KARNATAK LAW SOCIETY'S

GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG, BELAGAVI-590008

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

(APPROVED BY AICTE, NEW DELHI)

Department of Electrical and Electronics Engineering

Project Report on

SOLAR ENERGY PREDICTION USING MACHINE LEARNING

Submitted in partial fulfillment of the requirement for the award of the degree of

Bachelor of Engineering

in

Electrical and Electronics

Submitted by

Abhishek Khot	2GI19EE002
Archana Madalageri	2GI19EE007
Mihir Patel	2GI19EE022
Vinayak Biswagar	2GI19EE052

Guide

Prof. Rahul Suryavanshi

(Assistant Professor)

Co- Guide

Prof. Vinay Shetty

(Assistant Professor)

2022 – 2023

KARNATAK LAW SOCIETY'S

GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG, BELAGAVI-590008

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

(APPROVED BY AICTE, NEW DELHI)

Department of Electrical and Electronics Engineering

CERTIFICATE

Certified that the project entitled

SOLAR ENERGY PREDICTION USING MACHINE LEARNING

Carried out by

Mr. Abhishek Khot	USN 2GI19EE002
Ms. Archana Madalageri	USN 2GI19EE007
Mr. Mihir Patel	USN 2GI19EE022
Mr. Vinayak Biswagar	USN 2GI19EE052

students of KLS Gogte Institute of Technology, Belagavi, can be considered as a bonafide work for partial fulfillment for the award of **Bachelor of Engineering** in Electrical and Electronics Engineering of the Visvesvaraya Technological University, Belagavi during the year 2022- 2023

It is certified that all corrections/suggestions indicated have been incorporated in the report. The project report has been approved as it satisfies the academic requirements prescribed for the said Degree.

Guide	**Co-Guide**	**HOD**	**Principal**
Prof. Rahul G. Suryavanshi	Prof. Vinay Shetty	Dr. D. B. Kulkarni	Prof. D. A. Kulkarni

Date:

Final Viva-Voce

	Name of the examiners	**Date of Viva -voce**	**Signature**
1.			
2.			

DECLARATION BY THE STUDENT

We, *Abhishek Khot, Archana Madalageri, Mihir Patel, Vinayak Biswagar*, hereby declare that the project report entitled **"Solar Energy Prediction Using Machine Learning"** submitted by us to KLS Gogte Institute of Technology, Belagavi, in partial fulfillment of the Degree of **Bachelor of Engineering** in Department of **Electrical and Electronics Engineering** is a record of the project carried out at **KLS Gogte Institute of Technology**. This report is for the academic purpose.

We further declare that the report has not been submitted and will not be submitted, either in part or full, to any other institution and University for the award of any diploma or degree.

Name of the student	USN	Signature
Abhishek Khot	2GI19EE002	
Archana Madalageri	2GI19EE007	
Mihir Patel	2GI19EE022	
Vinayak Biswagar	2GI19EE057	

Place: Belgavi
Date:

ACKNOWLEDGEMENT

We would like to express our sincere gratitude and appreciation to all those who have contributed to the successful completion of the project on "Solar Energy Prediction using Machine Learning" Their support, guidance, and assistance were invaluable throughout the entire process. First and foremost, We would like to extend our deepest thanks to our project guide **Prof. Rahul Suryavanshi** and Co-guide **Prof. Vinay Shetty** for their continuous guidance, expertise, and valuable insights. Their mentorship played a crucial role in shaping the direction of this project and ensuring its successful execution Furthermore, we would like to thank our colleagues and peers for their valuable input, discussions, and collaboration during the course of this project. Their feedback and constructive criticism greatly enhanced the quality of the analysis and the final outcomes. We would also like to express our appreciation to our families and friends for their unwavering support, encouragement, and understanding throughout this endeavor. Their belief in us and their constant motivation were essential in keeping me focused and determined. Their pioneering work and published studies served as a solid foundation for this project. Once again, We extend our heartfelt gratitude to everyone who contributed to this project. Your assistance and support have been invaluable.

Abhishek Khot
Archana Madalageri
Mihir Patel
Vinayak Bisawgar

ABSTRACT

The increased competitiveness of solar photovoltaic (PV) panels as a renewable energy source has sparked a substantial surge in the installation of PV panels in recent years. This growth has been facilitated by advancements in technology, improved efficiency, and reduced costs associated with PV systems.

In parallel, the availability of vast amounts of data and the advancements in computational power have paved the way for the application of machine learning algorithms in the field of solar energy. Machine learning techniques, combined with time series models, offer promising solutions for accurately predicting solar PV energy output.

The accurate prediction of solar PV energy output is vital for various factors in the energy industry, including grid management, optimal resource allocation, energy pricing, and policy-making. Traditional methods for predicting solar PV energy output rely on physical models, but these often struggle to capture complex patterns and variations in real-world conditions.

In this study, we aim to address this challenge by employing various machine learning techniques to forecast solar PV energy output. We will compare and evaluate the performance of different models, such as Linear Regression and Decision Tree. By leveraging the wealth of available data and advanced computational capabilities, we seek to improve the accuracy and reliability of solar PV energy output predictions.

The outcomes of this study will contribute to the growing body of research on renewable energy forecasting and provide valuable insights for stakeholders in the energy industry. By harnessing the power of machine learning and time series models, we can make more informed decisions and drive the widespread adoption of solar PV systems as a sustainable energy source.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

Abbreviation	Description
PV	Photovoltaic
EDA	Exploratory Data Analysis
CI	Continuous Integration
CD	Continuous Deployment
SVMs	Support Vector Machines
ML	Machine Learning
NWS	National Weather Service
3D	Three-Dimensional
OLS	Ordinary Least Squares
RSS	Residual Sum of Squares
API	Application Protocol Interface
AWS	Amazon Web Services
EC2	Elastic Compute Cloud
ECR	Elastic Container Registry
EBS	Elastic Block Store
IAM	Identity and Access Management
UI	User Interface
UX	User Experience
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
DOM	Document Object Model
AJAX	Asynchronous JavaScript and XML

CHAPTER 1

1.1 INTRODUCTION

Imagine a college campus with a visionary solar energy plan. Vast arrays of solar panels adorn rooftops, capturing the abundant sunlight and generating clean, renewable electricity, as the campus strives to maximize its use of solar energy. Accurate solar power prediction is essential for efficient energy management, ensuring a reliable and stable power supply for the campus. Furthermore, the seamless integration of solar power into the existing grid infrastructure is crucial to optimize the utilization of renewable energy and minimize wastage. This is where our project comes into play, leveraging machine learning techniques to address this challenge and optimize the integration of solar power into the grid.

By harnessing the power of machine learning, we aim to develop a predictive model capable of accurately forecasting solar power generation. This model will consider various environmental factors such as solar irradiation, and temperature which influence the amount of energy generated by solar panels. Through an in-depth analysis of historical solar power generation data and environmental variables, we will train and fine-tune the model to provide accurate predictions of solar power output.

The integration of machine learning techniques into the prediction of solar power generation offers numerous benefits and implications for the local neighborhood. Firstly, it enables its energy usage by aligning power generation with demand patterns. This leads to cost savings, reduced reliance on non-renewable energy sources, and a more sustainable energy system.

Moreover, accurate solar power prediction facilitates the seamless integration of solar energy into the existing power grid. Providing reliable forecasts can ensure a smooth transition between solar power generation and grid supply, avoiding disruptions and maintaining a stable electricity supply. This integration paves the way for a more sustainable grid system, as excess solar energy can be fed back into the grid, reducing the overall carbon footprint and promoting a cleaner energy mix.

Furthermore, our project serves as a stepping stone towards a sustainable future, not only for the local surroundings but also as a model for other institutions and communities. By showcasing the successful integration of solar power generation through machine learning techniques, we inspire and encourage others to adopt renewable energy solutions, driving the global transition towards a low-carbon economy.

1.2 OBJECTIVES

1. Data collection: Collect reliable data from the solar PV plant, including parameters like solar irradiance, temperature, and power output.

2. Data wrangling: Clean, transform, and prepare the collected data for analysis by handling missing values, duplicates, and standardizing formats.

3. Exploratory Data Analysis (EDA): Gain insights from the data by visualizing patterns, trends, and relationships between variables.

4. Output generation prediction: Use appropriate machine learning techniques to build a model that predicts the solar PV plant's output generation based on input parameters.

5. CI-CD pipeline development: Establish a CI-CD pipeline for automated deployment, testing, and updates of the developed model or application.

6. Front-end and back-end setup: Develop a user-friendly front-end interface and a robust back-end system for real-time verification and monitoring of the solar PV plant's output generation.

1.3 METHODOLOGY

1.3.1 Machine Learning Techniques

Machine learning is a data analytics technique that teaches computers to do what comes naturally to humans and animals: learn from experience. Machine learning algorithms use computational methods to directly "learn" from data without relying on a predetermined equation as a model.

1.3.2 How does machine learning work?

Machine learning uses two techniques: supervised learning, which trains a model on known input and output data to predict future outputs, and unsupervised learning, which uses hidden patterns or internal structures in the input data.

