PREDICTING THE ENERGY OUTPUT OF WIND TURBINES BASED ON WEATHER CONDITIONS

DOMAIN : APPLIED DATA SCIENCE

TEAM ID: PNT2022TMID53075

TEAM MEMBERS:

MEGAN KIRUPA ROXANNE R - SSNCE195001063
GAYATHRI E - SSNCE195001030
KIRTHANNA RAJAN - SSNCE195001052
SANMATI P - SSNCE195001098

TABLE OF CONTENTS

1. 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. 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 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

Since 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.

This model can socially impact by increasing energy production as more power generation to the households, less power cuts.

This model can also provide solutions by collaborating with the power suppliers and the government.

2. LITERATURE SURVEY

2.1 Existing problem

1. **Improvement of wind power prediction from meteorological characterization with machine learning models - Renewable Energy, Volume 183, 2022**

Abstract

This work uses a decision tree machine learning model to assess the effectiveness of hub-height wind speed, rotor-equivalent wind speed, and lapse rate as variables in power prediction. Atmospheric data is used to train regression trees and correlate the power outputs to wind profiles and meteorological characteristics to be able to predict power responses according to physical patterns. The decision tree model was trained for four vertical wind profile classifications to showcase the need for multiple calculations of wind speed at various levels of the rotor layer. Results indicate that when compared to traditional power curve methods, the decision tree combining rotor-equivalent wind speed and lapse rate improves prediction accuracy by 22% for the given data-set, while also proving to be the most effective method in power

prediction for all classified vertical wind profile types. Models incorporating lapse rate into predictions performed better than those without it, showing the importance of considering atmospheric criteria in wind power prediction analyses.

Methods

Decision tree

Highlights

- Regression decision tree models used to improve wind power prediction.
- Correlate the power outputs to wind pro_les and meteorological charac-teristics.
- Machine learning models outperform the traditional turbine power curve approach.
- Decision tree with rotor equivalent wind speed and lapse rate improves predictions.

Drawbacks

- Ensure we have instrumentation installed at the new wind project site with the ability to gather atmospheric data and wind data at various heights for the length of the turbine.
- Develop the training model further with more points and turbine powers so that we can generalize the model to alternative locations.
- Drawbacks of predictor variable REWS
- There have been scenarios where the REWS method provides marginal to no improvement based on atmospheric conditions, turbine design, site location, etc.
- The usefulness of REWS depends on turbine dimension and wind shear regime, where if the ratio of turbine rotor diameter to hub-height is below 1.8 and the wind shear is constantly between −0.5 and 0.4, the REWS method may not be necessary
- This method does demonstrate the susceptibility of the turbine power curve by atmospheric conditions and the usefulness of measuring wind speed across the rotor layer as opposed to at a single instance, such as HHWS.
- Power prediction methods, such as the TPC and REWS, only utilize wind speed as a factor for prediction, without considering the surrounding atmospheric criteria. Studies have shown that variations in atmospheric conditions, such as temperature, atmospheric stability, wind shear, wind direction, and turbulence intensity can be factors in over or underestimation of turbine power output

2. Deep Learning-Based Prediction of Wind Power for Multi-turbines in a Wind Farm - Frontiers in Energy Research, 2021

Abstract

The prediction of wind power plays an indispensable role in maintaining the stability of the entire power grid. In this paper, a deep learning approach is proposed for the power prediction of multiple wind turbines. Starting from the time series of wind power, it is a two-stage modeling strategy, in which a deep neural network combines spatiotemporal correlation to simultaneously predict the power of multiple wind turbines. Specifically, the network is a joint model composed of Long Short-Term Memory Network (LSTM) and Convolutional Neural Network (CNN). Herein, the LSTM captures the temporal dependence of the historical power sequence, while the CNN extracts the spatial features among the data, thereby achieving the power prediction for multiple wind turbines. The proposed approach is validated by using the wind power data from an offshore wind farm in China, and the results in comparison with other approaches shows the high prediction preciseness achieved by the proposed approach.

Methods

Long Short-Term Memory Network (LSTM) - Convolutional Neural Network (CNN) joint model, Support Vector Machine (SVM), gradient-based training

Highlights

- The wind power of multiple wind turbines is predicted in this study, unlike predicting the total power of the wind farm or a single wind turbine in most studies.
- This study makes use of temporal correlation and spatial correlation (i.e., Spatiotemporal correlation) in a wind farm which could be helpful for multi-location wind power prediction using the joint LSTM-CNN model.
- Specifically, LSTM captures the temporal dependence between the wind power data of each single wind turbine, and CNN extracts the spatial correlation between the wind power data of multiple wind turbines.
- The corresponding output values of each turbine are put into a two-dimensional matrix according to the location of the wind turbines.

Drawbacks

- In the model, the length of the sliding window to obtain the input sequence (i.e., α) directly affects the effect of LSTM on the temporal correlation extraction of historical sequence data. So, a large number of experimental tests is needed to determine the value of α.
- The number of LSTM models is set as the same as the quantity of the selected wind turbines. This causes slowness in computation when scaling.
- CNN and SVM perform poorly. The reason is that, when facing the problem of power prediction of multiple wind turbines, the CNN can effectively capture the spatial features of the data, but it does not take the temporal correlation into account. Similarly, the LSTM has excellent performance when facing timing prediction problems, but ignores

the spatial features. Since the SVM only uses the global spatial and temporal information in the data, its prediction preciseness is noticeably lower than the three counterparts.

3. Forecasting of Wind Turbine Output Power Using Machine learning - 10th International Conference on Advanced Computer Information Technologies, Deggendorf, Germany, 2020

Abstract

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 predictions 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, the output power of the wind turbines is predicted using the random forest regressor algorithm.The SCADA data is collected for two years from a wind farm located in France. The wind direction, wind speed and outdoor temperature are used as input parameters to predict output power. The model is tested for two different capacity factors. The estimated mean absolute errors for the proposed model in this study were 3.6% and 7.3% for 0.4 and 0.2 capacity factors, respectively. The proposed model in this study offers an efficient method to predict the output power of a wind turbine with preferably low error.

Methods

Random Forest Regressor Algorithm

Highlights

- The elimination of variability of the wind is not possible, but by using machine learning, the outputs are typically a model that can address future issues of the same kind.
- This model has a low over-fitting tendency, simple and is fast to train.
- The model uses a design parameter called capacity factor which is used to evaluate the performance of a wind farm using real and rated values.
- The Mean Squared Error (MSE) and Mean Absolute Error (MAE) are calculated to evaluate the performance of the model.

Drawbacks

- The problem of missing data could have been solved using different imputation methods.
- The intermittency of wind speed introduces challenges in prediction, so this model did not pay too much attention to the variance of wind power supply due to it, when wind penetration went relatively low.

- The estimated mean absolute error for the proposed model for the capacity factor is nominal but could be minimized.

4. Wind power forecasting based on daily wind speed data using machine learning algorithms - Energy Conversion and Management, Volume 198, 2019

Abstract

Wind energy is a significant and eligible source that has the potential for producing energy in a continuous and sustainable manner among renewable energy sources. However, wind energy has several challenges, such as initial investment costs, the stationary property of wind plants, and the difficulty in finding wind-efficient energy areas. In this study, long-term wind power forecasting was performed based on daily wind speed data using five machine learning algorithms. We proposed a method based on machine learning algorithms to forecast wind power values efficiently. We conducted several case studies to reveal performances of machine learning algorithms. The results showed that machine learning algorithms could be used for forecasting long-term wind power values with respect to historical wind speed data. Furthermore, the results showed that machine learning-based models could be applied to a location different from model-trained locations. This study demonstrated that machine learning algorithms could be successfully used before the establishment of wind plants in an unknown geographical location whether it is logical by using the model of a base location.

Methods

Random Forest, support vector machines,Regression model, k means clustering and deep learning

Highlights

- In this paper Long-term wind power forecasting is being performed using different machine learning algorithms.
- When Wind power was being forecast for a different region, The results seem to show that using the training set and using the generated model with the training data could be used for other locations that have different wind characteristics.
- When wind power forecasting is based on only daily mean wind speed,it seems that machine learning algorithms could build reliable wind power models using only wind speed values even if there are no standard deviations to forecast long-term wind power values of a location.

Drawbacks

- However, using wind energy is challenging due to its initial investment costs, the requirement of careful analyses before establishing a wind plant, the distance of wind-efficient areas to the national grids, and its environmentally disruptive effects.

5. An Aggregative Machine Learning Approach for Output Power Prediction of Wind Turbines - IEEE TPEC, 2018

Abstract

Accurately forecasting power output of renewable sources is a necessity in operation of today's grid in order to achieve optimal energy utilization and carbon-free ecosystem. This study devises a stable, effective and accurate model for day ahead prediction of wind turbine power output through use of an aggregative approach. The method involves two types of Artificial Neural Network (Radial Basis and Conventional Feedforward Networks), Adaptive Neuro-Fuzzy Inference System (ANFIS) and Support Vector Machine (SVM) techniques. It is targeted at comparing the prediction models for their individual performances and finally coming upon an aggregative approach which outperforms the individual models through a strategic combination of them. Three techniques of combining (Simple Averaging, Regression and Outperformance) were tested. Though the individual models showed satisfactory performance by themselves, the combination techniques were able to outperform the individual models. Regression technique of combining was seen to be the most effective of all. The predicted output power through this technique was seen to greatly fit with the measured data with an NMSE of 1.03% for the test year. The combination techniques have also demonstrated more stable performance than the individual models while tested with the extreme cases of windy and less windy weeks.

Methods

Adaptive Neuro-Fuzzy Inference System (ANFIS), Radial Basis and Conventional Feedforward Networks (RBNN), Prediction, Back Propagation Neural Network(BPNN) and Support Vector Machine (SVM), Aggregating Ensemble Of Approaches

Highlights

- In this they have used soft computing approaches in predicting the power output from a wind turbine by using a strategic approach of combining the forecasts from individual models.
- The aggregative approach seems to exhibit better accuracy and also seems to show stable performances under extreme cases of windy and less windy periods.

- Though the individual models performed with a satisfactory forecast performance, the addition of the combining step seems to resolve the difficulty to identify a single best forecast model
- The aggregative approach seems to have alleviated issues of state-dependent performance of individual models and their systematic errors.

Drawbacks

- Lower performance of radial basis network when the 'power output of the previous hour' is added to the input matrix suggesting lower effectiveness of the approach in time series forecasting. Hence the paper has avoided the 'power output of the previous hour' for the radial basis NN
- SVM regression is seen to deteriorate in terms of prediction accuracy when more input parameters than wind speed are considered.Hence the paper only uses wind speed for the SVM model.

6. Machine learning ensembles for wind power prediction - Renewable Energy, Volume 89, 2015

Abstract

We first analyze homogeneous ensemble regressors that make use of a single base algorithm and compare decision trees to k-nearest neighbors and support vector regression. As the next step, we construct heterogeneous ensembles that make use of multiple base algorithms and benefit from a gain of diversity among the weak predictors. In the experimental evaluation, we show that a combination of decision trees and support vector regression outperforms state-of-the-art predictors (improvements of up to 37% compared to support vector regression) as well as homogeneous ensembles while requiring a shorter runtime (speed-ups from 1.60× to 8.78×). Furthermore, we show the heterogeneous ensemble prediction can be improved when using high-dimensional patterns by increasing the number of past steps considered and hereby the spatio-temporal information available by the measurements of the nearby turbines. The experiments are based on a large wind time series data set from simulations and real measurements.

Methods

Decision tree, Regression, K-nearest neighbors, Support vector machines, Ensemble machine learning

Highlights

- Use of heterogeneous machine learning ensembles for wind power prediction.

- A combination of decision trees and SVR outperforms state-of-the-art predictors.
- The prediction is improved by using high-dimensional spatio-temporal patterns.

Drawbacks

- Ensemble methods require less tuning and expert domain knowledge.
- Investing more computation time.
- There exist more sophisticated ensemble approaches like AdaBoost or Stacked Generalization, but, as we want to give a proof of concept with the possibility of heterogeneity, we limit ourselves here to bagging.
- Only if the data is diverse in nature the ensemble technique would be advantageous.
- We decided to implement a relatively simple bagging approach with weighting, which has some advantages. While the implementation is straight-forward and offers a moderate computational cost, we consider the approach sufficient for a proof of concept, which is also shown in the experimental evaluation.
- Because there are no computational dependencies between the ensemble members, the problem is embarrassingly parallel. To give an easy and fair comparison, in our experiments we only employ only one CPU core for the runtime measurements.
- With increasing number of features, the data become more and more challenging for the employed regression algorithms: First, the computational time is often dependent on the dimensionality of the data. Second, the prediction accuracy can get worse.

7. Using machine learning to predict wind turbine power output - IOPScience, 2013

Abstract

Wind turbine power output is known to be a strong function of wind speed, but is also affected by turbulence and shear. In this work, new aerostructural simulations of a generic 1.5 MW turbine are used to rank atmospheric influences on power output. Most significant is the hub height wind speed, followed by hub height turbulence intensity and then wind speed shear across the rotor disk. For a randomly selected atmospheric condition, the accuracy of the regression tree power predictions is three times higher than that from the traditional power curve methodology. The regression tree method can also be applied to turbine test data and used to predict turbine performance at a new site. No new data are required in comparison to the data that are usually collected for a wind resource assessment.

Methods

Regression Tree, Machine Learning

Highlights

Such an approach could significantly reduce bias in power predictions that arise because of the different turbulence and shear at the new site, compared to the test site.

Drawbacks

- Although this method has been demonstrated using simulated inflow and turbine response data, the method could be used to generate turbine performance models from turbine power testing data.
- Changes of wind direction with height, non-uniform shear, and the state of the turbine were not considered here but may impact turbine deployment sites, and their effect should be investigated using field data.

2.2 References

[1] Christiana Sasser, Meilin Yu, Ruben Delgado, “Improvement of wind power prediction from meteorological characterization with machine learning models”, Renewable Energy, Volume 183, 2022, Pages 491-501, ISSN 0960-148, https://doi.org/10.1016/j.renene.2021.10.034

[2] Chen Xiaojiao, Zhang Xiuqing, Dong Mi, Huang Liansheng, Guo Yan, He Shiying, “Deep Learning-Based Prediction of Wind Power for Multi-turbines in a Wind Farm”, Frontiers in Energy Research, Volume 9, 2021, ISSN 2296-598X, https://doi.org/10.3389/fenrg.2021.723775

[3] Rashid, Haroon & Haider, Waqar & Batunlu, Canras, “Forecasting of Wind Turbine Output Power Using Machine learning”, 09/2020, https://doi.org/10.1109/ACIT49673.2020.9208852

[4] Halil Demolli, Ahmet Sakir Dokuz, Alper Ecemis, Murat Gokcek, “Wind power forecasting based on daily wind speed data using machine learning algorithms”, Energy Conversion and Management, Volume 198, 2019, 111823, ISSN 0196-8904, https://doi.org/10.1016/j.enconman.2019.111823

[5] S. Netsanet, J. Zhang, D. Zheng, R. K. Agrawal and F. Muchahary, "An aggregative machine learning approach for output power prediction of wind turbines," 2018 IEEE Texas Power and Energy Conference (TPEC), 2018, pp. 1-6, https://doi.org/10.1109/TPEC.2018.8312085

[6] Justin Heinermann, Oliver Kramer, “Machine learning ensembles for wind power prediction”, Renewable Energy, Volume 89, 2016, Pages 671-679, ISSN 0960-1481, https://doi.org/10.1016/j.renene.2015.11.073

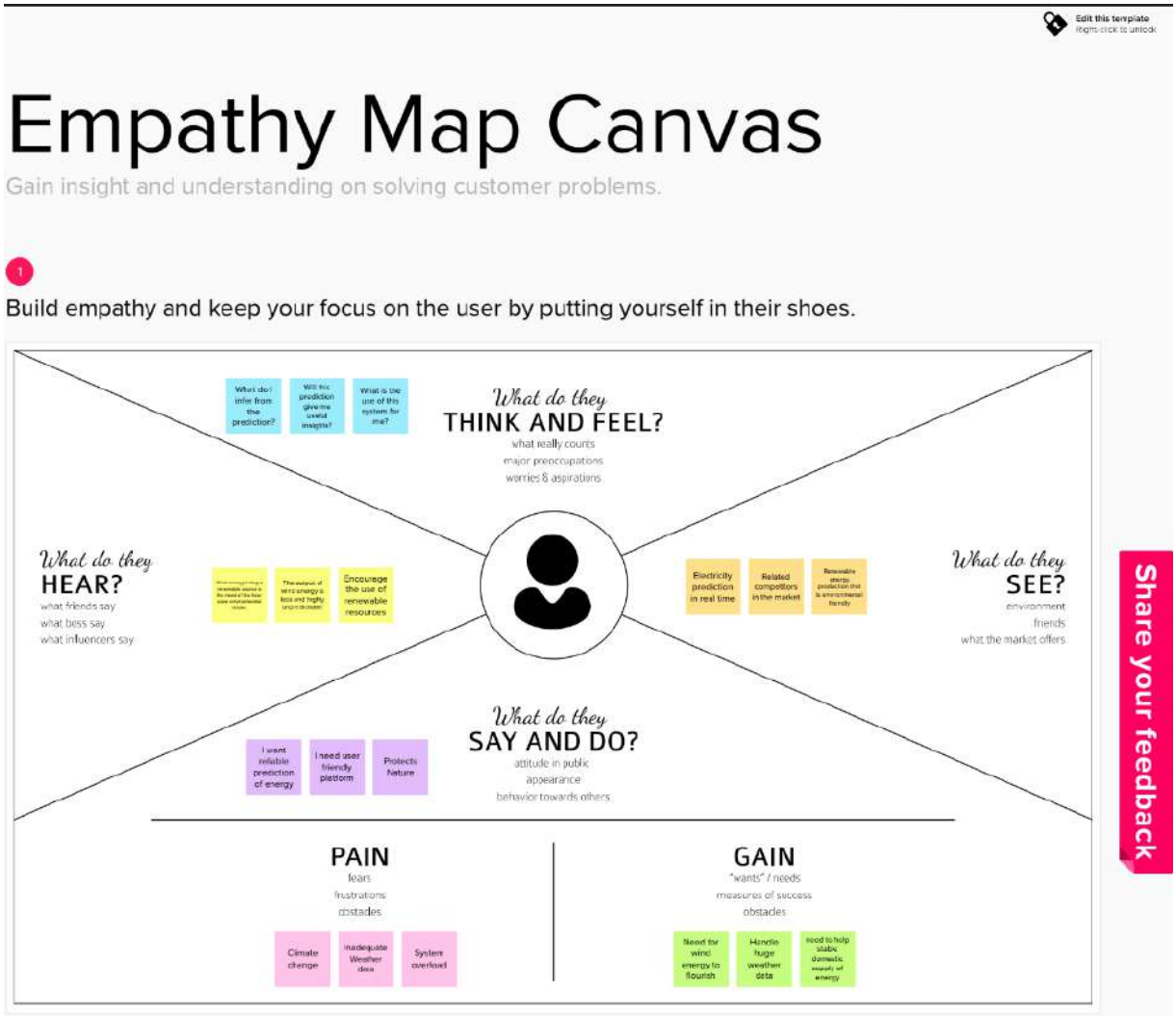
[7] Clifton A, Kilcher L, Lundquist J K, Fleming P, “Using machine learning to predict wind turbine power output”, Environ. Res. Lett. 8 024009, 2013, https://dx.doi.org/10.1088/1748-9326/8/2/024009

2.3 Problem Statement Definition

- The user needs a way to analyze weather conditions over a region so that the user can predict the windmill turbine's output.
- The user needs to find a solution that can predict the energy output of wind turbines so that the user can ensure a steady supply of electricity from the wind energy.
- The user needs to find a way to produce wind energy efficiently so that more households can be ensured of less power cuts by utilizing this energy.
- The organization needs to find a way to accurately predict the wind energy produced so that overproduction and cost of production of wind energy can be reduced.
- The manufacturer needs to find a way to do something that analyzes the weather conditions of a region so that the manufacturer can chose regions that produce high quality and quantities of wind energy.
- The manufacturer needs to find a way to collaborate with other power suppliers so that the wind energy produced in the market can be scaled up for more utilization.

3. IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation and Brainstorming

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

Need some inspiration?

See a finished version of this template to kickstart your work.

