PREDICTING THE ENERGY OUTPUT OF WIND TURBINE BASED ON WEATHER CONDITIONS

PROJECT REPORT

Submitted by

Akhil P	193002009
Harish R M	193002035
Meenatchi V	193002058
Melferd Fernando M	193002059

EEC ELECTIVE

Department of Electronics and Communication Engineering

Sri Sivasubramaniya Nadar College of Engineering

(An Autonomous Institution, Affiliated to Anna University)

Rajiv Gandhi Salai (OMR), Kalavakkam – 603 110

ODD SEM 2022 – 2023

Project Report Format

CHAPTER – I
INTRODUCTION

1.1 Project Overview

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 due to change in the weather conditions. 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

Renewable energy sources, in contrast to non-renewable sources are replenished in a short period of time. Windmill farms produce clean energy with zero waste and emissions thus making it a good alternative over conventional energy sources. The biggest challenge in making a windmill farm reliable is to determine how much power will be generated over a range of weather conditions so as to balance the supply and demand. Thus, our motivation for the project was to improve the existing wind farms by proposing a regression model to predict the energy output of the wind farms based on weather conditions. Our model will help to overcome one of the critical drawbacks in windmill farms thus making the system more reliable.

CHAPTER – II

LITERATURE SURVEY

2.1 Existing Problems

In paper [1], they have presented an efficient Bi – LSTM (Bidirectional – Long Short Term Memory) model for predicting the wind power generated based on weather conditions. For the dataset they have sourced it from National Renewable Energy Laboratory (NREL) which contains hourly wind data for about six years. The dataset contains the following features - timestamp (date and time), air temperature (in degree Celsius), pressure (in atmosphere), wind direction (in degrees), wind speed (in meter per second) and power generated by the system (kilo Watt). Root Mean Square Error (RMSE) have been computed to compare the effectiveness of different network models fitted to a particular time series. The system achieved good prediction results with a root mean square error of 1.358. The work also includes a Graphical User Interface (GUI) designed for the users.

2.2 References

[1] S. Preethi, H. Prithika, M. Pramila and S. Birundha, "Predicting the Wind Turbine Power Generation based on Weather Conditions," 2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA), 2021, pp. 132-139, doi: 10.1109/ICECA52323.2021.9676051.

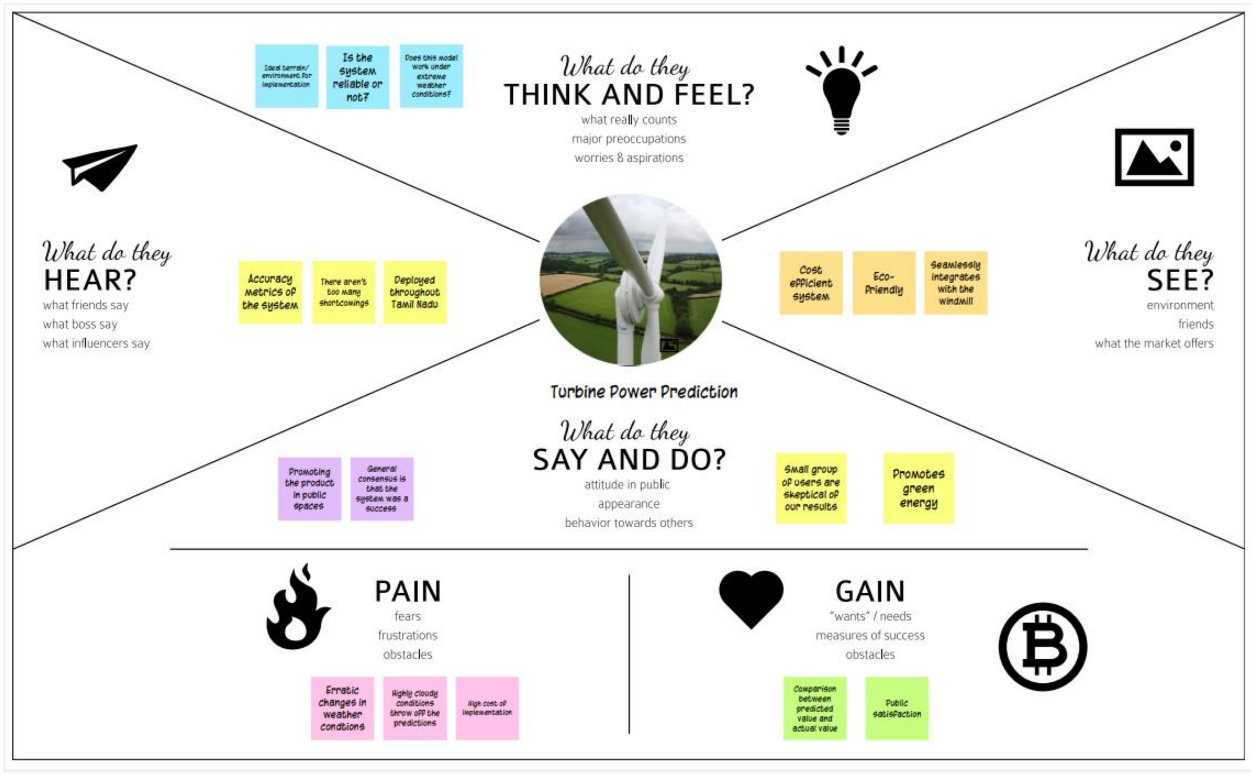
2.3 Problem Statement Definition

One of the most critical drawback in windmill farms is the uncertainty of the power it can generated thus reducing the reliability of the system. The power generated is directly proportional to the wind speed. Since, wind speed is affected by weather it is necessary to model the relationship between weather conditions and power generated.

CHAPTER – III

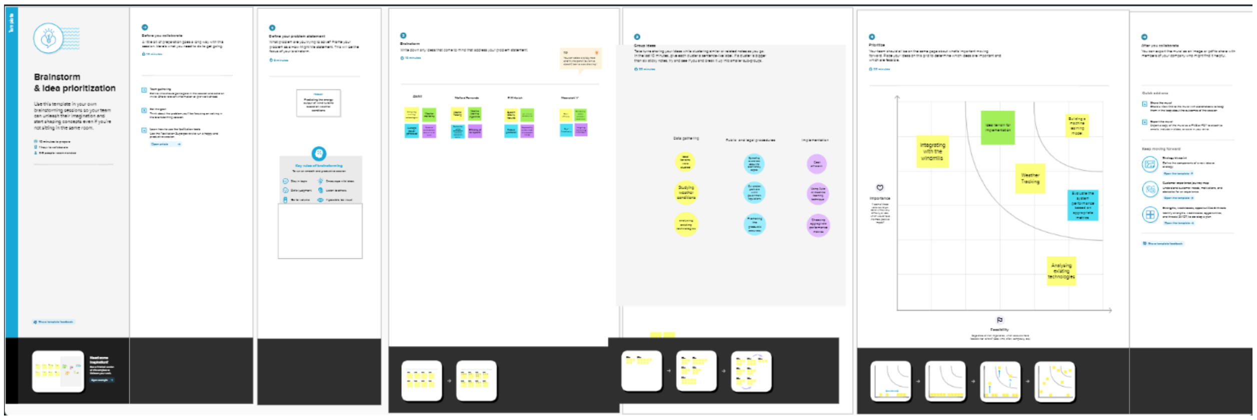
IDEATION AND PROPOSED SOLUTION

3.1 Empathy map canvas:


Ideal terrain/ environment for implementation
Is the system reliable or not?
Does this model work under extreme weather conditions?
What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
What do they HEAR?
what friends say
what boss say
what influencers say
Accuracy metrics of the system
There aren't too many shortcomings
Deployed throughout Tamil Nadu
Cost efficient system
Eco-friendly
Seamlessly integrates with the windmill
What do they SEE?
environment
friends
what the market offers
Turbine Power Prediction
Promoting the product in public spaces
General consensus is that the system was a success
What do they SAY AND DO?
attitude in public
appearance
behavior towards others
Small group of users are skeptical of our results
Promotes green energy
PAIN
fears
frustrations
obstacles
Erratic changes in weather condtions
Highly cloudy conditions throw off the predictions
High cost of implementation
GAIN
"wants" / needs
measures of success
obstacles
Comparison between predicted value and actual value
Public satisfaction

3.2 Ideation and brainstorming:

In this brainstorming session we defined our problem statement, addressed various challenges involved in making our model ready for use. We have also prioritized our ideas based on feasibility and importance.



3.3 Proposed solution:

Problem statement:

Wind power generation differs from conventional thermal power generation due to the stochastic nature of wind. 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.

