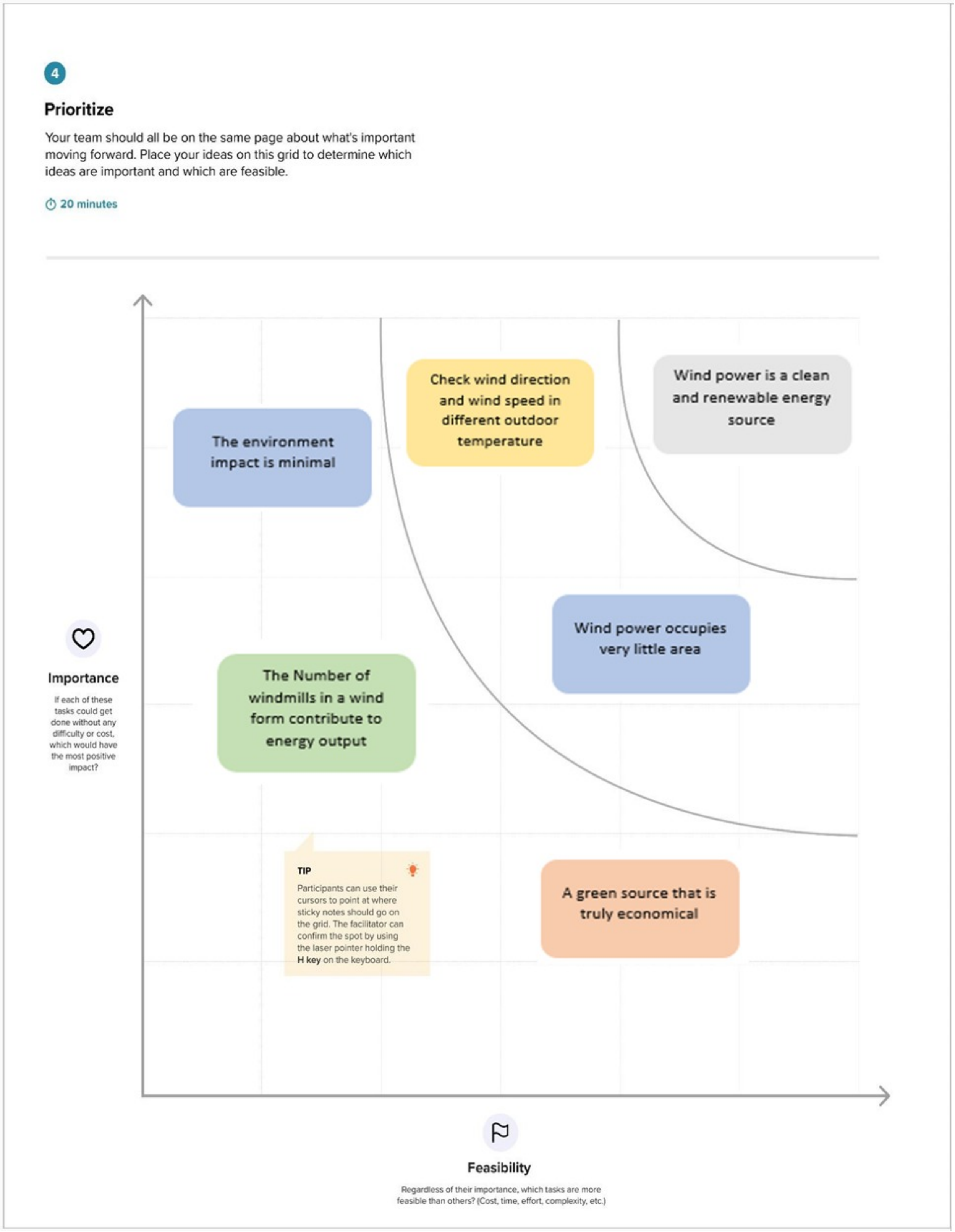
Temperature gradients cause high wind speed than normal

Diameter of a turbine motor plays a major role

Too little wind produces low electricity

Figure: 3.3 - Brainstorming

Step - 3: Idea Prioritization


4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

The environment impact is minimal

Check wind direction and wind speed in different outdoor temperature

Wind power is a clean and renewable energy source

Wind power occupies very little area

The Number of windmills in a wind form contribute to energy output

A green source that is truly economical

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.

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.)

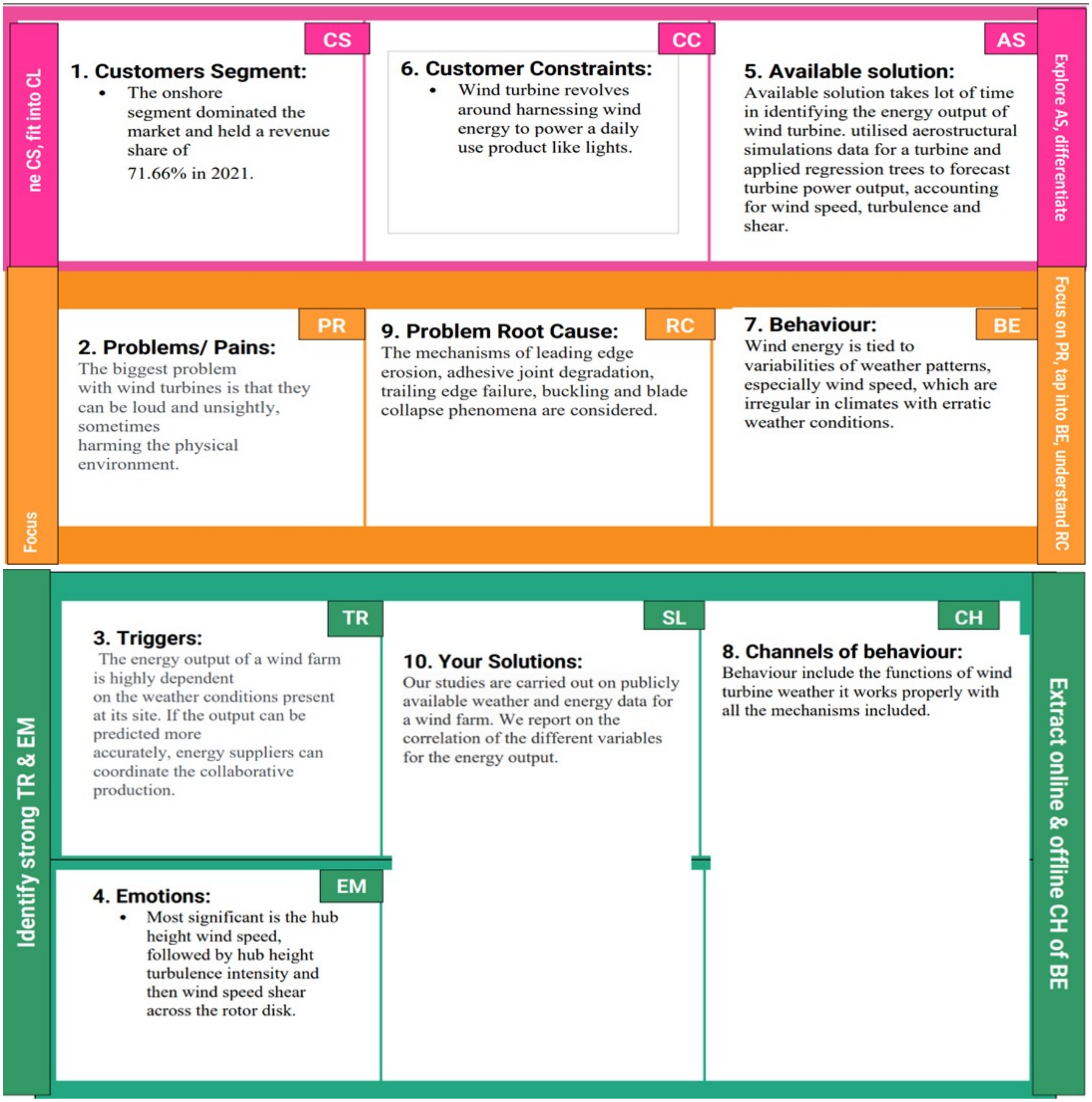
Figure: 3.3 - Idea Prioritization

3.3 PROPOSED SOLUTION

S.No.	PARAMETER	DESCRIPTION
1.	Problem Statement (Problem to be solved)	The wind speed on a site has a very powerful effect on the economics of a wind farm and wind provides both the fuel to generate electricity and, potentially, loads that can destroy the turbines.
2.	Idea / Solution description	One of the easiest solutions to predict the wind speed using Machine Learning techniques.
3.	Novelty / Uniqueness	This project provides the best accuracy for predicting the wind energy level.
4.	Social Impact / Customer Satisfaction	It helps to identify the speed of wind in effective way, reduce the cost and user friendly.
5.	Scalability of the Solution	This project can be improved by giving economy suggestion for energy sectors.

Table: 3.1 - Proposed Solution

3.4 PROBLEM SOLUTION FIT



ne CS, fit into CL

CS

1. Customers Segment:

- The onshore segment dominated the market and held a revenue share of 71.66% in 2021.

CC

6. Customer Constraints:

- Wind turbine revolves around harnessing wind energy to power a daily use product like lights.

AS

5. Available solution:

Available solution takes lot of time in identifying the energy output of wind turbine. utilised aerostructural simulations data for a turbine and applied regression trees to forecast turbine power output, accounting for wind speed, turbulence and shear.

Explore AS, differentiate

Focus

PR

2. Problems/ Pains:

The biggest problem with wind turbines is that they can be loud and unsightly, sometimes harming the physical environment.

RC

9. Problem Root Cause:

The mechanisms of leading edge erosion, adhesive joint degradation, trailing edge failure, buckling and blade collapse phenomena are considered.

BE

7. Behaviour:

Wind energy is tied to variabilities of weather patterns, especially wind speed, which are irregular in climates with erratic weather conditions.

Focus on PR, tap into BE, understand RC

Identify strong TR & EM

TR

3. Triggers:

The energy output of a wind farm is highly dependent on the weather conditions present at its site. If the output can be predicted more accurately, energy suppliers can coordinate the collaborative production.

EM

4. Emotions:

- Most significant is the hub height wind speed, followed by hub height turbulence intensity and then wind speed shear across the rotor disk.

SL

10. Your Solutions:

Our studies are carried out on publicly available weather and energy data for a wind farm. We report on the correlation of the different variables for the energy output.

CH

8. Channels of behaviour:

Behaviour include the functions of wind turbine weather it works properly with all the mechanisms included.

Extract online & offline CH of BE

Figure: 3.4 - Problem Solution Fit

CHAPTER 4
REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

FR No.	**Functional Requirement (Epic)**	**Sub Requirement (Story / Sub-Task)**
FR-1	Upload Data	The user can upload the pervious data. The data should be in structure data format.
FR-2	Process the Data	Raw data prepare for data processing Procedure.
FR-3	Train the Data	The Processed data is then transformed into a format that is needed by the data model
FR-4	Test the Data	The trained model creates a feature map of the uploaded data and predicts the output
FR-5	Determine and predict the output	The predicted output is then analyzed and converted to a user-friendly language
FR-6	Display the output	The analyzed result is then displayed to the user

Table: 4.1 - Functional Requirements

4.2 NON-FUNCTIONAL REQUIREMENTS

NFR No.	Non-Functional Requirement	Description
NFR-1	Usability	Datasets of all the model is used to detecting the speed that present in the data.
NFR-2	Security	The information belongs to the user and energy sector are secured highly.
NFR-3	Reliability	It is important for predicting the speed from the data.
NFR-4	Performance	The performance is based on the technology used for output prediction
NFR-5	Availability	It is available for all user to predict the speed of the wind.
NFR-6	Scalability	Increasing the prediction of the speed of the wind.