Open example →

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

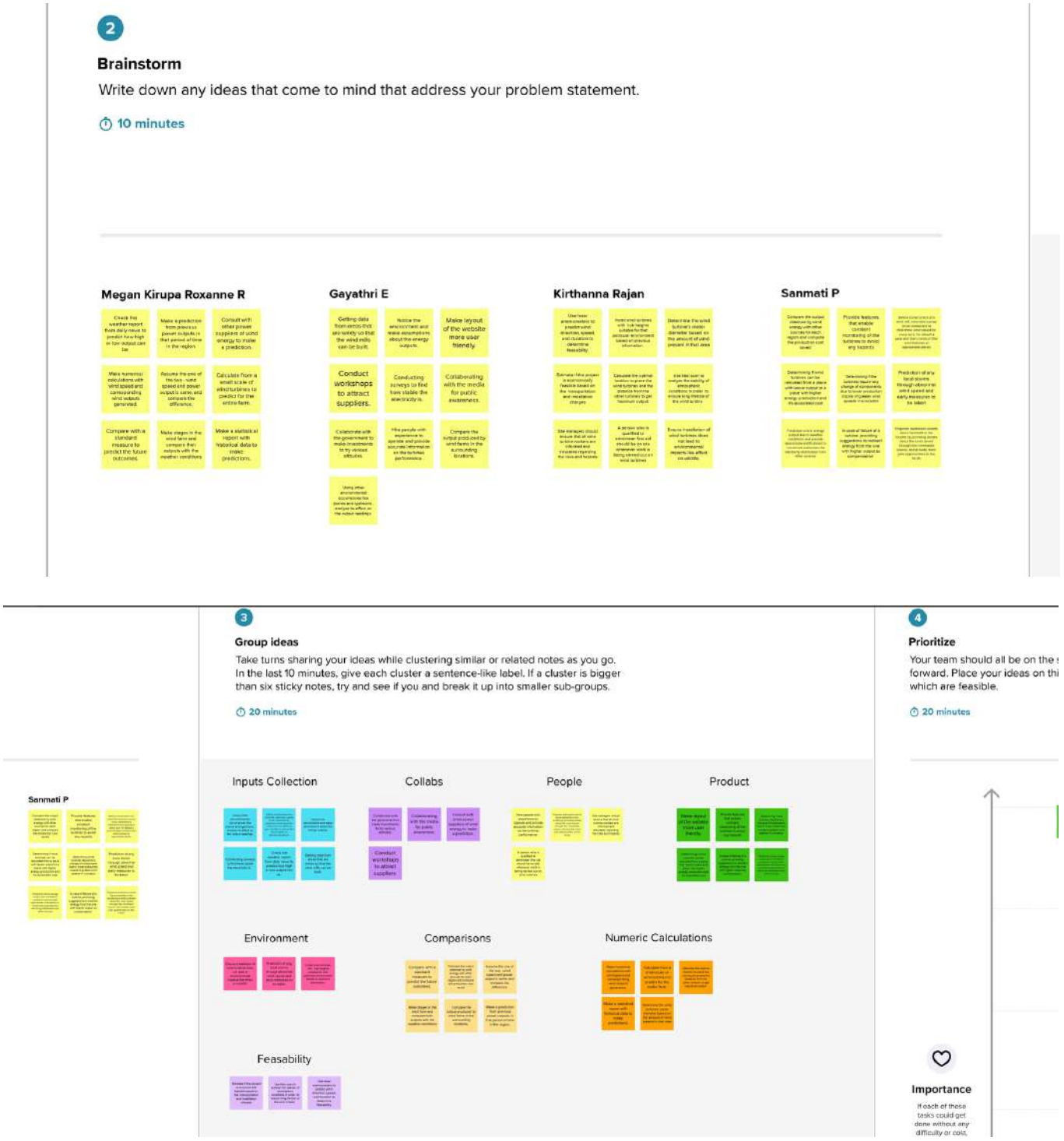
PROBLEM

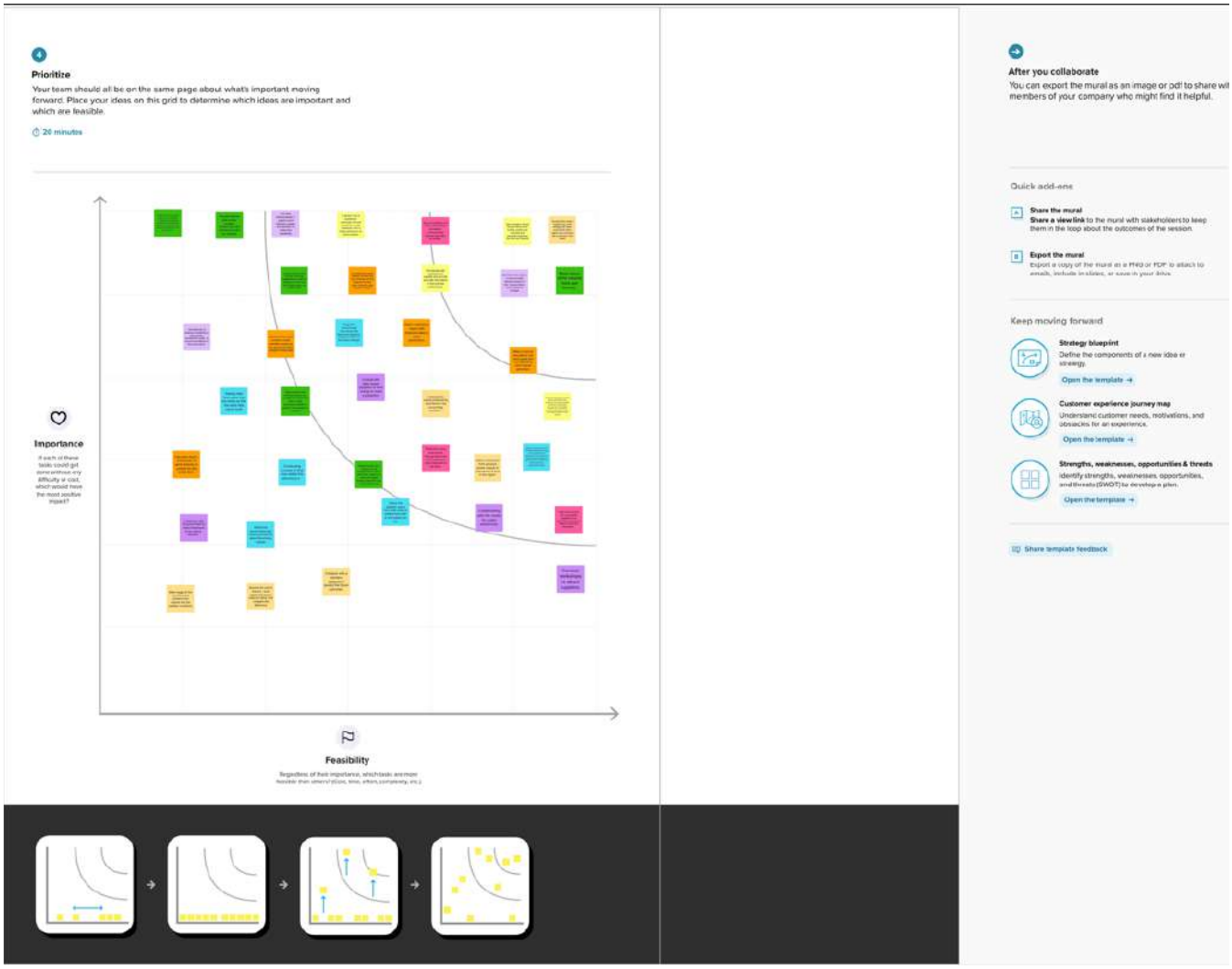
How might we analyze weather conditions over a region to predict the output a wind turbine?

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.


2
Brainstorm
Write down any ideas that come to mind that address your problem statement.
10 minutes
Megan Kirupa Roxanne R
Gayathri E
Conduct workshops to attract suppliers.
Kirthanna Rajan
Sanmati P
3
Group ideas
Take turns sharing your ideas while clustering similar or related notes as you go.
In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.
20 minutes
Sanmati P
Inputs Collection
Collabs
People
Product
Environment
Comparisons
Numeric Calculations
Feasability
4
Prioritize
Your team should all be on the
forward. Place your ideas on th
which are feasible.
20 minutes
Importance
If each of these
tasks could get
done without any
difficulty or cost,



3.3 Proposed Solution

Problem Statement

Wind power consists of converting the energy produced by the movement of wind turbine blades driven by the wind into electrical energy. Wind power generation differs due to the stochastic nature of wind. The prediction of wind power plays an indispensable role in maintaining the stability of the entire power grid. This solution aims to forecast the wind power values efficiently by correlating the parameters of weather conditions and wind turbines.

Idea / Solution Description

Wind energy is a significant and eligible source that has the potential for producing energy in a continuous and sustainable manner among renewable energy sources. However, wind energy has several challenges, such as initial investment costs, the stationary property of wind plants, and the difficulty in finding wind-efficient energy areas. Hence, long-term wind power

forecasting is to be performed based on daily wind speed data using machine learning algorithms. With the process of applying machine learning models along with statistical models to historical wind speed data of a region, we can obtain long-term wind power values. This architecture integrated with a weather forecasting API, furthermore assists in the prediction in any location. The model is trained using IBM Watson's machine learning service and its scoring endpoint is fed to the application developed using the Flask framework to process the API's and energy prediction requests from the user to render the results on the UI.

Novelty / Uniqueness

This solution is aimed to be architected in a way that makes it scalable and flexible in any kind of situations like unusual weather conditions, sudden influx of customers or users, analyzing in any type of location etc. This helps in dealing with the challenges of balancing supply and demand in any electricity system ,reducing the need for additional balancing energy and reserve power to integrate wind power by accurate wind power forecasting. Through statistical analysis combined with machine learning service over cloud as SAAS, the mentioned uniquenesses are desired to be achieved.

Social Impact / Customer Satisfaction

By creating an effective machine learning model we will be able to increase the power produced hence the num and a clean technology, and this is one of the main impacts that makes it such an attractive and promising energy supply solution to predict the energy output of wind turbines based on weather conditions.Main social and environmental benefits includes reduction in CO2 emissions and fossil fuels imports. In areas where wind energy is employed, job opportunities are provided to many families and hence increases the employment percentage providing regional development. The customer will be able to get an overall picture of the output hence will get a clear picture on whether to invest more in that particular region or to rectify by investigation in another region. The customer will also be able to analyze and make changes based on the output to maximize the output making him/her a happy customer.

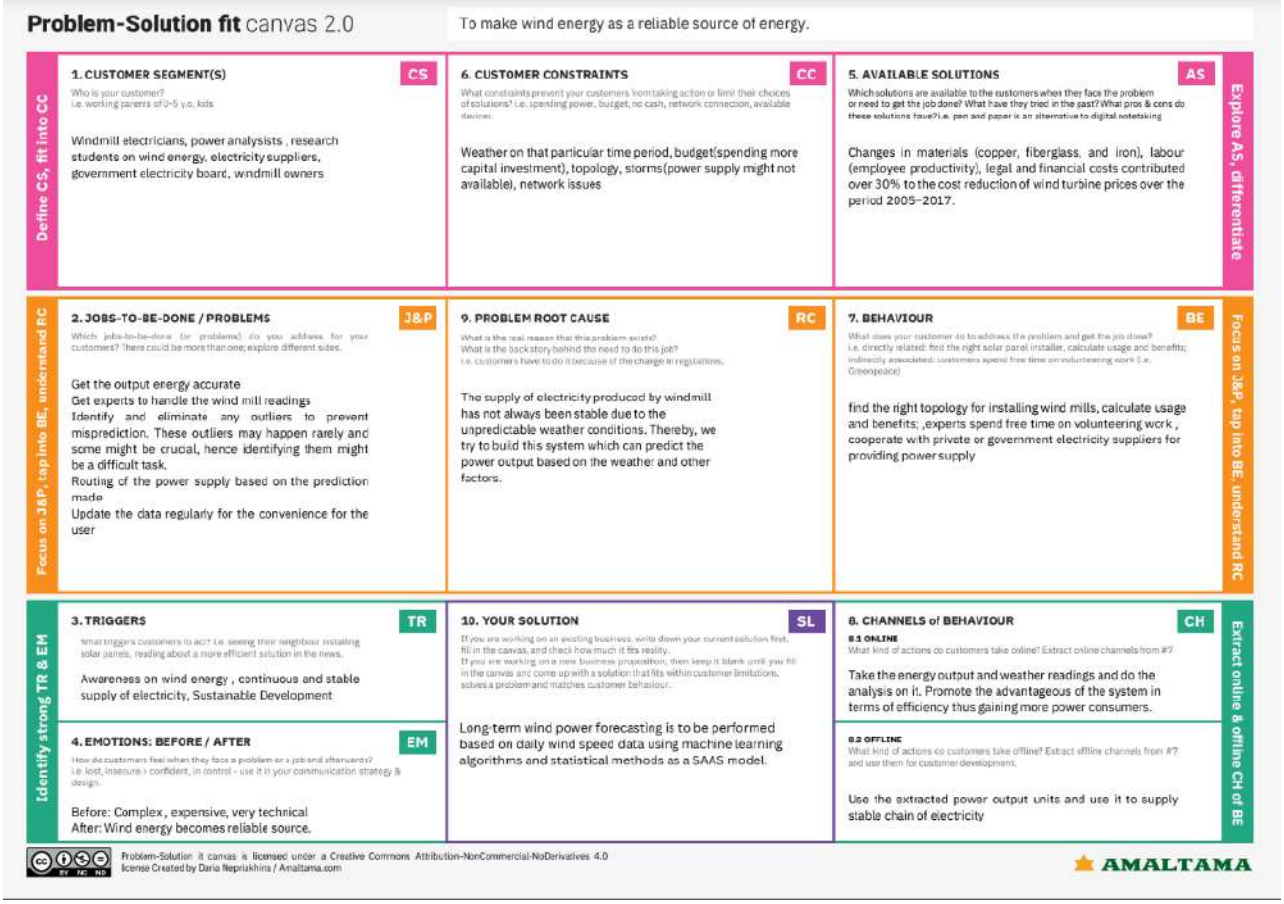
Business Model (Financial Benefit)

Opportunities to trade in the power produced are likely to expand significantly.Currently, it is possible for wind power producers to sell electricity to the grid, use it for captive consumption or sell it to third parties. With the emergence of independent power exchanges and with the likely liberalization and streamlining of power distribution across states, the opportunities to trade in power are likely to increase and become more lucrative.With the advent of the RPO/REC mechanism in India, there has been significant demand for non-solar (wind, small hydro, biomass etc.) over the past few months.

Scalability of Solution

Energy trading in liberalized markets is particularly interesting from the perspective of wind energy producers because of the non-dispatchable nature of wind. This means that wind energy producers need to forecast how much they will produce in the future in order to place their bids. Hence customers can use our SAAS model to accomplish the task.

3.4 Problem Solution Fit



4. REQUIREMENT ANALYSIS

4.1 Functional Requirements

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP

FR-3	User Login Into Website	Login using user entered credentials during registration Change/Forgot Password by updating user credentials
FR-4	User Profile Edit Option	Edit through Form
FR-5	Navigate Through Website Menu	Explore through Menu option in taskbar Explore through Search function in the home page
FR-6	Enter Wind And Other Required Parameters	Entry through Form
FR-7	Evaluate Wind Parameters	Model evaluating parameters submitted through form
FR-8	View Wind Power Prediction	Display prediction results in website window
FR-9	Download Prediction Results	Download as image Download as PDF Download and share to Email
FR-10	Make New Evaluations	Navigate through User profile Navigate through previous pages of website window
FR-11	User Log Out Of Website	Logout through option provided.
FR-12	Admin Role Login to the website	Login through the admin-only restricted Login page using login credentials.
FR-13	Admin adds content, new subpages and publishes content to the assigned pages	Add content to the super admin-assigned pages Make changes to the content Publish content in the super admin-assigned pages
FR-14	Super Admin adds content and access all items on the website	Add content to all pages Publish content on all pages Add new pages without restriction Edit site design and layout Edit site permission settings Monitor site traffic and make decisions
FR-15	Admin Role Logout from the website	Logout through option provided.

4.2 Non-Functional Requirements

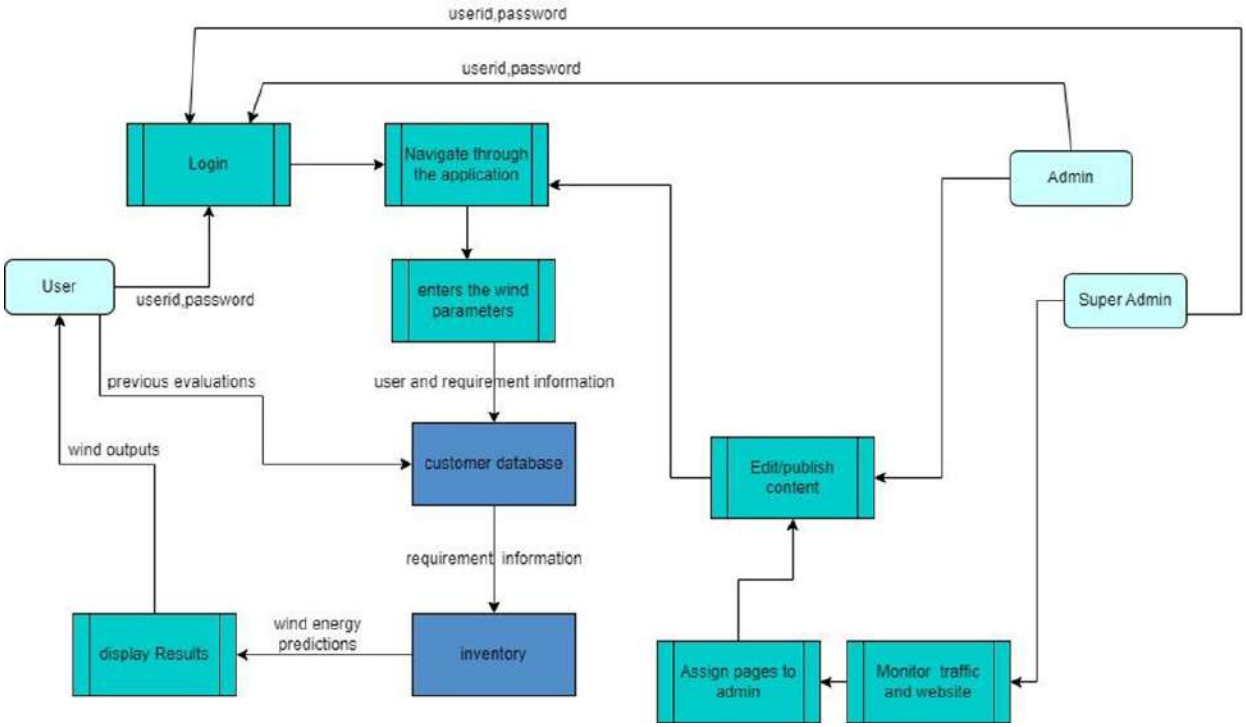
Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The website is responsive in various devices like computers and mobile devices.
NFR-2	**Security**	The user registration and login credentials are of single use - show error messages if credentials don't match an existing user profile. Limiting number of Login attempts. User credentials are stored with a unique ID - salting and hashing. Stored data can be protected using cloud security services.
NFR-3	**Reliability**	Well designed user interfaces to steer users through the website correctly. Historical data used for prediction is updated frequently. Building credibility by mentioning contact means of the website and through social proof. Install SSL Certificate. Website is fault-tolerant. Any critical failures are solved within 24 hours or less.
NFR-4	**Performance**	Website gives quick evaluation responses to user queries.
NFR-5	**Availability**	Website is available to users for use at all times of the day. Any maintenance work is priorly mentioned and done in less time.
NFR-6	**Scalability**	Website is equipped to handle requests from multiple users through cloud services. Volume of historic data from prediction is scaled.

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution and Technical Architecture

Solution Architecture:

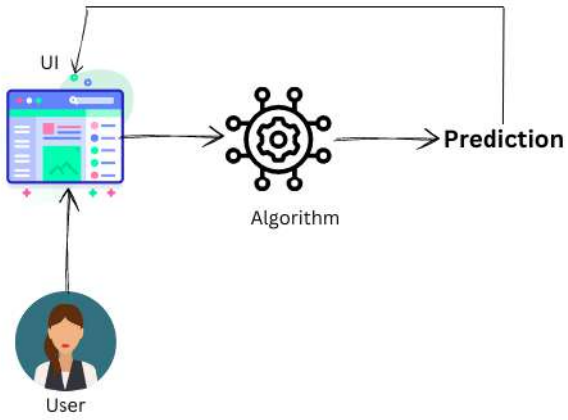
Problem Statement

Wind power consists of converting the energy produced by the movement of wind turbine blades driven by the wind into electrical energy. Wind power generation differs due to the stochastic nature of wind. The prediction of wind power plays an indispensable role in maintaining the stability of the entire power grid. This solution aims to forecast the wind power values efficiently by correlating the parameters of weather conditions and wind turbines.

Proposed Solution:

Long-term wind power forecasting is to be performed based on daily wind speed data using machine learning algorithms. A Minimal Viable Product is aimed to be built by integrating a machine learning algorithm with a front end UI to fetch the user inputs which will be evaluated and the wind power results are fed back to the UI. This architecture is further enhanced as the customer base expands by integrating with a weather forecasting API which assists in the prediction from any geographical location and by training the model using IBM Watson's machine learning service with its scoring endpoint fed to a Flask framework-built UI to process the API's and energy prediction requests from the user and rendering the results back to the UI.

Block Diagram Of MVA:



Block diagram

Conclusion:

The Minimal Viable Product is developed with basic features that provide the critical need of predicting wind power output based on weather conditions built with a simple UI powered by a regressor algorithm and statistical methods that can process the user's requests to predict the wind power values. The MVA is suitable for incremental development to augment the additional features or changes in the requirements to build a flexible and scalable version of the application architecture.

Technology Architecture:

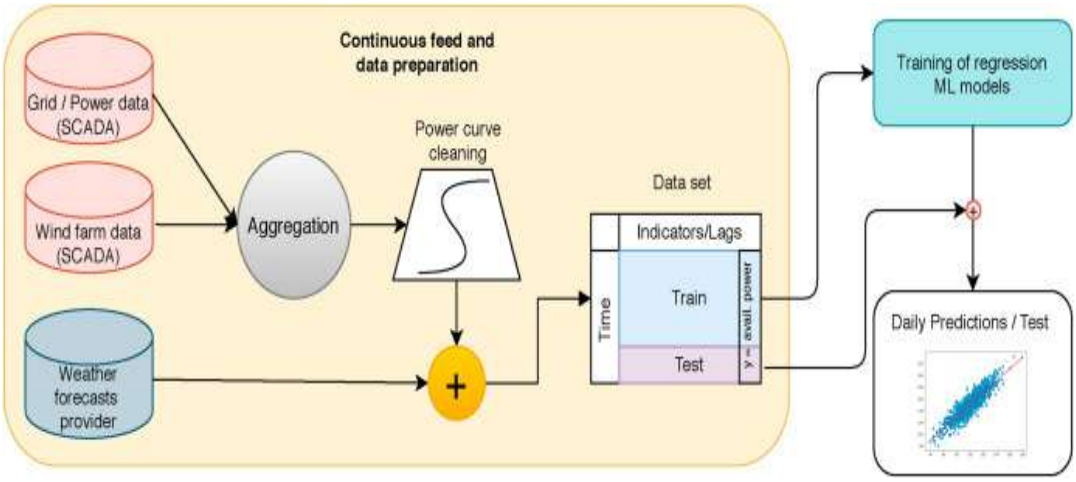


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	User can input their windmill parameters and request for the analysis	HTML, CSS, JavaScript
2.	Application Logic-1	Constructing the Model based on parameters input	Python
3.	Application Logic-2	Model Validation and Implementation	IBM Watson
4.	Machine Learning Model	Various ML techniques are used on samples of test data to be finally constructed into a single model	Ensemble Method

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Flask	Micro Web FrameWork
2.	Security Implementations	User logins are secured by credentials and limited login attempts. User data secured by cloud security services	Credentials are saved by salting and hashing. IBM watson security services
3.	Scalable Architecture	Handles multiple user requests	IBM Watson
4.	Availability	Ensured to be available all the time and maintenance work is priorly informed	Use of Multiple servers
5.	Performance	Quick response is ensured	IBM Watson

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & confirm	High	1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
		USN-3	As a user, I can register details in a cloud database	I can receive an OTP on my email and then register	Medium	4
		USN-4	As a user, I can update password on cloud database	I can register & access the dashboard with LinkedIN	Medium	4
	Login	USN-5	As a user, I can log into the application by entering email & password	I can access the website Home page	High	1
		USN-6	As a user, I can change my password in case I forget it through the reset password option.	I can receive a reset password email and login using those credentials	Medium	3
	Dashboard	USN-7	As a user, I can access the dashboard to View profile	I can view my profile	Low	2
		USN-8	As a user, I can access the dashboard to make the wind power prediction action	I can view the requirements form	High	2
		USN-9	As a user, I can new Evaluations with new data	I can see my history of searches	High	2
	Profile	USN-10	As a user, I can edit my profile time to time	I can update the profile & click submit	Medium	4
	Requirements	USN-11	As a user , I can enter the wind speed and other parameters through a form	I can enter the requirements	High	3
	Results	USN-12	As a user, I can view the results for my entered parameters	I can view the energy output	High	3
	Downloads	USN-13	As a user I can download the results as an image	I can download the image as png/jpg /jpeg	Medium	2

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
		USN-14	As a user, I can download the results as a pdf	I can download it as a document	High	2
		USN-15	As a user, I can download and share the results through email	I can enter the email addresses & click share	Low	2
Administrator	Login	USN-16	As an admin, I can log into the application by entering email & password	I can access the website	High	1
		USN-17	As an admin, I can change my password in case I forget it through the reset password option.	I can receive a reset password email and login using those credentials	Medium	3
	Website modification	USN-18	As an admin , I can add content and publish the pages on the application assign by the super admin	I can add, make changes to the content		4
Super Admin	Login	USN-19	As a super admin, I can log into the application by entering email & password	I can access the website Home	High	1
		USN-20	As a super admin, I can log into the application by receiving a reset email incase forgot password	I can receive a reset password email and login using those credentials	Medium	3
	Web Page assignment	USN-21	As a super admin , I can assign pages to be handles to the admin	I can assign pages to admin		3
	Website modification	USN-22	As a super admin , I can add content and publish on all the pages of the application	I can add, make changes to the content	Low	4

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
		USN-23	As a super admin, I can edit the design and layout of the website	I can access the design and program of the website	Mediu m	4

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password	2	High	Gayathri, Megan
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	2	High	Gayathri, Megan
Sprint-4		USN-3	As a user, I can register details in a cloud database	1	Medium	Megan, Sanmati
Sprint-4		USN-4	As a user, I can update password on cloud database	1	Medium	Gayathri, Kirthanna Rajan
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	2	High	Kirthanna Rajan, Sanmati