Idea/ Solution description:

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.

Novelty:

This project builds upon existing implementations and uses state of the art technologies to achieve highly accurate predictions.

Social impact:

Our project employs machine learning algorithms. This can be seamlessly integrated in existing windmill farms since it does not require drastic changes in the windmill infrastructure. This will attract customers.

Business model:

The revenue model for this project should consist of the following, initial investments which covers the construction and real estate costs, operating costs which includes maintenance and repair costs and profits. This can be modelled in a business performance dashboard for presenting the results.

Scalability:

The predictive analysis used here can be extended to other forms of energy generation such as hydroelectricity, tidal energy and solar energy.

CHAPTER – IV

REQUIREMENT ANALYSIS

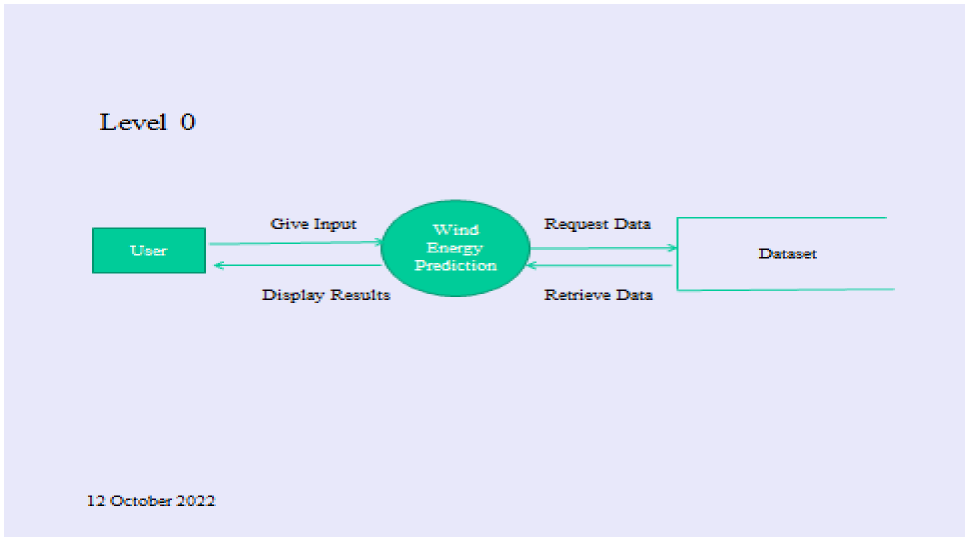
4.1 Functional requirement:

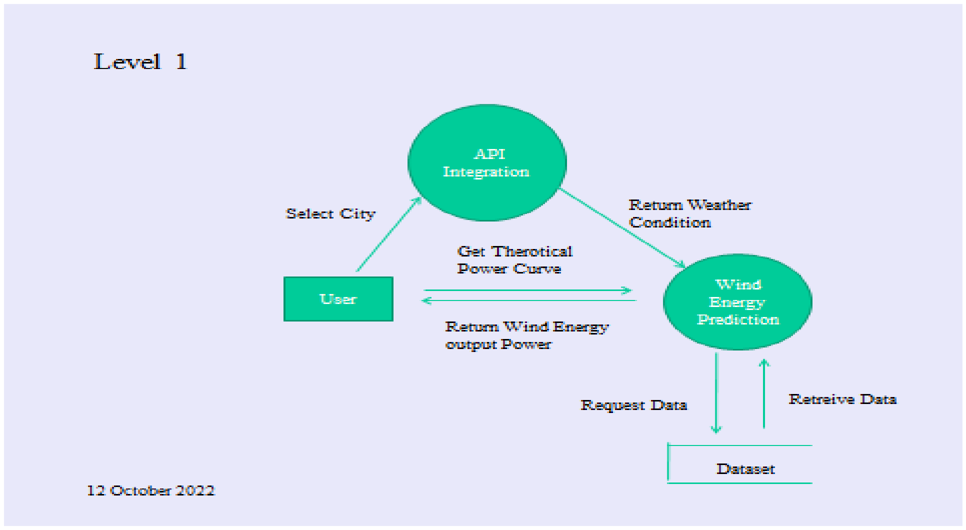
Most floating offshore wind turbines (FOWTs) have been designed in a sequential manner, with the controller being designed and optimized toward the end of the design process. Treating the controller design as an add-on to the constituent component designs does not capitalize on the inherent design coupling between the system dynamics and controller behavior, resulting in suboptimal system designs. The goal in controls co- design (CCD) is to bring all components, including the controller, together in a concurrent design and optimization approach that properly accounts for this coupling. CCD will be critical to realizing FOWT cost reductions that will position offshore wind as highly competitive with other energy sources. The Aerodynamic Turbines Lighter and Afloat with Nautical Technologies and Integrated Servo control (ATLANTIS)program funded by the U.S. Department of Energy (DOE) Advanced Research Projects Agency-Energy (ARPA-E) seeks to develop new technology pathways for the design of economically competitive FOWTs. Within ATLANTIS Topic Area 2 (Computer Tools), the National Renewable Energy Laboratory (NREL) and collaborators from the University of Illinois Urbana-Champaign and Colorado State University have been awarded a project that will develop the Wind Energy with Integrated Servo- control (WEIS) toolset, with the goal of providing the offshore wind industry and research communities with an open-source, user-friendly, flexible tool to enable true CCD of the FOWT physical design together with the controller.

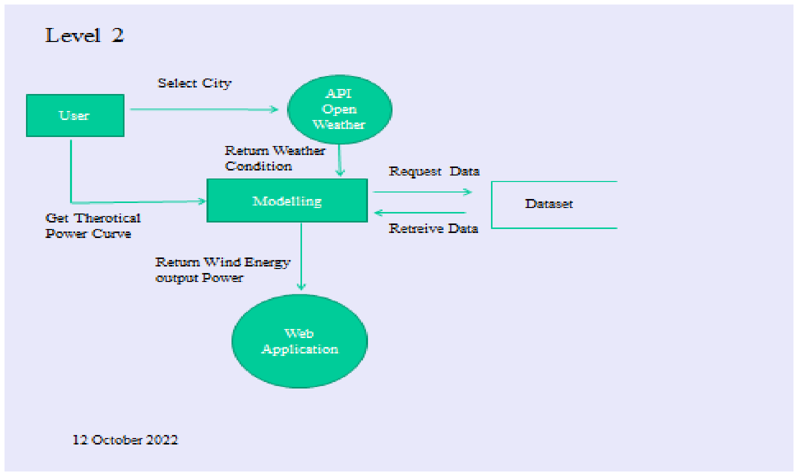
CHAPTER – V

PROJECT DESIGN

5.1 Data flow diagrams:

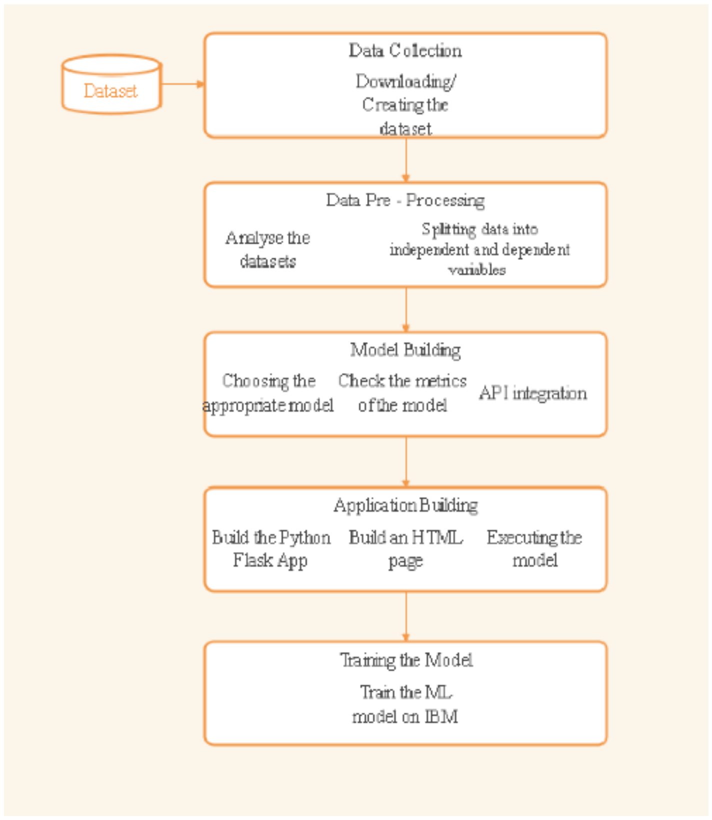






5.2 Solution & Technical Architecture:

Solution architecture diagram:


Dataset
Data Collection
Downloading/ Creating the dataset
Data Pre - Processing
Analyse the datasets
Splitting data into independent and dependent variables
Model Building
Choosing the appropriate model
Check the metrics of the model
API integration
Application Building
Build the Python Flask App
Build an HTML page
Executing the model
Training the Model
Train the ML model on IBM

Technical architecture:

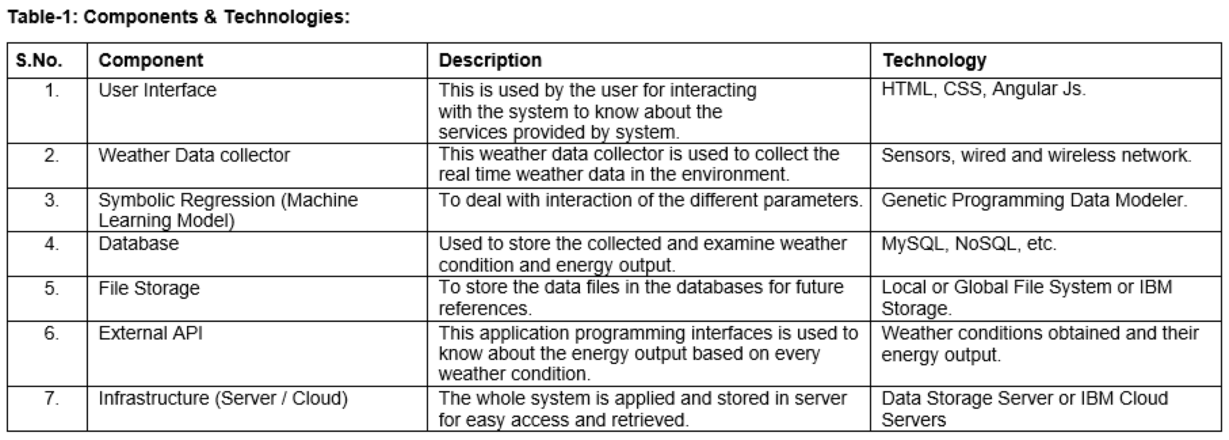


Table-1: Components & Technologies:

S.No.	Component	Description	Technology
1.	User Interface	This is used by the user for interacting with the system to know about the services provided by system.	HTML, CSS, Angular Js.
2.	Weather Data collector	This weather data collector is used to collect the real time weather data in the environment.	Sensors, wired and wireless network.
3.	Symbolic Regression (Machine Learning Model)	To deal with interaction of the different parameters.	Genetic Programming Data Modeler.
4.	Database	Used to store the collected and examine weather condition and energy output.	MySQL, NoSQL, etc.
5.	File Storage	To store the data files in the databases for future references.	Local or Global File System or IBM Storage.
6.	External API	This application programming interfaces is used to know about the energy output based on every weather condition.	Weather conditions obtained and their energy output.
7.	Infrastructure (Server / Cloud)	The whole system is applied and stored in server for easy access and retrieved.	Data Storage Server or IBM Cloud Servers

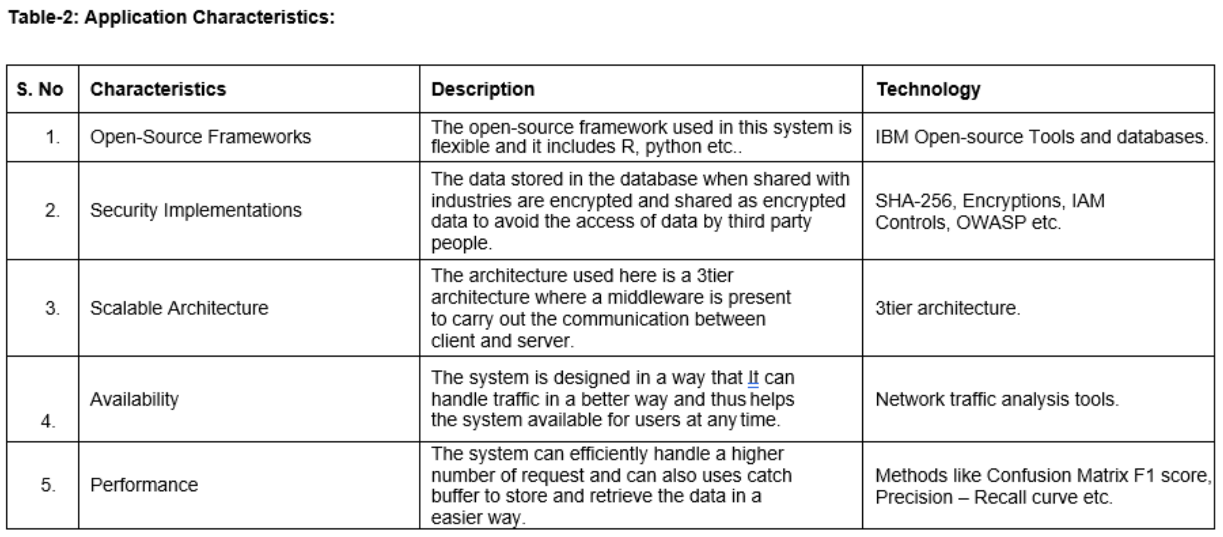


Table-2: Application Characteristics:

S. No	Characteristics	Description	Technology
1.	Open-Source Frameworks	The open-source framework used in this system is flexible and it includes R, python etc..	IBM Open-source Tools and databases.
2.	Security Implementations	The data stored in the database when shared with industries are encrypted and shared as encrypted data to avoid the access of data by third party people.	SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	The architecture used here is a 3tier architecture where a middleware is present to carry out the communication between client and server.	3tier architecture.
4.	Availability	The system is designed in a way that It can handle traffic in a better way and thus helps the system available for users at any time.	Network traffic analysis tools.
5.	Performance	The system can efficiently handle a higher number of request and can also uses catch buffer to store and retrieve the data in a easier way.	Methods like Confusion Matrix F1 score, Precision – Recall curve etc.

5.3 User Stories:

User Type	Functional Requirements	User Number Story	User Story/User Task	Acceptance Criteria	Priority	Release

Customer	Home (Application)	USN-1	As a user, I can view the guideline as well as the detailed information about the application	I can gain knowledge by practical method to use this application.	Low	Sprint-1
		USN-2	As a User, I can use this application by reading the instruction s	I can use this in user friendly method by reading the instruction .	Low	Sprint-1
		USN-3	As a User, I can login and by entering the correct username and password	If login is correctly entered, I can navigate to the next page.	Low	Sprint-2
		USN-4	As a user, I am allowed to select the city and can get the weather of the city.	I can select the city, if the city is correct, I can further enter the details.	Medium	Sprint-3

CHAPTER - VI

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	Akhil P Melferd M Harish R M Meenatchi V
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	5	High	Akhil P Melferd M Harish R M Meenatchi V
Sprint-1		USN-3	As a user, I can register for the application using phone number	2	Low	Akhil P Melferd M Harish R M Meenatchi V
Sprint-1		USN-4	As a user, I can register for the application through Gmail	3	Medium	Akhil P Melferd M Harish R M Meenatchi V
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	5	High	Akhil P Melferd M Harish R M Meenatchi V

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Dashboard	USN-6	Once I have logged in, I can see my dashboard.	6	Medium	Akhil P Melferd M Harish R M Meenatchi V
Sprint-2	Web Access	USN-7	As a customer I can access the website to predict the turbine power.	7	High	Akhil P Melferd M Harish R M Meenatchi V
Sprint-2	Prediction	USN-8	As a customer when I enter the weather details the website should predict the approximate turbine power.	7	High	Akhil P Melferd M Harish R M Meenatchi V
Sprint-3	Analysis	USN-9	As a customer, I wish to store my predictions and make analyses.	10	Medium	Akhil P Melferd M Harish R M Meenatchi V
Sprint-3	Security	USN-10	As a customer I expect my data to be secured.	10	Medium	Akhil P Melferd M Harish R M Meenatchi V
Sprint-4	Database Access	USN-11	As an administrator, I should maintain the website and update the website regularly.	20	Low	Akhil P Melferd M Harish R M Meenatchi V

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		
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022		
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022		
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022		

Velocity:

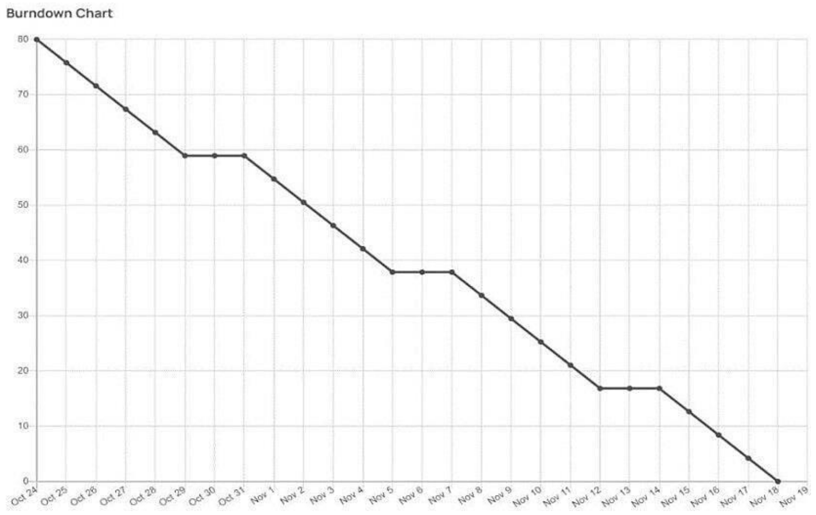
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

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

Burndown Chart:

A burndown chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

6.3 Reports from Jira:



CHAPTER – VII

CODING AND SOLUTIONING

7.1 Data visualization:

```
#Importing necessary libraries
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import joblib
import sklearn
import warnings
from sklearn.model_selection import train_test_split
path = 'C:\\Users\\Admin\\Desktop\\T1.csv'
df = pd.read_csv(path)
```

df.rename(columns={'Date/Time':'Time','LV ActivePower (kW)':'ActivePower(kW)',"Wind Speed (m/s)":"WindSpeed (m/s)","Wind Direction (°)":"Wind_Direction"},inplace=True)

7.2 Data pre-processing:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import Lasso
from sklearn.linear_model import Ridge
from sklearn.metrics import mean_squared_error , r2_score
import joblib



data = pd.read_csv('/content/drive/MyDrive/T1.csv')
data.rename(columns = {'LV ActivePower (kW)':'ActivePower(kW)',
                       "Wind Speed (m/s)":"WindSpeed(m/s)",
                       "Wind Direction (°)":"WindDirection","Theoretical_P
ower_Curve (KWh)":"TheoreticalPowerCurve(KWh)"},
            inplace = True)
data.head()



data['Date/Time'] = pd.to_datetime(data['Date/Time'],format='%d %m %Y %H:%
M')
data['year'] = data['Date/Time'].dt.year
data['month'] = data['Date/Time'].dt.month
data['day'] = data['Date/Time'].dt.day
data['Hour'] = data['Date/Time'].dt.hour
data['minute'] = data['Date/Time'].dt.minute
data.head()



data["Date/Time"] = pd.to_datetime(data["Date/Time"], format = "%d %m %Y %
H:%M", errors = "coerce")
data
X=data[['WindSpeed(m/s)','WindDirection']]
X.head()
y = data['ActivePower(kW)']
y.head()
X_train, X_test,y_train, y_test = train_test_split(X,y ,
                                   random_state=6,
                                   test_size=0.25)
```

7.3 Model training:

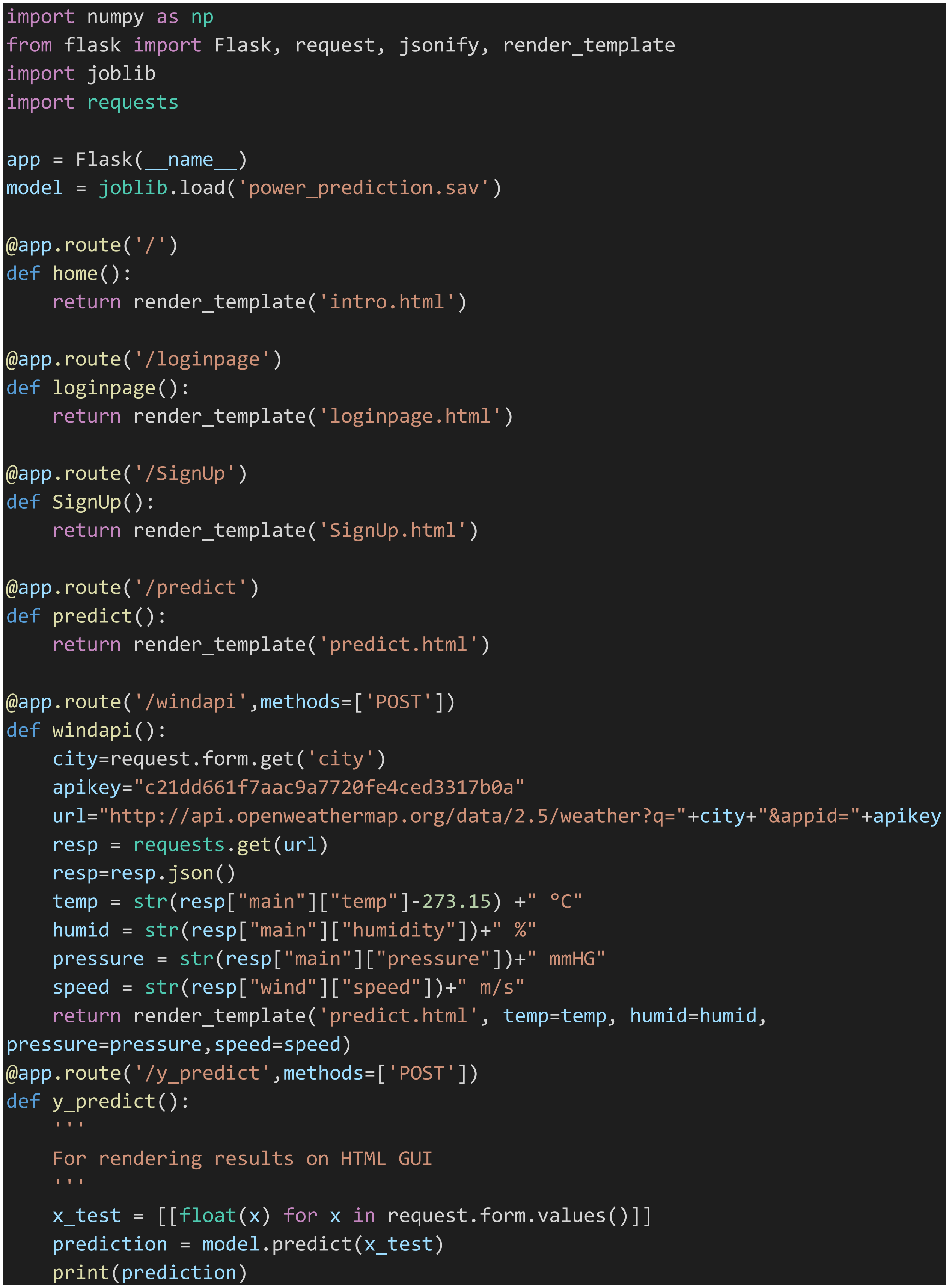
```
from sklearn.tree import DecisionTreeRegressor
from sklearn.svm import SVR
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
from sklearn.metrics import accuracy_score,r2_score,mean_squared_error
xgr=XGBRegressor()
rf=RandomForestRegressor()
lr=LinearRegression()
dt=DecisionTreeRegressor()
sm=SVR()



model_xg=xgr.fit(X_train,y_train)
y_xg=model_xg.predict(X_test)
model_rf=rf.fit(X_train,y_train)
y_rf=model_rf.predict(X_test)
model_lr=lr.fit(X_train,y_train)
y_lr=model_lr.predict(X_test)
model_dt=dt.fit(X_train,y_train)
y_dt=model_dt.predict(X_test)
model_sm=sm.fit(X_train,y_train)
y_sm=model_sm.predict(X_test)

params={
 "learning_rate"    : [0.05, 0.01,0.03,0.1, 0.15, 0.2] ,
 "n_estimators"     : [50, 100, 150, 200, 500, 800,1000,1500] ,
 "max_depth"        : [ 3, 4, 5, 6, 8, 10, 12, 15,20,25],
 "min_child_weight" : [ 1, 3, 5, 7 ,10,15,20,25],
 "gamma"            : [ 0.0, 0.1, 0.2 , 0.3, 0.4 ],
 "subsample"        : [ 0.1, 0.2 , 0.3, 0.4,0.6,0.8,1 ],
 "reg_lambda"       : [ 0.0, 0.1, 0.2 , 0.3, 0.4 ,0.6,0.8,1],
 "reg_alpha"        : [ 0.0, 0.1, 0.2 , 0.3, 0.4 ],
 "colsample_bytree" : [ 0.3, 0.4, 0.5 , 0.7,0.9 ],
 "colsample_bylevel" : [ 0.3, 0.4, 0.5 , 0.7,0.9 ]