Table: 4.2 - Non Functional Requirements

CHAPTER 5
PROJECT DESIGN

5.1 DATAFLOW DIAGRAMS

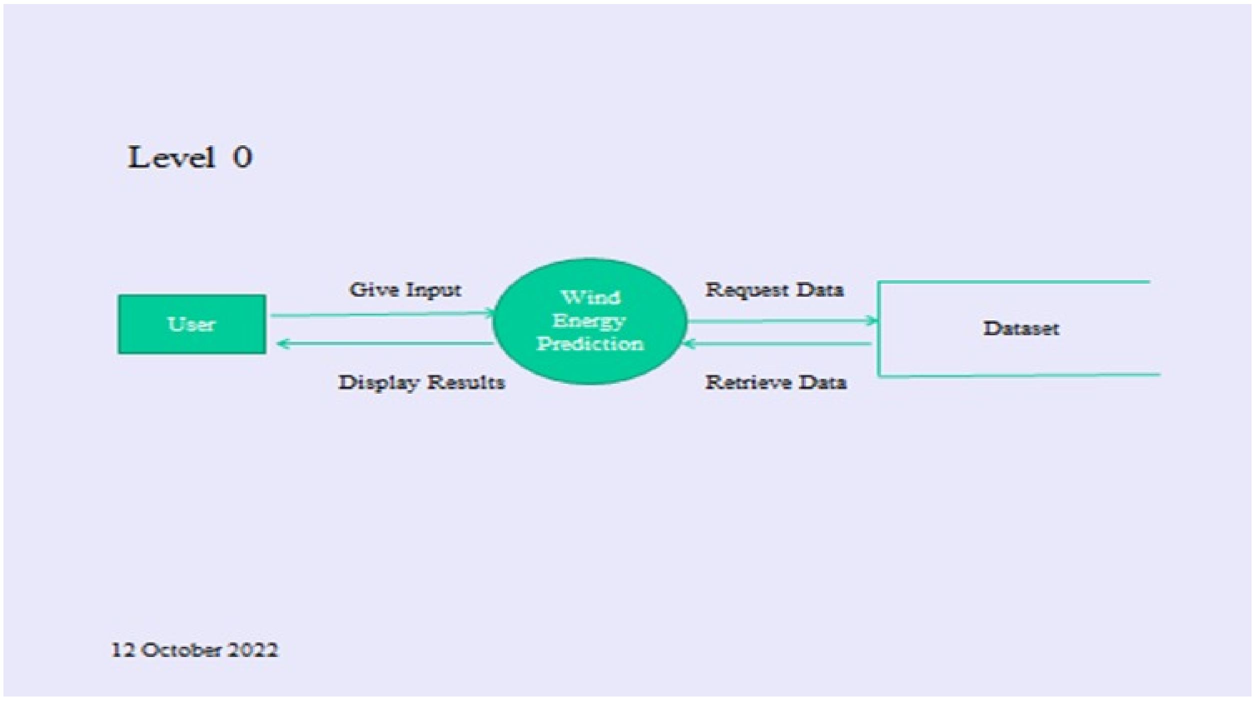


Figure: 5.1 - Dataflow Level 0

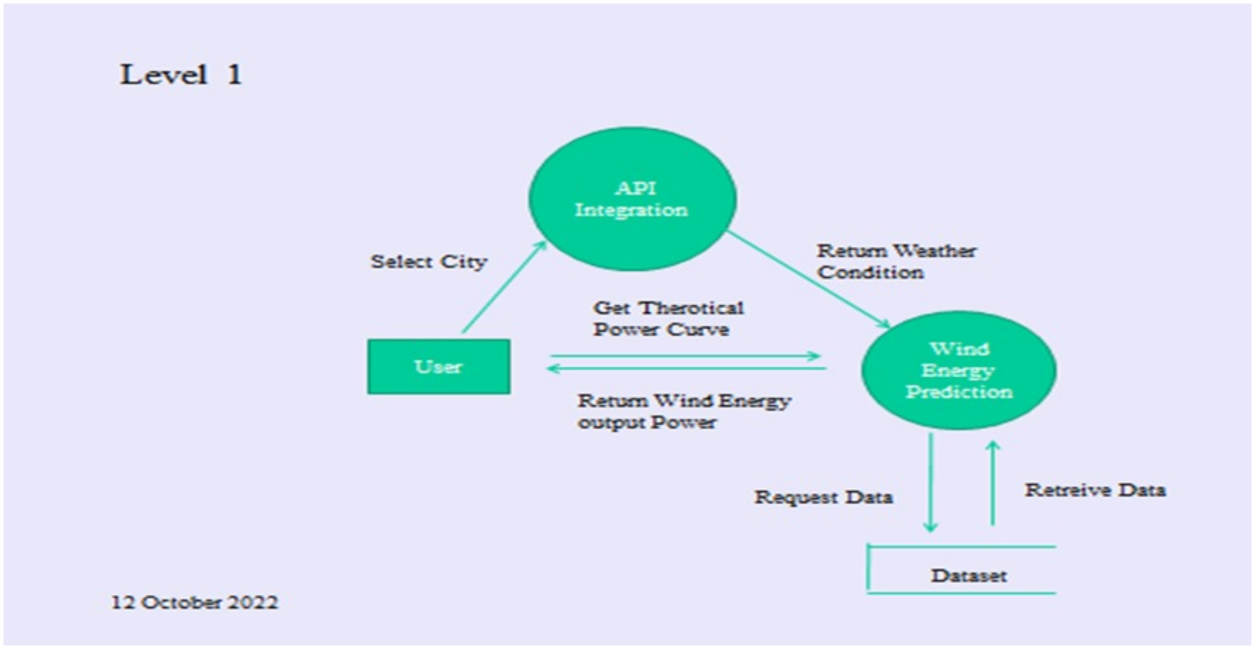


Figure: 5.2 - Dataflow Level 1

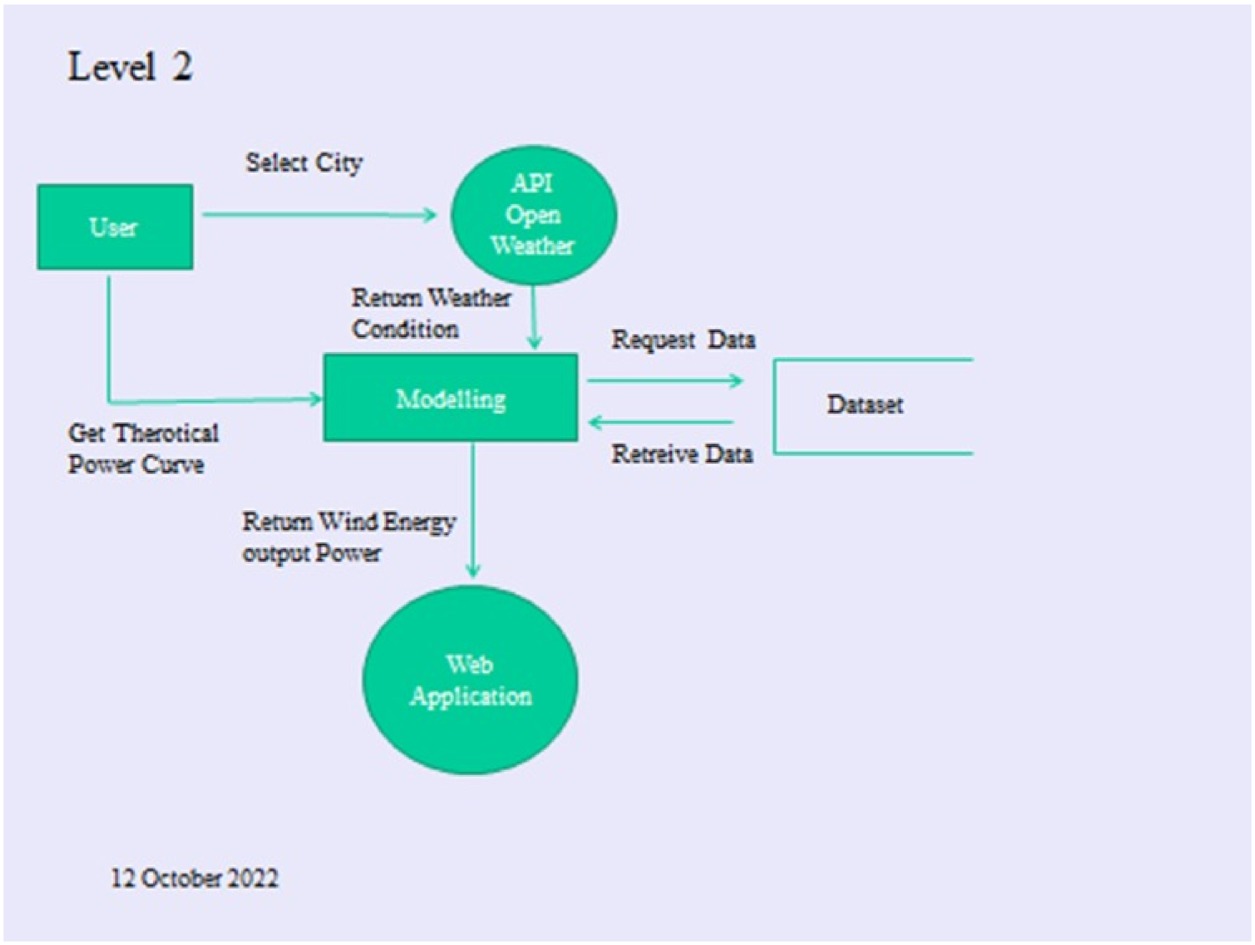


Figure: 5.3 - Dataflow Level 2

5.2 SOLUTION AND TECHNICAL ARCHITECTURE

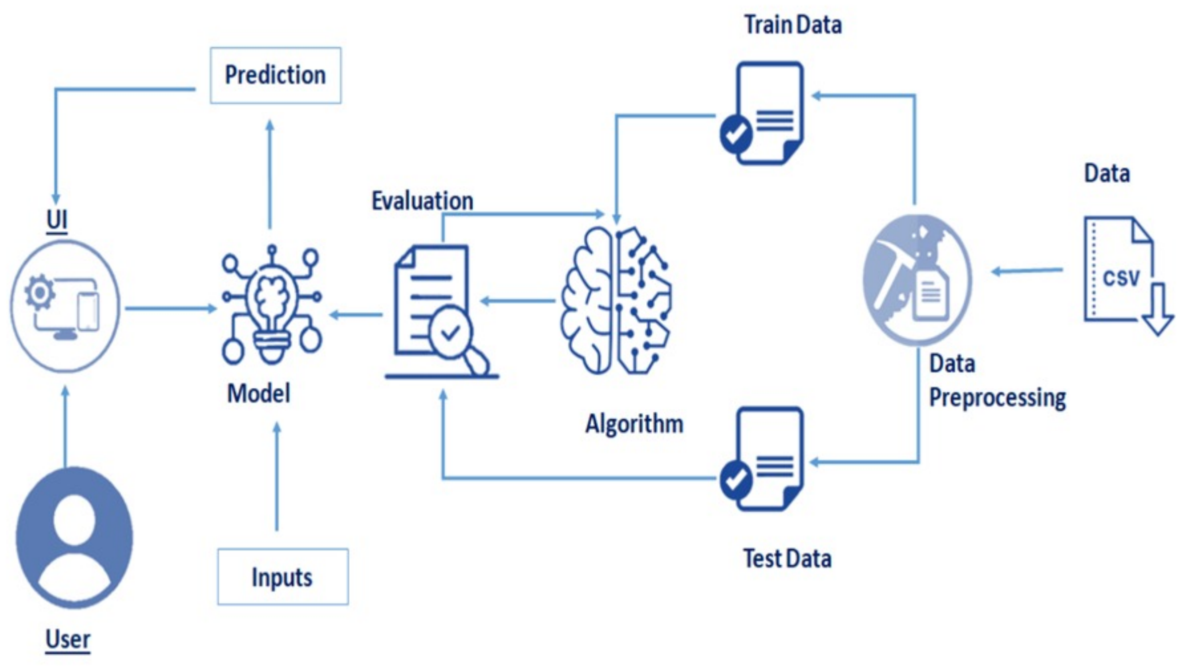


Figure: 5.4 - Solution Architecture

5.3 USER STORY

UserType	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 wellas the detailed information about the application	I can gain knowledge by practical method to usethis application.	Low	Sprint-1