Sprint-3		USN-6	As a user, I can change my password in case I forget it through the reset password option.	1	Medium	Megan, Kirthanna Rajan
Sprint-2	Dashboard	USN-7	As a user, I can access the dashboard to View profile	3	Low	Gayathri, Kirthanna Rajan
Sprint-2		USN-8	As a user, I can access the dashboard to make the wind power prediction action	3	High	Megan, Sanmati
Sprint-2		USN-9	As a user, I can new Evaluations with new data	2	High	Gayathri, Megan
Sprint-4	Profile	USN-10	As a user, I can edit my profile time to time	1	Medium	Gayathri, Kirthanna Rajan
Sprint-3	Requirements	USN-11	As a user, I can enter the wind speed and other inputs through a form	1	High	Gayathri, Kirthanna Rajan
Sprint-3	Results	USN-12	As a user, I can view the results for my entered inputs	1	High	Megan, Sanmati
Sprint-2	Downloads	USN-13	As a user I can download the results as an image	3	Medium	Megan, Sanmati
Sprint-2		USN-14	As a user, I can download the results as a pdf	3	High	Gayathri, Kirthanna Rajan

Sprint-2		USN-15	As a user, I can download and share the results through email	3	Low	Kirthanna Rajan, Sanmati
Sprint-1	Login	USN-16	As an admin, I can log into the application by entering email & password	3	High	Kirthanna Rajan, Sanmati
Sprint-3		USN-17	As an admin, I can change my password in case I forget it through the reset password option	3	Medium	Gayathri, Sanmati
Sprint-4	Website modification	USN-18	As an admin, I can add content and publish the pages on the application assign by the super admin	3	Medium	Megan, Sanmati
Sprint-1	Login	USN-19	As a super admin, I can log into the application by entering email & password	3	High	Kirthanna Rajan, Sanmati
Sprint-3		USN-20	As a super admin, I can log into the application by receiving a reset email incase forgot password	3	Medium	Kirthanna Rajan, Megan
Sprint-3	Web Page Assignment	USN-21	As a super admin, I can assign pages to be handled to the admin	2	High	Gayathri, Sanmati

Sprint-4	Website Modification	USN-22	As a super admin, I can add content and publish on all the pages of the application	4	Low	Gayathri, Sanmati
Sprint-4		USN-23	As a super admin, I can edit the design and layout of the website	4	Medium	Megan, Kirthanna Rajan

6.2 Sprint Delivery Schedule

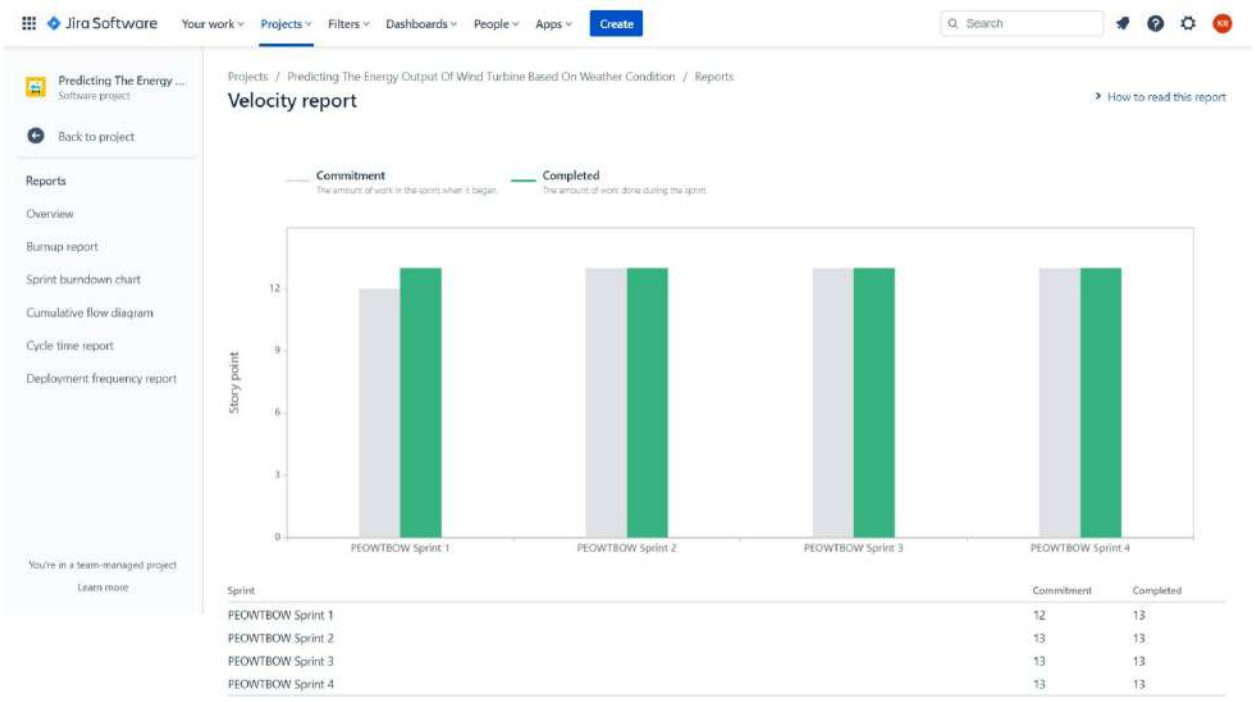
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	12	6 Days	24 Oct 2022	29 Oct 2022	13	29 Oct 2022
Sprint-2	17	6 Days	31 Oct 2022	05 Nov 2022	13	05 Nov 2022
Sprint-3	11	6 Days	07 Nov 2022	12 Nov 2022	13	12 Nov 2022
Sprint-4	14	6 Days	14 Nov 2022	19 Nov 2022	13	19 Nov 2022

6.3 Reports from JIRA

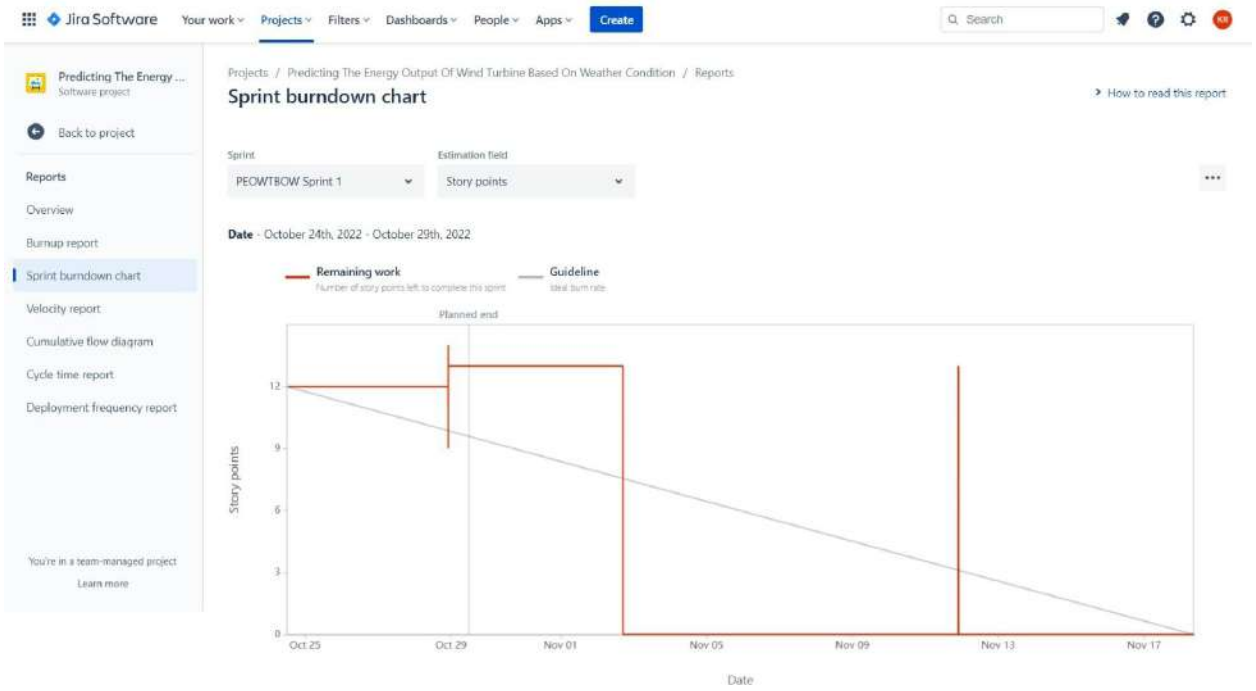
Project Tracker

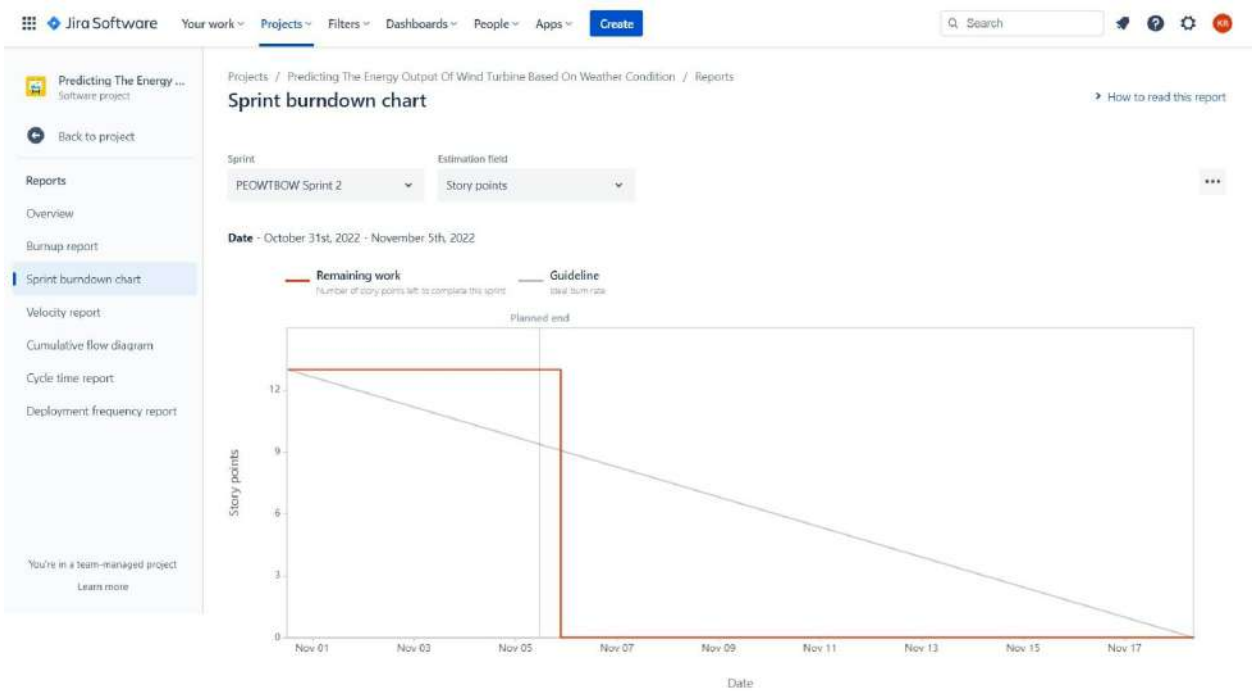
https://pnt2022tmid53075.atlassian.net/

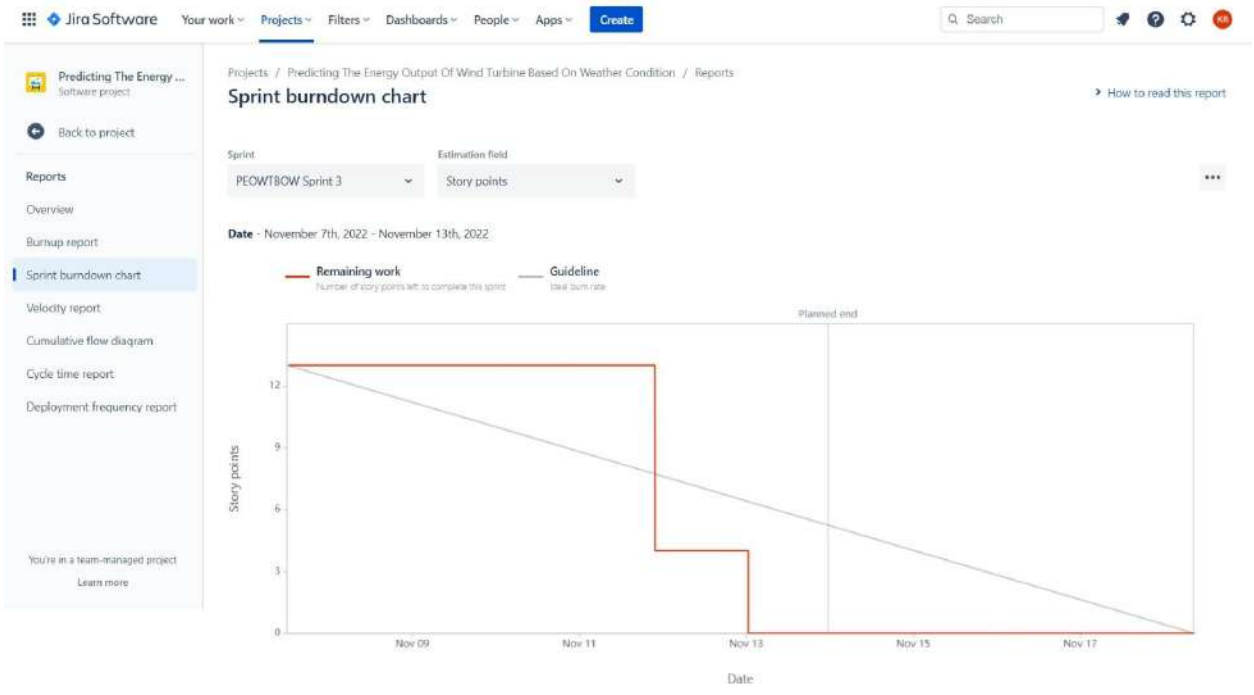
Velocity Chart

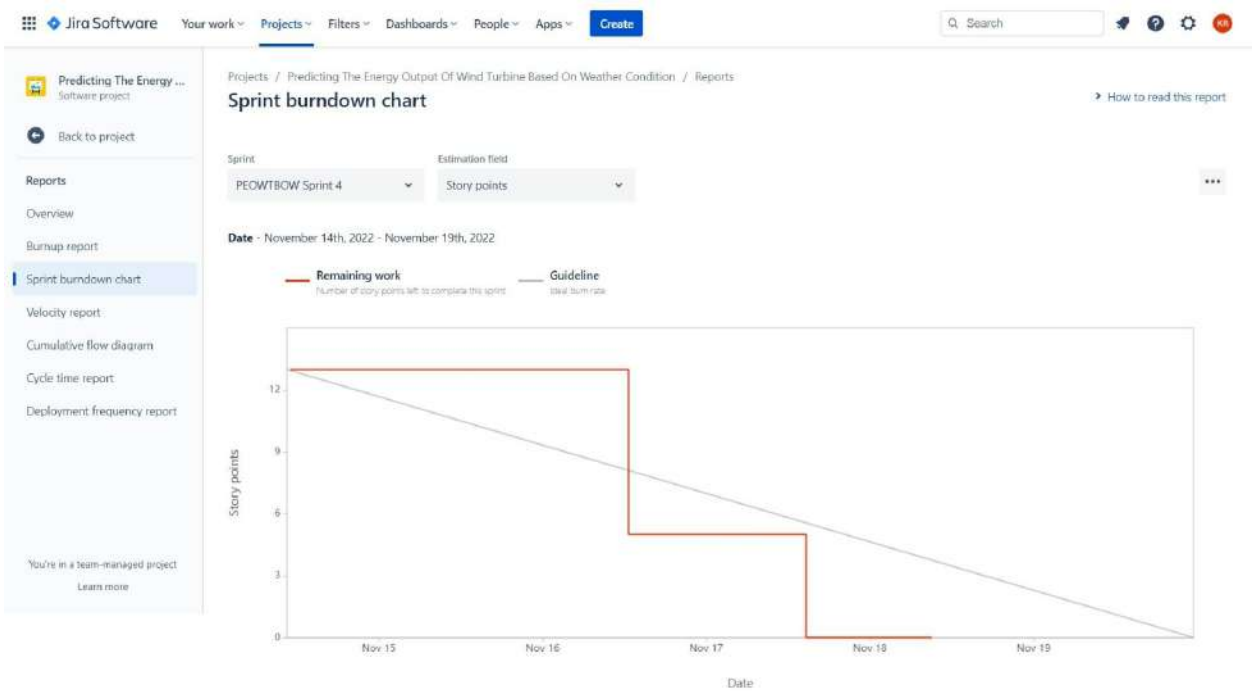


Burndown Chart




Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Predicting The Energy ...
Software project
Back to project
Reports
Overview
Burnup report
Sprint burndown chart
Velocity report
Cumulative flow diagram
Cycle time report
Deployment frequency report
You're in a team-managed project
Learn more
Projects / Predicting The Energy Output Of Wind Turbine Based On Weather Condition / Reports
Sprint burndown chart
How to read this report
Sprint
PEOWTBOW Sprint 2
Estimation field
Story points
Date - October 31st, 2022 - November 5th, 2022
Remaining work
Guideline
Planned end
Story points
0
3
6
9
12
Nov 01
Nov 03
Nov 05
Nov 07
Nov 09
Nov 11
Nov 13
Nov 15
Nov 17
Date


Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Predicting The Energy ...
Software project
Back to project
Reports
Overview
Burnup report
Sprint burndown chart
Velocity report
Cumulative flow diagram
Cycle time report
Deployment frequency report
You're in a team-managed project
Learn more
Projects / Predicting The Energy Output Of Wind Turbine Based On Weather Condition / Reports
Sprint burndown chart
How to read this report
Sprint
PEOWTBOW Sprint 3
Estimation field
Story points
Date - November 7th, 2022 - November 13th, 2022
Remaining work
Guideline
Planned end
Story points
0
3
6
9
12
Nov 09
Nov 11
Nov 13
Nov 15
Nov 17
Date



7. CODING & SOLUTIONING

7.1 Feature 1 - Registration

For the registration features we have created a database on cloud in which we will insert and extract the details. This will help us keep track of the users who are using this project. The passwords are converted to a hash value in the cloud database.

Register:

ibm_app.py

```
from flask import Flask, render_template, url_for, redirect, flash,
request
from flask_login import login_user, LoginManager, login_required,
logout_user
from wtforms.validators import InputRequired, Length, ValidationError
from flask_bcrypt import Bcrypt
from wtforms import StringField, PasswordField, SubmitField
from flask_wtf import FlaskForm

bcrypt = Bcrypt(app)
app.config['SECRET_KEY'] = 'B7-1A3E'
class RegisterForm(FlaskForm):
    email = StringField(validators=[InputRequired(), Length(min=4,
max=50)], render_kw={"placeholder":"Email"})
```

```
    username = StringField(validators=[InputRequired(), Length(min=4,
max=20)], render_kw={"placeholder": "Username"})
    rollnumber = StringField(validators=[InputRequired(), Length(min=5,
max=10)], render_kw={"placeholder":"RollNumber"})
    password = PasswordField(validators=[InputRequired(), Length(min=8,
max=20)],render_kw={"placeholder": "Password"})
    submit = SubmitField('Register')

    def validate_username(self, username):
        stmt = ibm_db.prepare(conn, 'SELECT * FROM user WHERE username=?')
        ibm_db.bind_param(stmt, 1, username.data)
        ibm_db.execute(stmt)
        existing_user_username = ibm_db.fetch_tuple(stmt)
        if existing_user_username:
            raise ValidationError('That username already exists. Try
another one.')
@ app.route('/register', methods=['GET', 'POST'])
def register():
    form = RegisterForm()
    if form.validate_on_submit():
        hashed_password =
bcrypt.generate_password_hash(form.password.data)
        stmt = ibm_db.prepare(conn, 'INSERT INTO user (email, username,
roll_number, pass_word) VALUES (?, ?, ?, ?)')
        ibm_db.bind_param(stmt, 1, form.email.data)
        ibm_db.bind_param(stmt, 2, form.username.data)
        ibm_db.bind_param(stmt, 3, form.rollnumber.data)
        ibm_db.bind_param(stmt, 4, hashed_password)
        #hash causes size to exceed VARCHAR size in DB2, hence made
VARCHAR(8000)
        ibm_db.execute(stmt)
        return redirect(url_for('login'))
    return render_template('register.html', form=form)
```

register.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.
css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoR
xT2MZw1T" crossorigin="anonymous">
    <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='style.css') }}" />
```

```
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Register</title>
</head>

<body>
    <h1>IBM-Predicting the energy output of wind turbine based on weather
condition</h1> <br>

    <h7>Register Page</h7> <br>


    <form method="POST" action="">
        {{ form.hidden_tag() }}
        <fieldset class="form-group">
            <!-- <legend class="border-bottom mb-4">Registration
Page</legend> -->

            <div>
                {{ form.email.label(class="form-control-label") }}

                {% if form.email.errors %}
                    {{ form.email(class="form-control form-control-lg
is-invalid") }}
                    <div class="invalid-feedback">
                        {% for error in form.email.errors %}
                            <span>{{ error }}</span>
                        {% endfor %}
                    </div>
                {% else %}
                    {{ form.email(class="form-control form-control-lg") }}
                {% endif %}
            </div>


            <div>
                {{ form.username.label(class="form-control-label") }}

                {% if form.username.errors %}
                    {{ form.username(class="form-control form-control-lg
is-invalid") }}
                    <div class="invalid-feedback">
                        {% for error in form.username.errors %}
                            <span>{{ error }}</span>
                        {% endfor %}
                    </div>
```

```
                {% else %}
                    {{ form.username(class="form-control form-control-lg")
}}
                {% endif %}
            </div>

            <div>
                {{ form.rollnumber.label(class="form-control-label") }}

                {% if form.rollnumber.errors %}
                    {{ form.rollnumber(class="form-control form-control-lg
is-invalid") }}
                    <div class="invalid-feedback">
                        {% for error in form.rollnumber.errors %}
                            <span>{{ error }}</span>
                        {% endfor %}
                    </div>
                {% else %}
                    {{ form.rollnumber(class="form-control
form-control-lg") }}
                {% endif %}
            </div>


            <div>
                {{ form.password.label(class="form-control-label") }}

                {% if form.password.errors %}
                    {{ form.password(class="form-control form-control-lg
is-invalid") }}
                    <div class="invalid-feedback">
                        {% for error in form.password.errors %}
                            <span>{{ error }}</span>
                        {% endfor %}
                    </div>
                {% else %}
                    {{ form.password(class="form-control form-control-lg")
}}
                {% endif %}
            </div>

        </fieldset>
        <div class="form-group">
            {{ form.submit(class="btn btn-outline-info") }}
        </div>

        <small class="text-muted ml-2"><br><br>
```

```
            <a href="{{ url_for('login') }}">Already have an account? Log
In</a>
        </small>

    </form>

    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE
1Pi6jizo" crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js"
integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86d
IHNDz0W1" crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js
"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xI
M+B07jRM" crossorigin="anonymous"></script>
</body>
<br><br><br><br><br>
<footer>
    @Team ID: PNT2022TMID53075
</footer>
</html>
```

7.2 Feature 2 - Login

For the Login feature, we have created a database on cloud in which we inserted the details of registration. This would be extracted from the cloud database and validated for further navigation into the project.

Login:

ibm_app.py

```
from flask import Flask, render_template, url_for, redirect, flash,
request
from flask_login import login_user, LoginManager, login_required,
logout_user
from wtforms.validators import InputRequired, Length, ValidationError
from flask_bcrypt import Bcrypt
from wtforms import StringField, PasswordField, SubmitField
from flask_wtf import FlaskForm

bcrypt = Bcrypt(app)
app.config['SECRET_KEY'] = 'B7-1A3E'
```

```
class LoginForm(FlaskForm):
    username = StringField(validators=[InputRequired(), Length(min=4,
max=20)], render_kw={"placeholder": "Username"})
    password = PasswordField(validators=[InputRequired(), Length(min=8,
max=20)], render_kw={"placeholder": "Password"})
    submit = SubmitField('Login')