}
```

```
## Ensemble Learning
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import VotingRegressor
from sklearn.svm import SVR
from sklearn.ensemble import RandomForestRegressor
import xgboost as xgb
from xgboost import XGBRegressor
from sklearn.metrics import accuracy_score,r2_score,mean_squared_error
xgr=XGBRegressor(base_score=0.5, booster='gbtree', colsample_bylevel=0.7,
             colsample_bynode=1, colsample_bytree=0.3, gamma=0.2,
             importance_type='gain', learning_rate=0.03, max_delta_step=0,
             max_depth=8, min_child_weight=25, missing=None, n_estimators=
800,
             n_jobs=1, nthread=None, objective='reg:linear', random_state=
0,
             reg_alpha=0.2, reg_lambda=0.8, scale_pos_weight=1, seed=None,
             silent=None, subsample=0.1, verbosity=1)
sm=SVR(gamma='auto',C=50,epsilon=0.3)
rf=RandomForestRegressor(n_estimators=500,max_depth=4)
lr=LinearRegression()
model=VotingRegressor([('lr',lr), ('rf',rf),('sm', sm),('xgr',xgr)],weight
s=[1,1,2,3])
```

7.4 Evaluation metrics:

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

print('R2-rf',r2_score(y_test,y_rf))
print('RMSE-rf',np.sqrt(mean_squared_error(y_test,y_rf)))

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

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

print('R2-svm',r2_score(y_test,y_sm))
print('RMSE-svm',np.sqrt(mean_squared_error(y_test,y_sm)))
```

7.5 Flask implementation:



```python
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('/loginpage')
def loginpage():
    return render_template('loginpage.html')

@app.route('/SignUp')
def SignUp():
    return render_template('SignUp.html')

@app.route('/predict')
def predict():
    return render_template('predict.html')

@app.route('/windapi',methods=['POST'])
def windapi():
    city=request.form.get('city')
    apikey="c21dd661f7aac9a7720fe4ced3317b0a"
    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"])+" m/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.predict(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)
```

HTML files:

```
<html>


	<head>
		<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/semantic-
ui@2.5.0/dist/semantic.min.css">
   <script src="https://cdn.jsdelivr.net/npm/semantic-ui@2.5.0/dist/semantic.min.js"></script>
	<title>Wind Energy Prediction</title>
	<style>


	body
	{
		width: 100%;
		height: auto;
		padding: 0;
		margin: 0;
		background:#edc77f;
	}
	#headbar
	{
		font-size: 30px;
```

```
        text-align: center;
        font-weight: bold;
        background-color:#6c493a;
        color: white;
        padding-left:0px;
        padding-top:30px;
        padding-bottom: 30px;
}
.second{
        top:80px;
        bottom:0px;
        margin:0px;
        left: 0px;
        right: 0px;
        position: fixed;
        padding: 0px;
        width: 100%;
        background-image:url(../static/m123.gif);
        background-repeat:no-repeat;
        background-size: contain;
}
.inside{
        top:80px;
        bottom:0px;
        margin:0px;
        left: 45%;
```

```
		right: 0%;

		position: fixed;

		padding-left: 40px;

		padding-top:8%;

		padding-right:40px;

		background-color:#F2D19A;

		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: 12px;

		background-color: #6c493a;

		padding: 10px 15px;

		margin: 0 auto;

		box-shadow: 0 5px 15px rgba(0,0,0,0.20);
```

```
        }

        </style>

        </head>

        <body>

                <div id="headbar">

                <header>Wind Turbine Energy Prediction Based on Weather Condition</header>

                <a href="{{url_for('loginpage')}}" id="loginPage">

                        <button style="position: relative; bottom:25px;left: 630px;" class="ui
    inverted button">Login</button>

                </a>

                </div>

                <div class="second">

                        <div class="inside">Renewable energy, such as wind and solar energy, plays
    an increasing role in the supply of energy worldwide. This trend will continue because global
    energy demand is increasing, and the use of nuclear power and traditional sources of energy such
    as coal and oil is unsafe and leads to a large amount of CO2 emission. Wind energy is a key player in
    the field of renewable energy. In Europe, the capacity of wind energy production has doubled from
    2009 to 2010. <br><br>

    However, levels of production of wind energy are hard to predict as they rely on potentially
    unstable weather conditions present at the wind farm. In particular, wind speed is crucial for
    energy production based on wind, and it may vary drastically over time. Energy suppliers are
    interested in accurate predictions, as they can avoid overproduction by coordinating the
    collaborative production of traditional power plants and weather-dependent energy sources. The
    energy can be predicted based on the power curve and the windspeed.

                <br><br><br>

                <a href="{{url_for('predict')}}" id="predict"><button type="button"
    class="myButton" >Want to predict the energy??</button></a>

                </div>




                </div>

        </body>
```

```
</html>

Login Page HTML page:

<!DOCTYPE html>

<html lang="en">

<head>

  <title>DayOut Guide</title>

  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/semantic-
ui@2.5.0/dist/semantic.min.css">

  <script src="https://cdn.jsdelivr.net/npm/semantic-ui@2.5.0/dist/semantic.min.js"></script>



  <!-- Final Script -->

  <script type="text/javascript">

   function valid()

   {

     var v1=document.getElementById("tt1");

     var er1=document.getElementById("un");

     if(v1.value=="")

     {

       er1.innerHTML="Username is empty";

       v1.focus();

       return false;

     }


     var v3=document.getElementById("pp2");

     var er3=document.getElementById("email");

     if(v3.value=="")
```

```
{
  er3.innerHTML="Email is empty";
  v3.focus();
  return false;
}
else
{
function myFunc() {
  location.href.document.getElementById("myBtn");
  }
}


var v2=document.getElementById("pp1");
var er2=document.getElementById("pwd");
if(v2.value=="")
{
  er2.innerHTML="Password is empty";
  v2.focus();
  return false;
}


if(v1.value=="Simon" && v2.value=="simon2512")
{
  alert("Logged in Successfully");
```

```
   return true;
  }
  else
  {
   alert("Invalid Password or Username");


   v1.value="";
   v2.value="";
   v3.value="";
   v1.focus();
   return false;
  }

 }
</script>

<style type="text/css">
 .colo
 {
  color: red;
  font-weight: bold;
  margin-left: 60px;
 }
</style>

<link rel="stylesheet" href="../static/loginpage.css">
```

```
   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">

</head>

<body >

  <header>Wind Turbine Energy Prediction Based on Weather Condition</header>

                            <div class="regform" >

     <h3>Login to Explore More!</h3>

     <form action="{{url_for('home')}}"id="/" method="get" onsubmit="return valid()"
autocomplete="off">

        <div class="input">

           <label class="textlabel" for="name"><b>User Name</b></label>

           <div class="inline field">

             <input type="text" id="tt1" name="t1" placeholder="Username" >

             <span id="un" class="colo"></span>

           </div>

        </div>

        <div class="input">

           <label class="textlabel" for="email"><b>Email</b></label>

           <div class="inline field">

             <input type="email" id="pp2" name="p2"  placeholder="email">

             <span id="email" class="colo"></span>

           </div>

        </div>


        <div class="input">

           <label class="textlabel" for="pwd"><b>Password</b></label>

           <div class="inline field">
```

```
             <input type="password" id="pp1" name="p1" placeholder="password" >

             <span id="pwd" class="colo"></span>

            </div>

          </div>

          <div class="button">

          <button id="loginlink" name="login" class="ui inverted button"  >

           login

          </button>

         </div>




        </form>

        <h4 id="create">Create an account?

         <button class="ui inverted button" id="myBtn" ><a style="color: white;"
    href="{{url_for('SignUp')}}">SignUp</button>

          <div id="myModal" class="modal">

           <div class="modal-content">

            <span class="close">&times;</span>

            <p>Welcome To Signup Page</p>

         </div>

        </div>


         <!-- <script>

          var modal = document.getElementById("myModal");

          var btn = document.getElementById("myBtn");

          var span = document.getElementsByClassName("close")[0];
          btn.onclick = function() {

            modal.style.display = "block";

          }

          span.onclick = function() {

            modal.style.display = "none";

          }

          window.onclick = function(event) {

            if (event.target == modal) {

              modal.style.display = "none";

            }

          }

          </script>          -->

</body>

</html>
```