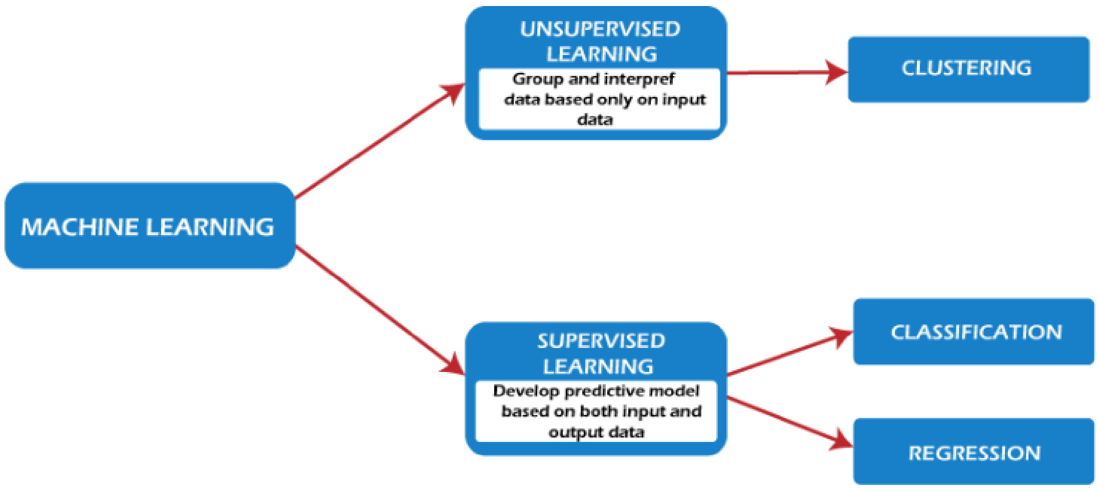

MACHINE LEARNING
UNSUPERVISED LEARNING
Group and interpref data based only on input data
CLUSTERING
SUPERVISED LEARNING
Develop predictive model based on both input and output data
CLASSIFICATION
REGRESSION

Figure 1.1: Machine learning types

1.3.2.1 Supervised learning

Supervised machine learning creates a model that makes predictions based on evidence in the presence of uncertainty. A supervised learning algorithm takes a known set of input data and known responses to the data (output) and trains a model to generate reasonable predictions for the response to the new data. Use supervised learning if you have known data for the output you are trying to estimate. Supervised learning uses classification and regression techniques to develop machine learning models. Classification models classify the input data. Classification techniques predict discrete responses. Common algorithms for performing classification include support vector machines (SVMs), boosted and bagged decision trees, k-nearest neighbors, Naive Bayes, discriminant analysis, logistic regression, and neural networks.

1.3.2.2 Unsupervised Learning

Detects hidden patterns or internal structures in unsupervised learning data. It is used to eliminate datasets containing input data without labeled responses. Clustering is a common unsupervised learning technique. It is used for exploratory data analysis to find hidden patterns and clusters in the data. Applications for cluster analysis include gene sequence analysis, market research, and commodity identification.

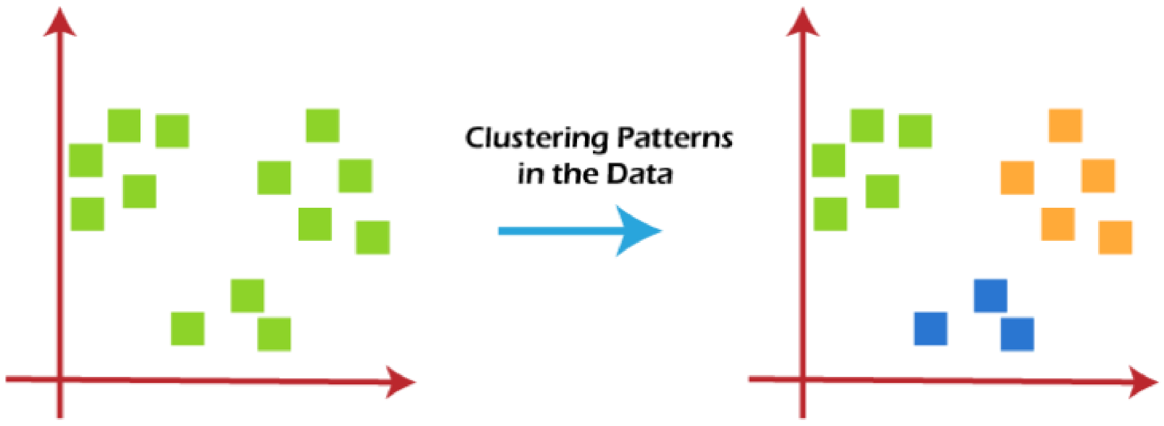


Figure1.2: Clustering patterns

Ten methods are described and it is a foundation you can build on to improve your machine learning knowledge and skills:

i. Regression
ii. Classification
iii. Clustering
iv. Dimensionality Reduction
v. Ensemble Methods
vi. Neural Nets and Deep Learning
vii. Transfer Learning
viii. Reinforcement Learning
ix. Natural Language Processing
x. Word Embedding's

Let's differentiate between two general categories of machine learning: supervised and unsupervised. We apply supervised ML techniques when we have a piece of data that we want to predict or interpret. We use the previous and output data to predict the output based on the new input.

1.3.3 Prediction of the Flow of Solar Radiation

Our project consisted of three parts data preprocessing, model building, and model prediction. The data preprocessing involved four steps: data quality control, dataset partitioning, data scaling, and variable selection. Among them, data quality control, dataset partitioning, and data scaling are described in Section "Study Area and Datasets," and variable selection is described in section.

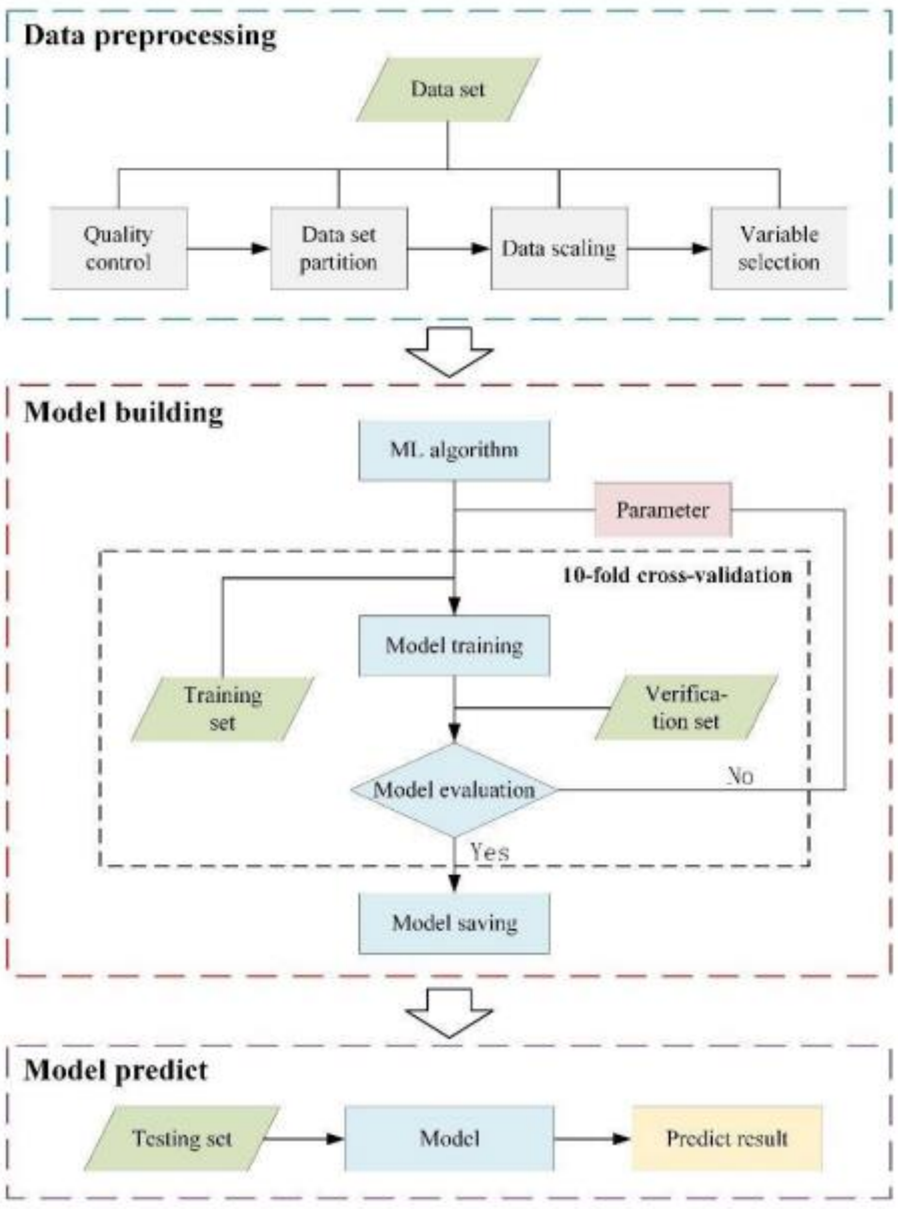


Figure 1.3: Flow chart of the machine learning models used to estimate solar radiation.

The process for the flow of solar radiation are:

i. Data collection and data preprocessing.
ii. Choose a machine learning algorithm from the two algorithms to predict solar radiation.
iii. Compare solar radiation predictive ability based on different parameters.
iv. If the best predictive ability is achieved, save the model.

v. Return to step (2) and choose another machine learning algorithm until all two algorithms have been subjected to machine learning model building.

vi. Input the preprocessing dataset (we prepared datasets on two timescales— daily and monthly—to estimate the solar radiation predictive performance of the two machine learning models) and use the two saved machine learning models to predict solar radiation and obtain the predicted results.

vii. Save predicted results and analyze.