@app.route('/login', methods=['GET', 'POST'])
def login():
    form = LoginForm()
    if form.validate_on_submit():
        stmt = ibm_db.prepare(conn, 'SELECT * FROM user WHERE username=?')
        ibm_db.bind_param(stmt, 1, form.username.data)
        ibm_db.execute(stmt)
        user = ibm_db.fetch_tuple(stmt)
        if user:
            if bcrypt.check_password_hash(user[4], form.password.data):
                usr_obj = User(user[0], user[1], user[2])
                login_user(usr_obj)
                return redirect(url_for('welcome'))
            else:
                print('Hi')
                flash(f'Invalid credentials, check and try logging in
again.', 'danger')
                return redirect(url_for('login'))
    return render_template('login.html', form=form)
```

login.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.
css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoR
xT2MZw1T" crossorigin="anonymous">
    <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='style.css') }}" />
</head>
```

```
<body>
    <div class="col-md-8">
        {% with messages = get_flashed_messages(with_categories=true) %}
            {% if messages %}
                {% for category, message in messages %}
                    <div class="alert alert-{{category}}">
                        {{ message }}
                    </div>
                {% endfor %}
            {% endif %}
        {% endwith %}
        {% block content %} {% endblock %}
    </div>
    <h1>IBM-Predicting the energy output of wind turbine based on weather
condition</h1> <br>

    <h7>Login Page</h7> <br>

    <form method="POST" action="">
        {{ form.hidden_tag() }}
        <fieldset class="form-group">
            <div>
                {{ form.username.label(class="form-control-label") }}

                {% if form.username.errors %}
                    {{ form.username(class="form-control form-control-lg
is-invalid") }}
                    <div class="invalid-feedback">
                        {% for error in form.username.errors %}
                            <span>{{ error }}</span>
                        {% endfor %}
                    </div>
                {% else %}
                    {{ form.username(class="form-control form-control-lg")
}}
                {% endif %}
            </div>


            <div>
                {{ form.password.label(class="form-control-label") }}

                {% if form.password.errors %}
                    {{ form.password(class="form-control form-control-lg
is-invalid") }}
                    <div class="invalid-feedback">
```

```
                        {% for error in form.password.errors %}
                            <span>{{ error }}</span>
                        {% endfor %}
                    </div>
                {% else %}
                    {{ form.password(class="form-control form-control-lg")
}}
                {% endif %}
            </div>

        </fieldset>
        <div class="form-group">
            {{ form.submit(class="btn btn-outline-info") }}
        </div>

        <small class="text-muted ml-2"><br>
            <a href='{{url_for('register')}}'>Do not have an account? Sign
Up?</a>
        </small>
    </form>
    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE
1Pi6jizo" crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js"
integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86d
IHNDz0W1" crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js
"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xI
M+B07jRM" crossorigin="anonymous"></script>
</body>
<br><br><br><br><br>
<footer>
    @Team ID: PNT2022TMID53075
</footer>
</html>
```

7.2 Feature 3 - Update password

The user can also update their password from time to time according to their convenience.

ibm_app.py

```
@ app.route('/update', methods=['GET', 'POST'])
```

```
def update():
    form = UpdateForm()
    if form.validate_on_submit():
        stmt = ibm_db.prepare(conn, 'SELECT * FROM user WHERE username=?')
        ibm_db.bind_param(stmt, 1, form.username.data)
        ibm_db.execute(stmt)
        user = ibm_db.fetch_tuple(stmt)
        if user:
            if bcrypt.check_password_hash(user[4], form.oldpassword.data):
                print(user)
                hashed_password1 =
bcrypt.generate_password_hash(form.password.data)
                stmt = ibm_db.prepare(conn, 'UPDATE user SET pass_word=?
WHERE username=?')
                ibm_db.bind_param(stmt, 1, hashed_password1)
                ibm_db.bind_param(stmt, 2, form.username.data)
                user = ibm_db.execute(stmt)
                flash(f'Password changed successfully.', 'success')
                return redirect(url_for('home'))
            else:
                flash(f'Invalid password, Enter valid password.',
'danger')
                return redirect(url_for('update'))
        else:
            flash(f'Invalid user, Enter valid User.', 'danger')
            return redirect(url_for('update'))
    return render_template('update.html', form=form)
```

update.html

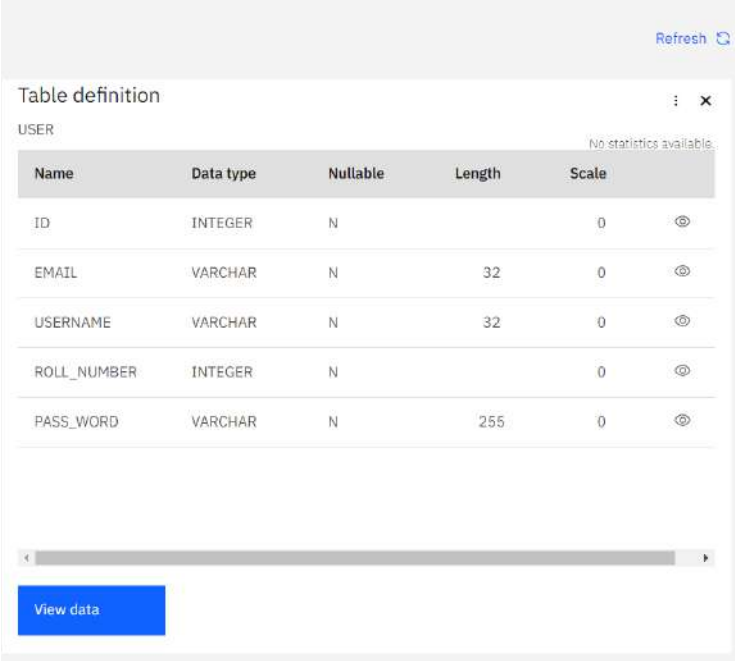
```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.
css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoR
xT2MZw1T" crossorigin="anonymous">
    <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='style.css') }}" />
</head>
```

```
<body>
    <div class="col-md-8">
        {% with messages = get_flashed_messages(with_categories=true) %}
            {% if messages %}
                {% for category, message in messages %}
                    <div class="alert alert-{{category}}">
                        {{ message }}
                    </div>
                {% endfor %}
            {% endif %}
        {% endwith %}
        {% block content %} {% endblock %}
    </div>
    <h1>IBM-Predicting the energy output of wind turbine based on weather
condition</h1> <br>
    <h7>Update Password</h7> <br>

    <form method="POST" action="">
        {{ form.hidden_tag() }}
        <fieldset class="form-group">
            <div>
                {{ form.username.label(class="form-control-label") }}

                {% if form.username.errors %}
                    {{ form.username(class="form-control form-control-lg
is-invalid") }}
                    <div class="invalid-feedback">
                        {% for error in form.username.errors %}
                            <span>{{ error }}</span>
                        {% endfor %}
                    </div>
                {% else %}
                    {{ form.username(class="form-control form-control-lg")
}}
                {% endif %}
            </div>

            <div>
                {{ form.oldpassword.label(class="form-control-label") }}

                {% if form.oldpassword.errors %}
                    {{ form.oldpassword(class="form-control
form-control-lg is-invalid") }}
                    <div class="invalid-feedback">
                        {% for error in form.oldpassword.errors %}
                            <span>{{ error }}</span>
                        {% endfor %}
```

```
                    </div>
                {% else %}
                    {{ form.oldpassword(class="form-control
form-control-lg") }}
                {% endif %}
            </div>


            <div>
                {{ form.password.label(class="form-control-label") }}

                {% if form.password.errors %}
                    {{ form.password(class="form-control form-control-lg
is-invalid") }}
                    <div class="invalid-feedback">
                        {% for error in form.password.errors %}
                            <span>{{ error }}</span>
                        {% endfor %}
                    </div>
                {% else %}
                    {{ form.password(class="form-control form-control-lg")
}}
                {% endif %}
            </div>

        </fieldset>
        <div class="form-group">
            {{ form.submit(class="btn btn-outline-info") }}
        </div>
    </form>

    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE
1Pi6jizo" crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js"
integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86d
IHNDz0W1" crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js
"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xI
M+B07jRM" crossorigin="anonymous"></script>
</body>
<br><br><br><br><br>
<footer>
    @Team ID: PNT2022TMID53075
```

```
</footer>
</html>
```

7.3 Database schema (cloud database):



8. TESTING

8.1 Test Cases

				Date	3-Nov-22								
				Team ID	PNT2022TMID53075								
				Project Name - Predicting the energy output of wind turbine based on weather condition									
				Maximum Marks	4 marks								
Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	Bla Coverage (Y/N)	BUG ID	Executed By
HomePage_TC_001	UI	Home Page	Verify user is able to see home page		1. Enter URL and click go		Home page should be visible with the following UI Elements: a. Login button b. Registration Button c. Update Password Button	Working as expected	Pass		N		
RegisterPage_TC_001	UI	Registration Page	Verify user is able to see Registration page		1. Enter URL and click go 2. Click on the Register Button present in the home page		Login page should be visible with the following UI Elements: a. Email Text Box b. Username Text Box c. Roll Number Text Box d. Password Text Box e. Register button f. Already have an account ? Log in- Link to Login Page	Working as expected	Pass		N		
RegisterPage_TC_002	Functional	Registration Page	Verify the user cannot leave any text box blank		1. Enter URL and click go 2. Click on the Register Button present in the home page 3. Do not enter Email ID 4. Enter a valid username 5. Enter a valid Roll Number 6. Enter a valid password 7. Click on Login Button	emailid: username: ABC roll number:12345 password: Test1234	The application should notify the user that "Please fill out this field"	Working as expected	Pass		N		
RegisterPage_TC_003	Functional	Registration Page	Verify the user cannot enter a username with length lesser than 4 Characters		1. Enter URL and click go 2. Click on the Register Button present in the home page 3. Enter a valid Email ID 4. Enter a username lesser than 4 characters 5. Enter a valid Roll Number 6. Enter a valid password 7. Click on Login Button	emailid: test@gmail.com username: ABC roll number:12345 password: Test1234	The application should notify the user that "Please lengthen this text to 4 characters or more (You are currently using 3 characters)"	Working as expected	Pass		N		
RegisterPage_TC_004	Functional	Registration Page	Verify the user cannot enter a roll number with length lesser than 5 Characters		1. Enter URL and click go 2. Click on the Register Button present in the home page 3. Enter a valid Email ID 4. Enter a valid username 5. Enter a Roll Number lesser than 5 characters 6. Enter a valid password 7. Click on Login Button	emailid: test@gmail.com username: TEST roll number:1234 password: Test1234	The application should notify the user that "Please lengthen this text to 5 characters or more (You are currently using 4 characters)"	Working as expected	Pass		N		
RegisterPage_TC_005	Functional	Registration Page	Verify the user cannot enter a password with length lesser than 8 Characters		1. Enter URL and click go 2. Click on the Register Button present in the home page 3. Enter a valid Email ID 4. Enter a valid username 5. Enter a valid roll number 6. Enter a password lesser than 8 characters 7. Click on Login Button	emailid: test@gmail.com username: TEST roll number:12345 password: Test	The application should notify the user that "Please lengthen this text to 8 characters or more (You are currently using 4 characters)"	Working as expected	Pass		N		
RegisterPage_TC_006	Functional	Registration Page	Verify the user cannot enter a username which is already exists (Used by another user)		1. Enter URL and click go 2. Click on the Register Button present in the home page 3. Enter a valid Email ID 4. Enter a username which already exists 5. Enter a valid roll number 6. Enter a valid password 7. Click on Login Button	emailid: test@gmail.com username: TEST roll number:12345 password: Test1234	The application should notify the user that "The username already exists.Try another one"	Working as expected	Pass		N		
RegisterPage_TC_007	UI	Registration Page	Verify the text box is highlighted when it is clicked		1. Enter URL and click go 2. Click on the Register Button present in the home page 3.Click on a text Box		The text box should be hightlighted in blue to indicate to the user that the have clicked on that particular text box	Working as expected	Pass		N		
RegisterPage_TC_008	UI	Registration Page	Verify the Register Button changes color when hoverd over		1. Enter URL and click go 2. Click on the Register Button present in the home page 3. Hover over Register Button		The Register button should change from white to blue to indiacte to the user that they are hovering over the register button	Working as expected	Pass		N		
RegisterPage_TC_009	Functional	Registration Page	Verify user can navigate to log in page		1. Enter URL and click go 2. Click on the Register Button present in the home page 3. Click on Already have an account ? Log in - Link to Login Page		The user should be taken to the login page	Working as expected	Pass		N		
RegisterPage_TC_010	Functional	Registration Page	Verify user is redirected to Login Page when Valid Registration Credentials are Given		1. Enter URL and click go 2. Click on the Register Button present in the home page 3. Enter a valid Email ID 4. Enter a valid username 5. Enter a valid roll number 6. Enter a valid password 7. Click on Login Button	emailid: test@gmail.com username: TEST roll number:12345 password: Test1234	The user should be redirected to the login page	Working as expected	Pass		N		
LoginPage_TC_001	UI	Login Page	Verify user is able to see Login page		1. Enter URL and click go 2. Click on the Login Button present in the home page		Login page should be visible with the following UI Elements: a. UsernameText Box b. Password Text Box c. Login button d. Do not have an account ? Sign Up- Link to Registration Page	Working as expected	Pass		N		
LoginPage_TC_002	Functional	Login Page	Verify the user cannot leave any text box blank		1. Enter URL and click go 2. Click on the Login Button present in the home page 3. Enter a valid username 4. Do not Enter password 5. Click on Login Button	username: TEST password:	The application should notify the user that "Please fill out this field"	Working as expected	Pass		N		

				Date	3-Nov-22								
				Team ID	PNT2022TMID53075								
				Project Name - Predicting the energy output of wind turbine based on weather condition									
				Maximum Marks	4 marks								
Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	Else automation only	BUG ID	Executed by
LoginPage_TC_003	Functional	Login Page	Verify the user cannot enter a username with length lesser than 4 Characters		1. Enter URL and click go 2. Click on the Login Button present in the home page 3. Enter a username lesser than 4 characters 4. Enter a valid password 5. Click on Login Button	username: ABC password: Test1234	The application should notify the user that "Please lengthen this text to 4 characters or more (You are currently using 3 characters)"	Working as expected	Pass		N		
LoginPage_TC_004	Functional	Login Page	Verify the user cannot enter a password with length lesser than 8 Characters		1. Enter URL and click go 2. Click on the Login Button present in the home page 3. Enter a valid username 4. Enter password lesser than 8 characters 5. Click on Login Button	username: TEST password: Test	The application should notify the user that "Please lengthen this text to 8 characters or more (You are currently using 4 characters)"	Working as expected	Pass		N		
LoginPage_TC_005	Functional	Login Page	Verify the user cannot login with invalid Login Credentials		1. Enter URL and click go 2. Click on the Login Button present in the home page 3. Enter a valid username 4. Enter wrong password 5. Click on Login Button	username: TEST password: WrongPass	The application should notify the user that "Invalid credentials, check and try logging in agian"	Working as expected	Pass		N		
LoginPage_TC_006	UI	Login Page	Verify the text box is highlighted when it is clicked		1. Enter URL and click go 2. Click on the Login Button present in the home page 3.Click on a text Box		The text box should be hightlighted in blue to indicate to the user that the have clicked on that particular text box	Working as expected	Pass		N		
LoginPage_TC_007	UI	Login Page	Verify the Login Button changes color when hoverd over		1. Enter URL and click go 2. Click on the Login Button present in the home page 3. Hover over Login Button		The Login button should change from white to blue to indicate to the user that they are hovering over the register button	Working as expected	Pass		N		
LoginPage_TC_008	Functional	Login Page	Verify user can navigate to Registration page		1. Enter URL and click go 2. Click on the Login Button present in the home page 3. Click on Do not have an account ? Sign Up- Link to Registration Page		The user should be talen to the Registration page	Working as expected	Pass		N		Team
LoginPage_TC_009	Functional	Login Page	Verify User redirected to welcome page when valid credentials have been entered		1. Enter URL and click go 2. Click on the Login Button present in the home page 3. Enter a valid username 4. Enter valiud password 5. Click on Login Button	username: TEST password: Test1234	The user should be redirected to the Welcome page	Working as expected	Pass		N		
WelcomePage_TC_001	UI	Welcome Page	Verify user is able to see Welcome page	User should have logged in to the application using Valid Login Credentials	1. Enter URL and click go 2. Click on the Login Button present in the home page 3. Enter a valid username 4. Enter valiud password 5. Click on Login Button		Welcome page should be visible with the following UI Elements: a. Dashboard Button b. Logout Button c. Text about the application	Working as expected	Pass		N		
WelcomePage_TC_002	Functional	Welcome Page	Verify user can log out of the application	User should have logged in to the application using Valid Login Credentials	1. Enter URL and click go 2. Click on the Login Button present in the home page 3. In Welcome page enter Logout Button		User should be logged out	Working as expected	Pass		N		
Dashboard_TC_001	UI	Dashboard	Verify user is able to see Dashboard	User should have logged in to the application using Valid Login Credentials	1. Enter URL and click go 2. Click on the Login Button present in the home page 3. Click on Dashboard		Dashboard page should be visible with the following UI Elements: a. Wind Direction Text Field b. Month Text Field c. Day Text Field d. Hour Text Field e. Mean Speed Text Field f. Evaluate Button	Working as expected	Pass		N		
Dashboard_TC_002	UI	Dashboard	Verify the text box is highlighted when it is clicked		1. Enter URL and click go 2. Click on the Login Button present in the home page 3. Click on Dashboard		The text box should be hightlighted in black to indicate to the user that the have clicked on that particular text box	Working as expected	Pass		N		
Dashboard_TC_003	Functional	Dashboard	Verify the user cannot leave any text box blank	User should have logged in to the application using Valid Login Credentials	1. Enter URL and click go 2. Click on the Login Button present in the home page 3. Click on Dashboard 4. Enter a valid Wind Direction 5. Enter a valid Month 6. Enter a valid Day 7. Do not enter Hour 8. Enter a valid Mean Speed 9. Click on the Evaluate Button	wind direction: 270 month: 1 day: 1 hour: mean speed: 5.5	The application should notify the user that "Please fill out this field"	Working as expected	Pass		N		
Dashboard_TC_004	Functional	Dashboard	Verify the user is redirected to the predict page	User should have logged in to the application using Valid Login Credentials	1. Enter URL and click go 2. Click on the Login Button present in the home page 3. Click on Dashboard 4. Enter a valid Wind Direction 5. Enter a valid Month 6. Enter a valid Day 7. Enter a valid Hour 8. Enter a valid Mean Speed 9. Click on the Evaluate Button	wind direction: 270 month: 1 day: 1 hour: 23 mean speed: 5.5	The user is redirected to the Predict Page	Working as expected	Pass		N		
PredictPage_TC_001	UI	Predict Page	Verify user is able to see Pridict page	User should have logged in to the application using Valid Login Credentials and The user has entered Valid paramteres for Prediction	1. Enter URL and click go 2. Click on the Login Button present in the home page 3. Click on Dashboard 4. Enter valid parameters in Dashboard and click evaluate		Login page should be visible with the a. predicted value in Watts b. Go Back - Link to Dashboard	Working as expected	Pass		N		
PredictPage_TC_002	Functional	Predict Page	Verify user can navigate to dashboard	User should have logged in to the application using Valid Login Credentials and The user has entered Valid paramteres for Prediction	1. Enter URL and click go 2. Click on the Login Button present in the home page 3. Click on Dashboard 4. Click on evaluate in dashboard 5. Click on Go Back -link to Dashboard		The user should be talen to the dashboard	Working as expected	Pass		N		

				Date	3-Nov-22								
				Team ID	PNT2022TMID53075								
				Project Name - Predicting the energy output of wind turbine based on weather condition									
				Maximum Marks	4 marks								
Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commants	Fix automation P/F	BUG ID	Executed By
UpdatePasswordPage_TC_001	UI	Update Password Page	Verify user is able to see Update Password Page		1. Enter URL and click go 2. Click on the Update Password Button present in the home page		Login page should be visible with the following UI Elements: a. UsernameText Box b. Old Passward Text Box c. Password Text Box d. Update button	Working as expected	Pass		N		
UpdatePasswordPage_TC_002	UI	Update Password Page	Verify the text box is highlighted when it is clicked		1. Enter URL and click go 2. Click on the Update Password Button present in the home page 3.Click on a text Box		The text box should be hightlighted in blueto indicate to the user that the have clicked on that particular text box	Working as expected	Pass		N		
UpdatePasswordPage_TC_003	UI	Update Password Page	Verify the Register Button changes color when hoverd over		1. Enter URL and click go 2. Click on the Update Password Button present in the home page 3. Hover over Update Password Button		The Update button should change from white to blue to indiacte to the user that they are hovering over the register button	Working as expected	Pass		N		
UpdatePasswordPage_TC_004	Functional	Update Password Page	Verify the user cannot leave any text box blank		1. Enter URL and click go 2. Click on the Update Password Button present in the home page 3. Do not enter username 4. Enter the correct old password 5. Enter avalid new password	username: password: Test1234 password: Test4321	The application should notify the user that "Please fill out this field"	Working as expected	Pass		N		
UpdatePasswordPage_TC_005	Functional	Update Password Page	Verify the user cannot enter a username with length lesser than 4 Characters		1. Enter URL and click go 2. Click on the Update Password Button present in the home page 3. Enter username lesser than 4 characters 4. Enter the correct old password 5. Enter avalid new password	username: ABC password: Test1234 password: Test4321	The application should notify the user that "Please lengthen this text to 4 charactersor more (You are currently using 3 characters)"	Working as expected	Pass		N		
UpdatePasswordPage_TC_006	Functional	Update Password Page	Verify the user cannot enter an old password with length lesser than 8 Characters		1. Enter URL and click go 2. Click on the Update Password Button present in the home page 3. Enter avalid username 4. Enter old password lesser than 8 characters 5. Enter avalid new password	username: TEST password: Test password: Test4321	The application should notify the user that "Please lengthen this text to 8 charactersor more (You are currently using 4 characters)"	Working as expected	Pass		N		
UpdatePasswordPage_TC_007	Functional	Update Password Page	Verify the user cannot enter a password with length lesser than 8 Characters		1. Enter URL and click go 2. Click on the Update Password Button present in the home page 3. Enter avalid username 4. Enter avalid old password 5. Enter password lesser than 8 characters	username: TEST password: Test1234 password: Test	The application should notify the user that "Please lengthen this text to 8 charactersor more (You are currently using 4 characters)"	Working as expected	Pass		N		
UpdatePasswordPage_TC_008	Functional	Update Password Page	Verify the user cannot enter an invalid username(non existant)		1. Enter URL and click go 2. Click on the Update Password Button present in the home page 3. Enter ainvalid username 4. Enter avalid old password 5. Enter avalid password	username: XYZ password: Test1234 password: Test4321	The application should notify the user that "Invalid user, Enter valid User"	Working as expected	Pass		N		
UpdatePasswordPage_TC_009	Functional	Update Password Page	Verify the user cannot enter an wrong password		1. Enter URL and click go 2. Click on the Update Password Button present in the home page 3. Enter avalid username 4. Enter awrong old password 5. Enter avalid password	username: TEST password: WrongPass password: Test4321	The application should notify the user that "Invalid password, Enter valid password"	Working as expected	Pass		N		
UpdatePasswordPage_TC_010	Functional	Update Password Page	Verify the password is changed on entering all details right		1. Enter URL and click go 2. Click on the Update Password Button present in the home page 3. Enter avalid username 4. Enter acorrect old password 5. Enter avalid password	username: TEST password: Test1234 password: Test4321	The application should notify the user that "Password changed successfuly."	Working as expected	Pass		N		

8.2 User Acceptance Testing

The purpose of this document is to briefly explain the test coverage and open issues of the Predicting the energy output of wind turbines based on weather conditions project at the time of the release to User Acceptance Testing (UAT).

Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved.

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	9	2	0	0	11
Duplicate	1	0	0	0	1
External	0	0	0	0	0
Fixed	6	2	1	8	15
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	16	4	1	8	27

Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested.

Section	Total Cases	Not Tested	Fail	Pass
Client Application	38	0	0	38
Security	2	0	0	2
Cloud Deployment	2	0	0	2
Exception Reporting	0	0	0	0
Final Report Output	6	0	0	6
Version Control	2	0	0	2

9. RESULTS

9.1 Performance Metrics

Model Performance Testing:

S.No.	Parameter	Values	Screenshot
1.	Metrics	**Lasso Regression Model:** **Training:** MAE -390.1986862434888 , MSE -285030.7951416801 , RMSE - 533.8827541152459, R2 score -0.834908161147255 **Testing:** MAE -392.2440666053982 , MSE -291411.88240853307 , RMSE - 53.8257889435564, R2 score -0.8292153379352329	Training MSE and MAE score: 285030.7951416801 390.1986862434888 Testing MSE and MAE score: 291411.88240853307 392.2440666053982 Training RMSE and R2 score: 533.8827541152459 0.834908161147255 Testing RMSE and R2 score: 539.8257889435564 0.8292153379352329

10. ADVANTAGES & DISADVANTAGES

Our system enables smoother and efficient prediction of wind energy from any turbine provided the necessary readings. Without having to manually predict the output for any weather condition our system is much easier. Some advantages that can be listed are as follows:

1. Easy to input the wind parameters.
2. Computation of the wind energy in a short time using our machine learning model.
3. Simple method to create login for your account.
4. Security of your account and the data input is ensured.
5. Handles a large load of requests since we deploy our system in the cloud.

Some negligible disadvantages of our system are listed below:

1. Any user needs to create an account to use our system even if it's for once. Free trial needs to be created for anonymous users with certain limitations in accessing the features of our system.
2. User needs to manually input the wind parameters. They might feel it better to view just the prediction without having to input data. This can be achieved combining the direct access to the wind sensors with our system.
3. While registering in as a new user, the user needs to again enter their login credentials which shows redundancy.
4. Feedback from users isn't present, hence the user experience cannot be improved further.

11. CONCLUSION

Wind Energy predicting systems are a very helpful system in the current world due to increasing needs for renewable energy and the unpredictable nature of the weather. With the aim of a better tomorrow, we have chosen this problem statement to create this system that predicts the energy that can be generated when we are supplied with the necessary values for the wind

parameters. In order to ensure the security of data of various wind energy suppliers we have created the account facility in the system. To improve the user experience without any delay we have deployed our system in the cloud. Our machine learning model also gives good performance over a wide range of the parameter values. We can now use this efficient system with a simple and easy to use user interface to predict the energy produced given a weather condition.

12. FUTURE SCOPE

Although our system has its maximum efficiency in predicting the wind energy output for the given weather condition, we can still improve our system by adding more features to enhance the user's experience and ease of use. Our system requires manual input from the user which can be automated by directly linking the weather sensor's output to the system, thereby making predictions in real time. Another enhancement to our system can be providing free trial for anonymous users in order to create more awareness about our system and its usefulness. Finally we can add space for users feedback so that we can understand their requirements better.

13. APPENDIX

Source code:

B7-Wind-Power.ipynb

[4]:
```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It␣
 ↪includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='eUF7Yi_sNCXK_DqT9SVffQWmstw__Z7B3SYHhdCwZmBo',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'b7windpower-donotdelete-pr-c6dgxiz3ahhiqh'
object_key = 'Wind_Dataset.csv'

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

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

[4]:

	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		
2	01 01 2018 00:20	306.376587	5.216037		
3	01 01 2018 00:30	419.645905	5.659674		
4	01 01 2018 00:40	380.650696	5.577941		

```
1                519.917511          268.641113
2                390.900016          272.564789
3                516.127569          271.258087
4                491.702972          265.674286
```

[5]:
```
import warnings
warnings.filterwarnings('ignore')
```

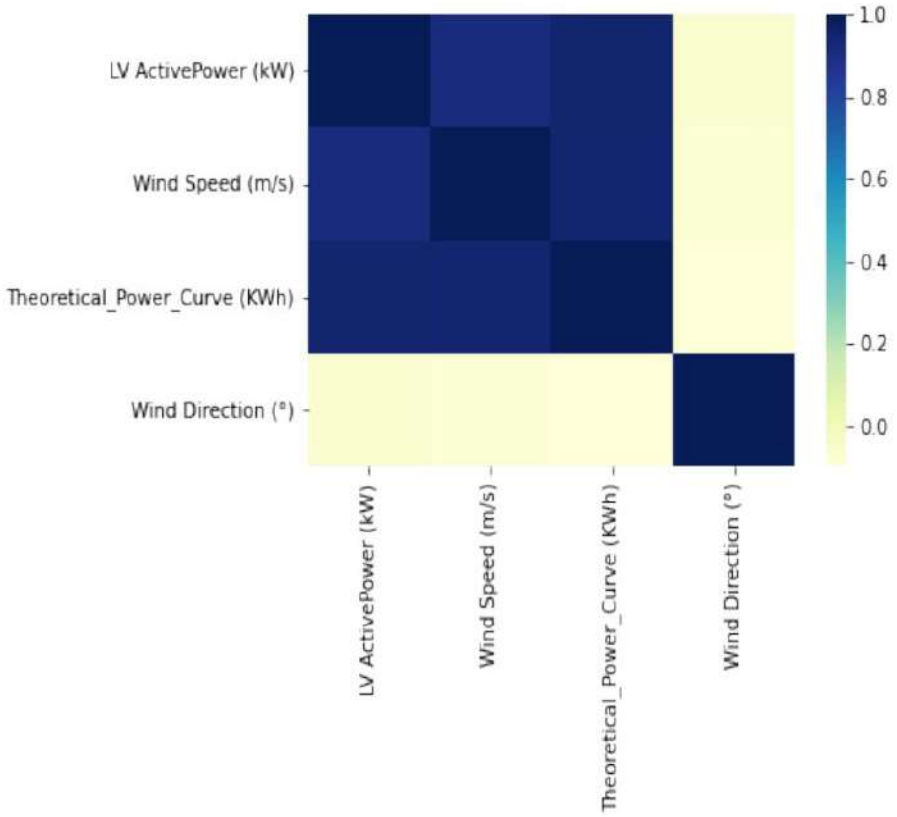
[6]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from collections import Counter
import seaborn as sns
import os
```

[7]:
```
data.shape
```

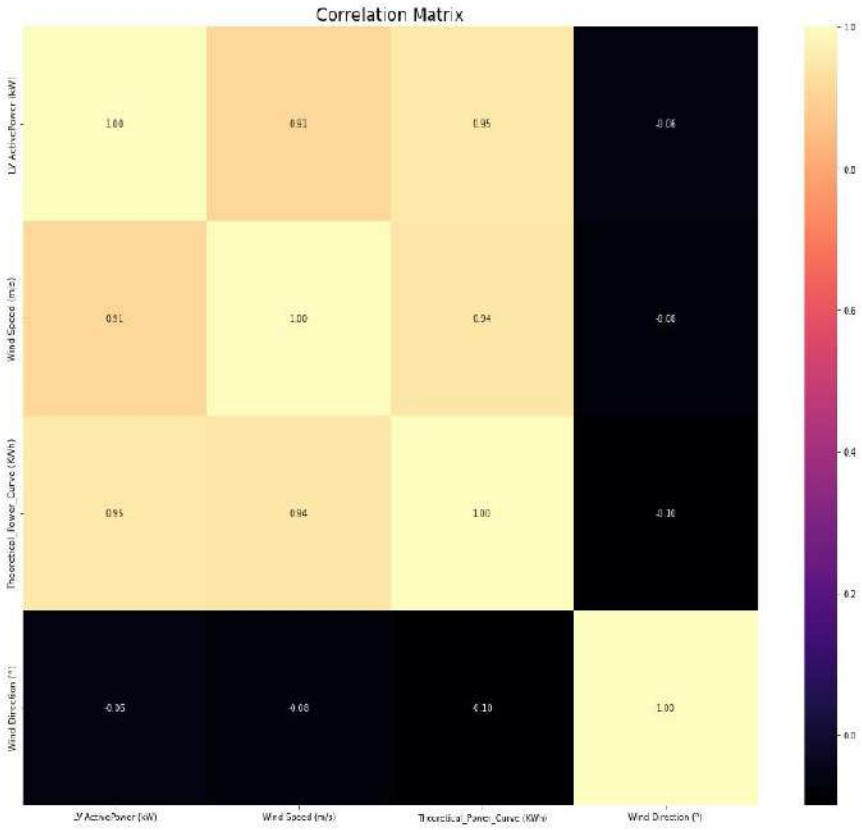
[7]:
```
(50530, 5)
```

[8]:
```
from matplotlib import rcParams
sns.pairplot(data)
```

[8]:
```
<seaborn.axisgrid.PairGrid at 0x7f1f2b36c340>
```



[10]:
```
f, ax = plt.subplots(figsize= [20,15])
sns.heatmap(data.corr(), annot=True, fmt=".2f", ax=ax, cmap = "magma" )
ax.set_title("Correlation Matrix", fontsize=20)
plt.show()
```



[11]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50530 entries, 0 to 50529
Data columns (total 5 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   Date/Time                      50530 non-null  object
 1   LV ActivePower (kW)            50530 non-null  float64
 2   Wind Speed (m/s)               50530 non-null  float64
 3   Theoretical_Power_Curve (KWh)  50530 non-null  float64
 4   Wind Direction (°)             50530 non-null  float64
dtypes: float64(4), object(1)
memory usage: 1.9+ MB
```

[12]:
```
data.describe
```

[12]:
```
<bound method NDFrame.describe of              Date/Time  LV ActivePower (kW)
Wind Speed (m/s)  \
0       01 01 2018 00:00           380.047791          5.311336
1       01 01 2018 00:10           453.769196          5.672167
2       01 01 2018 00:20           306.376587          5.216037
3       01 01 2018 00:30           419.645905          5.659674
4       01 01 2018 00:40           380.650696          5.577941
...                  ...                ...               ...
50525   31 12 2018 23:10          2963.980957         11.404030
50526   31 12 2018 23:20          1684.353027          7.332648
50527   31 12 2018 23:30          2201.106934          8.435358
50528   31 12 2018 23:40          2515.694092          9.421366
50529   31 12 2018 23:50          2820.466064          9.979332

       Theoretical_Power_Curve (KWh)  Wind Direction (°)
0                          416.328908          259.994904
1                          519.917511          268.641113
2                          390.900016          272.564789
3                          516.127569          271.258087
4                          491.702972          265.674286
...                               ...                 ...
50525                     3397.190793           80.502724
50526                     1173.055771           84.062599
50527                     1788.284755           84.742500
50528                     2418.382503           84.297913
50529                     2779.184096           82.274620

[50530 rows x 5 columns]>
```

[13]:
```
data.isnull().any()
```

[13]:
```
Date/Time                       False
LV ActivePower (kW)             False
Wind Speed (m/s)                False
Theoretical_Power_Curve (KWh)   False
Wind Direction (°)              False
dtype: bool
```

[14]:
```
data.rename(columns = {'LV ActivePower (kW)':'ActivePower(kW)',
                       "Wind Speed (m/s)":"WindSpeed(m/s)",
                       "Wind Direction (°)":
 ↪"WindDirection","Theoretical_Power_Curve (KWh)":"TheoreticalPowerCurve(KWh)"},
            inplace = True)
data.head()
```

[14]:
```
          Date/Time  ActivePower(kW)  WindSpeed(m/s)  \
0  01 01 2018 00:00       380.047791        5.311336
```

```
1  01 01 2018 00:10       453.769196        5.672167
2  01 01 2018 00:20       306.376587        5.216037
3  01 01 2018 00:30       419.645905        5.659674
4  01 01 2018 00:40       380.650696        5.577941

   TheoreticalPowerCurve(KWh)  WindDirection
0                  416.328908     259.994904
1                  519.917511     268.641113
2                  390.900016     272.564789
3                  516.127569     271.258087
4                  491.702972     265.674286
```

[15]:
```
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
```

[16]:
```
data['Hour'] = data['Date/Time'].dt.hour
data['minute'] = data['Date/Time'].dt.minute
data.head()
```

[16]:
```
            Date/Time  ActivePower(kW)  WindSpeed(m/s)  \
0 2018-01-01 00:00:00       380.047791        5.311336
1 2018-01-01 00:10:00       453.769196        5.672167
2 2018-01-01 00:20:00       306.376587        5.216037
3 2018-01-01 00:30:00       419.645905        5.659674
4 2018-01-01 00:40:00       380.650696        5.577941

   TheoreticalPowerCurve(KWh)  WindDirection  year  month  day  Hour  minute
0                  416.328908     259.994904  2018      1    1     0       0
1                  519.917511     268.641113  2018      1    1     0      10
2                  390.900016     272.564789  2018      1    1     0      20
3                  516.127569     271.258087  2018      1    1     0      30
4                  491.702972     265.674286  2018      1    1     0      40
```

[17]:
```
def mean_speed(x):
    x = round(x,2)
    a = x//1
    a,b = a+0.25,a+0.75
    if x < a:
        x = a - 0.25
    else:
        x = b -0.25
    return x
```

[18]:
```
data['meanSpeed'] = data['WindSpeed(m/s)'].apply(mean_speed)
data.head(100)
```

[18]:

	Date/Time	ActivePower(kW)	WindSpeed(m/s)	\
0	2018-01-01 00:00:00	380.047791	5.311336	
1	2018-01-01 00:10:00	453.769196	5.672167	
2	2018-01-01 00:20:00	306.376587	5.216037	
3	2018-01-01 00:30:00	419.645905	5.659674	
4	2018-01-01 00:40:00	380.650696	5.577941	
..	...	...	...	
95	2018-01-01 15:50:00	2820.512939	10.772420	
96	2018-01-01 16:00:00	2812.279053	10.647520	
97	2018-01-01 16:10:00	2530.447021	9.982661	
98	2018-01-01 16:20:00	2399.121094	9.874386	
99	2018-01-01 16:30:00	2335.587891	9.785480	

	TheoreticalPowerCurve(KWh)	WindDirection	year	month	day	Hour	minute	\
0	416.328908	259.994904	2018	1	1	0	0	
1	519.917511	268.641113	2018	1	1	0	10	
2	390.900016	272.564789	2018	1	1	0	20	
3	516.127569	271.258087	2018	1	1	0	30	
4	491.702972	265.674286	2018	1	1	0	40	
..	...	...	...	...	...	...	...	
95	3186.029883	225.276398	2018	1	1	15	50	
96	3133.259224	224.680603	2018	1	1	16	0	
97	2781.274041	225.519501	2018	1	1	16	10	
98	2711.492458	227.273804	2018	1	1	16	20	
99	2651.341009	229.255493	2018	1	1	16	30	

	meanSpeed
0	5.5
1	5.5
2	5.0
3	5.5
4	5.5
..	...
95	10.5
96	10.5
97	9.5
98	9.5
99	9.5

[100 rows x 11 columns]

[19]:
```
def mean_direction(x):
    list=[]
    i=15
    while i<=375:
        list.append(i)
        i+=30
```

```
    for i in list:
        if x < i:
            x=i-15
            if x==360:
                return 0
            else:
                return x
```

[20]:
```
data["meanDirection"]=data["WindDirection"].apply(mean_direction)
data.head(100)
```

[20]:

	Date/Time	ActivePower(kW)	WindSpeed(m/s)	\
0	2018-01-01 00:00:00	380.047791	5.311336	
1	2018-01-01 00:10:00	453.769196	5.672167	
2	2018-01-01 00:20:00	306.376587	5.216037	
3	2018-01-01 00:30:00	419.645905	5.659674	
4	2018-01-01 00:40:00	380.650696	5.577941	
..	...	...	...	
95	2018-01-01 15:50:00	2820.512939	10.772420	
96	2018-01-01 16:00:00	2812.279053	10.647520	
97	2018-01-01 16:10:00	2530.447021	9.982661	
98	2018-01-01 16:20:00	2399.121094	9.874386	
99	2018-01-01 16:30:00	2335.587891	9.785480	

	TheoreticalPowerCurve(KWh)	WindDirection	year	month	day	Hour	minute	\
0	416.328908	259.994904	2018	1	1	0	0	
1	519.917511	268.641113	2018	1	1	0	10	
2	390.900016	272.564789	2018	1	1	0	20	
3	516.127569	271.258087	2018	1	1	0	30	
4	491.702972	265.674286	2018	1	1	0	40	
..	...	...	...	...	...	...	...	
95	3186.029883	225.276398	2018	1	1	15	50	
96	3133.259224	224.680603	2018	1	1	16	0	
97	2781.274041	225.519501	2018	1	1	16	10	
98	2711.492458	227.273804	2018	1	1	16	20	
99	2651.341009	229.255493	2018	1	1	16	30	

	meanSpeed	meanDirection
0	5.5	270
1	5.5	270
2	5.0	270
3	5.5	270
4	5.5	270
..	...	...
95	10.5	240
96	10.5	210

```
97        9.5            240
98        9.5            240
99        9.5            240

[100 rows x 12 columns]
```

[21]:
```
directiondict = {0:"N", 30:"NNE", 60:"NEE", 90:"E", 120:"SEE", 150:"SSE", 180:
↪"S", 210:"SSW", 240:"SWW", 270:"W", 300:"NWW", 330:"NNW"}

def wind_direction(x):
  for x in directiondict:
    return directiondict[x]
```

[22]:
```
data['windCDirection'] = data['meanDirection'].apply(wind_direction)
```

[23]:
```
data.head(10)
```

[23]:

	Date/Time	ActivePower(kW)	WindSpeed(m/s)	\
0	2018-01-01 00:00:00	380.047791	5.311336	
1	2018-01-01 00:10:00	453.769196	5.672167	
2	2018-01-01 00:20:00	306.376587	5.216037	
3	2018-01-01 00:30:00	419.645905	5.659674	
4	2018-01-01 00:40:00	380.650696	5.577941	
5	2018-01-01 00:50:00	402.391998	5.604052	
6	2018-01-01 01:00:00	447.605713	5.793008	
7	2018-01-01 01:10:00	387.242188	5.306050	
8	2018-01-01 01:20:00	463.651215	5.584629	
9	2018-01-01 01:30:00	439.725708	5.523228	

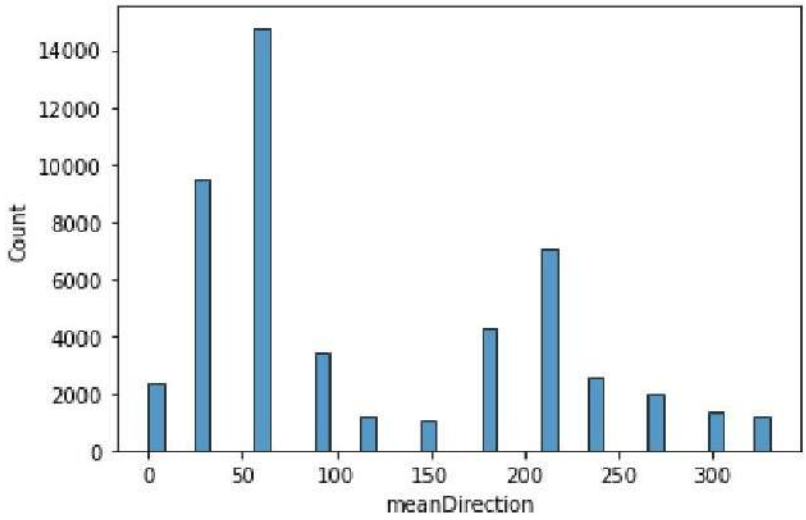
	TheoreticalPowerCurve(KWh)	WindDirection	year	month	day	Hour	minute	\
0	416.328908	259.994904	2018	1	1	0	0	
1	519.917511	268.641113	2018	1	1	0	10	
2	390.900016	272.564789	2018	1	1	0	20	
3	516.127569	271.258087	2018	1	1	0	30	
4	491.702972	265.674286	2018	1	1	0	40	
5	499.436385	264.578613	2018	1	1	0	50	
6	557.372363	266.163605	2018	1	1	1	0	
7	414.898179	257.949493	2018	1	1	1	10	
8	493.677652	253.480698	2018	1	1	1	20	
9	475.706783	258.723785	2018	1	1	1	30	

	meanSpeed	meanDirection	windCDirection
0	5.5	270	N
1	5.5	270	N
2	5.0	270	N
3	5.5	270	N
4	5.5	270	N

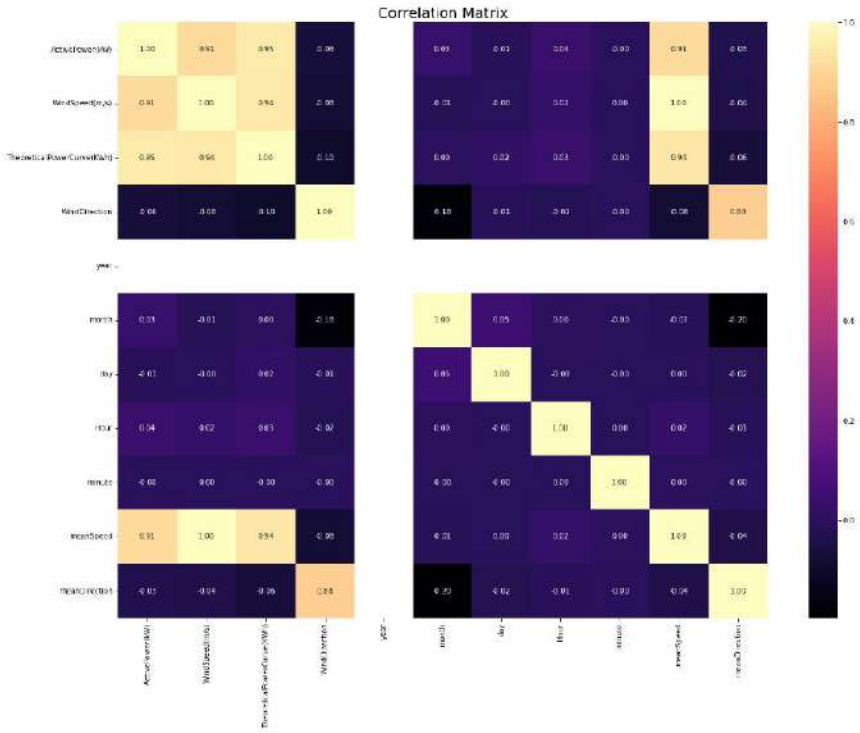
5	5.5	270	N
6	5.5	270	N
7	5.5	270	N
8	5.5	240	N
9	5.5	270	N

[24]:
```
sns.histplot(x=data.meanDirection)
```

[24]:
```
<AxesSubplot:xlabel='meanDirection', ylabel='Count'>
```



[25]:
```
f, ax = plt.subplots(figsize= [20,15])
sns.heatmap(data.corr(), annot=True, fmt=".2f", ax=ax, cmap = "magma" )
ax.set_title("Correlation Matrix", fontsize=20)
plt.show()
```



[26]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50530 entries, 0 to 50529
Data columns (total 13 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Date/Time                   50530 non-null  datetime64[ns]
 1   ActivePower(kW)             50530 non-null  float64
 2   WindSpeed(m/s)              50530 non-null  float64
 3   TheoreticalPowerCurve(KWh)  50530 non-null  float64
 4   WindDirection               50530 non-null  float64
 5   year                        50530 non-null  int64
 6   month                       50530 non-null  int64
 7   day                         50530 non-null  int64
 8   Hour                        50530 non-null  int64
 9   minute                      50530 non-null  int64
 10  meanSpeed                   50530 non-null  float64
```

```
 11  meanDirection              50530 non-null  int64
 12  windCDirection             50530 non-null  object
dtypes: datetime64[ns](1), float64(5), int64(6), object(1)
memory usage: 5.0+ MB
```