		USN-2	As a User, I canuse this application by readingthe instructions	I can use this in user friendly method by reading the instruction.	Low	Sprint-1
		USN-3	As a User, I canlogin and by entering the correct username and password	If login is correctly entered, I can navigate to the nextpage.	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 thecity, If the city is correct, I can further enter the details.	Medium	Sprint-3
		USN-5	As a user I am allowed to viewthe weather of the selected city.	If correct cityis selected , then the weather of the particular city will be displayed.	Medium	Sprint-4
		USN-6	As a User, I canview the Power generated by the wind	If all values areentered correctly, I can view the power by the wind	High	Sprint-5

		USN-7	As a User, I canuse the web application virtually anywhere	I can use the application portably	High	Sprint-2
		USN-8	As it is open source, I can useitcost freely.	I can use it without any payment to access	Medi um	Sprint-2

Table: 5.1 - User Story

CHAPTER 6
PROJECT PLANNING AND SCHEDULING
6.1 SPRINT PLANNING 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	Elavarasan S Gandhikumar SGowsik MP Harish KumarS
Sprint-1	Registration	USN-2	As a user, I will receive confirmation email once I haveregistered for the application	5	High	Elavarasan S Gandhikumar SGowsik M P Harish Kumar S
Sprint-1	Registration	USN-3	User should verify the email once they havecreated their account.	2	Low	Elavarasan S Gandhikumar SGowsik M P Harish Kumar S
Sprint-1	Registration	USN-4	As a user,I can register for the application through Gmail	3	Medium	Elavarasan S Gandhikumar SGowsik M P Harish Kumar S
Sprint-1	Login	USN-5	As a user,I can log into the application by entering email & password	5	High	Elavarasan S Gandhikumar S Gowsik M P Harish Kumar S
Sprint-2	Dashboard	USN-6	Once I have logged in, I can see my dashboard.	6	Medium	Elavarasan S Gandhikumar S Gowsik M P Harish Kumar S
Sprint-2	Web access	USN-7	As a customer I can accessthe website to predict the turbine power	7	High	Elavarasan S Gandhikumar S Gowsik M P Harish Kumar S

Sprint-2	Prediction	USN-8	As a customer when I enterthe weather details, the website should predict the approximate turbine power	7	High	Elavarasan S Gandhikumar S Gowsik M P Harish Kumar S
Sprint-3	Plotting	USN-9	Customer can also provide the latitude and longitude of any location, and our web app will predict the wind power based on the wind speed and wind direction of the location given.	10	Medium	Elavarasan S Gandhikumar S Gowsik M P Harish Kumar S
Sprint-3	Plotting	USN-11	Website provides various charts to make the customer understand the speed,direction and powervisually.	3	Low	Elavarasan S Gandhikumar S Gowsik M P Harish Kumar S
Sprint-3	Security	USN-12	As a customer I expect my data to be secured	2	Low	Elavarasan S Gandhikumar S Gowsik M P Harish Kumar S
Sprint-4	Database Access	USN-13	As an Administrator, I should maintain the website. And update the website regularly.	20	High	Elavarasan S Gandhikumar S Gowsik M P Harish Kumar S

Table: 6.1 - Sprint Planning

PROJECT TRACKER, VELOCITY AND BURNDOWN CHART

Sprint	Total Story Points	Duration	Sprint StartDate	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	8	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov2022	7	05 Nov 2022
Sprint-3	20	6 Days	07 Nov2022	12 Nov2022	8	12 Nov 2022
Sprint-4	20	6 Days	14 Nov2022	19 Nov2022	7	19 Nov 2022

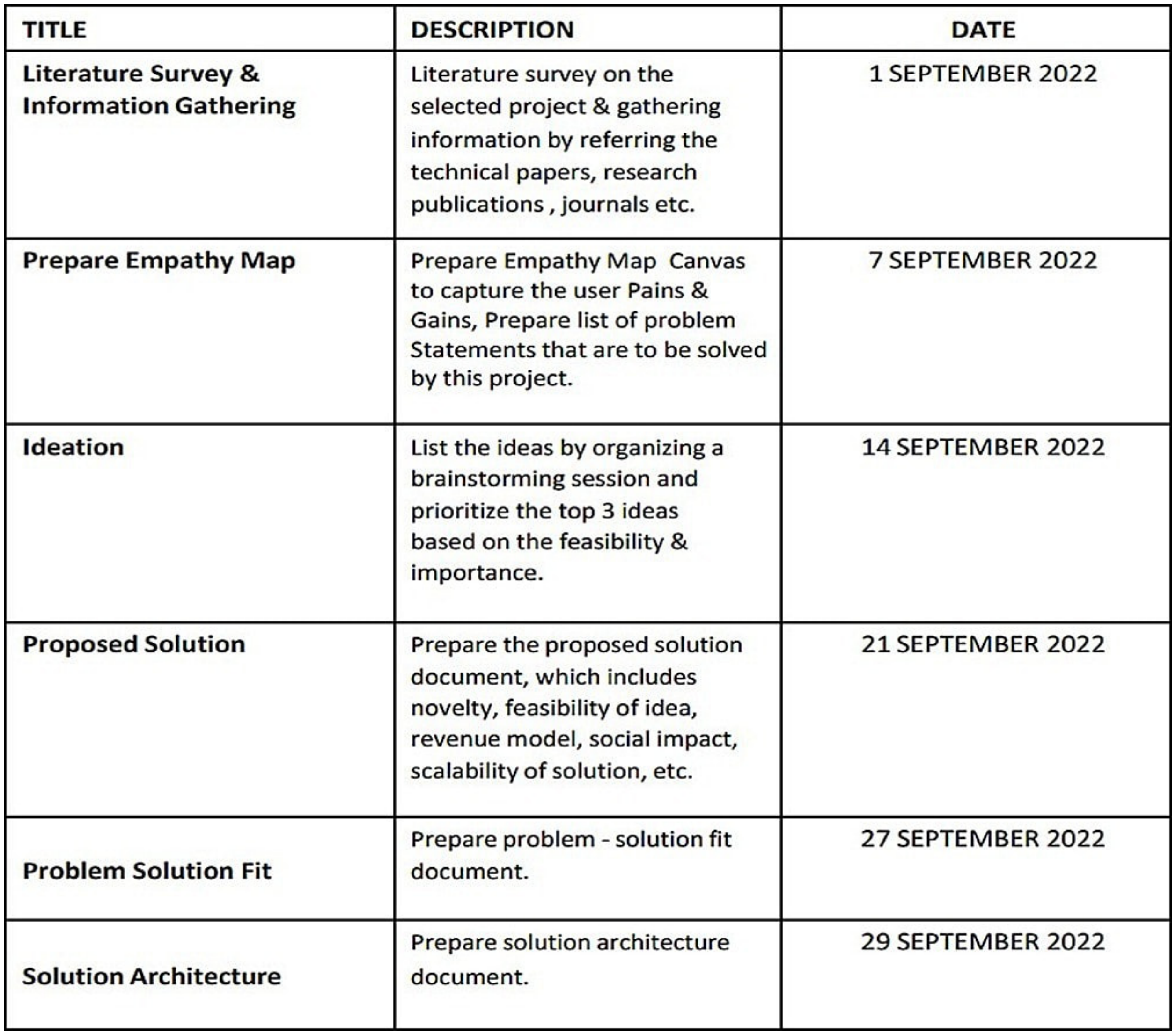
Table: 6.2- Project Tracker Chart

Velocity:

Imagine we have a 10-daysprint duration, and the velocity of the team is 20 (points per sprint). Let’scalculate the team’s average velocity(AV) per iterationunit (story points per day)

$$AV = \frac{\textit{sprint duration}}{\textit{velocity}} = \frac{20}{10} = 2$$

6.2 PROJECT DELIVERY SCHEDULE



TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the technical papers, research publications , journals etc.	1 SEPTEMBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem Statements that are to be solved by this project.	7 SEPTEMBER 2022
Ideation	List the ideas by organizing a brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	14 SEPTEMBER 2022
Proposed Solution	Prepare the proposed solution document, which includes novelty, feasibility of idea, revenue model, social impact, scalability of solution, etc.	21 SEPTEMBER 2022
Problem Solution Fit	Prepare problem - solution fit document.	27 SEPTEMBER 2022
Solution Architecture	Prepare solution architecture document.	29 SEPTEMBER 2022

Figure: 6.1.i - Project Delivery Schedule

Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).	4 OCTOBER 2022
Functional Requirement	Prepare the functional requirement document.	5 OCTOBER 2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review.	5 OCTOBER 2022
Technology Architecture	Prepare the technology architecture diagram.	13 OCTOBER 2022
Prepare Milestone & Activity List	Prepare the milestones & activity list of the project.	22 OCTOBER 2022
Project Development - Delivery of Sprint-1, 2, 3 & 4	Develop & submit the developed code by testing it.	IN PROGRESS..