1.3.4 Data Gathering for Solar Energy Prediction

In order to design for solar energy prediction, it is important to analyze the main factors that affect solar energy generation. These factors will have to be considered by the model as inputs, which will need to be adequately tailored for each particular situation we wish to predict. Aside from the energy carried by solar rays, the amount of electric solar energy generated also varies depending on various parameters and conditions that vary depending on the moment of the day. Among these factors, the most notable ones are:

i. Solar azimuth angle
ii. Air temperature
iii. Wind speed
iv. Performance of the photovoltaic (PV) panel
v. Size of the generating installation

In order to provide a better overview of all factors involved in energy generation, we present an overview

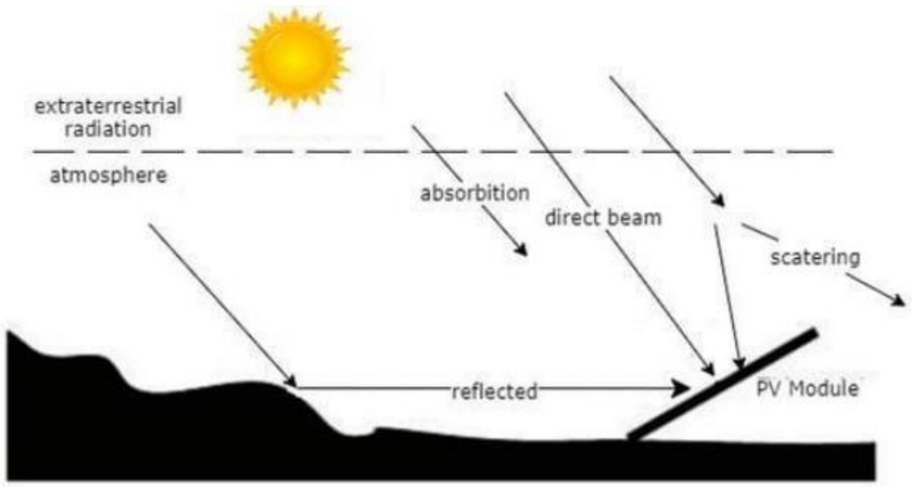


Figure 1.4: Solar radiation in Earth’s atmosphere.

Once we have summarized the factors that affect solar energy generation, in the next section we present how we make use of these factors to predict solar energy output taking into account 9 the characteristics of the solar panel installation as well as its location and other particular conditions. Hence the data can be collected using the weather station along with the weather monitoring system

1.3.5 Data Cleaning

Data cleaning is one of the important parts of machine learning. It plays a significant part in building a model. If we have a well-cleaned dataset, there are chances that we can get achieve good results with simple algorithms also, which can prove very beneficial at times especially in terms of computation when the dataset size is large. Obviously, different types of data will require different types of cleaning. However, this systematic approach can always serve as a good starting point.

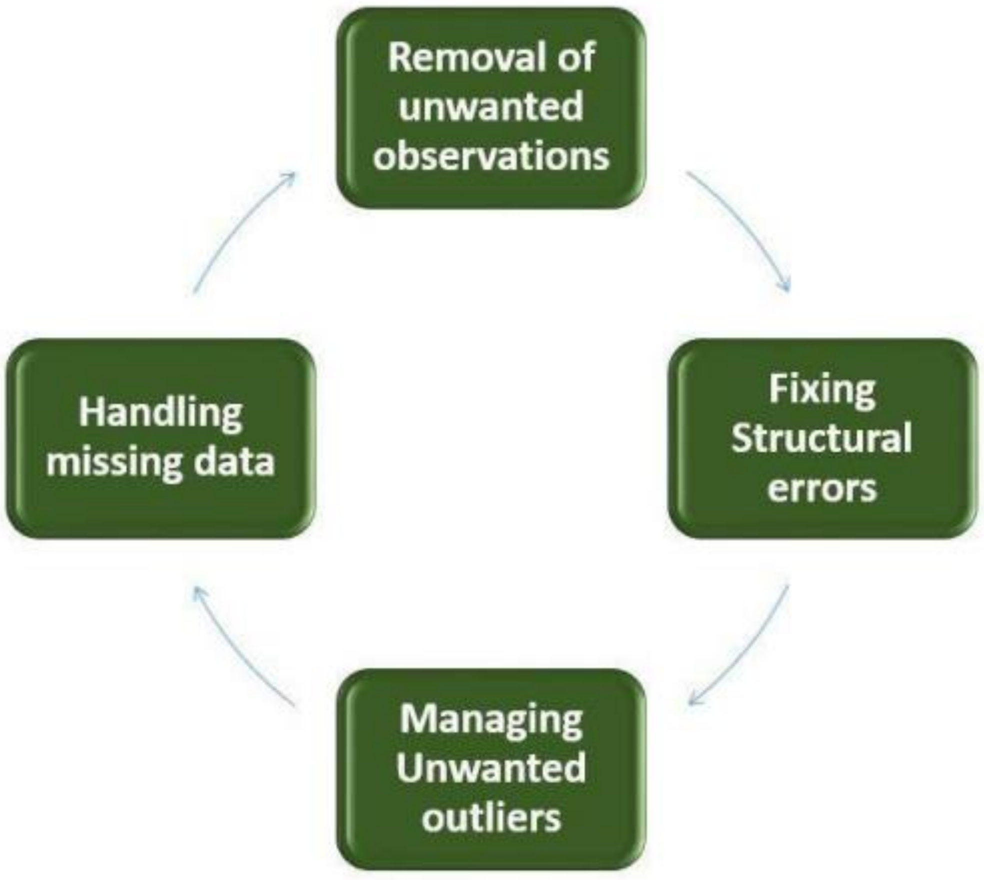


Figure 1.5: Data cleaning process

Steps involved in Data Cleaning:

i. Removal of unwanted observations

   This includes deleting duplicate/ redundant or irrelevant values from your dataset. Redundant observations alter the efficiency by a great extent as the data repeats and may add towards the correct side or towards the incorrect side, thereby producing unfaithful results. Irrelevant observations are any type of data that is of no use to us and can be removed directly.

ii. Fixing Structural errors

   The errors that arise during measurement, transfer of data, or other similar situations are called structural errors. Structural errors include typos in the name of features, the same attribute with a different name, mislabeled classes, i.e. separate classes that should really be the same, or inconsistent capitalization

iii. Managing unwanted outliers

   Outliers can cause problems with certain types of models. Sometimes, removing them improves performance, sometimes not. So, one must have a good reason to remove the outlier, such as suspicious measurements that are unlikely to be part of real data.

iv. Handling missing data

   Missing data is a deceptively tricky issue in machine learning. We cannot just ignore or remove the missing observation. They must be handled carefully as they can be an indication of something important. The two most common ways to deal with missing data are

   - Dropping observations with missing values.
   - Imputing the missing values from past observations.

   So, missing data is always an informative and an indication of something important. And we must be aware of our algorithm of missing data by flagging it. By using this technique of flagging and filling, you are essentially allowing the algorithm to estimate the optimal constant for messiness, instead of just filling it in with the mean.

CHAPTER 2

2.1 LITERATURE REVIEW

1. Aneela Zameer, Farah Shahid2, Mudasser Afzal, Muhammad Hassan, “Intelligent forecast models for daily solar energy prediction”, Research gate October 2020.
   The above paper shows the exploration of solar energy characteristics and comparison of linear and non-linear machine learning methodologies to improve their generalization ability for better adaption and reduced prediction error. For this purpose, various machine learning based regression models such as multivariate linear regression, ridge regression, and lasso regression are implemented for daily solar energy forecast

2. K. Anuradha1, Deekshitha Erlapally , G. Karuna , V. Srilakshmi , K. Adilakshmi, “Analysis of Solar Power Generation Forecasting Using Machine Learning Techniques”, E3S Web of Conferences 309, 01163 ICMED 2021.
   The above paper concentrates on solar power generation using photovoltaic (PV) systems all over the world. Because the output power of PV systems is alternating and highly dependent on environmental circumstances, solar power sources are unpredictable in nature. The impacts of various environmental conditions on the output of a PV system are discussed. Machine Learning (ML) algorithms have shown great results in time series forecasting and so can be used to anticipate power with weather conditions as model inputs. The use of multiple machine learning, Deep learning and artificial neural network techniques to perform solar power forecasting.

3. Bouchaib Zazoum, “Solar photovoltaic power prediction using different machine learning methods”, 2021 8th International Conference on Power and Energy Systems Engineering (CPESE 2021).
   The above paper concentrates on the problem of automatically generating models that accurately predict renewable generation based on National Weather Service forecasts (NWS). Using historical NWS forecast data and data generated by solar panels, we experiment with a variety of machine learning techniques to develop prediction models. For developing prediction models, a variety of regression

algorithms are tested, including linear least squares and support vector machines with various kernel functions. In these tests it shows that a machine learning approach can correctly anticipate solar power radiations.

4. C. Vennila, Anita Titus, T. Sri Sudha, U. Sreenivasulu, N. Pandu Ranga Reddy, K. Jamal, Dayadi Lakshmaiah, P. Jagadeesh and Assefa Belay, "Forecasting Solar Energy Production Using Machine Learning", Hindawi International Journal of Photo energy Volume 2022, Article ID 7797488, 7 pages https://doi.org/10.1155/2022/7797488
   The above paper concentrates on large range of datasets and time steps, prediction ranges, settings, and performance measurements, a single machine learning model cannot improve forecasting performance on a single dataset or time step. With the use of hybrid machine learning algorithms, projections for solar power have also been enhanced. In order to effectively predict the availability of solar power radiations, significant attention has lately been drawn to support vector machines (SVMs) and deep learning algorithms.

5. Mohsen El-hawary, "Short-Term Solar Energy Forecasting Using Machine Learning Techniques", IEEE Conference 2019.
   This paper provides a comprehensive overview of machine learning techniques employed for short-term solar energy forecasting. It discusses the challenges associated with accurate solar energy prediction, such as data availability and model selection. The authors explore various machine learning algorithms, including artificial neural networks, support vector regression, and decision trees, and evaluate their performance in short-term solar energy forecasting. The paper highlights the importance of feature selection, data preprocessing, and model optimization for improving prediction accuracy. It also discusses the potential of hybrid models and ensemble techniques in enhancing forecasting results.