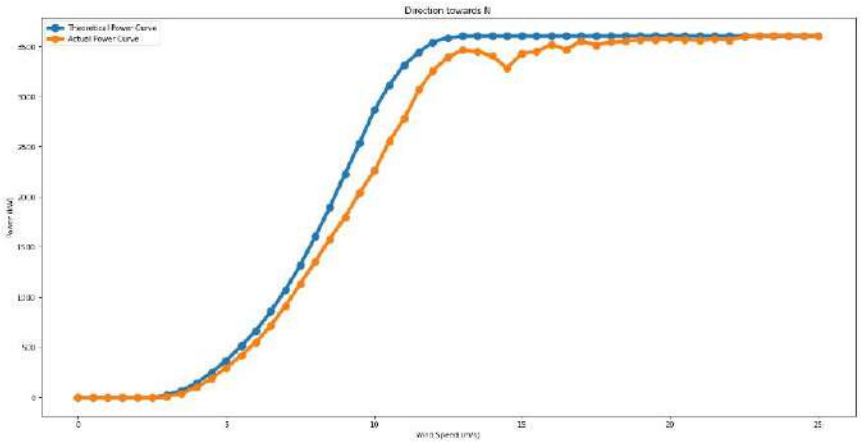
[27]:
```
list_data=[]
list_yon=["N","NNE","NEE","E","SEE","SSE","S","SSW","SWW","W","NWW","NNW"]

for i in range(0,12):
    data1T_A=data[data["windCDirection"] == list_yon[i]]

    DepGroup_A = data1T_A.groupby("meanSpeed")
    data_T_A = DepGroup_A.mean()

    data_T_A.drop(columns = {"WindSpeed(m/s)",
                             "WindDirection",
                             "meanDirection"},
                  inplace = True)

    listTA_WS = data_T_A.index.copy()
    data_T_A["WindSpeed(m/s)"] = listTA_WS

    data_T_A = data_T_A[["WindSpeed(m/s)",
                         "ActivePower(kW)",
                         "TheoreticalPowerCurve(KWh)"]]

    data_T_A["Index"] = list(range(1,len(data_T_A.index)+1))

    data_T_A.set_index("Index", inplace = True)

    data_T_A = data_T_A.round({'ActivePower(kW)': 2,
                               'TheoreticalPowerCurve(KWh)': 2})

    data_T_A["count"] = [len(data1T_A["meanSpeed"][data1T_A["meanSpeed"] == x])␣
 ↪for x in data_T_A["WindSpeed(m/s)"]]
    list_data.append(data_T_A)
```

[28]:
```
data_T_N = list_data[0]
data_T_NNE = list_data[1]
data_T_NEE = list_data[2]
data_T_E = list_data[3]
data_T_SEE = list_data[4]
data_T_SSE = list_data[5]
data_T_S = list_data[6]
data_T_SSW = list_data[7]
data_T_SWW = list_data[8]
data_T_W = list_data[9]
```

```
data_T_NWW = list_data[10]
data_T_NNW = list_data[11]

list_table=[data_T_N,data_T_NNE,data_T_NEE,data_T_E,data_T_SEE,data_T_SSE,data_T_S,
            data_T_SSW,data_T_SWW,data_T_W,data_T_NWW,data_T_NNW]

list_tableName=["N","NNE","NEE","E","SEE","SSE","S","SSW","SWW","W","NWW","NNW"]
```

[29]:
```
def graph_T(i):
    fig = plt.figure(figsize=(20,10))
    plt.plot(list_table[i]["WindSpeed(m/s)"],
             list_table[i]["TheoreticalPowerCurve(KWh)"],
             label = "Theoretical Power Curve",
             marker = "o", markersize = 10, linewidth = 5)

    plt.plot(list_table[i]["WindSpeed(m/s)"],
             list_table[i]["ActivePower(kW)"],
             label = "Actual Power Curve",
             marker = "o", markersize = 10, linewidth = 5)

    plt.xlabel("Wind Speed (m/s)")
    plt.ylabel("Power (kW)")
    plt.title("Direction towards {}".format(list_tableName[i]))
    plt.legend()
    plt.show()
    fig.savefig("{}_Powercurve.jpeg".format(list_tableName[i]))
    plt.close(fig)
```

[30]:
```
for i in range(0,12):
    graph_T(i)
```



[31]:
```
X = data[[ 'WindDirection', 'month', 'day', 'Hour', 'meanSpeed']]
```

[32]:
```
X
```

[32]:

	WindDirection	month	day	Hour	meanSpeed
0	259.994904	1	1	0	5.5
1	268.641113	1	1	0	5.5
2	272.564789	1	1	0	5.0
3	271.258087	1	1	0	5.5
4	265.674286	1	1	0	5.5
...	...	...	...	...	...
50525	80.502724	12	31	23	11.5
50526	84.062599	12	31	23	7.5
50527	84.742500	12	31	23	8.5
50528	84.297913	12	31	23	9.5
50529	82.274620	12	31	23	9.5

[50530 rows x 5 columns]

[33]:
```
y = data['ActivePower(kW)']
```

[34]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import Lasso
from sklearn.metrics import mean_squared_error , r2_score
import joblib
```

[35]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20,␣
↪random_state = 42)
```

[36]:
```
lasso = Lasso(alpha = 0.01)
model = lasso.fit(X_train, y_train)
pred_train_lasso= lasso.predict(X_train)

print("Training RMSE and R2 score:")
print(np.sqrt(mean_squared_error(y_train,pred_train_lasso)))
print(r2_score(y_train, pred_train_lasso))

pred_test_lasso= lasso.predict(X_test)
print("Testing RMSE and R2 score:")
print(np.sqrt(mean_squared_error(y_test,pred_test_lasso)))
print(r2_score(y_test, pred_test_lasso))
```

```
Training RMSE and R2 score:
533.8827541152459
0.834908161147255
Testing RMSE and R2 score:
539.8257889435564
0.8292153379352329
```

[38]:
```
print("Training MSE and MAE score:")
print(mean_squared_error(y_train,pred_train_lasso))
print(mean_absolute_error(y_train, pred_train_lasso))

pred_test_lasso= lasso.predict(X_test)
print("Testing MSE and MAE score:")
print(mean_squared_error(y_test,pred_test_lasso))
print(mean_absolute_error(y_test, pred_test_lasso))
```

```
Training MSE and MAE score:
285030.7951416801
390.1986862434888
Testing MSE and MAE score:
291411.88240853307
392.2440666053982
```

[39]:
```
print(*model.predict([[270,1,1,23,5.5]]))
```

```
745.2822237012499
```

[40]:
```
!pip install -U ibm-watson-machine-learning
```

```
Requirement already satisfied: ibm-watson-machine-learning in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.0.257)
Requirement already satisfied: pandas<1.5.0,>=0.24.2 in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-
learning) (1.3.4)
Requirement already satisfied: importlib-metadata in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-
```

```
learning) (4.8.2)
Requirement already satisfied: requests in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-
learning) (2.26.0)
Requirement already satisfied: lomond in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-
learning) (0.3.3)
Requirement already satisfied: packaging in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-
learning) (21.3)
Requirement already satisfied: certifi in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-
learning) (2022.9.24)
Requirement already satisfied: urllib3 in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-
learning) (1.26.7)
Requirement already satisfied: ibm-cos-sdk==2.11.* in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-
learning) (2.11.0)
Requirement already satisfied: tabulate in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-
learning) (0.8.9)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-
sdk==2.11.*->ibm-watson-machine-learning) (0.10.0)
Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-
sdk==2.11.*->ibm-watson-machine-learning) (2.11.0)
Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-
sdk==2.11.*->ibm-watson-machine-learning) (2.11.0)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk-
core==2.11.0->ibm-cos-sdk==2.11.*->ibm-watson-machine-learning) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from
pandas<1.5.0,>=0.24.2->ibm-watson-machine-learning) (2021.3)
Requirement already satisfied: numpy>=1.17.3 in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from
pandas<1.5.0,>=0.24.2->ibm-watson-machine-learning) (1.20.3)
Requirement already satisfied: six>=1.5 in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-
dateutil<3.0.0,>=2.1->ibm-cos-sdk-core==2.11.0->ibm-cos-sdk==2.11.*->ibm-watson-
machine-learning) (1.15.0)
Requirement already satisfied: charset-normalizer~=2.0.0 in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->ibm-
watson-machine-learning) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in
```

```
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->ibm-
watson-machine-learning) (3.3)
Requirement already satisfied: zipp>=0.5 in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from importlib-
metadata->ibm-watson-machine-learning) (3.6.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from packaging->ibm-
watson-machine-learning) (3.0.4)
```

[41]:
```
from ibm_watson_machine_learning import APIClient
import json
```

[42]:
```
wml_credentials = {
    "apikey":"05v3AdW4glComLU-hp_MrtEo9fn3RLu27kzVIttUAQ6l",
    "url":"https://us-south.ml.cloud.ibm.com"
}
```

[43]:
```
wml_client = APIClient(wml_credentials)
```

[44]:
```
wml_client.spaces.list()
```

```
Note: 'limit' is not provided. Only first 50 records will be displayed if the
number of records exceed 50
------------------------------------  -------------  ------------------------
ID                                    NAME           CREATED
39f5e3f7-78bf-441e-af6a-040607518328  Wind-Power-B7  2022-11-14T09:34:34.326Z
------------------------------------  -------------  ------------------------
```

[45]:
```
SPACE_ID= "39f5e3f7-78bf-441e-af6a-040607518328"
```

[46]:
```
wml_client.set.default_space(SPACE_ID)
```

[46]:
```
'SUCCESS'
```

[47]:
```
wml_client.software_specifications.list(100)
```

```
------------------------------  ------------------------------------  ----
NAME                            ASSET_ID                              TYPE
default_py3.6                   0062b8c9-8b7d-44a0-a9b9-46c416adcbd9  base
kernel-spark3.2-scala2.12       020d69ce-7ac1-5e68-ac1a-31189867356a  base
pytorch-onnx_1.3-py3.7-edt      069ea134-3346-5748-b513-49120e15d288  base
scikit-learn_0.20-py3.6         09c5a1d0-9c1e-4473-a344-eb7b665ff687  base
spark-mllib_3.0-scala_2.12      09f4cff0-90a7-5899-b9ed-1ef348aebdee  base
pytorch-onnx_rt22.1-py3.9       0b848dd4-e681-5599-be41-b5f6fccc6471  base
ai-function_0.1-py3.6           0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda  base
shiny-r3.6                      0e6e79df-875e-4f24-8ae9-62dcc2148306  base
tensorflow_2.4-py3.7-horovod    1092590a-307d-563d-9b62-4eb7d64b3f22  base
pytorch_1.1-py3.6               10ac12d6-6b30-4ccd-8392-3e922c096a92  base
tensorflow_1.15-py3.6-ddl       111e41b3-de2d-5422-a4d6-bf776828c4b7  base
```

```
autoai-kb_rt22.2-py3.10          125b6d9a-5b1f-5e8d-972a-b251688ccf40  base
runtime-22.1-py3.9               12b83a17-24d8-5082-900f-0ab31fbfd3cb  base
scikit-learn_0.22-py3.6          154010fa-5b3b-4ac1-82af-4d5ee5abbc85  base
default_r3.6                     1b70aec3-ab34-4b87-8aa0-a4a3c8296a36  base
pytorch-onnx_1.3-py3.6           1bc6029a-cc97-56da-b8e0-39c3880dbbe7  base
kernel-spark3.3-r3.6             1c9e5454-f216-59dd-a20e-474a5cdf5988  base
pytorch-onnx_rt22.1-py3.9-edt    1d362186-7ad5-5b59-8b6c-9d0880bde37f  base
tensorflow_2.1-py3.6             1eb25b84-d6ed-5dde-b6a5-3fbdf1665666  base
spark-mllib_3.2                  20047f72-0a98-58c7-9ff5-a77b012eb8f5  base
tensorflow_2.4-py3.8-horovod     217c16f6-178f-56bf-824a-b19f20564c49  base
runtime-22.1-py3.9-cuda          26215f05-08c3-5a41-a1b0-da66306ce658  base
do_py3.8                         295addb5-9ef9-547e-9bf4-92ae3563e720  base
autoai-ts_3.8-py3.8              2aa0c932-798f-5ae9-abd6-15e0c2402fb5  base
tensorflow_1.15-py3.6            2b73a275-7cbf-420b-a912-eae7f436e0bc  base
kernel-spark3.3-py3.9            2b7961e2-e3b1-5a8c-a491-482c8368839a  base
pytorch_1.2-py3.6                2c8ef57d-2687-4b7d-acce-01f94976dac1  base
spark-mllib_2.3                  2e51f700-bca0-4b0d-88dc-5c6791338875  base
pytorch-onnx_1.1-py3.6-edt       32983cea-3f32-4400-8965-dde874a8d67e  base
spark-mllib_3.0-py37             36507ebe-8770-55ba-ab2a-eafe787600e9  base
spark-mllib_2.4                  390d21f8-e58b-4fac-9c55-d7ceda621326  base
autoai-ts_rt22.2-py3.10          396b2e83-0953-5b86-9a55-7ce1628a406f  base
xgboost_0.82-py3.6               39e31acd-5f30-41dc-ae44-60233c80306e  base
pytorch-onnx_1.2-py3.6-edt       40589d0e-7019-4e28-8daa-fb03b6f4fe12  base
pytorch-onnx_rt22.2-py3.10       40e73f55-783a-5535-b3fa-0c8b94291431  base
default_r36py38                  41c247d3-45f8-5a71-b065-8580229facf0  base
autoai-ts_rt22.1-py3.9           4269d26e-07ba-5d40-8f66-2d495b0c71f7  base
autoai-obm_3.0                   42b92e18-d9ab-567f-988a-4240ba1ed5f7  base
pmml-3.0_4.3                     493bcb95-16f1-5bc5-bee8-81b8af80e9c7  base
spark-mllib_2.4-r_3.6            49403dff-92e9-4c87-a3d7-a42d0021c095  base
xgboost_0.90-py3.6               4ff8d6c2-1343-4c18-85e1-689c965304d3  base
pytorch-onnx_1.1-py3.6           50f95b2a-bc16-43bb-bc94-b0bed208c60b  base
autoai-ts_3.9-py3.8              52c57136-80fa-572e-8728-a5e7cbb42cde  base
spark-mllib_2.4-scala_2.11       55a70f99-7320-4be5-9fb9-9edb5a443af5  base
spark-mllib_3.0                  5c1b0ca2-4977-5c2e-9439-ffd44ea8ffe9  base
autoai-obm_2.0                   5c2e37fa-80b8-5e77-840f-d912469614ee  base
spss-modeler_18.1                5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b  base
cuda-py3.8                       5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e  base
runtime-22.2-py3.10-xc           5e8cddff-db4a-5a6a-b8aa-2d4af9864dab  base
autoai-kb_3.1-py3.7              632d4b22-10aa-5180-88f0-f52dfb6444d7  base
pytorch-onnx_1.7-py3.8           634d3cdc-b562-5bf9-a2d4-ea90a478456b  base
spark-mllib_2.3-r_3.6            6586b9e3-ccd6-4f92-900f-0f8cb2bd6f0c  base
tensorflow_2.4-py3.7             65e171d7-72d1-55d9-8ebb-f813d620c9bb  base
spss-modeler_18.2                687eddc9-028a-4117-b9dd-e57b36f1efa5  base
pytorch-onnx_1.2-py3.6           692a6a4d-2c4d-45ff-a1ed-b167ee55469a  base
spark-mllib_2.3-scala_2.11       7963efe5-bbec-417e-92cf-0574e21b4e8d  base
spark-mllib_2.4-py37             7abc992b-b685-532b-a122-a396a3cdbaab  base
caffe_1.0-py3.6                  7bb3dbe2-da6e-4145-918d-b6d84aa93b6b  base
pytorch-onnx_1.7-py3.7           812c6631-42b7-5613-982b-02098e6c909c  base
```

```
cuda-py3.6                        82c79ece-4d12-40e6-8787-a7b9e0f62770  base
tensorflow_1.15-py3.6-horovod     8964680e-d5e4-5bb8-919b-8342c6c0dfd8  base
hybrid_0.1                        8c1a58c6-62b5-4dc4-987a-df751c2756b6  base
pytorch-onnx_1.3-py3.7            8d5d8a87-a912-54cf-81ec-3914adaa988d  base
caffe-ibm_1.0-py3.6               8d863266-7927-4d1e-97d7-56a7f4c0a19b  base
runtime-22.2-py3.10-cuda          8ef391e4-ef58-5d46-b078-a82c211c1058  base
spss-modeler_17.1                 902d0051-84bd-4af6-ab6b-8f6aa6fdeabb  base
do_12.10                          9100fd72-8159-4eb9-8a0b-a87e12eefa36  base
do_py3.7                          9447fa8b-2051-4d24-9eef-5acb0e3c59f8  base
spark-mllib_3.0-r_3.6             94bb6052-c837-589d-83f1-f4142f219e32  base
cuda-py3.7-opence                 94e9652b-7f2d-59d5-ba5a-23a414ea488f  base
nlp-py3.8                         96e60351-99d4-5a1c-9cc0-473ac1b5a864  base
cuda-py3.7                        9a44990c-1aa1-4c7d-baf8-c4099011741c  base
hybrid_0.2                        9b3f9040-9cee-4ead-8d7a-780600f542f7  base
spark-mllib_3.0-py38              9f7a8fc1-4d3c-5e65-ab90-41fa8de2d418  base
autoai-kb_3.3-py3.7               a545cca3-02df-5c61-9e88-998b09dc79af  base
spark-mllib_3.0-py39              a6082a27-5acc-5163-b02c-6b96916eb5e0  base
runtime-22.1-py3.9-do             a7e7dbf1-1d03-5544-994d-e5ec845ce99a  base
default_py3.8                     ab9e1b80-f2ce-592c-a7d2-4f2344f77194  base
tensorflow_rt22.1-py3.9           acd9c798-6974-5d2f-a657-ce06e986df4d  base
kernel-spark3.2-py3.9             ad7033ee-794e-58cf-812e-a95f4b64b207  base
autoai-obm_2.0 with Spark 3.0     af10f35f-69fa-5d66-9bf5-acb58434263a  base
runtime-22.2-py3.10               b56101f1-309d-549b-a849-eaa63f77b2fb  base
default_py3.7_opence              c2057dd4-f42c-5f77-a02f-72bdbd3282c9  base
tensorflow_2.1-py3.7              c4032338-2a40-500a-beef-b01ab2667e27  base
do_py3.7_opence                   cc8f8976-b74a-551a-bb66-6377f8d865b4  base
spark-mllib_3.3                   d11f2434-4fc7-58b7-8a62-755da64fdaf8  base
autoai-kb_3.0-py3.6               d139f196-e04b-5d8b-9140-9a10ca1fa91a  base
spark-mllib_3.0-py36              d82546d5-dd78-5fbb-9131-2ec309bc56ed  base
autoai-kb_3.4-py3.8               da9b39c3-758c-5a4f-9cfd-457dd4d8c395  base
kernel-spark3.2-r3.6              db2fe4d6-d641-5d05-9972-73c654c60e0a  base
autoai-kb_rt22.1-py3.9            db6afe93-665f-5910-b117-d879897404d9  base
tensorflow_rt22.1-py3.9-horovod   dda170cc-ca67-5da7-9b7a-cf84c6987fae  base
autoai-ts_1.0-py3.7               deef04f0-0c42-5147-9711-89f9904299db  base
tensorflow_2.1-py3.7-horovod      e384fce5-fdd1-53f8-bc71-11326c9c635f  base
default_py3.7                     e4429883-c883-42b6-87a8-f419d64088cd  base
do_22.1                           e51999ba-6452-5f1f-8287-17228b88b652  base
autoai-obm_3.2                    eae86aab-da30-5229-a6a6-1d0d4e368983  base
runtime-22.2-r4.2                 ec0a3d28-08f7-556c-9674-ca7c2dba30bd  base
tensorflow_rt22.2-py3.10          f65bd165-f057-55de-b5cb-f97cf2c0f393  base
do_20.1                           f686cdd9-7904-5f9d-a732-01b0d6b10dc5  base
-------------------------------   ------------------------------------  ----
```

[48]:
```
import sklearn
sklearn.__version__
```

[48]: '1.0.2'

[49]:
```
MODEL_NAME = 'B7-Wind-Power'
DEPLOYMENT_NAME = 'Wind-Power-B7'
DEMO_MODEL = model
```

[50]:
```
software_spec_uid = wml_client.software_specifications.
 ↪get_id_by_name('runtime-22.1-py3.9')
```

[51]:
```
# Setup model meta
model_props = {
    wml_client.repository.ModelMetaNames.NAME: MODEL_NAME,
    wml_client.repository.ModelMetaNames.TYPE: 'scikit-learn_1.0',
    wml_client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_spec_uid
}
```

[52]:
```
#Save model

model_details = wml_client.repository.store_model(
    model=DEMO_MODEL,
    meta_props=model_props,
    training_data=X_train,
    training_target=y_train
)
```