Predict HTML file:

```
<!DOCTYPE html>

<html lang="en">

<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="https://cdn.jsdelivr.net/npm/semantic-
ui@2.5.0/dist/semantic.min.css">
```

```
<script src="https://cdn.jsdelivr.net/npm/semantic-ui@2.5.0/dist/semantic.min.js"></script>

  <style>
    body
    {
      width: 100%;
      height: auto;
      padding: 0;
      margin: 0;
      background:#edc77f;
    }
    header
    {
      font-size: 30px;
      text-align: center;
      font-weight: bold;
      background-color:#6c493a;
      color: white;
      padding-left:0px;
      padding-top:30px;
      padding-bottom: 30px;
    }
    #citycontent
    {
      margin-left: 150px;
      margin-top: 50px;
    }
    #dropdown
    {
      margin-left: 110px;
    }
    #button{
                        border: none;
                        text-align: center;
                        cursor: pointer;
                        outline: none;
                        overflow: hidden;
                        color: #fff;
                        font-weight: 700;
                        font-size: 12px;
                        background-color: #50d838;
                        padding: 10px 15px;
                        margin: 0 auto;
                        margin-left:17%;
               }
    #button1
    {
      margin-top: 50px;
      margin-left: 75px;
      color: white;
      font-weight: bold;
    }
```

```
#button2
{
  margin-top: 50px;
  margin-left: 95px;
  color: white;
  font-weight: bold;
  background-color: green;
}
#calculations
{
   background-color: rgb(0, 0, 0,0.5);
   width: 440px;
   margin-left: 130px;
   margin-top: 30px;
   padding: 20px;
   color: white;
   border: 2px solid black;
   border-radius: 30px;
   height: 310px;
}
#weathercalculationcontent
{
   font-size: 20px;
   margin-bottom: 30px;
 }
```

```
#table
{
   font-size: 18px;
}
.label
{
   padding-bottom: 20px;
}
.input
{
   padding-bottom: 20px;

}
.input1
{
   border-radius: 8px;
   border: 1px solid white;
   box-sizing: border-box;
   box-shadow: 2px;
   box-shadow: 2px;
   padding: 6px 6px;
   margin: 0px 0px 0px 45px;
   width: 90%;
   font-size: 15px;
   color: white;
   font-weight: bold;
```

```
}
.ip
{
   border-radius: 8px;
   border: 1px solid white;
   box-sizing: border-box;
   box-shadow: 2px;
   padding: 6px 6px;
   /* margin: 0px 0px 0px 45px; */
   width: 52%;
   font-size: 15px;
}
.ip1
{
   border-radius: 8px;
   border: 1px solid white;
   box-sizing: border-box;
   box-shadow: 2px;
   padding: 6px 6px;
   /* margin: 0px 0px 0px 45px; */
   width: 100%;
   font-size: 15px;
}
#output
{
   float: right;
```

```
        /* background-color: rgb(0, 0, 0,0.5); */
        /* border: 1px solid black; */
        height: 420px;
        width: 440px;
        margin-right: 200px;
        border-radius: 30px;
        position: relative;
        bottom: 80px;
        padding: 40px;
        color: #000000;
      }
    </style>
    <title>Document</title>
  </head>
  <body>
    <header>Wind Turbine Energy Prediction Based on Weather Condition</header>
    <form action="{{ url_for('windapi')}}"method="post" >
    <div id="citycontent">
      <h3>Give your city name to know weather conditions..</h3>
      <br>
      <div id="dropdown">
        <select name="city"  required  class="ui dropdown">
          <option disabled value="">Select City</option>
            <option value ="Agartala">Agartala</option>
            <option value ="Ariyalur"    >          Ariyalur           </option>
                                                   <option  value ="Aizawl">Aizawl</option>
```

```
                                              <option  value ="Bengaluru">Bengaluru</option>
                                              <option  value ="Bhopal"        >         Bhopal </option>
                                              <option  value ="Bhubaneswar"        >         Bhubaneswar
</option>
  <option value ="Cuddalore">         Cuddalore       </option>
                                              <option value ="Chandigarh"   >         Chandigarh
</option>
  <option value ="Coimbatore"         >         Coimbatore     </option>
                                              <option value ="Chennai"        >         Chennai
</option>
                                              <option value ="Daman"          >         Daman </option>
                                              <option value ="Dehradun"     >         Dehradun
</option>
                                              <option value ="Delhi">         Delhi    </option>
                                              <option value ="Dispur"          >         Dispur  </option>
  <option value ="Erode"       >         Erode   </option>
                                              <option value ="Gandhinagar">         Gandhinagar
</option>

                                              <option value ="Gangtok"       >         Gangtok
</option>

                                              <option value ="Hyderabad"   >         Hyderabad
</option>

                                              <option value ="Imphal"          >         Imphal </option>
                                              <option value ="Itanagar"       >         Itanagar
</option>

                                              <option value ="Jaipur"           >         Jaipur   </option>
                                              <option value ="Kavaratti"      >         Kavaratti
</option>

                                              <option value ="Kohima"         >         Kohima
</option>

                                              <option value ="Kolkata"         >         Kolkata</option>
```

```
    <option value ="Kanyakumari"        >           Kanyakumari   </option>
                                           <option value ="Lucknow"       >           Lucknow
</option>
    <option value ="Namakkal" >           Namakkal         </option>
                                           <option value ="Mumbai"        >           Mumbai
</option>
    <option value ="Madurai"    >           Madurai           </option>
                                           <option value ="Panaji"           >           Panaji   </option>
                                           <option value ="Patna"            >           Patna    </option>
                                           <option value ="Pondicherry"  >           Pondicherry
</option>
                                           <option value ="Port Blair"      >           Port Blair
</option>
                                           <option value ="Raipur"          >           Raipur  </option>
                                           <option value ="Ranchi"          >           Ranchi  </option>
                                           <option value ="Shillong"        >           Shillong
</option>
                                           <option value ="Shimla"          >           Shimla  </option>
                                           <option value ="Silvassa"        >           Silvassa
</option>
                                           <option value ="Srinagar"       >           Srinagar
</option>
    <option value ="Salem"        >           Salem   </option>
                                           <option value ="Thiruvananthapuram"             >
Thiruvananthapuram   </option>
                                           <option value ="Tirupati"        >           Tirupati
</option>
    <option value ="Vellore"      >           Vellore </option>
    <option value ="Virudhunagar"       >           Virudhunagar   </option>
</select>
```

```
        </div>

        <div>

            <button  id="button1" class="ui inverted green button">Check the weather
Condition</button>

        </div>

    </div>

</form>

<form action="{{ url_for('y_predict')}}" method="post">

    <div id="output">

        <div>

        <h2>Predict the wind Energy!!</h2>

        </div>

        <div class="ui divider"></div>

        <div class="inline field">

          <input class="ip" type="text" name="theo" required placeholder=" Theoritical Power">

          <div class="ui left pointing label">

            Theoritical Power in kwh

          </div>

        </div>

        <div class="ui divider"></div>

        <div class="inline field">

          <input class="ip" type="text" name="wind"required placeholder="Windspeed">

          <div class="ui left pointing label">

            Windspeed in m/s

          </div>

        </div>
        <br><br>

        <div>

          <button style="color: black;" type="submit" id="button"  class="ui inverted standard button"
>Predict the Energy</button>

          <br>

                        <br>

          <br>

                        <h3>{{ prediction_text }}</h3>

        </div>

                      </form>


    </div>




    <div id="calculations">

        <div id="weathercalculationcontent">

          <h3> The weather conditions of the city are</h3>

        </div>

        <table id="table">

          <tr>

            <td class="label"> <label  for="Temperature">Temperature</label></td>

            <td class="input"> <input class="input1" disabled  value="{{temp}}" type="text" ></td>

          </tr>

          <tr>

             <td class="label"> <label  for="Humidity">Humidity</label></td>

             <td class="input"> <input class="input1" disabled  type="text" value="{{humid}}"></td>
            </tr>
            <tr>
              <td class="label"> <label  for="Pressure">Pressure</label></td>
              <td class="input"> <input class="input1" disabled  type="text"  value="{{pressure}}"></td>
            </tr>
            <tr>
              <td class="label"> <label  for="windspeed">Windspeed</label></td>
              <td class="input"> <input class="input1" disabled type="text" value="{{speed}}"></td>
            </tr>
        </table>
    </div>


</body>
</html>
```