6. J. Lee, "Machine learning methods for solar power forecasting" IEEE PES Conference, 2018.
   This paper presents a comprehensive survey of machine learning methods employed in solar power forecasting. It examines various algorithms, such as artificial neural networks, support vector machines, and random forests, and evaluates their

effectiveness in predicting solar energy generation. The study discusses the influence of different meteorological and environmental factors on solar power forecasting accuracy. It also addresses issues related to data preprocessing, model training, and performance evaluation. The paper concludes with recommendations for selecting appropriate machine learning techniques based on the specific characteristics and requirements of the solar energy prediction task.

7. S.Ravikumar, "Forecasting solar power generation using hybrid intelligent models" IEEE Conference, 2017.
   This research paper focuses on the application of hybrid intelligent models for solar power generation forecasting. It investigates the performance of various hybrid models that combine artificial neural networks with other techniques such as fuzzy logic, genetic algorithms, and particle swarm optimization. The study compares and analyzes the accuracy and robustness of different hybrid models in predicting solar power generation. It discusses the advantages and limitations of each approach and provides insights into selecting appropriate hybrid intelligent models for solar energy prediction tasks.

8. Hossain, "Solar energy prediction using machine learning techniques", a systematic literature survey 2020.
   This systematic literature paper explores the use of machine learning techniques for solar energy prediction. It provides an overview of the methodologies, datasets, and evaluation metrics employed in existing studies. The paper summarizes the performance and limitations of different machine learning algorithms, such as artificial neural networks, support vector machines, and decision trees. It also discusses the challenges of data collection, feature selection, and model training in solar energy prediction. The survey highlights the need for standardized evaluation frameworks and datasets to facilitate further advancements in the field.

9. S. Zhang, "Solar power forecasting using machine learning algorithms", a comprehensive paper 2020.
   This comprehensive paper presents an extensive analysis of machine learning algorithms used in solar power forecasting. It discusses the advantages and disadvantages of different techniques, including decision trees, random forests, and

support vector machines. The study highlights the importance of feature selection, data preprocessing, and ensemble methods for improving the accuracy of solar power prediction. It also explores the impact of various weather variables and temporal factors on forecasting performance. The paper concludes with suggestions for future research directions and potential applications of machine learning in solar energy prediction.

10. F. Qiang, “Solar power forecasting using machine learning techniques” IEEE Conference, 2021.
    This systematic paper provides an in-depth analysis of machine learning techniques employed in solar power forecasting. It evaluates the performance of different algorithms, including artificial neural networks, k-nearest neighbors, and ensemble methods, in predicting solar energy generation. The study explores the impact of weather variables, temporal factors, and data preprocessing techniques on prediction accuracy. It also discusses the challenges associated with model selection, feature engineering, and uncertainty quantification in solar energy forecasting. The paper highlights the potential of emerging techniques, such as deep learning and hybrid models, and suggests future research directions in the field.

CHAPTER 3

3.1 IPR PRIOR ART SEARCH

1. IPR 1: Machine learning based solar power generation prediction apparatus and method that does not use future meteorological forecast data

   Patent number: KR102092860B1

   Application number: KR1020190007968A

   Applicant Name: Kwanho Kim, Donghoon Lee

   Abstract:

   Disclosed are a machine learning based photovoltaic power generation value prediction device which does not use future weather forecast data and a method thereof. The present invention suggests a technique of predicting not a photovoltaic power generation amount in a peak time zone by using the future weather forecast data with uncertainty, but a photovoltaic power generation amount in the peak time zone of the day based on weather data measured in a time zone before the peak time zone of the day. Therefore, the present invention can support a manager of a photovoltaic power generation plant to more accurately predict the photovoltaic power generation amount in the peak time zone.

   Claims:

i. A set of different types of weather data measured at predetermined unit time intervals in a time period prior to a present peak time period of the day and solar power generation data measured at the unit time intervals in the peak time period for each day. A measurement table holding unit that stores and maintains the divided and recorded data table;

ii. A set of weather data classified and recorded in the data table for each day and solar power generation data are randomly classified into a training set for machine learning and a test set for verification of machine learning results, but the number of training sets is the test. A set classifying unit that classifies the number of sets to exceed;

iii. A first neural network-based model, a second neural network-based model, and a third neural network-based model by designating a set of weather data classified as the training set as an input and designating a solar power generation data of a corresponding date as an output. The first neural network-based prediction model, the second neural network-based prediction model, and the second neural network-based prediction model for predicting photovoltaic power generation in the peak time zone by performing supervised learning-based machine learning for generating A prediction model generator for generating a prediction model based on the third neural network, respectively

iv. Weights on the first neural network-based prediction model, the second neural network-based prediction model, and the third neural network-based prediction model based on a ratio between the first prediction accuracy, the second prediction accuracy, and the third prediction accuracy -The sum of the weights allocated to each prediction model is 1-a weight allocator assigning;

v. A display unit that displays on the screen solar power generation data of the unit time interval in the peak time zone of the first date calculated by the prediction unit as prediction data corresponding to the prediction command

2. IPR 2: solar power forecasting
   Patent Number: WO2017193172A1
   Application Number: PCT/AU2017/050433
   Applicant Name: Sam West, Ryan Lagerstrom, Changming Sun, Li RONGXIN, Joel WONG

   Abstract:
   A method for determining a level of solar radiation at a point of interest (POI). Multiple sky images are captured by a distributed network of digital cameras. Sun location parameters are determined. A three-dimensional (3D) sky model is generated based on the sky images. Generating the 3D sky model includes generating 3D object data based on the sky images to model one or more objects in a region of sky, and generating position data to model a position of the one or more objects in the region of sky. A level of solar radiation at the POI is determined based on the position data and 3D object data of the 3D sky model and the sun location parameters.

Claims:

i. A method for determining a level of solar radiation at a point of interest (POI), the method comprising: capturing multiple sky images by a distributed network of digital cameras; determining sun location parameters; generating a three-dimensional (3D) sky model based on the sky images, wherein generating the 3D sky model comprises generating 3D object data based on the sky images to model one or more objects in a region of sky and generating position data to model a position of the one or more objects in the region of sky; determining a level of solar radiation at the POI based on the position data and 3D object data of the 3D sky model and the sun location parameters.
ii. The method of claim 1, wherein determining a level of solar radiation at a POI comprises: computing a light path between the POI and a location of the sun based on the sun location parameters; determining if the one or more objects modelled in the 3D sky model occludes the light path; and determining the level of solar radiation at the POI based on a level of occlusion of the light path by the one or more objects.
iii. The method of claim 1 or 2, wherein the 3D object data comprises information regarding one or more of the following for each of the one or more objects: a continuous 3D surface mesh of the object; a continuous 3D volumetric mesh of the object; a thickness of the object; a 3D shape of the object; a volume of the object; an optical depth of the object.
iv. The method of any preceding claim, wherein the distributed network of digital cameras comprises multiple pairs or sets of digital cameras capturing sky images of the region of the sky.

3. IPR 3: Solar irradiation modeling and forecasting using community based terrestrial sky imaging

Patent Number: US10692013B2

Application Number: US15/175,620

Applicant Name: Mani Abedini, Rahil Garnavi, Timothy M. Lynar, Christopher I. E. Mesiku, John M. Wagner

Abstract:

Solar irradiation may be predicted based on input terrestrial sky images comprising cloud images, the terrestrial sky images taken from a plurality of geographic locations by a plurality of devices; for example, wherein the terrestrial sky images are crowd sourced from the plurality of devices. A model may be generated that predicts solar irradiation in a geographic area based on the input terrestrial sky images and the geographic locations from where the terrestrial sky images were taken. A signal representing the solar irradiation predicted by the model is output.

Claims:

i. The method of claim 1, wherein the solar irradiation and another solar irradiation are combined based on weights to provide a weighted solar irradiation combination.
ii. The method of claim 2, further comprising determining solar power based on the weighted solar irradiation combination and solar panel characteristics associated with a solar panel in the geographic area.
iii. The method of claim 1, wherein the terrestrial sky images are crowd sourced from a community network including at least the plurality of devices.
iv. The method of claim 1, wherein the method is performed at a central service platform receiving the terrestrial sky images uploaded from the plurality of devices.
v. The method of claim 1, wherein the method is performed by one of the plurality of devices sharing the terrestrial sky images from the plurality of devices.

4. IPR 4: Apparatus and method for predicting solar irradiance variation
Patent Number: US20130152997A1
Application Number: US13/329,450
Applicant Name: Yi Yao, Peter Tu, Ming-Ching Chang, Li Guan, Yan Tong

Abstract:

An apparatus and method, as may be used for predicting solar irradiance variation, are provided. The apparatus may include a solar irradiance predictor processor configured to process a sequence of images (e.g., sky images). The irradiance predictor processor may include a cloud classifier module configured to classify respective pixels of an image of a cloud to indicate a solar irradiance-passing characteristic of at least a portion of the cloud. A cloud motion predictor may be configured to predict motion of the cloud over a time horizon. An event predictor may be configured to predict over the time horizon occurrence of a solar obscuration event. The prediction of the solar obscuration event may be based on the predicted motion of the cloud. The event predictor may include an irradiance variation prediction for the obscuration event based on the solar irradiance-passing characteristic of the cloud.

Claims:

i. Apparatus comprising a solar irradiance predictor processor configured to process a sequence of images, the irradiance predictor processor comprising a cloud classifier module configured to classify respective pixels of an image of at least one cloud to indicate a solar irradiance-passing characteristic of at least a portion of said at least one cloud a cloud motion predictor configured to predict motion of said at least one cloud over a time horizon and an event predictor configured to predict over the time horizon occurrence of a solar obscuration event, wherein the prediction of the solar obscuration event is based on the predicted motion of said at least one cloud, wherein the event predictor includes an irradiance variation prediction for the obscuration event based on the solar irradiance-passing characteristic of said at least one cloud.
ii. The apparatus of claim 1, wherein said irradiance predictor processor further comprises an image segmentation module configured to identify at least a cloud segment and/or a sky segment for the image of said at least one cloud.
iii. The apparatus of claim 1, wherein the motion prediction from the cloud motion predictor is based on a flow motion model.
iv. The apparatus of claim 1, wherein the event predictor is configured to predict said irradiance variation by back projecting a solar disk image based on an average speed vector computed from the predicted motion of said at least one cloud.
v. The apparatus of claim 1, further comprising at least one image acquisition device arranged to acquire the sequence of images.