[53]:
```
model_details
```

[53]:
```
{'entity': {'hybrid_pipeline_software_specs': [],
  'label_column': 'ActivePower(kW)',
  'schemas': {'input': [{'fields': [{'name': 'WindDirection',
       'type': 'float64'},
      {'name': 'month', 'type': 'int64'},
      {'name': 'day', 'type': 'int64'},
      {'name': 'Hour', 'type': 'int64'},
      {'name': 'meanSpeed', 'type': 'float64'}],
     'id': '1',
     'type': 'struct'}],
   'output': []},
  'software_spec': {'id': '12b83a17-24d8-5082-900f-0ab31fbfd3cb',
   'name': 'runtime-22.1-py3.9'},
  'type': 'scikit-learn_1.0'},
 'metadata': {'created_at': '2022-11-18T03:14:11.115Z',
  'id': '77f9c8ed-2a3e-40ef-a3ef-af0b04301423',
  'modified_at': '2022-11-18T03:14:14.254Z',
  'name': 'B7-Wind-Power',
  'owner': 'IBMid-6640040VGB',
  'resource_key': '5e34b0c8-31c2-4854-b1d5-58318218c0a0',
  'space_id': '39f5e3f7-78bf-441e-af6a-040607518328'},
 'system': {'warnings': []}}
```

```
[54]: model_id = wml_client.repository.get_model_id(model_details)
      model_id

[54]: '77f9c8ed-2a3e-40ef-a3ef-af0b04301423'

[55]: # Set meta
      deployment_props = {
          wml_client.deployments.ConfigurationMetaNames.NAME:DEPLOYMENT_NAME,
          wml_client.deployments.ConfigurationMetaNames.ONLINE: {}
      }

[56]: # Deploy
      deployment = wml_client.deployments.create(
          artifact_uid=model_id,
          meta_props=deployment_props
      )

      ##########################################################################################
      #######

      Synchronous deployment creation for uid: '77f9c8ed-2a3e-40ef-a3ef-af0b04301423'
      started

      ##########################################################################################
      #######


      initializing
      Note: online_url is deprecated and will be removed in a future release. Use
      serving_urls instead.

      ready


      ------------------------------------------------------------------------------
      ----------------
      Successfully finished deployment creation,
      deployment_uid='4e141cac-848d-44c4-a8a0-b9bd034cbf7d'
      ------------------------------------------------------------------------------
      ----------------


[ ]:
```

style.css :

```
.header {
  padding: 5px 120px;
  width: 150px;
  height: 70px;
  background-color: #236b8e;
}
footer{
  background-color: #06182b;
```

```
  color: #ffffff;
  text-align: center;
  font-size: 20px;
}
.border {
  padding: 80px 50px;
  width: 400px;
  height: 450px;
  border: 1px solid #236b8e;
  border-radius: 0px;
  background-color: #9ac0cd;
}

.btn {
  padding: 10px 40px;
  background-color: #236b8e;
  color: #ffffff;
  font-style: oblique;
  font-weight: bold;
  border-radius: 10px;
  text-align: center;
}

.textbox {
  padding: 10px 40px;
  background-color: #236b8e;
  text-shadow: #ffffff;
  border-radius: 10px;
}

::placeholder {
  color: #ffffff;
  opacity: 1;
  font-style: oblique;
  font-weight: bold;
}

.word {
  color: #ffffff;
  font-style: oblique;
  font-weight: bold;
}

.bottom {
  color: #236b8e;
  font-style: oblique;
  font-weight: bold;
```

```
}

body {
  background: #fafafa;
  color: #ffffff;
  margin-top: 5rem;
  background-image:
url('https://tees.tamu.edu/news/2018/09/_news-images/isen-news-wind-turbin
e-25sept18tees.png');
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-size: cover;
  text-align: center;
}

h1, h2, h3, h4, h5, h6 {

padding: 30px;
text-align: center;
font-size: 40px;
color: white;
}
h7{
  padding: 30px;
text-align: center;
font-size: 30px;
color: white;
}
p{
  padding: 40px;
color: white;
text-align: center;
}
.bg-steel {
  background-color: #5f788a;
}

.site-header .navbar-nav .nav-link {
  color: #cbd5db;
}

.site-header .navbar-nav .nav-link:hover {
  color: #ffffff;
}

.site-header .navbar-nav .nav-link.active {
  font-weight: 500;
```

```
}

.content-section {
  background: #ffffff;
  padding: 10px 20px;
  border: 1px solid #dddddd;
  border-radius: 3px;
  margin-bottom: 20px;
}

div{
  text-align: center;
  display: inline-block;

}
.btn {
  outline: none;
  border: none;
  cursor: pointer;
  display: inline-block;
  margin: 0 auto;
  padding: 0.9rem 2.5rem;
  text-align: center;
  background-color:white;
  color: #2B547E;
  border-radius: 4px;
  box-shadow: 0px 3px 6px rgba(0, 0, 0, 0.16);
}

.article-title {
  color: #444444;
}

a.article-title:hover {
  color: #428bca;
  text-decoration: none;
}

.article-content {
  white-space: pre-line;
}

.article-img {
  height: 65px;
  width: 65px;
  margin-right: 16px;
}
```

```
.article-metadata {
  padding-bottom: 1px;
  margin-bottom: 4px;
  border-bottom: 1px solid #e3e3e3
}

.article-metadata a:hover {
  color: #333;
  text-decoration: none;
}

.article-svg {
  width: 25px;
  height: 25px;
  vertical-align: middle;
}

.account-img {
  height: 125px;
  width: 125px;
  margin-right: 20px;
  margin-bottom: 16px;
}

.account-heading {
  font-size: 2.5rem;
}
```

HTML Templates:

home.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.
css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoR
xT2MZw1T" crossorigin="anonymous">
    <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='style.css') }}" />
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home</title>
</head>

<body>
    <div class="col-md-8">
        {% with messages = get_flashed_messages(with_categories=true) %}
            {% if messages %}
                {% for category, message in messages %}
                    <div class="alert alert-{{category}}">
                        {{ message }}
                    </div>
                {% endfor %}
            {% endif %}
        {% endwith %}
        {% block content %} {% endblock %}
    </div>
    <h1>IBM-Predicting the energy output of wind turbine based on weather
condition</h1> <br>
    <br><br><br>
    <a href="{{ url_for('login') }}">
        <button class="btn" type="submit" >Login</button>
    </a>
    <a href="{{ url_for('register') }}">
        <button class="btn" type="submit">Register</button>
    </a>
    <a href="{{ url_for('update') }}">
        <button class="btn" type="submit">Update Password</button>
    </a>



    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE
1Pi6jizo" crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js"
integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86d
IHNDz0W1" crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js
"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xI
M+B07jRM" crossorigin="anonymous"></script>
</body>
<br><br><br><br><br><br><br><br>
<footer>
```

```
    @Team ID: PNT2022TMID53075
</footer>
</html>
```

register.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.
css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoR
xT2MZw1T" crossorigin="anonymous">
    <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='style.css') }}" />
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Register</title>
</head>

<body>
    <h1>IBM-Predicting the energy output of wind turbine based on weather
condition</h1> <br>

    <h7>Register Page</h7> <br>


    <form method="POST" action="">
        {{ form.hidden_tag() }}
        <fieldset class="form-group">
            <!-- <legend class="border-bottom mb-4">Registration
Page</legend> -->

            <div>
                {{ form.email.label(class="form-control-label") }}

                {% if form.email.errors %}
                    {{ form.email(class="form-control form-control-lg
is-invalid") }}
                    <div class="invalid-feedback">
                        {% for error in form.email.errors %}
                            <span>{{ error }}</span>
                        {% endfor %}
                    </div>
```

```
                {% else %}
                    {{ form.email(class="form-control form-control-lg") }}
                {% endif %}
            </div>


            <div>
                {{ form.username.label(class="form-control-label") }}

                {% if form.username.errors %}
                    {{ form.username(class="form-control form-control-lg
is-invalid") }}
                    <div class="invalid-feedback">
                        {% for error in form.username.errors %}
                            <span>{{ error }}</span>
                        {% endfor %}
                    </div>
                {% else %}
                    {{ form.username(class="form-control form-control-lg")
}}
                {% endif %}
            </div>

            <div>
                {{ form.rollnumber.label(class="form-control-label") }}

                {% if form.rollnumber.errors %}
                    {{ form.rollnumber(class="form-control form-control-lg
is-invalid") }}
                    <div class="invalid-feedback">
                        {% for error in form.rollnumber.errors %}
                            <span>{{ error }}</span>
                        {% endfor %}
                    </div>
                {% else %}
                    {{ form.rollnumber(class="form-control
form-control-lg") }}
                {% endif %}
            </div>


            <div>
                {{ form.password.label(class="form-control-label") }}

                {% if form.password.errors %}
                    {{ form.password(class="form-control form-control-lg
is-invalid") }}
```

```
                    <div class="invalid-feedback">
                        {% for error in form.password.errors %}
                            <span>{{ error }}</span>
                        {% endfor %}
                    </div>
                {% else %}
                    {{ form.password(class="form-control form-control-lg")
}}
                {% endif %}
            </div>

        </fieldset>
        <div class="form-group">
            {{ form.submit(class="btn btn-outline-info") }}
        </div>

        <small class="text-muted ml-2"><br><br>
            <a href="{{ url_for('login') }}">Already have an account? Log
In</a>
        </small>

    </form>

    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE
1Pi6jizo" crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js"
integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86d
IHNDz0W1" crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js
"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xI
M+B07jRM" crossorigin="anonymous"></script>
</body>
<br><br><br><br><br>
<footer>
    @Team ID: PNT2022TMID53075
</footer>
</html>
```

login.html

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.
css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoR
xT2MZw1T" crossorigin="anonymous">
    <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='style.css') }}" />
</head>

<body>
    <div class="col-md-8">
        {% with messages = get_flashed_messages(with_categories=true) %}
            {% if messages %}
                {% for category, message in messages %}
                    <div class="alert alert-{{category}}">
                        {{ message }}
                    </div>
                {% endfor %}
            {% endif %}
        {% endwith %}
        {% block content %} {% endblock %}
    </div>
    <h1>IBM-Predicting the energy output of wind turbine based on weather
condition</h1> <br>

    <h7>Login Page</h7> <br>

    <form method="POST" action="">
        {{ form.hidden_tag() }}
        <fieldset class="form-group">
            <div>
                {{ form.username.label(class="form-control-label") }}

                {% if form.username.errors %}
                    {{ form.username(class="form-control form-control-lg
is-invalid") }}
                    <div class="invalid-feedback">
                        {% for error in form.username.errors %}
                            <span>{{ error }}</span>
                        {% endfor %}
                    </div>
                {% else %}
```

```
                    {{ form.username(class="form-control form-control-lg")
}}
                {% endif %}
            </div>


            <div>
                {{ form.password.label(class="form-control-label") }}

                {% if form.password.errors %}
                    {{ form.password(class="form-control form-control-lg
is-invalid") }}
                    <div class="invalid-feedback">
                        {% for error in form.password.errors %}
                            <span>{{ error }}</span>
                        {% endfor %}
                    </div>
                {% else %}
                    {{ form.password(class="form-control form-control-lg")
}}
                {% endif %}
            </div>

        </fieldset>
        <div class="form-group">
            {{ form.submit(class="btn btn-outline-info") }}
        </div>

        <small class="text-muted ml-2"><br>
            <a href='{{url_for('register')}}'>Do not have an account? Sign
Up?</a>
        </small>

    </form>

    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE
1Pi6jizo" crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js"
integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86d
IHNDz0W1" crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js
"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xI
M+B07jRM" crossorigin="anonymous"></script>
```

```
</body>
<br><br><br><br><br>
<footer>
    @Team ID: PNT2022TMID53075
</footer>
</html>
```

update.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.
css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoR
xT2MZw1T" crossorigin="anonymous">
    <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='style.css') }}" />
</head>

<body>
    <div class="col-md-8">
        {% with messages = get_flashed_messages(with_categories=true) %}
            {% if messages %}
                {% for category, message in messages %}
                    <div class="alert alert-{{category}}">
                        {{ message }}
                    </div>
                {% endfor %}
            {% endif %}
        {% endwith %}
        {% block content %} {% endblock %}
    </div>
    <h1>IBM-Predicting the energy output of wind turbine based on weather
condition</h1> <br>
    <h7>Update Password</h7> <br>

    <form method="POST" action="">
        {{ form.hidden_tag() }}
        <fieldset class="form-group">
            <div>
```

```
                {{ form.username.label(class="form-control-label") }}

                {% if form.username.errors %}
                    {{ form.username(class="form-control form-control-lg
is-invalid") }}
                    <div class="invalid-feedback">
                        {% for error in form.username.errors %}
                            <span>{{ error }}</span>
                        {% endfor %}
                    </div>
                {% else %}
                    {{ form.username(class="form-control form-control-lg")
}}
                {% endif %}
            </div>

            <div>
                {{ form.oldpassword.label(class="form-control-label") }}

                {% if form.oldpassword.errors %}
                    {{ form.oldpassword(class="form-control
form-control-lg is-invalid") }}
                    <div class="invalid-feedback">
                        {% for error in form.oldpassword.errors %}
                            <span>{{ error }}</span>
                        {% endfor %}
                    </div>
                {% else %}
                    {{ form.oldpassword(class="form-control
form-control-lg") }}
                {% endif %}
            </div>


            <div>
                {{ form.password.label(class="form-control-label") }}

                {% if form.password.errors %}
                    {{ form.password(class="form-control form-control-lg
is-invalid") }}
                    <div class="invalid-feedback">
                        {% for error in form.password.errors %}
                            <span>{{ error }}</span>
                        {% endfor %}
                    </div>
                {% else %}
```

```
                    {{ form.password(class="form-control form-control-lg")
}}
                {% endif %}
            </div>

        </fieldset>
        <div class="form-group">
            {{ form.submit(class="btn btn-outline-info") }}
        </div>
    </form>

    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE
1Pi6jizo" crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js"
integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86d
IHNDz0W1" crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js
"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xI
M+B07jRM" crossorigin="anonymous"></script>
</body>
<br><br><br><br><br>
<footer>
    @Team ID: PNT2022TMID53075
</footer>
</html>
```

welcome.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.
css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoR
xT2MZw1T" crossorigin="anonymous">
    <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='style.css') }}" />
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Welcome Page</title>
```

```
</head>

<body>
    <h1>IBM-Predicting the energy output of wind turbine based on weather
condition</h1> <br>
    <h7>Welcome!</h7> <br>
    <a href="{{url_for('dashboard')}}">
        <button class="btn" type="submit" >Dashboard</button>
    </a>
    <a href="{{url_for('logout')}}">
        <button class="btn" type="submit" >Logout</button>
    </a>
    <p><b>Wind power or wind energy is mostly the use of wind turbines to
generate electricity. Wind power is a popular, sustainable, renewable
energy source that has a much smaller impact on the environment than
burning fossil fuels. Historically, wind power has been used in sails,
windmills and windpumps but today it is mostly used to generate
electricity. Wind farms consist of many individual wind turbines, which
are connected to the electric power transmission network.Wind energy
conversion devices like wind turbines are used for converting wind energy
into mechanical energy. Wind turbines consist basically of a few sails,
vanes or blades radiating from a central axis. When wind blows against the
blades or vanes, they rotate about the axis. This rotational motion is
utilised to perform some useful work. By connecting the wind turbine to an
electric generator wind energy can be converted into electric energy.
        In India, winds are relatively low (5 km/hr to 15/20 km/hr) and
vary appreciably with season. This makes the exploitation of wind energy
an expensive project.</b></p>

    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE
1Pi6jizo" crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/popper.min.js"
integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86d
IHNDz0W1" crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootstrap.min.js
"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xI
M+B07jRM" crossorigin="anonymous"></script>
</body><br><br><br><br><br>
<footer>
    @Team ID: PNT2022TMID53075
</footer>
</html>
```

dashboard.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.
css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoR
xT2MZw1T" crossorigin="anonymous">
    <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='style.css') }}" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>DashBoard</title>
</head>
<body>
    <h1>IBM-Predicting the energy output of wind turbine based on weather
condition</h1> <br>

    <h7>Dashboard</h7>
    <form method="POST" action="/predict">
        <p>Wind Direction</p>
        <input name="wdir" required>
        <p>Month</p>
        <input name="mnt" required>
        <p>Day</p>
        <input name="dy" required>
        <p>Hour</p>
        <input name="hr" required>
        <p>Mean Speed</p>
        <input name="mspd" required>
        <br>
        <br>
        <button type="submit">Evaluate</button>
    </form>
</body>
<br><br><br><br><br>
<footer>
    @Team ID: PNT2022TMID53075
</footer>
</html>
```

predict.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.
css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoR
xT2MZw1T" crossorigin="anonymous">
    <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='style.css') }}" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>DashBoard</title>
</head>
<body>
    <h1>IBM-Predicting the energy output of wind turbine based on weather
condition</h1> <br>
    <h1>The predicted power(in watts) is</h1>
    <h1>{{predict}}</h1>
    <a href="{{url_for('dashboard')}}">
        <button class="btn" type="submit" >Go to Dashboard</button>
<br><br><br>
    </a>
    <a href="{{url_for('logout')}}">
        <button class="btn" type="submit" >Logout</button>
    </a>
</body>
<br><br><br><br><br>
<footer>
    @Team ID: PNT2022TMID53075
</footer>
</html>
```

ibm_app.py

```
from flask import Flask, render_template, url_for, redirect, flash,
request
from flask_login import login_user, LoginManager, login_required,
logout_user
from wtforms.validators import InputRequired, Length, ValidationError
from flask_bcrypt import Bcrypt
from wtforms import StringField, PasswordField, SubmitField
from flask_wtf import FlaskForm
from flask_cors import CORS
```

```
import joblib
import ibm_db
import requests

# NOTE: you must manually set API_KEY below using information retrieved
from your IBM Cloud account.
API_KEY = "05v3AdW4glComLU-hp_MrtEo9fn3RLu27kzVIttUAQ6l"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token',
data={"apikey": API_KEY, "grant_type":
'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' +
mltoken}

app = Flask(__name__)
CORS(app)

bcrypt = Bcrypt(app)
app.config['SECRET_KEY'] = 'B7-1A3E'

conn =
ibm_db.connect("DATABASE=bludb;HOSTNAME=1bbf73c5-d84a-4bb0-85b9-ab1a4348f4
a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=32286;SECURITY=SSL;
SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=jqx32302;PWD=xB6z3lioK6c
EBx7c", '', '')

login_manager = LoginManager()
login_manager.init_app(app)
login_manager.login_view = 'login'

@login_manager.user_loader
def load_user(user_id):
    stmt = ibm_db.prepare(conn, 'SELECT * FROM user WHERE id=?')
    ibm_db.bind_param(stmt, 1, user_id)
    ibm_db.execute(stmt)
    user = ibm_db.fetch_tuple(stmt)
    usr_obj = User(user[0], user[1], user[2])
    return usr_obj

class User:
    def __init__(self, id, email, username):
        self.id = id
        self.username = username
        self.email = email

    def to_json(self):
```

```
        return {"username": self.username, "email": self.email}

    def is_authenticated(self):
        return True

    def is_active(self):
        return True

    def is_anonymous(self):
        return False

    def get_id(self):
        return str(self.id)


class RegisterForm(FlaskForm):
    email = StringField(validators=[InputRequired(), Length(min=4,
max=50)], render_kw={"placeholder":"Email"})
    username = StringField(validators=[InputRequired(), Length(min=4,
max=20)], render_kw={"placeholder": "Username"})
    rollnumber = StringField(validators=[InputRequired(), Length(min=5,
max=10)], render_kw={"placeholder":"RollNumber"})
    password = PasswordField(validators=[InputRequired(), Length(min=8,
max=20)],render_kw={"placeholder": "Password"})
    submit = SubmitField('Register')

    def validate_username(self, username):
        stmt = ibm_db.prepare(conn, 'SELECT * FROM user WHERE username=?')
        ibm_db.bind_param(stmt, 1, username.data)
        ibm_db.execute(stmt)
        existing_user_username = ibm_db.fetch_tuple(stmt)
        if existing_user_username:
            raise ValidationError('That username already exists. Try
another one.')

class LoginForm(FlaskForm):
    username = StringField(validators=[InputRequired(), Length(min=4,
max=20)], render_kw={"placeholder": "Username"})
    password = PasswordField(validators=[InputRequired(), Length(min=8,
max=20)], render_kw={"placeholder": "Password"})
    submit = SubmitField('Login')

class UpdateForm(FlaskForm):
    username = StringField(validators=[InputRequired(), Length(min=4,
max=20)], render_kw={"placeholder": "Username"})
    oldpassword = PasswordField(validators=[InputRequired(), Length(min=8,
max=20)], render_kw={"placeholder":"Previous Password"})
```

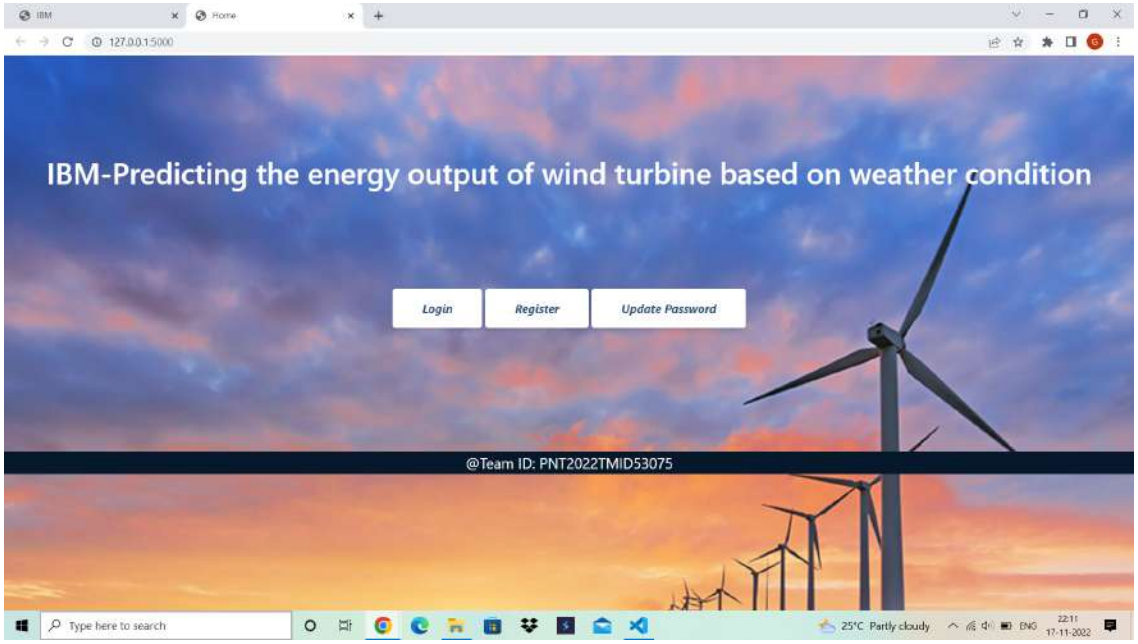
```
    password = PasswordField(validators=[InputRequired(), Length(min=8,
max=20)], render_kw={"placeholder": "Password"})
    submit = SubmitField('Update')

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

@app.route('/login', methods=['GET', 'POST'])
def login():
    form = LoginForm()
    if form.validate_on_submit():
        stmt = ibm_db.prepare(conn, 'SELECT * FROM user WHERE username=?')
        ibm_db.bind_param(stmt, 1, form.username.data)
        ibm_db.execute(stmt)
        user = ibm_db.fetch_tuple(stmt)
        if user:
            if bcrypt.check_password_hash(user[4], form.password.data):
                usr_obj = User(user[0], user[1], user[2])
                login_user(usr_obj)
                return redirect(url_for('welcome'))
            else:
                print('Hi')
                flash(f'Invalid credentials, check and try logging in
again.', 'danger')
                return redirect(url_for('login'))
    return render_template('login.html', form=form)

@app.route('/welcome', methods=['GET', 'POST'])
@login_required
def welcome():
    return render_template('welcome.html')
@app.route('/dashboard')
def dashboard():
    return render_template('dashboard.html')

@app.route('/logout', methods=['GET', 'POST'])
@login_required
def logout():
    logout_user()
    return redirect(url_for('home'))

@ app.route('/register', methods=['GET', 'POST'])
def register():
    form = RegisterForm()
    if form.validate_on_submit():
```

```
        hashed_password =
bcrypt.generate_password_hash(form.password.data)
        stmt = ibm_db.prepare(conn, 'INSERT INTO user (email, username,
roll_number, pass_word) VALUES (?, ?, ?, ?)')
        ibm_db.bind_param(stmt, 1, form.email.data)
        ibm_db.bind_param(stmt, 2, form.username.data)
        ibm_db.bind_param(stmt, 3, form.rollnumber.data)
        ibm_db.bind_param(stmt, 4, hashed_password)
        #hash causes size to exceed VARCHAR size in DB2, hence made
VARCHAR(8000)
        ibm_db.execute(stmt)
        return redirect(url_for('login'))
    return render_template('register.html', form=form)

@ app.route('/update', methods=['GET', 'POST'])
def update():
    form = UpdateForm()
    if form.validate_on_submit():
        stmt = ibm_db.prepare(conn, 'SELECT * FROM user WHERE username=?')
        ibm_db.bind_param(stmt, 1, form.username.data)
        ibm_db.execute(stmt)
        user = ibm_db.fetch_tuple(stmt)
        if user:
            if bcrypt.check_password_hash(user[4], form.oldpassword.data):
                print(user)
                hashed_password1 =
bcrypt.generate_password_hash(form.password.data)
                stmt = ibm_db.prepare(conn, 'UPDATE user SET pass_word=?
WHERE username=?')
                ibm_db.bind_param(stmt, 1, hashed_password1)
                ibm_db.bind_param(stmt, 2, form.username.data)
                user = ibm_db.execute(stmt)
                flash(f'Password changed successfully.', 'success')
                return redirect(url_for('home'))
            else:
                flash(f'Invalid password, Enter valid password.',
'danger')
                return redirect(url_for('update'))
        else:
            flash(f'Invalid user, Enter valid User.', 'danger')
            return redirect(url_for('update'))
    return render_template('update.html', form=form)