Figure: 6.1.ii - Project Delivery Schedule

6.3 REPORTS FROM JIRA

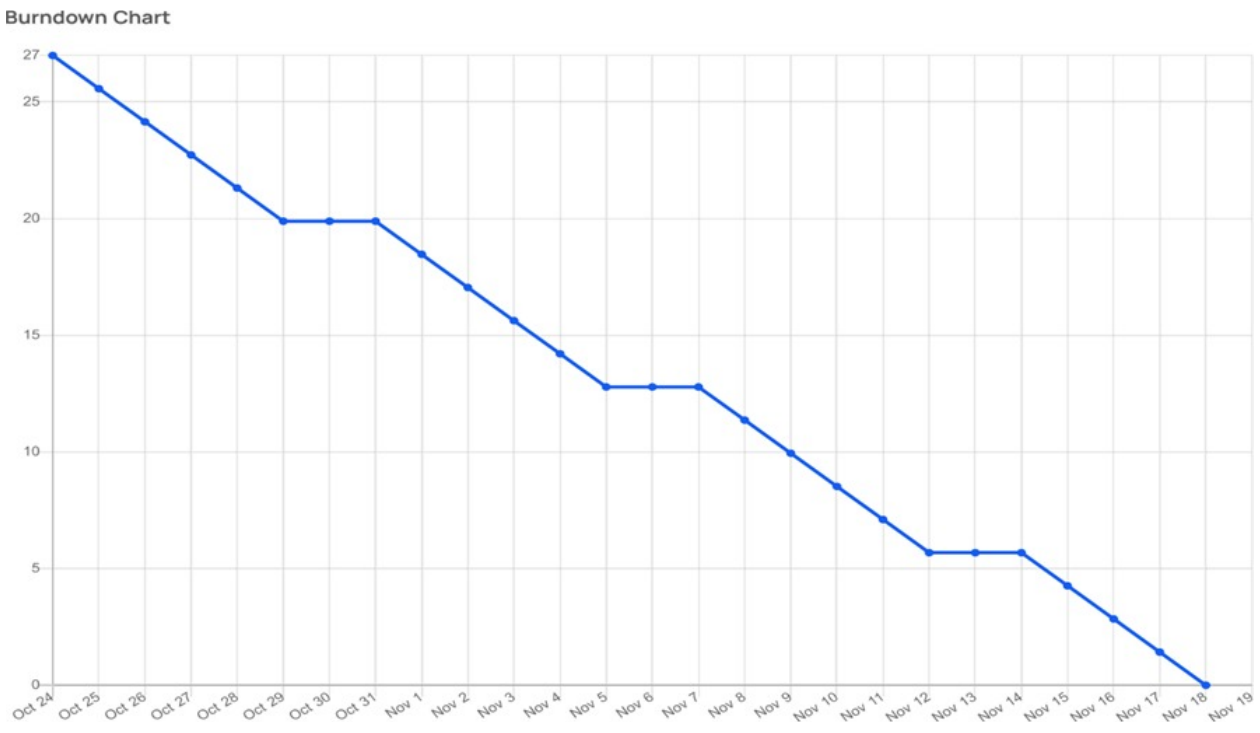


Figure 6.2 - Burndown chart

BURNUP CHART:

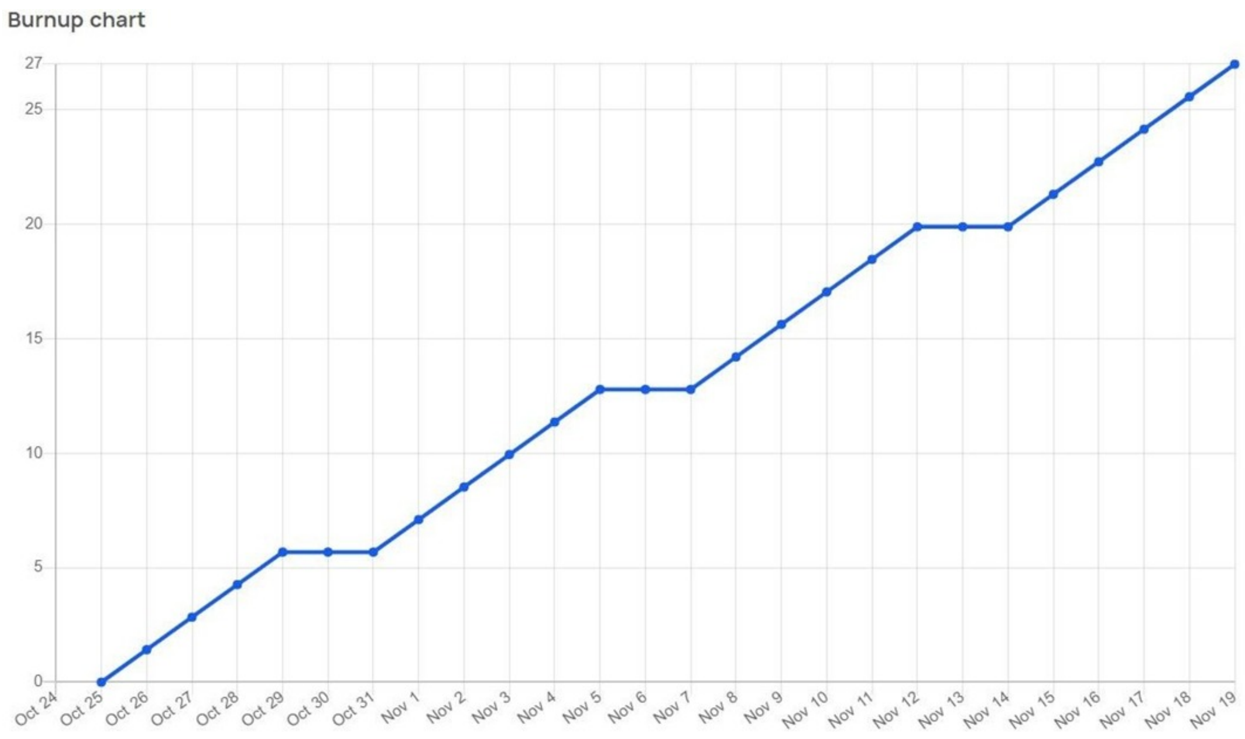


Figure 6.3 - Burnup chart

CHAPTER 7
CODING AND SOLUTION

7.1 FEATURE 1

Random Forest Classifier is used to train and test the model for detecting the wind speed prediction with the help of collected and pre-processed dataset collections. NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. Moreover, NumPy forms the foundation of the Machine Learning stack. Pandas is an open-source Python package that is most widely used for data science/data analysis and machine learning tasks. Sea born is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. For a brief introduction to the ideas behind the library, you can read the introductory notes or the paper. Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. Create publication quality plots. Make interactive figures that can zoom, pan, updated is applied to investigate the data and summarize the key insights. It will give you the basic understanding of your data, it is distribution, null values and much more. You can either explore data using graphs or through some python functions. There will be two types of analysis. Descriptive statistics are brief informational coefficients that summarize a given data set, which can be either a representation of the entire population or a sample of a population. Descriptive statistics are broken down into measures of central tendency and measures of variability. Measures of central tendency include the mean, median, and mode, while measures of variability include standard deviation, variance, minimum and maximum variables, kurtosis, and Skewness. Label Encoding refers to converting the labels into a numeric form to convert them into the machine-readable form. Machine learning algorithms can then decide in a better way how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning. “Pickling” is the process whereby a Python object hierarchy is converted into a byte stream, and “unpickling” is the inverse operation, whereby a byte stream is converted back into an object hierarchy. XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable. It implements machine learning algorithms under the Gradient Boosting framework.

# RandomForestClassifier: from sklearn.ensemble import RandomForestClassifier RandomForest = RandomForestClassifier() RandomForest =

RandomForest.fit(X_train,y_train) # Predictions: y_pred = RandomForest.predict(X_test) # Performance: print('Accuracy:', accuracy_score(y_test,y_pred)) print(confusion_matrix(y_test,y_pred))

print(classification_report(y_test,y_pred))

Gradient boosting classifiers are a group of machine learning algorithms that combine many weak learning models together to create a strong predictive model. Decision trees are usually used when doing gradient boosting.

# GradientBoostingClassifier: from sklearn.ensemble import GradientBoostingClassifier GradientBoost = GradientBoostingClassifier() GradientBoost = GradientBoost.fit(X_train,y_train) # Predictions: y_pred = GradientBoost.predict(X_test) # Performance: print('Accuracy:', accuracy_score(y_test,y_pred)) print(confusion_matrix(y_test,y_pred)) print(classification_report(y_test,y_pred))

AdaBoost can be used to boost the performance of any machine learning algorithm. It is best used with weak learners. These are models that achieve accuracy just above random chance on a classification problem. The most suited and therefore most common algorithm used with AdaBoost are decision trees with one level.

# AdaBoostClassifier: from sklearn.ensemble import AdaBoostClassifier AdaBoost = AdaBoostClassifier() AdaBoost = AdaBoost.fit(X_train,y_train)

# Predictions: y_pred = AdaBoost.predict(X_test) # Performance: print('Accuracy:', accuracy_score(y_test,y_pred)) print(confusion_matrix(y_test,y_pred)) print(classification_report(y_test,y_pred))

7.2 FEATURE 2

The framework is the basis upon which software programs are built. It serves as a foundation for software developers, allowing them to create a variety of applications for certain platforms. It is a set of functions and predefined classes used to connect with the system software and handle inputs and outputs. It simplifies the life of a developer while giving them the ability to use certain extensions and makes the online applications scalable and maintainable. Flask is a web application framework written in Python. A Web Application Framework or a simply a Web Framework represents a collection of libraries and modules that enable web application developers to write applications without worrying about low-level details such as protocol, thread management, among other examples. Flask is a web application framework written in Python. Flask is based on the Werkzeg WSGI toolkit and the

Jinja2 template engine. Both are Pocco projects. The Web Server Gateway Interface (Web Server Gateway Interface, WSGI) has been used as a standard for Python web application development. WSGI is the specification of a common interface between web servers and web applications. Flask is often referred to as a micro-framework. It is designed to keep the core of the application simple and scalable. Instead of an abstraction layer for database support, Flask supports extensions to add such capabilities to the application. Unlike the Django framework, Flask is very Pythonic. It’s easy to get started with Flask, because it doesn’t have a huge learning curve.HTML stands for Hyper Text Markup Language. HTML is the standard markup language for creating Web pages. HTML describes the structure of a Web page. HTML consists of a series of elements. HTML elements tell the browser how to display the content. Flask is used for developing web applications using python, implemented on Werkzeug and Jinja2. Advantages of using Flask framework are: There is a built-in development server and a fast debugger provided. The model deployed using Flask is used to predict the wind speed. Hypertext markup language (HTML) is the basic language used to create documents for the Web and, along with HTTP (hypertext transfer protocol) and URLs (universal resource locators), is one of the three main protocols of the Web. Hypertext is text that contains hyperlinks. A hyperlink is an automated cross-reference to another location on the same document or to another document which, when selected by a user, causes the computer to display the linked location or document within a concise period. A markup language is a set of tags that can be embedded in digital text to provide additional information about it, including its content, structure and appearance. This information facilitates automated operations on the text, including formatting it for display, searching it and even modifying it. Some type of markup language is employed by every word processing program and by nearly every other program that displays text, although such languages and their tags are typically hidden from the user.HTML consists of a set of predefined tags that can be embedded in text by web site designers in order to indicate the details of how web pages are rendered (i.e., converted into a final, easily usable, form) by web browsers. These details include paragraphing, margins, fonts (including style and size), columns, colors (background and text), links, the location of images, text flow around images, tables, and user input form elements (such as spaces for adding text and submit buttons).