5. IPR 5: The Forecasting Methodology and device of photo-thermal power generation power

   Patent Number: CN108171363A

   Application Number: CN201711328455.6A

   Applicant Name: Dong Chenhui, Zuo Liye, Han Zifen

Abstract:

An embodiment of the present invention provides a kind of Forecasting Methodologies and device of photo-thermal power generation power. The Forecasting Methodology of the photo-thermal power generation power includes According to meteorological data and heliostat control parameter, the temperature of heat transfer medium is predicted According to the temperature of the heat transfer medium and the flow of the heat transfer medium, the generated output of solar-thermal generating system is predicted. The Forecasting Methodology and device of photo-thermal power generation power provided by the invention for photo thermal conversion link, according to meteorological data and heliostat control parameter, predict the temperature of heat transfer medium, realize the prediction for the thermal energy that photo thermal conversion link is converted to light energy. For turbine power generation link, according to the temperature of heat transfer medium and the flow of heat transfer medium, the generated output of solar-thermal generating system is predicted, realize the prediction of electric energy that turbine power generation link is converted to thermal energy. To sum up, the Forecasting Methodology and device of photo-thermal power generation power provided by the invention can accurately predict the generated output of solar-thermal generating system.

Claims:

i. A kind of Forecasting Methodology of photo-thermal power generation power, which is characterized in that including：
ii. According to meteorological data and heliostat control parameter, the temperature of heat transfer medium is predicted；
iii. According to the temperature of the heat transfer medium and the flow of the heat transfer medium, the generated output of solar-thermal generating system is predicted.
iv. Forecasting Methodology according to claim 1, which is characterized in that described according to meteorological data and heliostat control ginseng Number predicts the temperature of heat transfer medium, including：
v. According to the meteorological data and the heliostat control parameter, based on photo-thermal numerical model, the heat transfer medium is predicted temperature.

CHAPTER 4

4.1 PROBLEM DEFINATION AND ALGORITHMS

4.1.1 Problem Definition

The problem addressed in this project is the accurate prediction of solar energy generation using machine learning techniques. The goal is to develop models that can forecast solar energy production within specific timeframes by leveraging factors such as weather conditions and historical data. Improving the accuracy of solar energy predictions is crucial for optimizing energy utilization, making informed decisions in renewable energy integration, and effectively managing the grid.

4.1.2 Initial Design using Design Thinking approach

To address the challenge of accurately predicting solar energy generation, the project will employ machine learning techniques.

The initial layout of the work was as follows

Step 1: Data Collection: Gather relevant data, including historical solar energy generation data, weather conditions (such as solar irradiance, temperature, and cloud cover), and other influential factors (geographical location, time of day, etc.).

Step 2: Data Pre-processing: Clean and pre-process the collected data, handling missing values, normalizing variables, and performing feature engineering if necessary. This step ensures the data is in a suitable format for machine learning algorithms.

Step 3: Model Selection: Explore and select appropriate machine learning algorithms, such as artificial neural networks, support vector machines, or decision trees, based on the nature of the problem and available data.

Step 4: Model Deployment: Once a suitable model is selected, deploy it to predict solar energy generation based on real-time or future input data. Continuously monitor and validate the model's performance to ensure its accuracy and effectiveness.

During the course of the project, several major problems may arise when predicting solar energy generation using machine learning. One common challenge is the availability and accuracy of data. Insufficient or inaccurate data can have a detrimental impact on the performance of machine learning models. To address this, it is crucial to collect data from reliable sources and ensure its accuracy through proper calibration. Additionally, employing data preprocessing techniques, such as handling missing values and outlier detection, can help mitigate data quality issues.

Another significant challenge is feature selection. With a potentially large set of predictors, it can be challenging to determine which features are the most relevant for accurate solar energy predictions. To tackle this, feature selection techniques such as correlation analysis can be applied. These techniques help identify the most informative features that contribute to the accurate prediction of solar energy generation.

By recognizing and addressing these challenges, the project can overcome obstacles that may hinder the accurate prediction of solar energy generation using machine learning techniques. These solutions ensure the availability of high-quality data, appropriate feature selection, and well-regularized models, contributing to improved predictions and informed decision-making in the field of solar energy generation.

4.1.3 Linear Regression Model

Linear regression is a widely used supervised learning algorithm in machine learning and statistics. It is used to model the relationship between a dependent variable (target variable) and one or more independent variables (input features). The goal of linear regression is to find the best-fitting linear relationship between the input variables and the target variable.

The linear regression model assumes a linear relationship between the input features and the target variable. It represents this relationship with a straight line defined by a slope and an intercept. The slope determines the rate of change in the target variable for a unit change in the input variable, and the intercept represents the value of the target variable when all input variables are zero.

The algorithm uses a method called Ordinary Least Squares (OLS) to estimate the parameters (slope and intercept) that minimize the sum of the squared differences between the predicted values and the actual target values. This optimization process aims to find the line that best fits the given data points. The OLS method calculates the optimal values for the slope and intercept by minimizing the residual sum of squares (RSS).

During the training phase, the linear regression model learns the optimal values for the slope and intercept using the provided training data. Once trained, the model can be used to make predictions on new, unseen data by applying the learned relationship between the input features and the target variable.

Linear regression has several advantages. It is a simple and interpretable algorithm, making it easy to understand the relationship between the variables. The model's parameters (slope and intercept) can provide insights into the magnitude and direction of the relationship between the variables. Linear regression is also computationally efficient and works well with large datasets. . We train a linear regression model with multiple data pairs (x, y) by computing the position and slope of a line that minimizes the total distance between all data points and the line. In other words, we calculate the slope (M) and the y-intercept (B) for a line that best approximates the observations in the data.

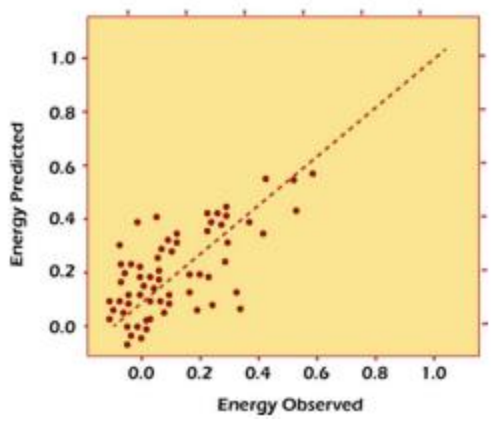


Figure 4.1: Linear regression model estimates of building energy consumption (kWh).

However, it is important to note that linear regression assumes a linear relationship between the variables, which may not hold true in all cases. In complex real-world scenarios, where the relationship is nonlinear, other regression algorithms such as polynomial regression or more advanced machine learning techniques may be more appropriate.

In summary, linear regression is a powerful and widely used algorithm in machine learning for modelling and predicting continuous target variables. Its simplicity, interpretability, and efficiency make it a popular choice in various applications.