SignUp HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Signup</title>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/semantic-
ui@2.5.0/dist/semantic.min.css">
    <script src="https://cdn.jsdelivr.net/npm/semantic-ui@2.5.0/dist/semantic.min.js"></script>
    <link rel="stylesheet" href="../static/SignupCss.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
</head>
<body>
```

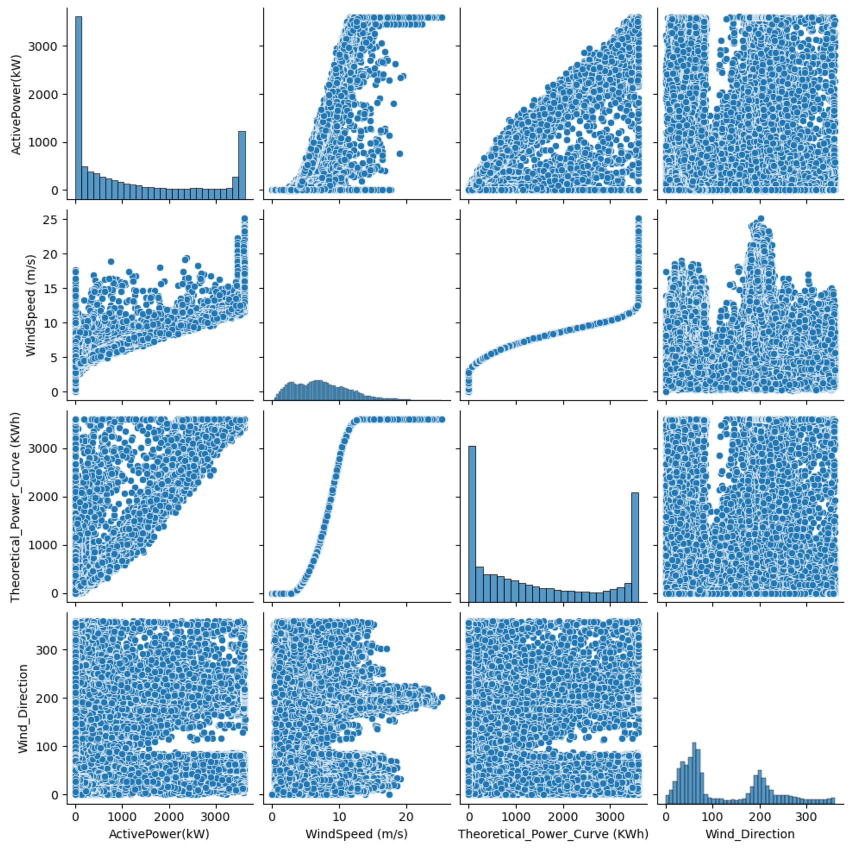
```
<header>Wind Turbine Energy Prediction Based on Weather Condition</header>


 <div class="regform">
    <h3>SignUp</h3>
 <form action="">
    <div class="input">
        <label class="textlabel" for="name"><b>Name</b></label>
        <div class="inline field">
          <input type="text" placeholder=" Name">
        </div>
    </div>



    <div class="input">
        <label class="textlabel" for="name"><b>Email Id</b></label>
        <div class="inline field">
          <input type="email" placeholder="email">
        </div>
    </div>
    <div class="input">
        <label class="textlabel" for="name"><b>Password</b></label>
        <div class="inline field">
          <input type="password" placeholder="Password">
        </div>
    </div>
    <div class="input">
```

```
            <label class="textlabel" for="name"><b>Confirm Password</b></label>

            <div class="inline field">

              <input type="password" placeholder="Confirm Password">

            </div>

        </div>



        <button id="signupbtn" ><h5><a href="{{url_for('home')}}">Sign-up</a></h5></button>


        <p id="create">Already have an account?&nbsp;<a id="signup"
href="{{url_for('loginpage')}}">login</a></p>

    </div>

    </form>

</body>

</html>
```

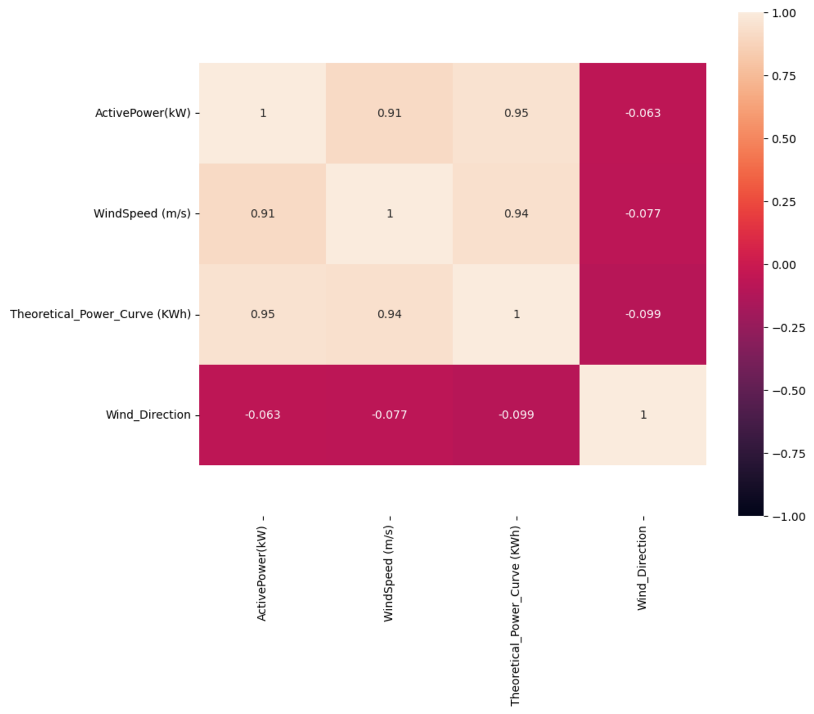
CHAPTER – IX

RESULTS

Data Visualization:



ActivePower(kW)
WindSpeed (m/s)
Theoretical_Power_Curve (KWh)
Wind_Direction



ActivePower(kW)
WindSpeed (m/s)
Theoretical_Power_Curve (KWh)
Wind_Direction
1 0.91 0.95 -0.063
0.91 1 0.94 -0.077
0.95 0.94 1 -0.099
-0.063 -0.077 -0.099 1

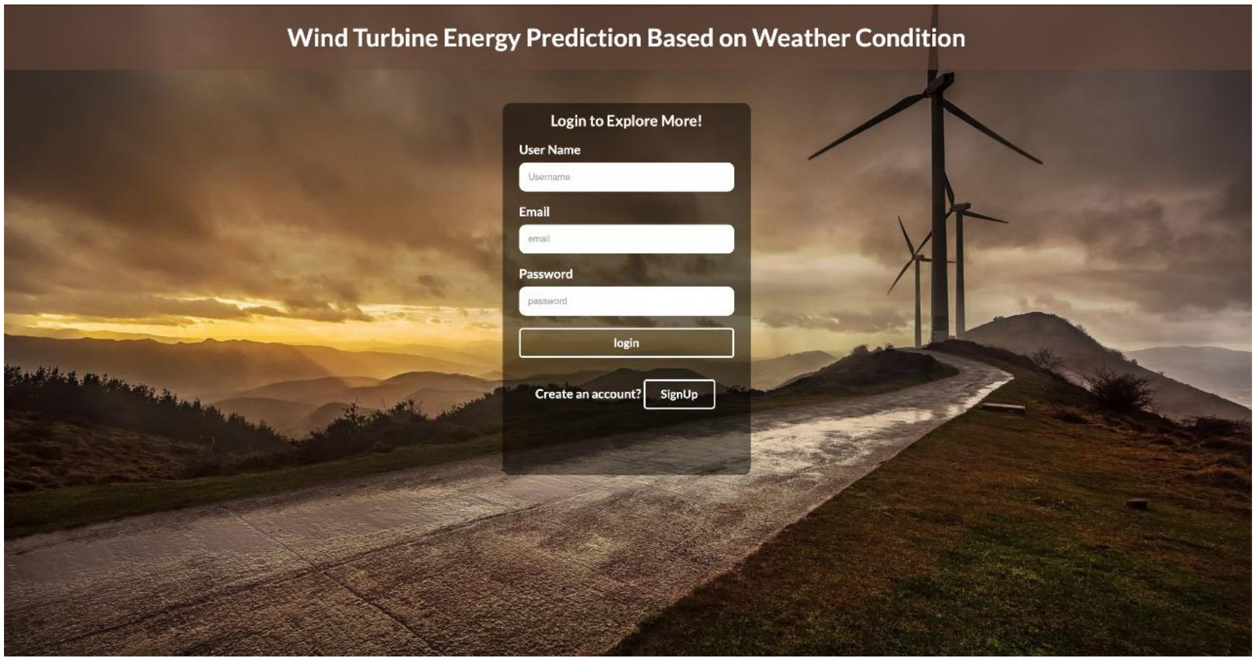
Performance metrics results:

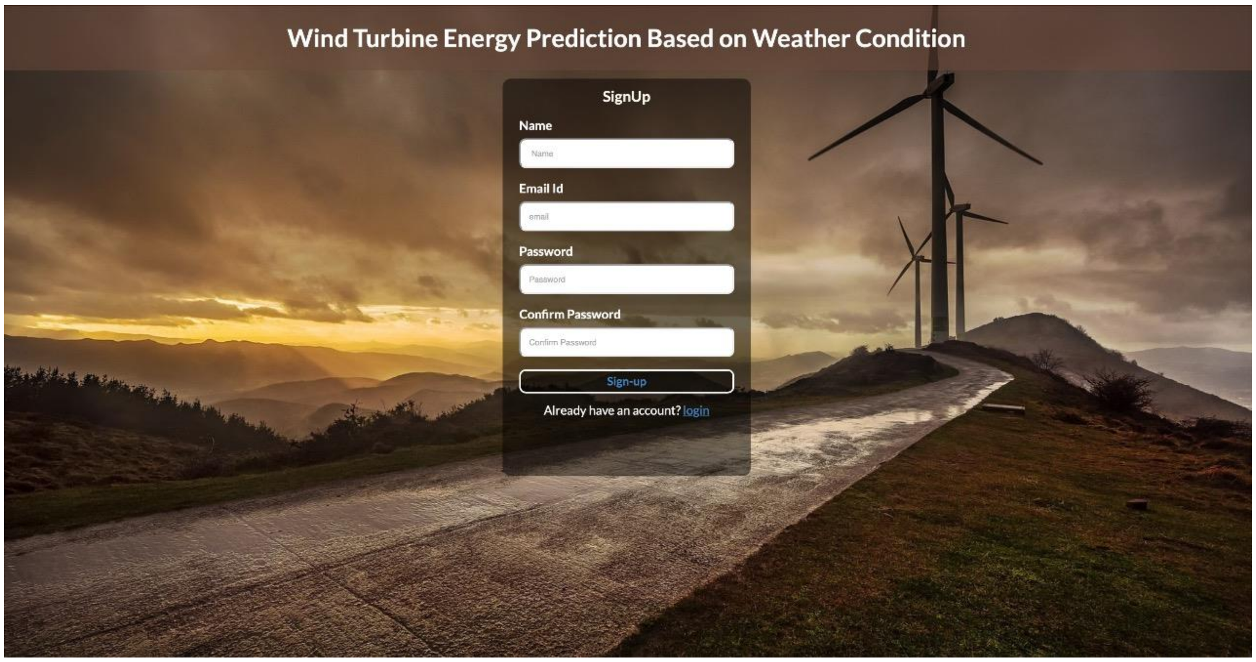
```
R2-xgb 0.9222746826171284
RMSE-xgb 364.85477293970644
R2-rf 0.9098970861855176
RMSE-rf 392.83321179326947
R2-lr 0.8368251429450982
RMSE-lr 528.6465476346768
R2-dt 0.8423138686115776
RMSE-dt 519.6794327141452
R2-svm 0.005368134807760105
RMSE-svm 1305.1786596858901
```

Hyperparameter tuning:

```
XGBRegressor(colsample_bylevel=0.4, colsample_bytree=0.5, gamma=0.2,
             learning_rate=0.01, min_child_weight=15, n_estimators=1000,
             reg_alpha=0.3, subsample=0.3)
```

Flask app





Wind Turbine Energy Prediction Based on Weather Condition

SignUp

Name

Name

Email Id

email

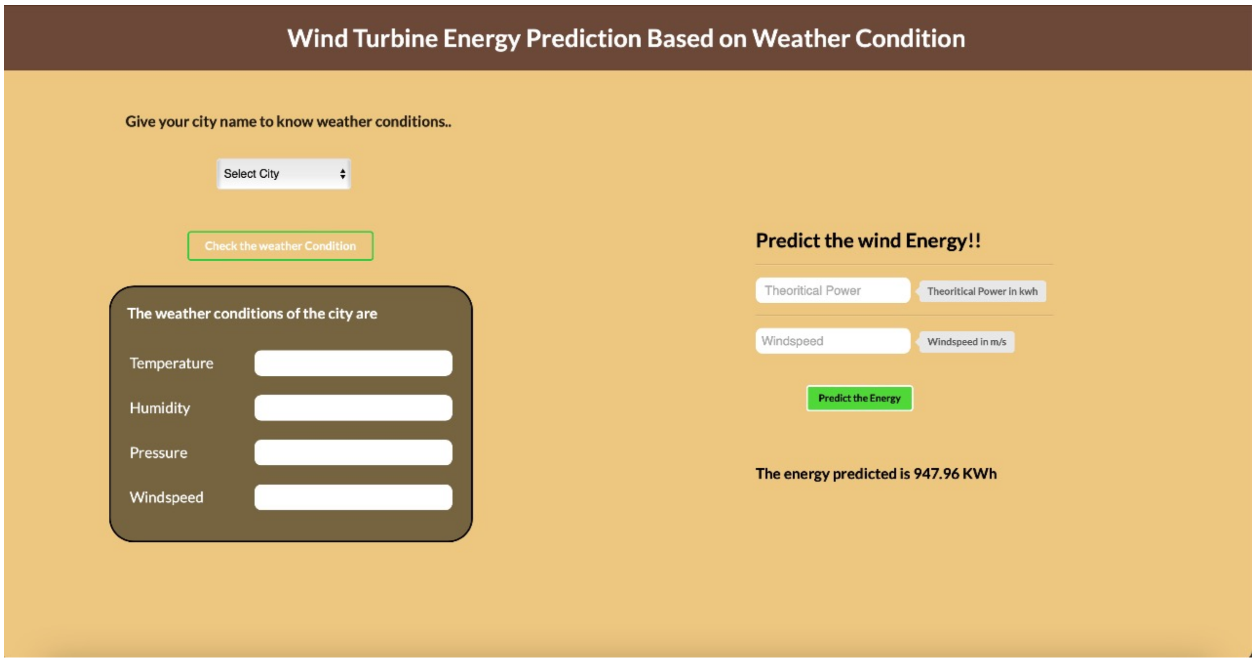
Password

Password

Confirm Password

Confirm Password

Already have an account? login



Wind Turbine Energy Prediction Based on Weather Condition

Give your city name to know weather conditions..

The weather conditions of the city are

Temperature

Humidity

Pressure

Windspeed

Predict the wind Energy!!

Theoritical Power | Theoritical Power in kwh

Windspeed | Windspeed in m/s

The energy predicted is 947.96 KWh

CHAPTER X

ADVANTAGES AND DISADVANTAGES

Advantages:

Our regression model helps in predicting the power output of the wind turbine based on weather conditions this aids in balancing the supply and demand for power with other conventional power sources. Our model can be seamlessly integrated with existing system with almost no modifications required since our system is software oriented. The maintenance cost of our model is very low with only software maintenance expenses.

Disadvantages:

Since, our model heavily relies on weather patterns for accurate prediction it is necessary to have data covering diverse weather conditions in the area at which the windmill farms are established. Machine learning models need to address issues related to training such as overfitting. It also necessary to update the model to the latest trends in machine learning to keep improving the system for accurate and reliable predictions.

CHAPTER XI

CONCLUSION

Renewable sources of energy is the need of the hour considering it offers an environment friendly alternative over conventional energy sources. We have experimented with five different regression models for predicting the energy output and settled with XGB regressor which showed good results among the five when evaluated using Root Mean Square (RMSE) and R squared metrics. We also developed an app using Flask to create an intuitive interface for the users which also

CHAPTER XII

FUTURE SCOPE

In the future, we plan to concentrate on the hardware side by placing sensors to measure the weather patterns in the area so as to have a rich data consisting of the weather patterns to make our model more reliable. We also aim to explore other machine learning and deep learning models to achieve even better prediction results.

CHAPTER – XIII

APPENDIX

GitHub repository link : IBM-EPBL/IBM-Project-20642-1659758825: Predicting the energy output of wind turbine based on weather condition (github.com)

Project demo video:

https://drive.google.com/file/d/10ckA8i2Dxq0tBnvRF2YOZyPInBifAOqp/view?usp=share_link