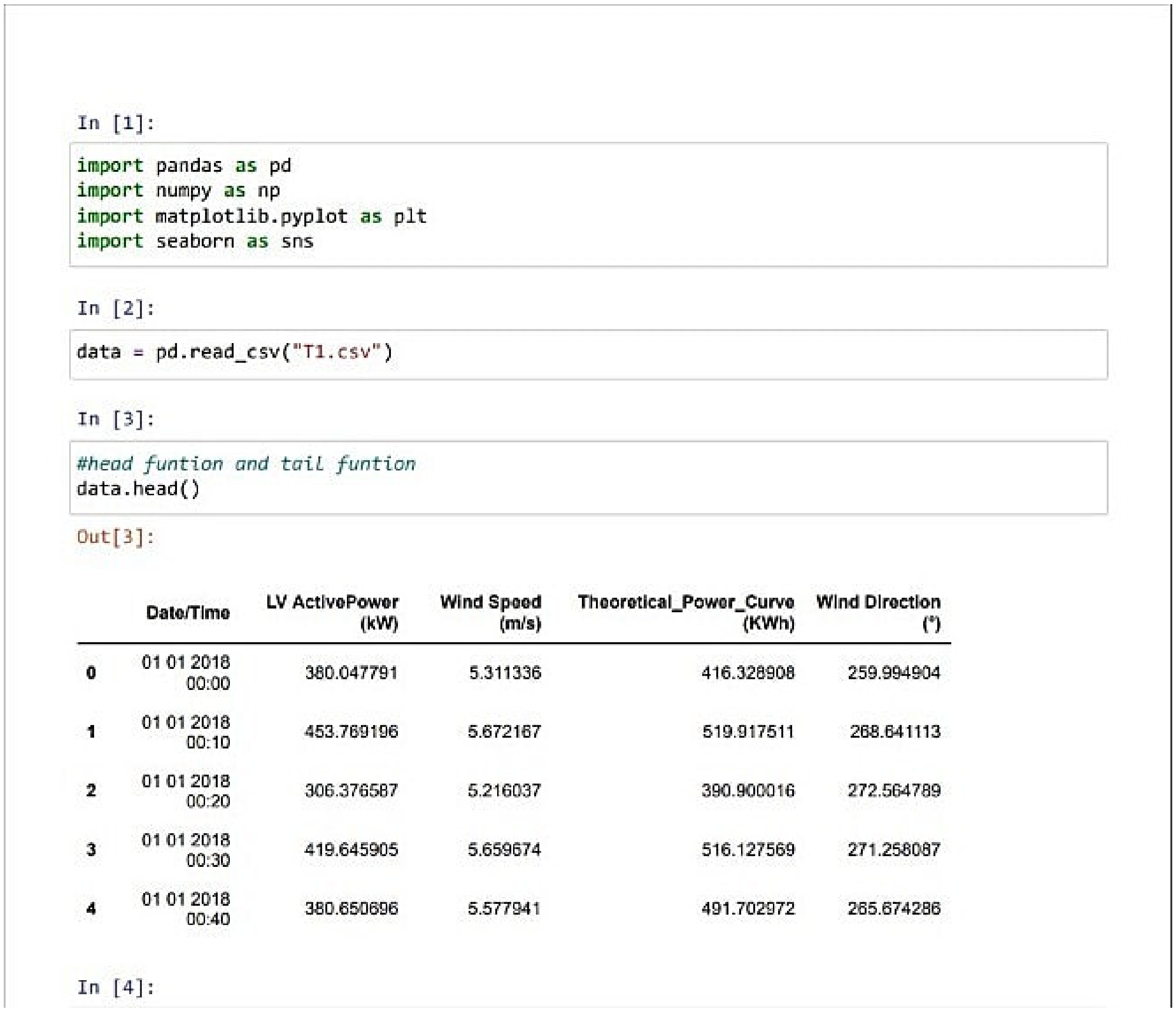
from flask import Flask, render_template, request import numpy as np import pickle import requests import json Here we import all the necessary features of this project involving in Python flask.

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken} app = Flask(_name_) model = pickle.load(open('liver2.pkl', 'rb')) @app.route('/',methods=['GET']) def Home(): return render_template('index.html')

```
        @app.route("/predict", methods=['POST']) def predict(): if request.method == 'POST':
Age = int(request.form['Age']) Gender = int(request.form['Gender']) Total_Bilirubin =
float(request.form['Total_Bilirubin']) Alkaline_Phosphotase =
int(request.form['Alkaline_Phosphotase']) Alamine_Aminotransferase =
int(request.form['Alamine_Aminotransferase']) Aspartate_Aminotransferase =
int(request.form['Aspartate_Aminotransferase']) Total_Protiens =
float(request.form['Total_Protiens']) Albumin = float(request.form['Albumin'])
Albumin_and_Globulin_Ratio = float(request.form['Albumin_and_Globulin_Ratio'] values =
np.array([[Age,Gender,Total_Bilirubin,Alkaline_Phosphotase,Alamine_Aminotran
sferase,Aspartate_Aminotransferase,Total_Protiens,Albumin,Albumin_and_Glob
ulin_Ratio]]) prediction = model.predict(values) return render_template('result.html',
prediction=prediction if _name_ == "_main_": app.run(debug=True)
```

CHAPTER 8
TESTING

8.1 TEST CASES



In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
data = pd.read_csv("T1.csv")
```

In [3]:

```
#head funtion and tail funtion
data.head()
```

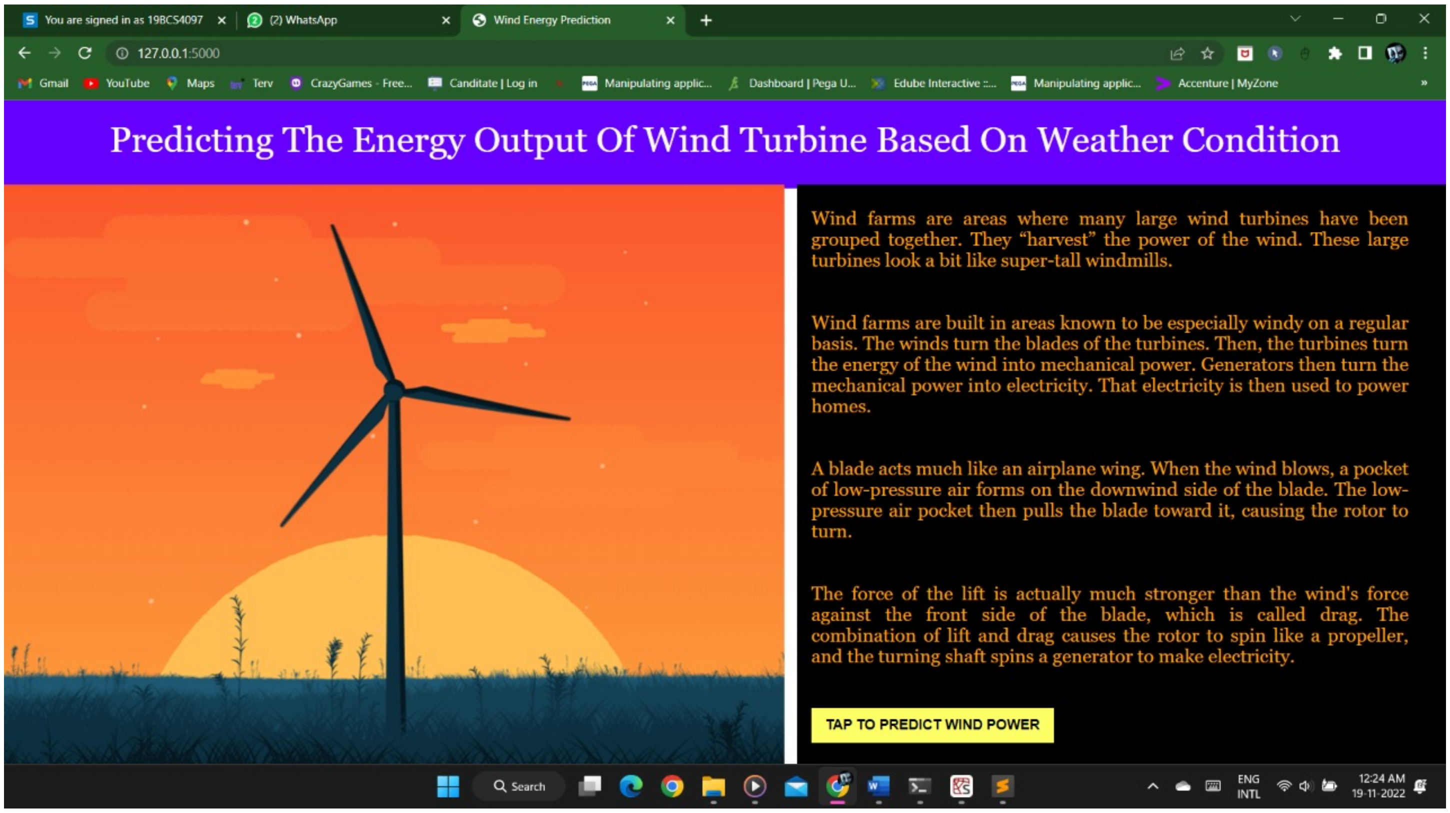
Out[3]:

	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 [4]:

Figure 8.1 - Test Case

8.2 USER ACCEPTANCE TESTING



Predicting The Energy Output Of Wind Turbine Based On Weather Condition

Wind farms are areas where many large wind turbines have been grouped together. They “harvest” the power of the wind. These large turbines look a bit like super-tall windmills.

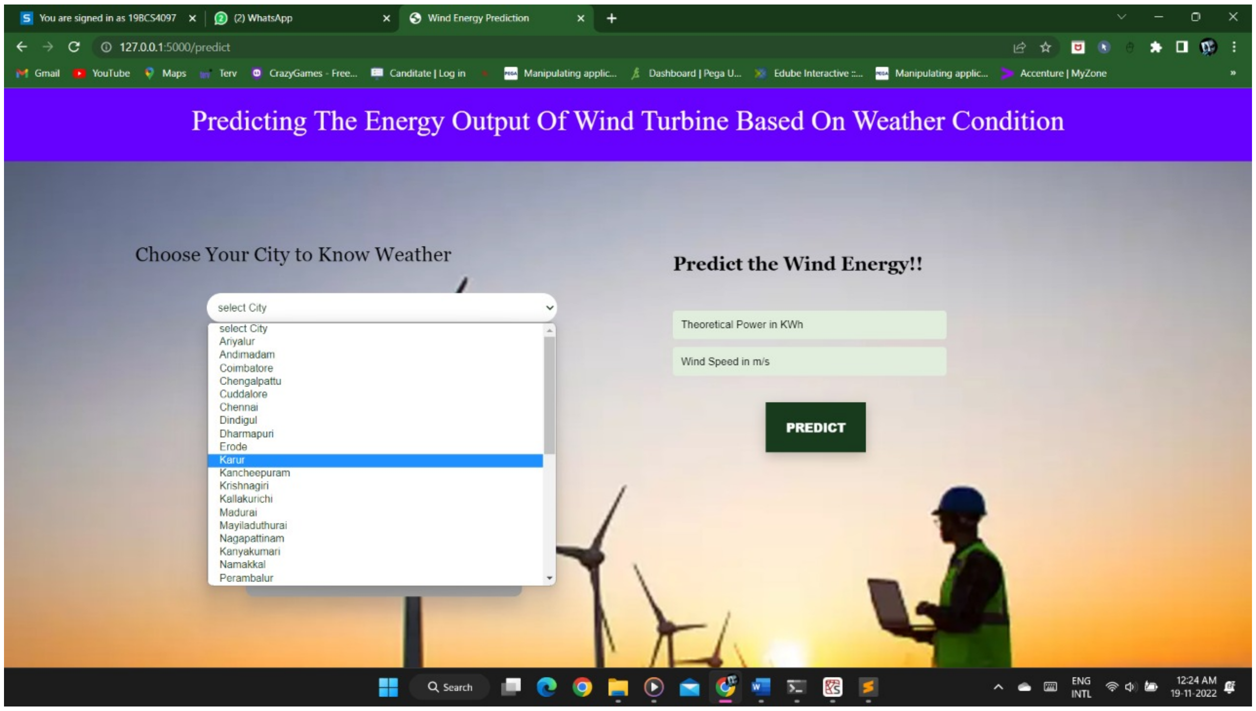
Wind farms are built in areas known to be especially windy on a regular basis. The winds turn the blades of the turbines. Then, the turbines turn the energy of the wind into mechanical power. Generators then turn the mechanical power into electricity. That electricity is then used to power homes.

A blade acts much like an airplane wing. When the wind blows, a pocket of low-pressure air forms on the downwind side of the blade. The low-pressure air pocket then pulls the blade toward it, causing the rotor to turn.

The force of the lift is actually much stronger than the wind's force against the front side of the blade, which is called drag. The combination of lift and drag causes the rotor to spin like a propeller, and the turning shaft spins a generator to make electricity.

TAP TO PREDICT WIND POWER

Figure 8.2 - Welcome Page



Predicting The Energy Output Of Wind Turbine Based On Weather Condition

Choose Your City to Know Weather

select City

select City
Ariyalur
Andimadam
Coimbatore
Chengalpattu
Cuddalore
Chennai
Dindigul
Dharmapuri
Erode
Karur
Kancheepuram
Krishnagiri
Kallakurichi
Madurai
Mayiladuthurai
Nagapattinam
Kanyakumari
Namakkal
Perambalur

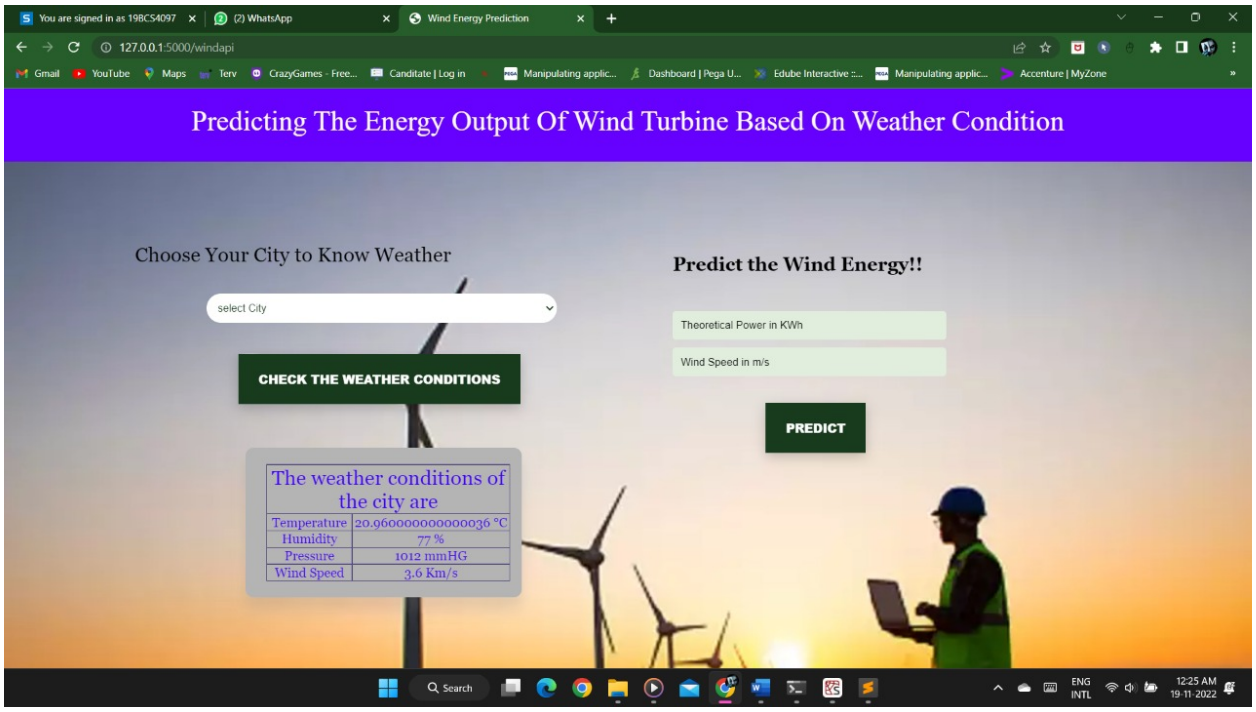
Predict the Wind Energy!!

Theoretical Power in KWh

Wind Speed in m/s

PREDICT

Figure 8.3 - Selecting City



Predicting The Energy Output Of Wind Turbine Based On Weather Condition

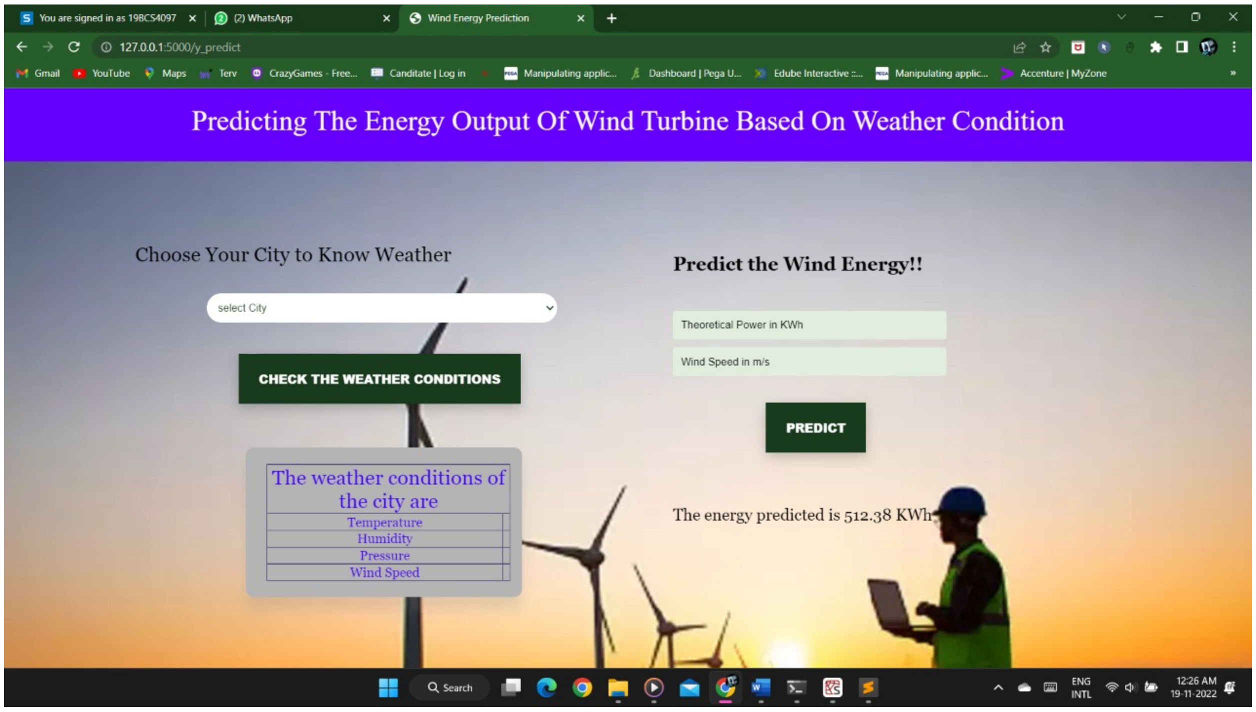
Choose Your City to Know Weather

Predict the Wind Energy!!

The weather conditions of the city are

Temperature	20.960000000000036 °C
Humidity	77 %
Pressure	1012 mmHG
Wind Speed	3.6 Km/s

Figure 8.4 - Current Weather Status



Predicting The Energy Output Of Wind Turbine Based On Weather Condition

Choose Your City to Know Weather

Predict the Wind Energy!!

The weather conditions of the city are

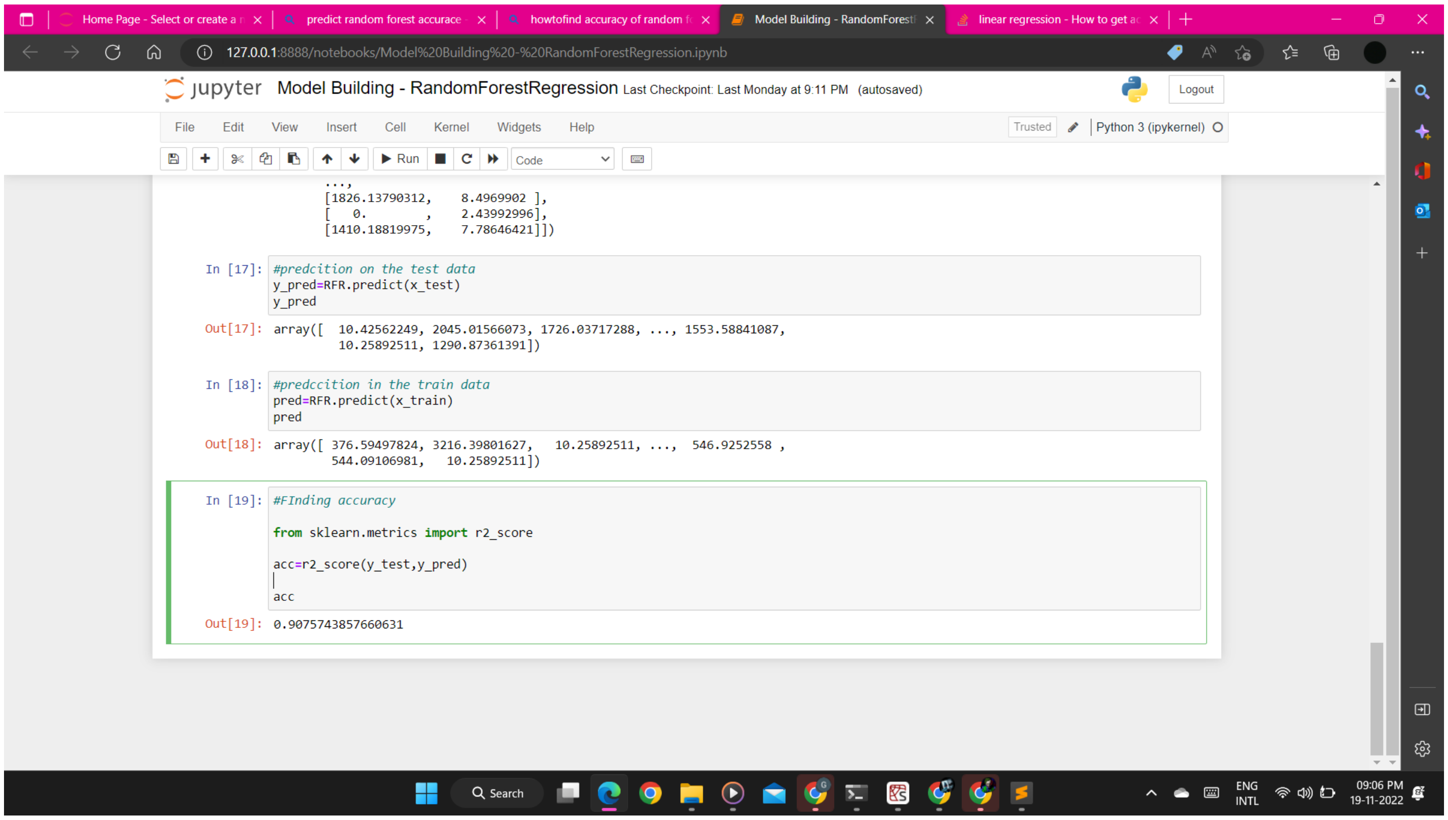
Temperature	
Humidity	
Pressure	
Wind Speed	

The energy predicted is 512.38 KWh

Figure 8.5 - Energy Prediction

CHAPTER 9
RESULTS

9.1 PERFORMANCE METRICS



Model Building - RandomForestRegression Last Checkpoint: Last Monday at 9:11 PM (autosaved)