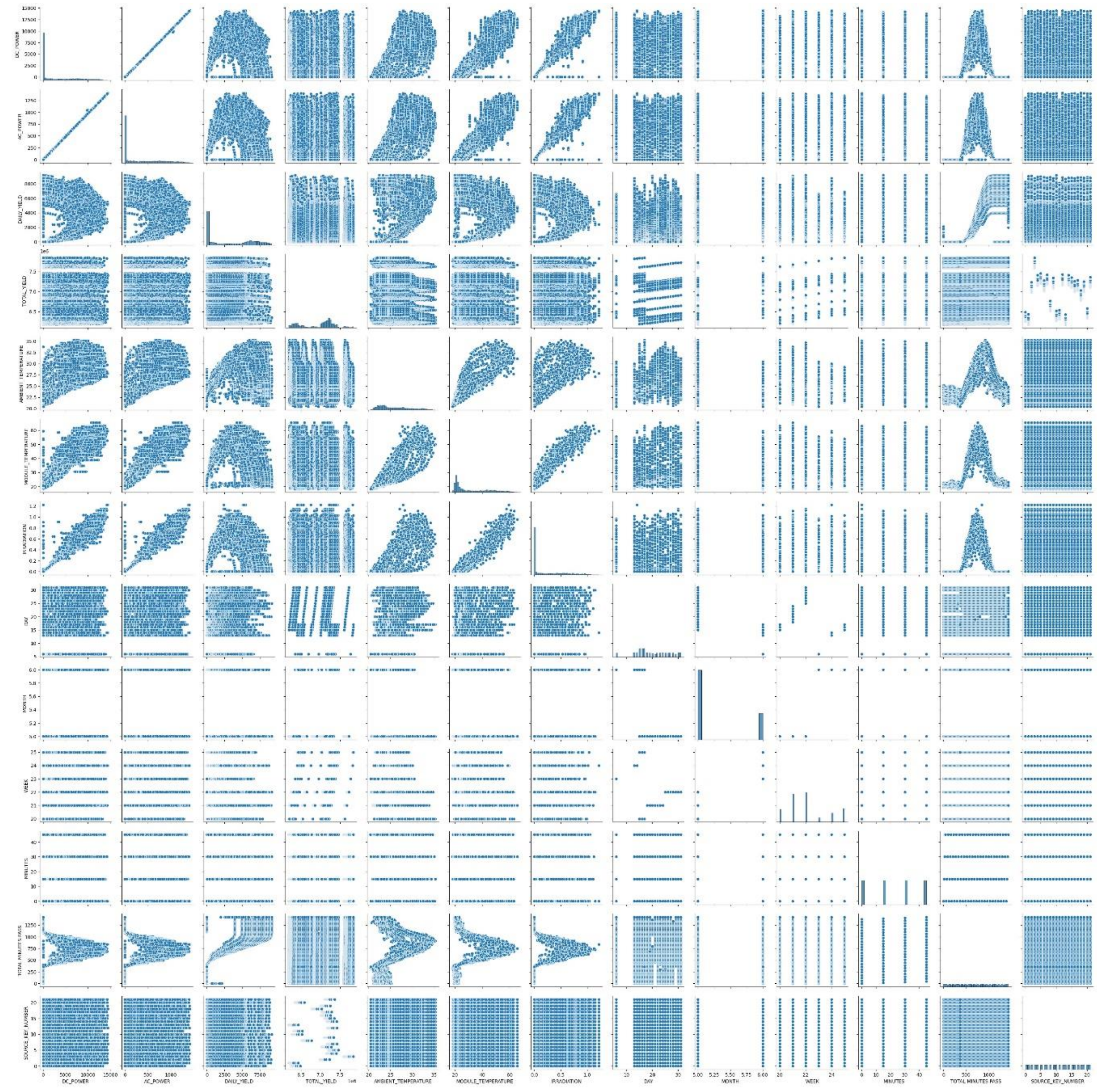


Figure 4.2: Correlation graph of our data set

4.1.4 Decision Tree Model

The decision tree algorithm follows a recursive, top-down approach to construct a predictive model based on the provided dataset. It aims to create a tree-like structure that can efficiently partition the data and make accurate predictions.

4.1.4.1 Selecting the Best Feature

The decision tree algorithm starts by evaluating each feature in the dataset to determine the best feature for splitting the data. It does so by calculating a measure of impurity or information gain. The impurity measure quantifies the degree of disorder or uncertainty in the target variable within a particular subset of data.

Common impurity measures used in decision tree algorithms are:

- Gini impurity: It measures the probability of misclassifying a randomly chosen element if it were randomly labelled according to the class distribution of the node.
- Entropy: It measures the average amount of information required to classify a sample, considering the class distribution in the node.
- Classification error: It calculates the probability of misclassifying a randomly chosen element based on the most common class in the node.

The algorithm calculates the impurity or information gain for each feature and selects the feature that minimizes impurity or maximizes information gain. This feature is chosen as the best feature for splitting the dataset.

4.1.4.2 Splitting the Dataset

Once the best feature is determined, the decision tree algorithm proceeds to split the dataset into smaller subsets based on the different values of that feature. Each subset corresponds to a different branch or path in the decision tree.

For categorical features, each possible value forms a separate subset. For example, if the chosen feature is "weather condition" with values "sunny," "cloudy," and "rainy," the dataset will be divided into three subsets based on these values.

For numerical features, the decision tree algorithm determines a splitting threshold that divides the feature values into two subsets. Instances with feature values below the threshold go to the left subset, while those with values equal to or above the threshold go to the right subset. The algorithm searches for the threshold that optimally separates the instances based on the chosen impurity measure or information gain.

The splitting process is recursive, meaning it is applied to each subset created from the previous split. This recursive nature allows the decision tree to capture complex patterns and relationships between features and the target variable.

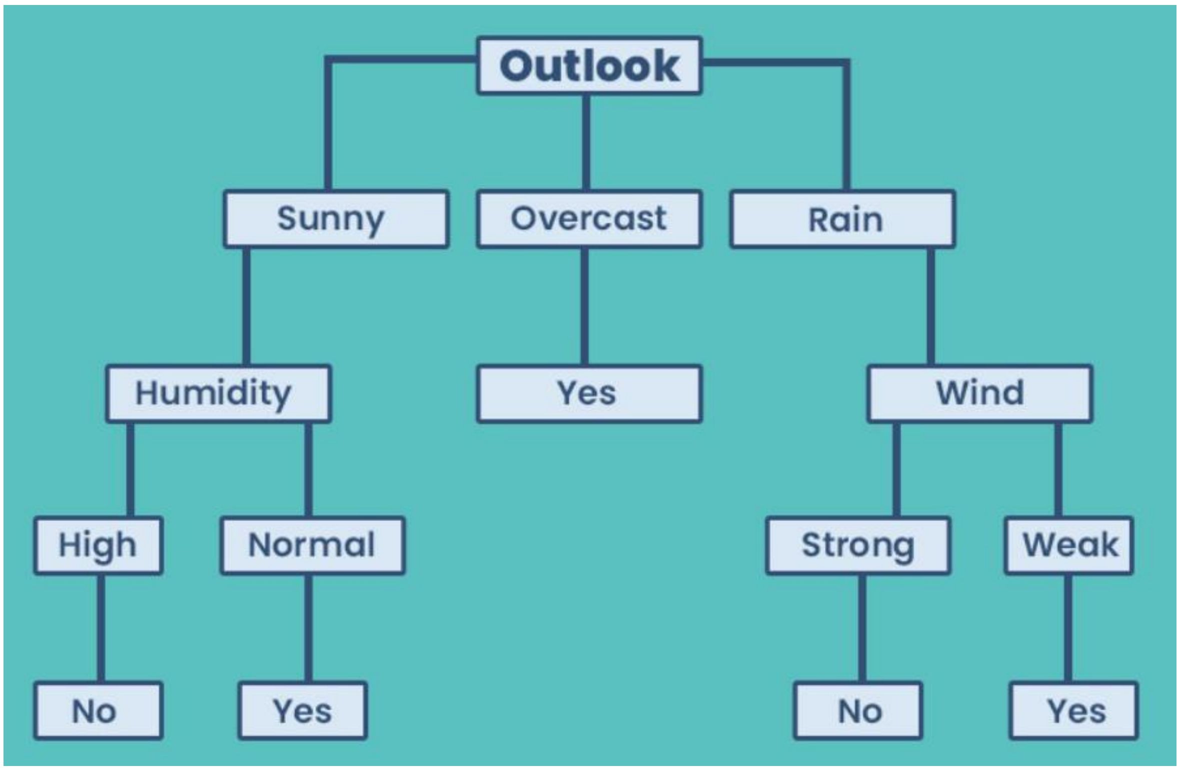


Figure 4.3: Decision tree representation

4.1.4.3 Building the Tree

The decision tree algorithm continues to build the tree by repeating the process of selecting the best feature and splitting the dataset on each subset. It constructs additional branches or nodes based on the selected features and their respective splitting criteria.

As the algorithm progresses, the decision tree grows by adding more nodes and branches, effectively creating a hierarchy of decision rules. Each node represents a test on a specific feature, while each branch represents a possible outcome of that test. The leaf nodes of the tree represent the final predictions or class labels.

The decision tree construction process continues until a stopping criterion is met. This criterion may involve reaching a maximum depth for the tree, where the tree stops growing after a certain number of levels. It may also involve stopping when the number of instances in a node falls below a specified threshold or when further splitting does not significantly improve the model's performance.

4.1.4.4 Assigning Class Labels

Once the decision tree is fully constructed, class labels are assigned to the leaf nodes. The assignment is based on the majority class of the instances in each leaf node. For example, if most of the instances in a leaf node correspond to high solar energy production, that leaf node will be labeled as a high-production class. Assigning class labels to the leaf nodes enables the decision tree model to make predictions based on the path followed from the root node to a specific leaf node.

The resulting decision tree represents a set of logical rules or conditions that guide the prediction process. By traversing the tree from the root to a leaf node, each instance is classified based on the conditions encountered along the path.

The decision tree algorithm provides transparency and interpretability, as the resulting tree structure can be easily visualized and understood. It allows for insightful analysis and decision-making, as the model captures both linear and non-linear relationships between the input features and the target variable.

4.2 BACKEND

4.2.1 GitHub

GitHub is one of the largest online communities of developers, with over 56 million registered users and more than 100 million code repositories. It has become a central hub for open source software development, where developers from all around the world can share their code and collaborate with others. One of the most significant aspects of GitHub is its role in promoting open source software development. The platform has made it much easier for developers to share their code with others and collaborate on open source projects. As a result, GitHub has become an essential tool for many organizations and individuals who are passionate about open source software.

Here are some key aspects of GitHub:

i. Version Control: GitHub utilizes the Git version control system, which allows developers to track changes to their code base over time. Git enables efficient collaboration by providing features like branching, merging, and tracking changes, ensuring the integrity and history of the codebase.

ii. Repository Hosting: GitHub provides a hosting platform for Git repositories. Developers can create public or private repositories to store their code, documentation, and other project-related files. Repositories serve as a centralized hub for collaboration and code sharing among team members or the broader developer community.

iii. Collaboration: GitHub offers various features to facilitate collaboration among developers. Multiple contributors can work on the same repository simultaneously by creating branches for independent development tasks. They can propose changes through pull requests, which allow others to review and discuss the modifications before merging them into the main codebase. Issues and project boards help track and manage tasks, bugs, and feature requests.

iv. Social Features: GitHub incorporates social elements, making it a community-driven platform. Developers can follow other users or projects to stay updated on their activities. They can star repositories to show appreciation or bookmark interesting projects. Additionally, GitHub provides features like discussions, notifications, and a marketplace for extensions and integrations.