@app.route('/predict', methods=['POST'])
def predictSpecies():
    wdir = float(request.form['wdir'])
    mnt = float(request.form['mnt'])
```

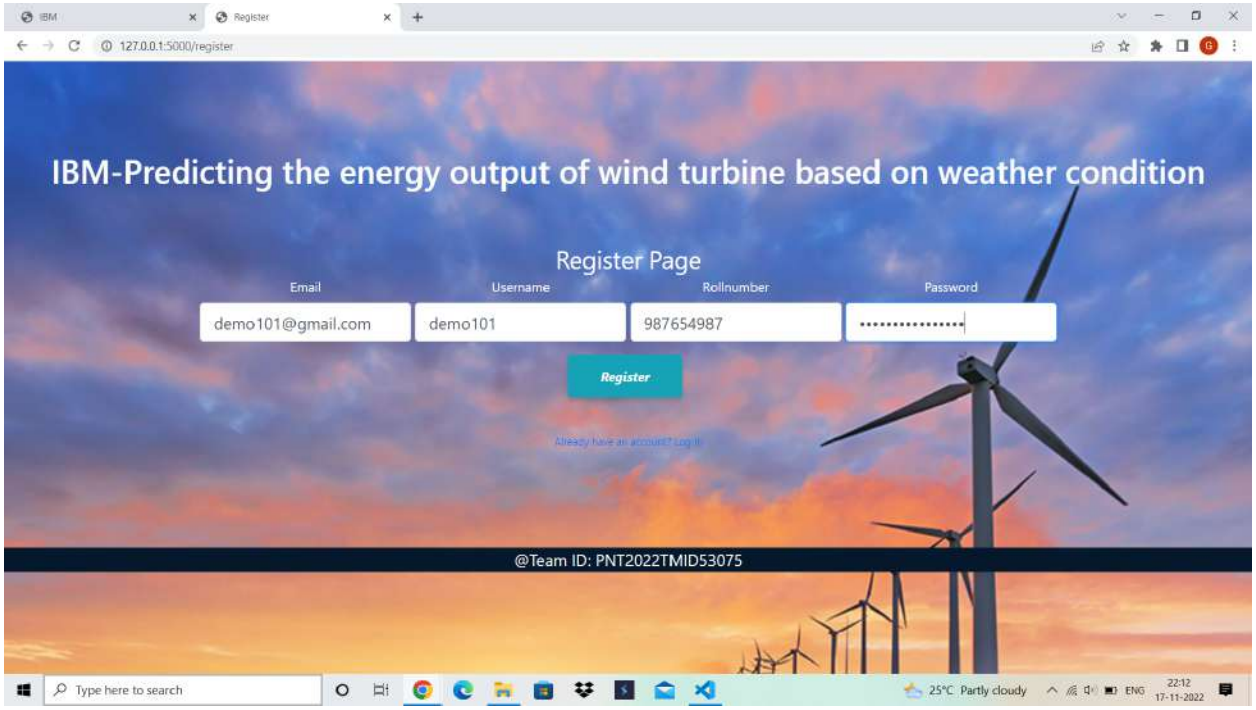
```
    dy = float(request.form['dy'])
    hr = float(request.form['hr'])
    mspd = float(request.form['mspd'])
    X = [[wdir, mnt, dy, hr, mspd]]

    ## NOTE: manually define and pass the array(s) of values to be scored
in the next line
    payload_scoring = {"input_data": [{"field": [[wdir, mnt, dy, hr,
mspd]], "values": X}]}

    response_scoring =
requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/4e141ca
c-848d-44c4-a8a0-b9bd034cbf7d/predictions?version=2022-11-18',
json=payload_scoring,headers={'Authorization': 'Bearer ' + mltoken})
    print(response_scoring)
    predictions = response_scoring.json()
    predict = predictions['predictions'][0]['values'][0][0]
    print("Final prediction : ",predict)

    return render_template('predict.html',predict=predict)

if __name__ == '__main__':
    app.run()
```

Output screenshots:

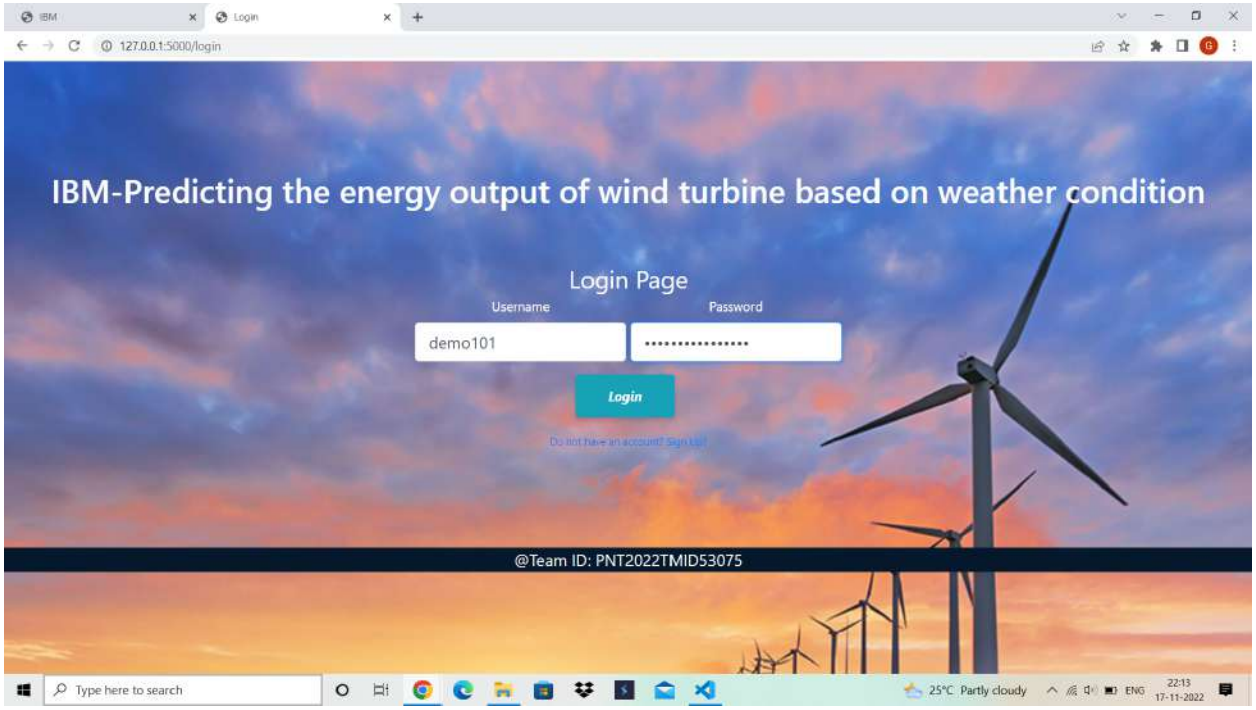
home page:



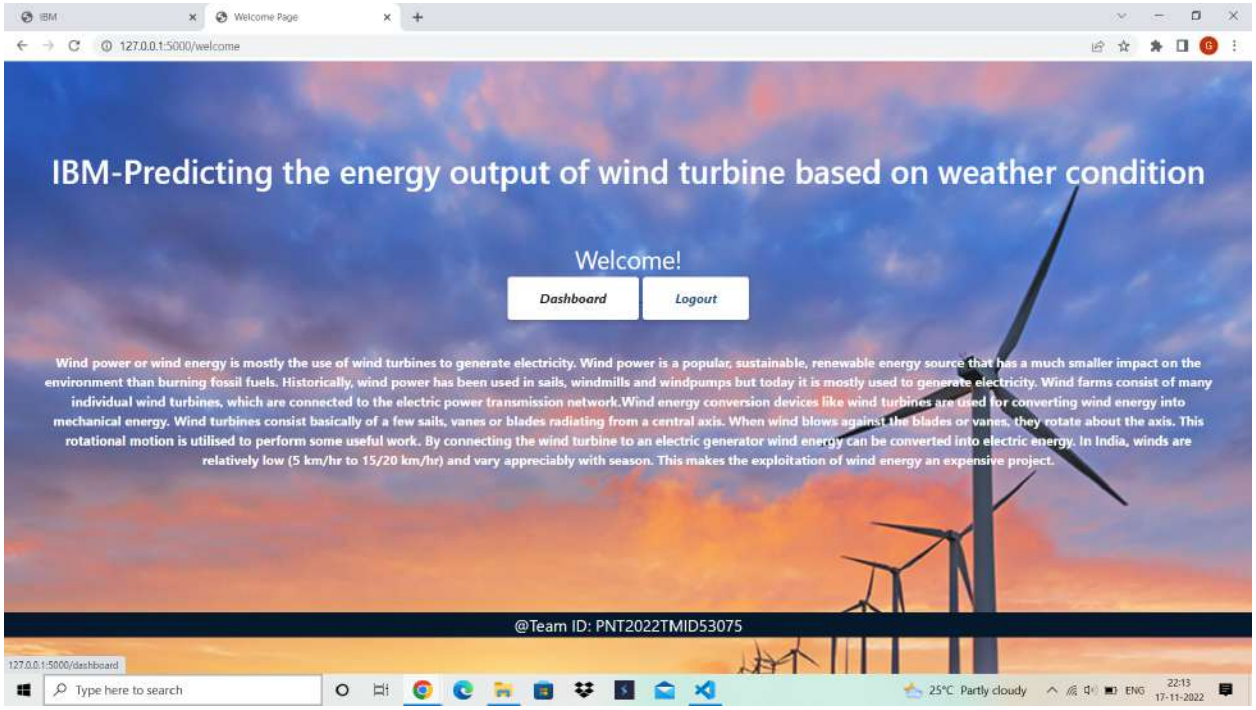
Registration page:



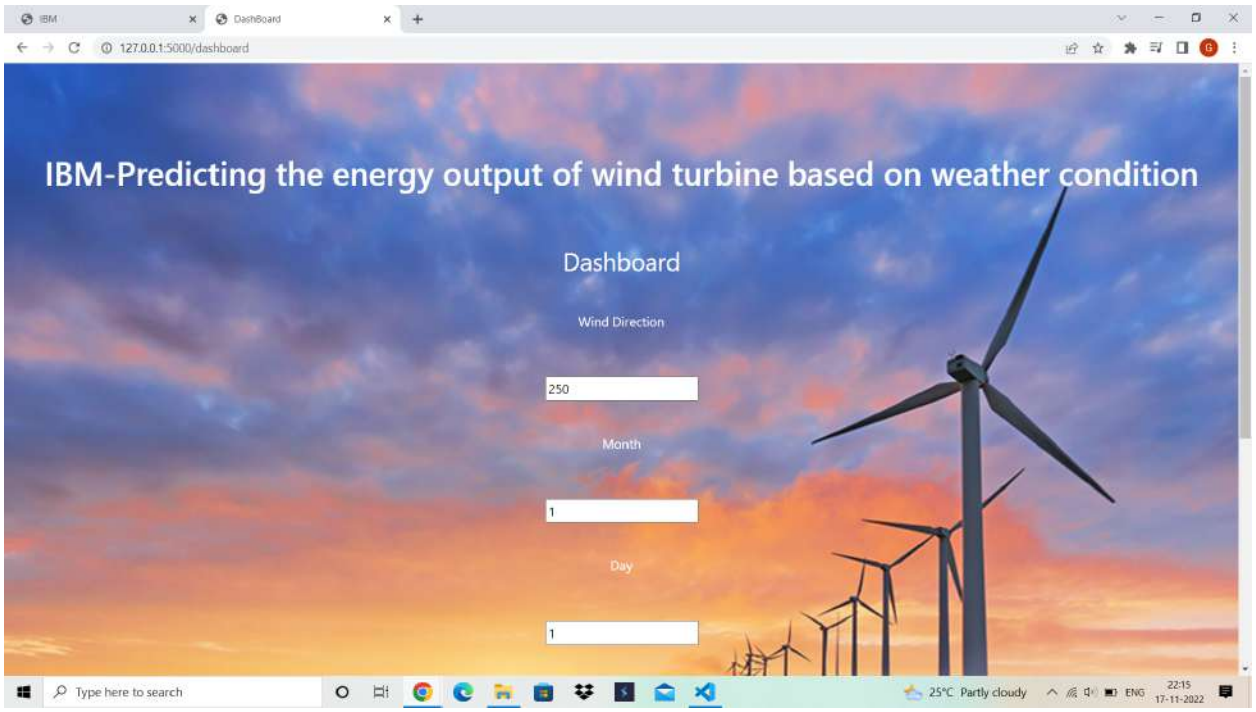
Login page:

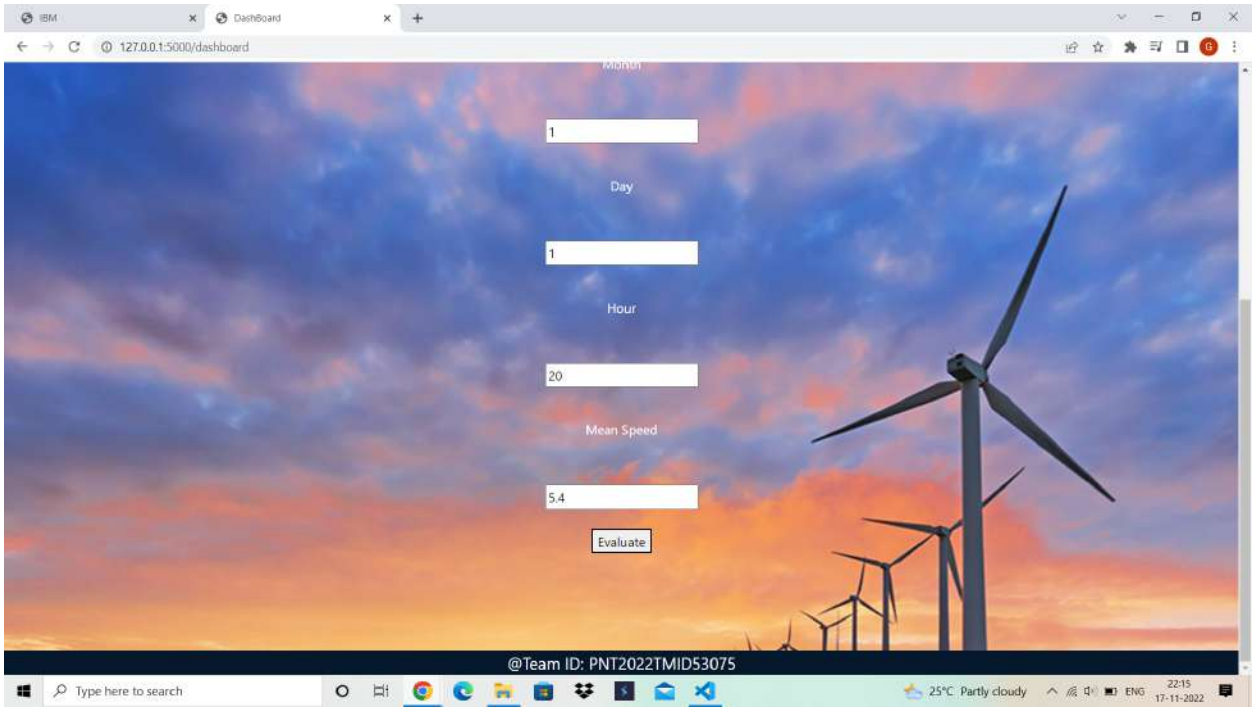


welcome page:

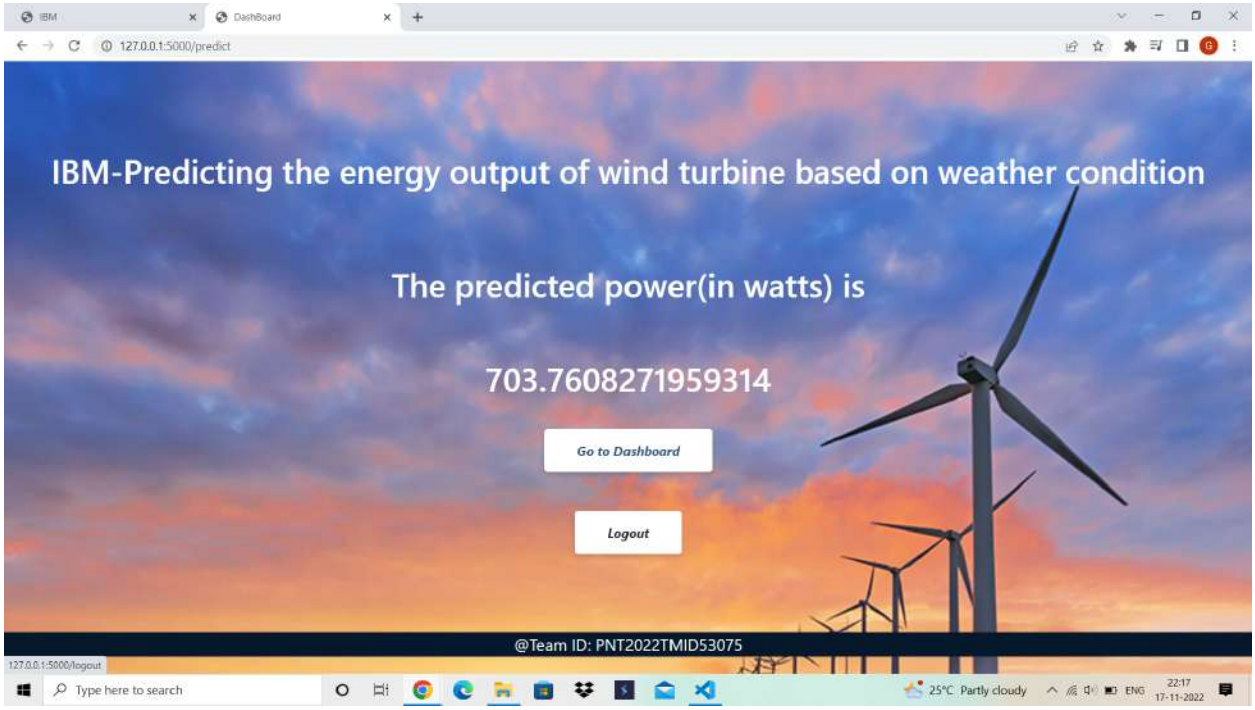


Dashboard page:

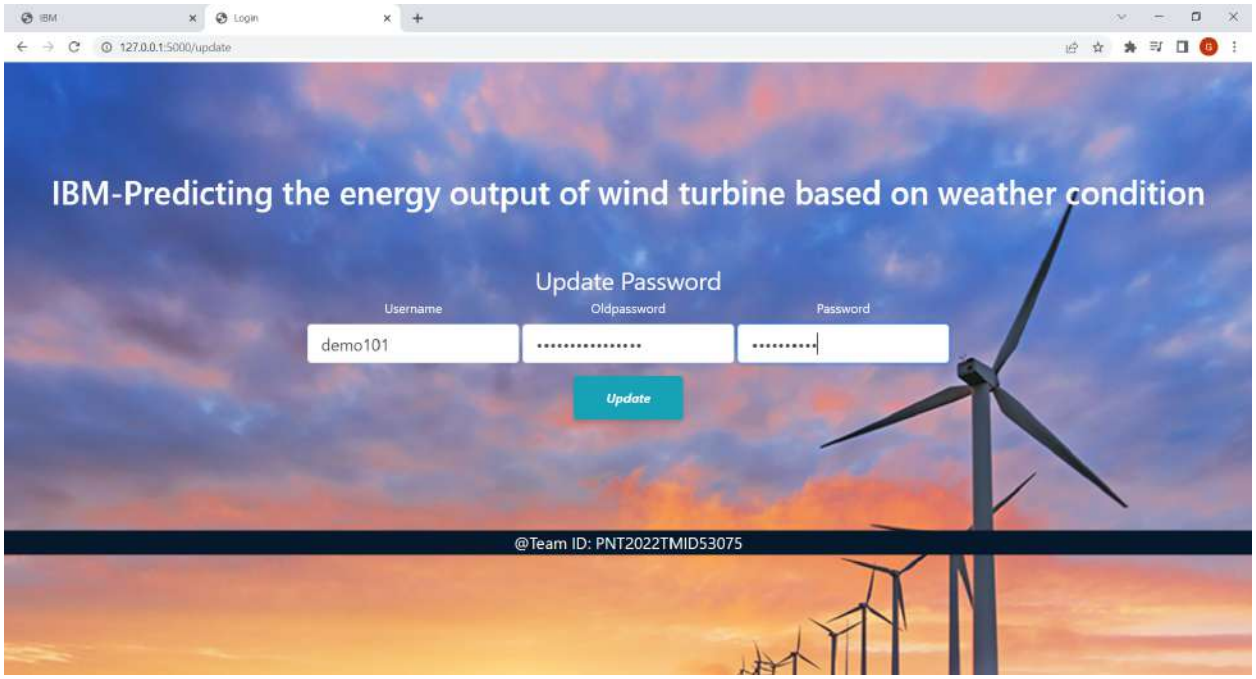


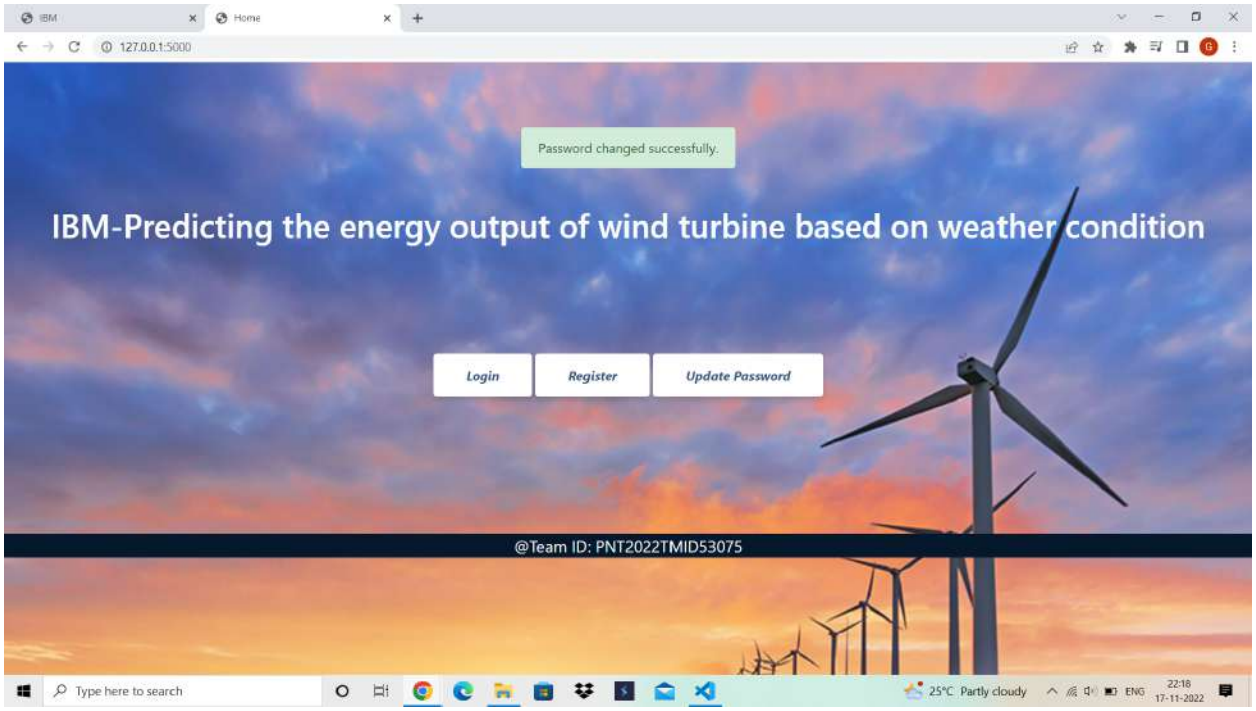


Prediction page:



Update password page:





GitHub & Project Demo Link

Project Demo Link:
https://drive.google.com/file/d/1HYb2RmR8xo7NoVQF9tu8fYA2K1Dufs7D/view?usp=sharing

Github Link:

https://github.com/IBM-EPBL/IBM-Project-17312-1659633887.git