```
         ...,
         [1826.13790312,    8.4969902 ],
         [   0.         ,    2.43992996],
         [1410.18819975,    7.78646421]])

In [17]: #predcition on the test data
         y_pred=RFR.predict(x_test)
         y_pred

Out[17]: array([  10.42562249, 2045.01566073, 1726.03717288, ..., 1553.58841087,
                  10.25892511, 1290.87361391])

In [18]: #predccition in the train data
         pred=RFR.predict(x_train)
         pred

Out[18]: array([ 376.59497824, 3216.39801627,   10.25892511, ...,  546.9252558 ,
                 544.09106981,   10.25892511])

In [19]: #FInding accuracy

         from sklearn.metrics import r2_score

         acc=r2_score(y_test,y_pred)

         acc

Out[19]: 0.9075743857660631
```

Figure: 9.1 - Performances Metrics

The proposed communication network architecture for the Smart-WPF consists of three networks: the turbine area network (TAN), the farm area network (FAN), and the control area network (CAN). It consists of hierarchical architectures where Level 1 is a sensor network in a single wind turbine, Level 2 is the wind turbine to-wind turbine interaction in the WPF, Level 3 is the local control centre to wind turbine interaction, and Level 4 is the farm-to-farm interaction to optimize grid operation. In order to implement hierarchical network architectures, a hybrid communication solution is considered. EPON-based architecture represents a wired solution, while ZigBee-Pro is considered for the wireless solution. In this work, Levels 1 and 2 are explained in more detail, while Levels 3 and 4 are out the scope.

CHAPTER 10
ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

- 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
- Predict the variation in the long-term wind speed over the site at the hub height of the machines, based on the long-term wind speeds at the mast locations; Predict the wake losses that arise as a result of one turbine operating behind another – in other words in its wake; and calculate or estimate the other losses.

DISADVANTAGES:

- In order to achieve better generalization for wind speed prediction, the input and output are to be modelled and the hidden neuron number should be appropriately selected for the neural network design.
- In the current scenario many prediction research fields have been heuristic. While numerous researchers have developed prediction models for accurate wind speed prediction, no perfect model has been achieved. However, an accurate wind speed prediction model based on RRBFNN for a long-term forecast horizon (1 day to 1 week ahead or more) is presented in this paper.

CHAPTER 11
CONCLUSION

Kernel methods and neural techniques have proven well in analysis and modelling dynamic systems. This seems natural as neural techniques are motivated by mechanisms of natural neural systems that have to cope with dynamic environments. Wind is a very dynamic system. An important aspect is to manage its volatile and dynamic nature. The integration of wind energy into smart grids affords balancing capabilities, and balancing affords understanding and forecasting. The examples presented in this work have shown how kernel techniques can help to cope with the dynamics of wind time series data. They turn out to be successful methods in modelling, forecasting and monitoring of wind energy time series data. Efficient implementations allow their application in real-time scenarios. It is subject to future projects to show the success of these and other kernel methods in real-world energy applications.

CHAPTER 12
FUTURE SCOPE

Most wind power forecasting models study ‘regular’ wind conditions. The EU funded project called ‘Safe wind’ aims to improve wind power prediction over challenging and extreme weather periods and at different temporal and spatial scales. Development activities are on-going to reduce error in wind power prediction, to improve regionalized wind power forecasting for on shore wind farms and to derive methods for wind power prediction for offshore wind farms. It is possible that the use of ensemble and combined weather prediction methods together may enhance forecasting. Offshore wind farms pose more of a challenge in terms of accurate wind power forecasting because the environment is typically flat and smooth with very few obstacles so changes in wind speed and thermal effects are felt more acutely than on land as weather fronts pass over the wind farm. A review of published data has gleaned very little knowledge of methods in use for offshore wind power prediction.

CHAPTER 13
APPENDIX

Source Code:

Algorithm:

```
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="a802b0f626c637d04185e582b5ad0d58"
url="http://api.openweathermap.org/data/2.5/weather?q="+city+"&appid="+apikey
    resp = requests.get(url)
    resp=resp.json()
    temp = str((resp["main"]["temp"])-273.15) +" °C"
    humid = str(resp["main"]["humidity"])+" %"
    pressure = str(resp["main"]["pressure"])+" mmHG"
```

```
        speed = str((resp["wind"]["speed"])*3.6)+" Km/s"
        return render_template('predict.html', temp=temp, humid=humid,
pressure=pressure,speed=speed)
    @app.route('/y_predict',methods=['POST'])
    def y_predict():
        '''
        For rendering results on HTML GUI
        '''
        x_test = [[float(x) for x in request.form.values()]]
        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)
    {
     "cells": [
      {
       "cell_type": "code",
       "execution_count": 1,
       "id": "f862682f",
       "metadata": {},
       "outputs": [],
       "source": [
        "import pandas as pd\n",
   "import numpy as np\n",
   "import matplotlib.pyplot as plt\n",
   "import seaborn as sns"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 2,
  "id": "c52e127f",
  "metadata": {},
  "outputs": [],
  "source": [
   "data = pd.read_csv(\"T1.csv\")"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 3,
  "id": "5ec998e6",
  "metadata": {},
  "outputs": [
   {
    "data": {
     "text/html": [
      "<div>\n",
      "<style scoped>\n",
"    .dataframe tbody tr th:only-of-type {\n",
"        vertical-align: middle;\n",
"    }\n",
"\n",
"    .dataframe tbody tr th {\n",
"        vertical-align: top;\n",
"    }\n",
"\n",
"    .dataframe thead th {\n",
"        text-align: right;\n",
"    }\n",
"</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"  <thead>\n",
"    <tr style=\"text-align: right;\">\n",
"      <th></th>\n",
"      <th>Date/Time</th>\n",
"      <th>LV ActivePower (kW)</th>\n",
"      <th>Wind Speed (m/s)</th>\n",
"      <th>Theoretical_Power_Curve (KWh)</th>\n",
"      <th>Wind Direction (°)</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>01 01 2018 00:00</td>\n",
"      <td>380.047791</td>\n",
"      <td>5.311336</td>\n",
"      <td>416.328908</td>\n",
"      <td>259.994904</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>01 01 2018 00:10</td>\n",
"      <td>453.769196</td>\n",
"      <td>5.672167</td>\n",
"      <td>519.917511</td>\n",
"      <td>268.641113</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>01 01 2018 00:20</td>\n",
"      <td>306.376587</td>\n",
"      <td>5.216037</td>\n",
"      <td>390.900016</td>\n",
"      <td>272.564789</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>01 01 2018 00:30</td>\n",
"      <td>419.645905</td>\n",
"      <td>5.659674</td>\n",
"      <td>516.127569</td>\n",
"      <td>271.258087</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>4</th>\n",
"      <td>01 01 2018 00:40</td>\n",
"      <td>380.650696</td>\n",
"      <td>5.577941</td>\n",
"      <td>491.702972</td>\n",
"      <td>265.674286</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>"
],
"text/plain": [
"          Date/Time  LV ActivePower (kW)  Wind Speed (m/s)  \\\n",
"0  01 01 2018 00:00           380.047791          5.311336   \n",
"1  01 01 2018 00:10           453.769196          5.672167   \n",
"2  01 01 2018 00:20           306.376587          5.216037   \n",
"3  01 01 2018 00:30           419.645905          5.659674   \n",
"4  01 01 2018 00:40           380.650696          5.577941   \n",
"\n",
"   Theoretical_Power_Curve (KWh)  Wind Direction (°)  \n",
"0                     416.328908          259.994904  \n",
      "1                    519.917511        268.641113  \n",
      "2                    390.900016        272.564789  \n",
      "3                    516.127569        271.258087  \n",
      "4                    491.702972        265.674286  "
     ]
    },
    "execution_count": 3,
    "metadata": {},
    "output_type": "execute_result"
   }
  ],
  "source": [
   "#head funtion and tail funtion\n",
   "data.head()"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 4,
  "id": "b93a47a2",
  "metadata": {},
  "outputs": [],
  "source": [
   "data = data.rename(columns = {\"Date/Time\":\"Date\",\n",
   "                           \"LV ActivePower (kW)\":\"Active_Power\",\n",
   "                           \"Wind Speed (m/s)\":\"Wind_Speed\",\n",
    "                           \"Theoretical_Power_Curve (KWh)\":\"Theoretical_Power\",\n",
    "                           \"Wind Direction (°)\" :\"Wind_Direction\"\n",
    "                          })"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 5,
   "id": "2e944a5c",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/html": [
       "<div>\n",
       "<style scoped>\n",
       "    .dataframe tbody tr th:only-of-type {\n",
       "        vertical-align: middle;\n",
       "    }\n",
       "\n",
       "    .dataframe tbody tr th {\n",
       "        vertical-align: top;\n",
       "    }\n",
       "\n",
       "    .dataframe thead th {\n",
       "        text-align: right;\n",
"    }\n",
"</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"  <thead>\n",
"    <tr style=\"text-align: right;\">\n",
"      <th></th>\n",
"      <th>Date</th>\n",
"      <th>Active_Power</th>\n",
"      <th>Wind_Speed</th>\n",
"      <th>Theoretical_Power</th>\n",
"      <th>Wind_Direction</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>50525</th>\n",
"      <td>31 12 2018 23:10</td>\n",
"      <td>2963.980957</td>\n",
"      <td>11.404030</td>\n",
"      <td>3397.190793</td>\n",
"      <td>80.502724</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>50526</th>\n",
"      <td>31 12 2018 23:20</td>\n",
"      <td>1684.353027</td>\n",
"      <td>7.332648</td>\n",
"      <td>1173.055771</td>\n",
"      <td>84.062599</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>50527</th>\n",
"      <td>31 12 2018 23:30</td>\n",
"      <td>2201.106934</td>\n",
"      <td>8.435358</td>\n",
"      <td>1788.284755</td>\n",
"      <td>84.742500</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>50528</th>\n",
"      <td>31 12 2018 23:40</td>\n",
"      <td>2515.694092</td>\n",
"      <td>9.421366</td>\n",
"      <td>2418.382503</td>\n",
"      <td>84.297913</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>50529</th>\n",
"      <td>31 12 2018 23:50</td>\n",
"      <td>2820.466064</td>\n",
"      <td>9.979332</td>\n",
"      <td>2779.184096</td>\n",
  "     <td>82.274620</td>\n",
  "    </tr>\n",
  "  </tbody>\n",
  "</table>\n",
  "</div>"
 ],
 "text/plain": [
  "                  Date  Active_Power  Wind_Speed  Theoretical_Power  \\\n",
  "50525  31 12 2018 23:10   2963.980957   11.404030       3397.190793   \n",
  "50526  31 12 2018 23:20   1684.353027    7.332648       1173.055771   \n",
  "50527  31 12 2018 23:30   2201.106934    8.435358       1788.284755   \n",
  "50528  31 12 2018 23:40   2515.694092    9.421366       2418.382503   \n",
  "50529  31 12 2018 23:50   2820.466064    9.979332       2779.184096   \n",
  "\n",
  "       Wind_Direction  \n",
  "50525       80.502724  \n",
  "50526       84.062599  \n",
  "50527       84.742500  \n",
  "50528       84.297913  \n",
  "50529       82.274620  "
 ]
},
"execution_count": 5,
"metadata": {},
"output_type": "execute_result"
}
```