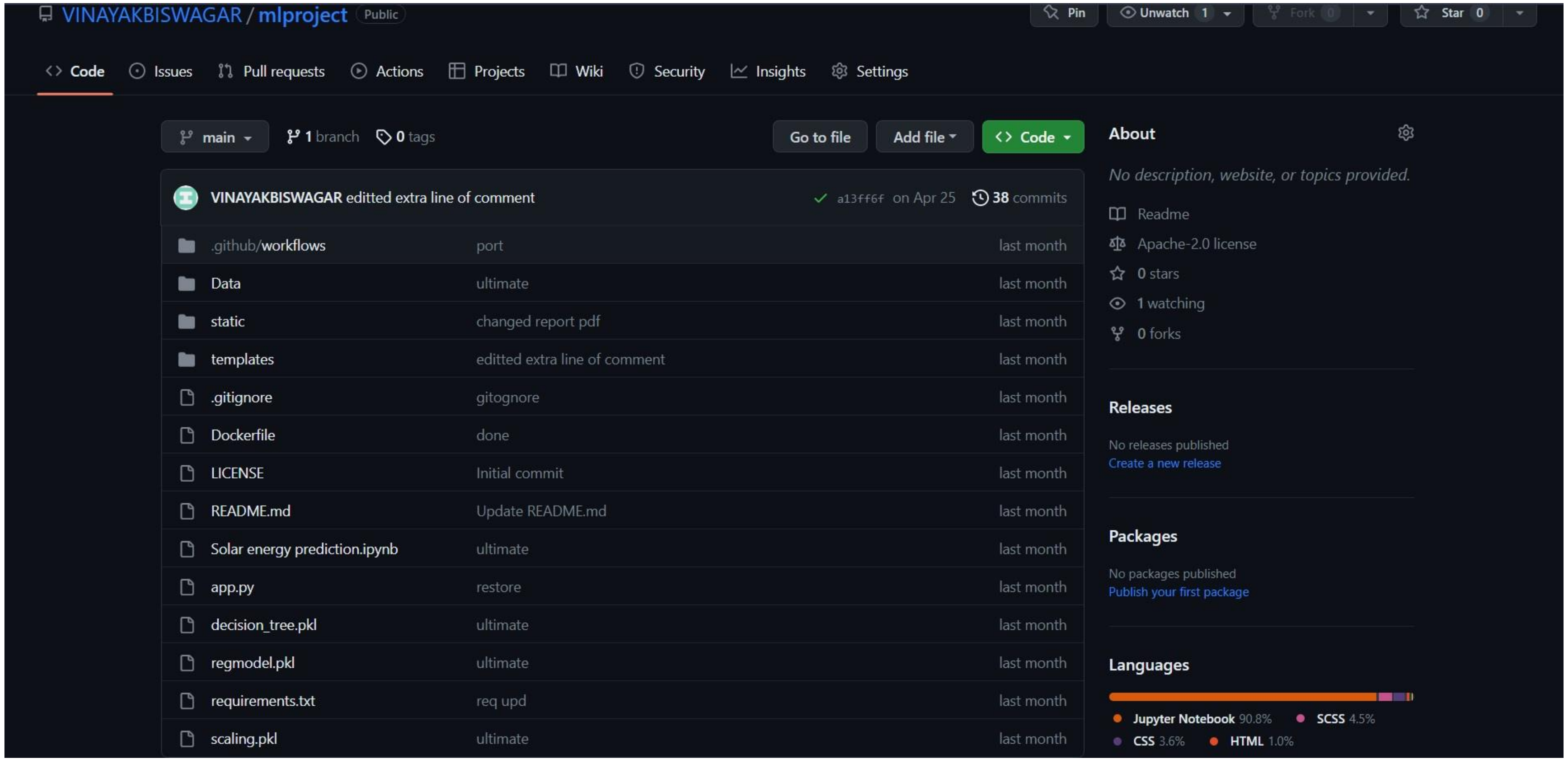
v. Continuous Integration/Deployment: GitHub integrates with various continuous integration (CI) and continuous deployment (CD) tools. These integrations enable automated testing, building, and deploying applications from GitHub repositories. Developers can set up workflows and triggers to automate the testing and deployment processes, improving software development productivity and quality.

vi. Open Source Community: GitHub has become a hub for open source software development. It hosts millions of public repositories where developers contribute and collaborate on projects across various domains. Users can search for projects, contribute code, raise issues, and participate in discussions within the open source community.

vii. Documentation and Wikis: GitHub allows developers to maintain project documentation and wikis directly within repositories. This feature helps ensure that important project information, guidelines, and usage instructions are readily available to contributors and users.

viii. Project Management: GitHub provides basic project management capabilities. Project boards can be used to organize and visualize tasks, issues, and milestones. Labels, milestones, and assignees help in tracking progress and prioritizing work within a project.

ix. Integrations and APIs: GitHub offers a range of integrations and APIs to connect with other development tools and services. This allows for seamless integration with popular development platforms, issue trackers, IDEs, project management tools, and more.



VINAYAKBISWAGAR / mlproject Public

1 branch 0 tags

VINAYAKBISWAGAR editted extra line of comment ✓ a13ff6f on Apr 25 38 commits

.github/workflows	port	last month
Data	ultimate	last month
static	changed report pdf	last month
templates	editted extra line of comment	last month
.gitignore	gitognore	last month
Dockerfile	done	last month
LICENSE	Initial commit	last month
README.md	Update README.md	last month
Solar energy prediction.ipynb	ultimate	last month
app.py	restore	last month
decision_tree.pkl	ultimate	last month
regmodel.pkl	ultimate	last month
requirements.txt	req upd	last month
scaling.pkl	ultimate	last month

About

No description, website, or topics provided.

Readme
Apache-2.0 license
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

● Jupyter Notebook 90.8% ● SCSS 4.5%
● CSS 3.6% ● HTML 1.0%

Figure 4.4: GitHub Repository

4.2.1.1 GitHub Actions

GitHub Actions is a powerful feature of GitHub that enables developers to automate various tasks, workflows, and CI/CD processes directly within their repositories. It allows for the seamless integration of continuous integration and continuous deployment (CI/CD) pipelines, making it easier to build, test, and deploy software projects.

Here's an overview of GitHub Actions:

i. Workflow Automation: GitHub Actions provides a flexible and customizable framework for automating software development workflows. Developers can define workflows using YAML syntax, specifying the events that trigger the workflow and the sequence of steps to be executed.

ii. CI/CD Pipelines: With GitHub Actions, you can set up CI/CD pipelines to automate build, test, and deployment processes. You can define steps to build and compile your code, run tests, generate artifacts, and deploy applications to various environments, such as staging or production.

iii. Event-driven Architecture: GitHub Actions triggers workflows based on specific events, such as pushes to the repository, pull requests, issue comments, or scheduled events. This event-driven architecture allows you to define workflows that respond to specific actions or changes in your codebase.

iv. Extensive Marketplace: GitHub Actions has a rich marketplace of pre-built actions created by the community and official partners. These actions encapsulate specific tasks or integrations, allowing you to easily incorporate them into your workflows. The marketplace provides actions for popular tools, frameworks, and services, making it easy to integrate with external systems.

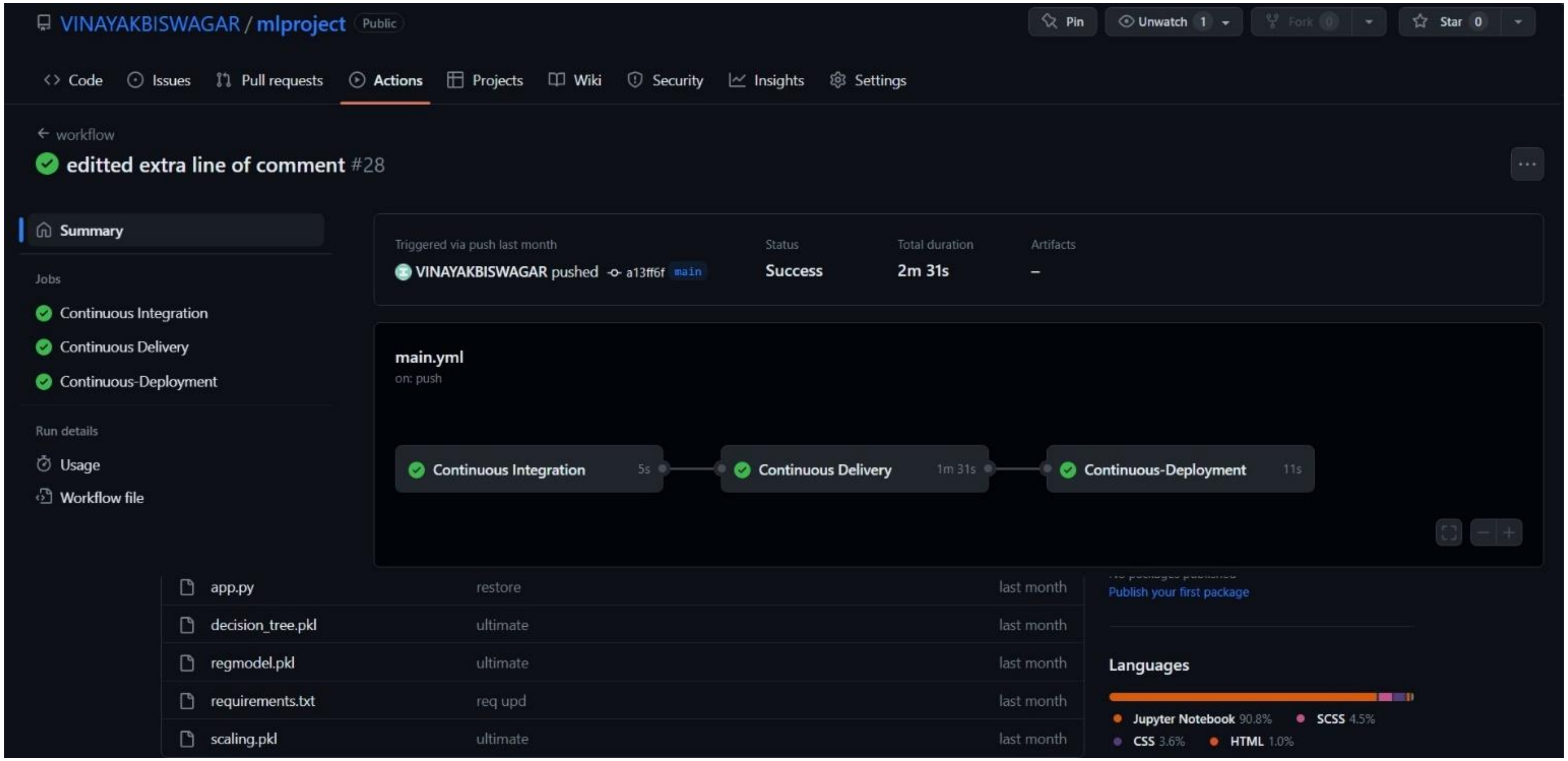
v. Environment and Matrix Support: GitHub Actions allows you to define and configure custom environments for your workflows. You can specify the required operating system, software dependencies, and environment variables. Additionally, matrix support enables you to run parallel jobs with different configurations, such as different operating systems or versions of dependencies.