```
   ],
   "source": [
    "data.tail() #last 5 rows of the dataset"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 6,
   "id": "e9cf34ba",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "(50530, 5)"
      ]
     },
     "execution_count": 6,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "#shape of the dataset\n",
    "data.shape"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 7,
   "id": "f10a67fe",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "Date                 0\n",
       "Active_Power         0\n",
       "Wind_Speed           0\n",
       "Theoretical_Power    0\n",
       "Wind_Direction       0\n",
       "dtype: int64"
      ]
     },
     "execution_count": 7,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "#missing values\n",
    "\n",
    "data.isna().sum()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 8,
   "id": "aeb2e781",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/html": [
       "<div>\n",
       "<style scoped>\n",
       "    .dataframe tbody tr th:only-of-type {\n",
       "        vertical-align: middle;\n",
       "    }\n",
       "\n",
       "    .dataframe tbody tr th {\n",
       "        vertical-align: top;\n",
       "    }\n",
       "\n",
       "    .dataframe thead th {\n",
       "        text-align: right;\n",
       "    }\n",
       "</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"  <thead>\n",
"    <tr style=\"text-align: right;\">\n",
"      <th></th>\n",
"      <th>count</th>\n",
"      <th>mean</th>\n",
"      <th>std</th>\n",
"      <th>min</th>\n",
"      <th>25%</th>\n",
"      <th>50%</th>\n",
"      <th>75%</th>\n",
"      <th>max</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>Active_Power</th>\n",
"      <td>50530.0</td>\n",
"      <td>1307.684332</td>\n",
"      <td>1312.459242</td>\n",
"      <td>-2.471405</td>\n",
"      <td>50.677890</td>\n",
"      <td>825.838074</td>\n",
"      <td>2482.507568</td>\n",
"      <td>3618.732910</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>Wind_Speed</th>\n",
"      <td>50530.0</td>\n",
"      <td>7.557952</td>\n",
"      <td>4.227166</td>\n",
"      <td>0.000000</td>\n",
"      <td>4.201395</td>\n",
"      <td>7.104594</td>\n",
"      <td>10.300020</td>\n",
"      <td>25.206011</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>Theoretical_Power</th>\n",
"      <td>50530.0</td>\n",
"      <td>1492.175463</td>\n",
"      <td>1368.018238</td>\n",
"      <td>0.000000</td>\n",
"      <td>161.328167</td>\n",
"      <td>1063.776283</td>\n",
"      <td>2964.972462</td>\n",
"      <td>3600.000000</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>Wind_Direction</th>\n",
"      <td>50530.0</td>\n",
"      <td>123.687559</td>\n",
     "     <td>93.443736</td>\n",
     "     <td>0.000000</td>\n",
     "     <td>49.315437</td>\n",
     "     <td>73.712978</td>\n",
     "     <td>201.696720</td>\n",
     "     <td>359.997589</td>\n",
     "    </tr>\n",
     "  </tbody>\n",
     "</table>\n",
     "</div>"
    ],
    "text/plain": [
     "                   count        mean          std       min        25%  \\\\\n",
     "Active_Power       50530.0  1307.684332  1312.459242 -2.471405   50.677890
\n",
     "Wind_Speed        50530.0     7.557952     4.227166  0.000000    4.201395   \n",
     "Theoretical_Power  50530.0  1492.175463  1368.018238  0.000000  161.328167
\n",
     "Wind_Direction     50530.0   123.687559    93.443736  0.000000   49.315437   \n",
     "\n",
     "                          50%          75%          max  \n",
     "Active_Power        825.838074  2482.507568  3618.732910  \n",
     "Wind_Speed            7.104594    10.300020    25.206011  \n",
     "Theoretical_Power  1063.776283  2964.972462  3600.000000  \n",
     "Wind_Direction       73.712978   201.696720   359.997589  "
    ]
    },
    "execution_count": 8,
    "metadata": {},
    "output_type": "execute_result"
   }
  ],
  "source": [
   "#statisticak overview of the data\n",
   "data.describe().T"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 9,
  "id": "b7a7f7d7",
  "metadata": {},
  "outputs": [
   {
    "data": {
     "text/plain": [
      "<matplotlib.collections.PathCollection at 0x1ff83401ac0>"
     ]
    },
    "execution_count": 9,
    "metadata": {},
    "output_type": "execute_result"
   },
   {
    "data": {
     "image/png":
     "text/plain": [
      "<Figure size 640x480 with 1 Axes>"
     ]
    },
    "metadata": {},
    "output_type": "display_data"
   }
  ],
  "source": [
   "#scatterplot\n",
   "plt.scatter(data['Theoretical_Power'],data['Wind_Speed'])"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 10,
  "id": "6bae2d41",
  "metadata": {},
  "outputs": [],
  "source": [
   "#split the data\n",
   "\n",
   "x=x = data[[\"Theoretical_Power\", \"Wind_Speed\"]]\n",
   "y=data[\"Active_Power\"]"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 11,
  "id": "e7858244",
  "metadata": {},
  "outputs": [],
  "source": [
   "x=x = data[[\"Theoretical_Power\", \"Wind_Speed\"]].values\n",
   "y=data[\"Active_Power\"].values"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 12,
  "id": "b28bf883",
  "metadata": {},
  "outputs": [
   {
    "data": {
     "text/plain": [
      "array([[ 416.32890782,    5.31133604],\n",
      "       [ 519.91751106,    5.67216682],\n",
     "       [ 390.90001581,    5.2160368 ],\n",
     "       ...,\n",
     "       [1788.28475526,    8.43535805],\n",
     "       [2418.38250336,    9.42136574],\n",
     "       [2779.18409628,    9.97933197]])"
    ]
   },
   "execution_count": 12,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "x"
 ]
},
{
 "cell_type": "code",
 "execution_count": 13,
 "id": "aa6454bb",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "array([ 380.04779053,  453.76919556,  306.37658691, ..., 2201.10693359,\n",
     "      2515.6940918 , 2820.46606445])"
    ]
   },
   "execution_count": 13,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "y"
 ]
},
{
 "cell_type": "code",
 "execution_count": 14,
 "id": "5ab973ba",
 "metadata": {},
 "outputs": [],
 "source": [
  "from sklearn.model_selection import train_test_split\n",
  "\n",
  "x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)"
 ]
},
{
 "cell_type": "code",
   "execution_count": 15,
   "id": "7f6de812",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "RandomForestRegressor(max_depth=4, max_leaf_nodes=500,
n_estimators=750,\n",
       "                      random_state=1)"
      ]
     },
     "execution_count": 15,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "from sklearn.ensemble import RandomForestRegressor\n",
    "\n",
    "RFR= RandomForestRegressor(n_estimators = 750, max_depth = 4, max_leaf_nodes
= 500, random_state = 1)\n",
    "\n",
    "RFR.fit(x_train,y_train) "
   ]
  },
{
 "cell_type": "code",
 "execution_count": 16,
 "id": "5255d9d2",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "array([[7.16452560e+02, 6.26323986e+00],\n",
     "       [1.66507070e+02, 4.22783899e+00],\n",
     "       [9.64137061e+02, 6.88290691e+00],\n",
     "       ...,\n",
     "       [3.15609958e+03, 1.07004499e+01],\n",
     "       [0.00000000e+00, 2.68328810e+00],\n",
     "       [1.00601774e+02, 3.84762692e+00]])"
    ]
   },
   "execution_count": 16,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "x_test"
 ]
```

```
},
{
 "cell_type": "code",
 "execution_count": 17,
 "id": "e4835b77",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "array([ 546.66390077,  153.93564142,  767.7112221 , ..., 2592.02145298,\n",
     "         10.19952956,   26.09517064])"
    ]
   },
   "execution_count": 17,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "#predcition on the test data\n",
  "y_pred=RFR.predict(x_test)\n",
  "y_pred"
 ]
},
{
 "cell_type": "code",
 "execution_count": 18,
 "id": "03827c33",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "array([1755.21422586, 1415.80367242,  486.00579377, ...,   10.19952956,\n",
     "       1619.69721272, 2897.79356349])"
    ]
   },
   "execution_count": 18,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "#predccition in the train data \n",
  "pred=RFR.predict(x_train)\n",
  "pred"
 ]
},
{
 "cell_type": "code",
 "execution_count": 19,
 "id": "58207376",
 "metadata": {},
 "outputs": [
  {
   "data": {
    "text/plain": [
     "0.9098600729923303"
    ]
   },
   "execution_count": 19,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "#FInding accuracy\n",
  "\n",
  "from sklearn.metrics import r2_score\n",
  "\n",
  "acc=r2_score(y_test,y_pred)\n",
  "\n",
  "acc"
 ]
},
{
 "cell_type": "code",
   "execution_count": 20,
   "id": "22a39454",
   "metadata": {},
   "outputs": [
    {
     "data": {
      "text/plain": [
       "['Power_Prediction.sav']"
      ]
     },
     "execution_count": 20,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "import joblib\n",
    "joblib.dump(RFR, \"Power_Prediction.sav\")"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 21,
   "id": "6f8c0d0f",
   "metadata": {},
   "outputs": [],
    "source": [
     "joblib.dump(RFR, open(r'C:\\Users\\balas\\Desktop\\IBM BALA PROJECT\\Project
Development Phase\\App Build Flask\\Power_Prediction.sav', 'wb'))"
    ]
   }
  ],
  "metadata": {
   "kernelspec": {
    "display_name": "Python 3 (ipykernel)",
    "language": "python",
    "name": "python3"
   },
   "language_info": {
    "codemirror_mode": {
     "name": "ipython",
     "version": 3
    },
    "file_extension": ".py",
    "mimetype": "text/x-python",
    "name": "python",
    "nbconvert_exporter": "python",
    "pygments_lexer": "ipython3",
    "version": "3.9.13"
   }
  },
  "nbformat": 4,
"nbformat_minor": 5}
```

Github and Project Video Demo Link

Github Link:

https://github.com/IBM-EPBL/IBM-Project-21166-1659774471

Project Video Demo Link:

https://drive.google.com/drive/folders/1K5JrMbKoMU7QTVI7VxukRjaoIZ6E1Qq-?usp=share_link