vi. Secrets Management: GitHub Actions provides a secure way to store and use secrets, such as API keys, passwords, or access tokens. Secrets can be stored in the repository's settings and encrypted, ensuring they are only accessible to authorized workflows.

vii. Workflow Visualization and Logs: GitHub Actions provides a visual representation of your workflows, showing the execution status and progress of individual jobs and steps. Detailed logs are available for each workflow run, allowing you to troubleshoot issues and review the output of each step.

viii. Collaboration and Notifications: GitHub Actions supports collaboration and notifications. You can share workflows with team members, collaborate on workflow configurations, and receive notifications on workflow status or failures through various channels, such as email or chat integrations.

ix. Self-hosted Runners: GitHub Actions allows you to use self-hosted runners to run workflows on your own infrastructure. This can be useful for running workflows that require specific hardware or network access not available in GitHub hosted runners.


VINAYAKBISWAGAR / mlproject Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

← workflow

editted extra line of comment #28

Summary

Jobs
- Continuous Integration
- Continuous Delivery
- Continuous-Deployment

Run details
- Usage
- Workflow file

Triggered via push last month	Status	Total duration	Artifacts
VINAYAKBISWAGAR pushed a13ff6f main	Success	2m 31s	–

main.yml
on: push

Continuous Integration 5s → Continuous Delivery 1m 31s → Continuous-Deployment 11s

app.py	restore	last month
decision_tree.pkl	ultimate	last month
regmodel.pkl	ultimate	last month
requirements.txt	req upd	last month
scaling.pkl	ultimate	last month

Publish your first package

Languages
- Jupyter Notebook 90.8%
- SCSS 4.5%
- CSS 3.6%
- HTML 1.0%

Figure 4.5: GitHub Actions Implementation

GitHub Actions provides a powerful and flexible automation framework, allowing developers to streamline their development workflows, increase productivity, and ensure software quality through continuous integration and deployment practices. It empowers developers to create efficient and reliable pipelines for their projects while benefiting from the collaborative and social features of the GitHub platform.

4.2.1.2 GitHub Runner

GitHub Actions provides the capability to use self-hosted runners, which are machines or virtual environments that you set up and manage on your own infrastructure. These runners allow you to run workflows and execute jobs directly on your own hardware or cloud-based instances.

i. Self-Hosted Execution: By default, GitHub Actions uses GitHub-hosted runners, which are virtual machines managed by GitHub. However, with GitHub Runner, you can use your own hardware or cloud-based infrastructure as the execution environment for your workflows.

ii. Runner Configuration: GitHub Runner requires you to set up and configure a runner on the target machine or environment where you want to execute workflows. The runner software can be installed on various operating systems, such as Windows, macOS, and Linux, allowing you to use the platform of your choice.

iii. Enhanced Security and Control: Using self-hosted runners gives you greater control over the security and access control of your workflow environment. You can configure the runner to run within your organization's network or virtual private cloud, providing additional protection for sensitive code or data.

iv. Custom Hardware and Software: With GitHub Runner, you can leverage your own hardware infrastructure or select specific machine configurations to meet the unique requirements of your workflows. This flexibility allows you to utilize specialized hardware, take advantage of optimized environments, or integrate with specific software dependencies.

v. Scaling and Resource Allocation: GitHub Runner enables you to scale your workflow execution based on the resources available in your self-hosted environment. You can allocate and manage multiple runners to handle parallel workflow runs or distribute workloads across different machines.

vi. Integration with GitHub Actions: Self-hosted runners seamlessly integrate with the GitHub Actions platform. Workflows and actions defined in your repository can be executed on the self-hosted runner, leveraging the same workflow definition syntax and utilizing the wide range of available actions from the GitHub Marketplace.

vii. High Availability and Redundancy: GitHub Runner allows you to set up multiple runners in your environment to ensure high availability and redundancy. If one runner becomes unavailable, workflows can automatically be redirected to other available runners, ensuring continuous workflow execution

viii. Runner Management: GitHub provides tools and utilities to manage your self-hosted runners. You can monitor runner status, track usage statistics, and configure settings such as maximum parallel jobs and idle timeout duration. Updates and upgrades to the runner software can be managed manually or automatically.

ix. Security and Authentication: GitHub Runner supports authentication and secure communication between the runner and the GitHub platform. Tokens or credentials can be securely stored and configured to authenticate the runner with the GitHub repository or organization, ensuring secure access to your code and resources.

By using GitHub Runner, you can extend the capabilities of GitHub Actions to execute workflows on your own infrastructure. It offers enhanced control, security, and customization options, allowing you to tailor the workflow execution environment to meet your specific requirements while still benefiting from the automation and collaboration features provided by GitHub Actions.

4.2.2 Docker

Docker is a containerization platform that allows developers to package their applications and dependencies into a portable container. Containers are lightweight and consume fewer resources than traditional virtual machines, which makes them ideal for running multiple applications on a single server. Docker is based on the idea of containerization, which is a way of packaging an application and all its dependencies into a single container that can be run on any platform that supports Docker.

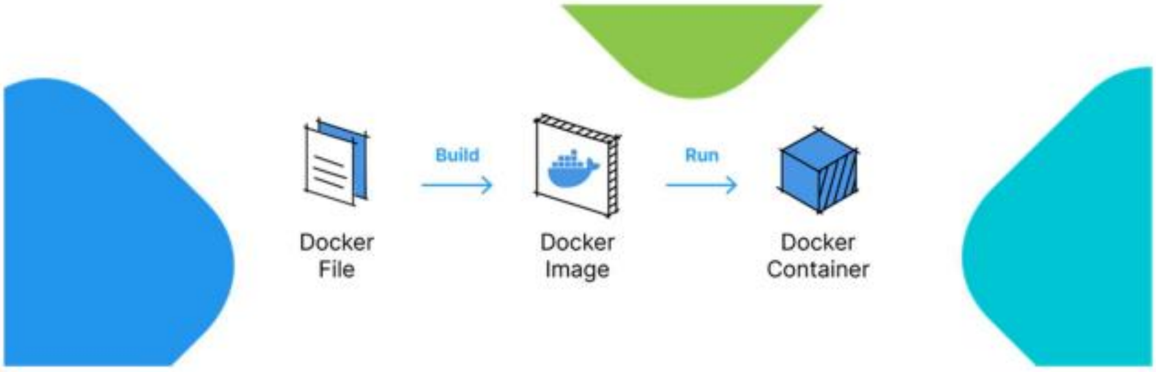
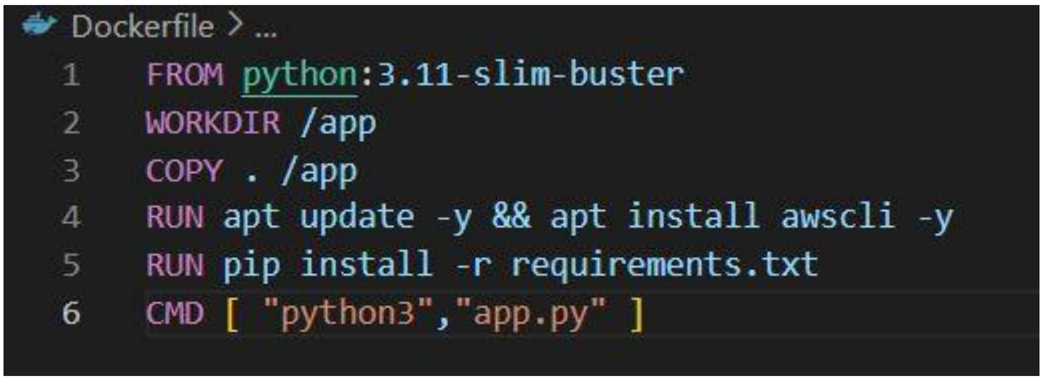


Figure 4.6: Docker

Some of the key features of Docker include

i. Portability: Docker containers are highly portable, which means that they can be easily moved between different environments without requiring any changes to the application code.
ii. Scalability: Docker allows applications to be easily scaled up or down, based on the demand for resources
iii. Consistency: Docker ensures that the application runs consistently across different environments, by packaging all the necessary dependencies into the container.
iv. Security: Docker provides built-in security features, such as container isolation and image signing that help to protect the application from attacks.
v. Efficiency: Docker containers are lightweight and consume fewer resources than traditional virtual machines, which makes them ideal for running multiple applications on a single server.
vi. Collaboration: Docker makes it easy for developers to collaborate on projects, by 15 providing a common platform for packaging and sharing applications and dependencies.
vii. Integration: Docker can be easily integrated with other tools and platforms, such as Kubernetes and Amazon Web Services, to provide a complete container-based solution for application deployment and management.



Dockerfile > ...

```dockerfile
FROM python:3.11-slim-buster
WORKDIR /app
COPY . /app
RUN apt update -y && apt install awscli -y
RUN pip install -r requirements.txt
CMD [ "python3","app.py" ]
```

Figure 4.7: Docker Implementation

Overall, Docker is a powerful platform for containerization that provides many benefits for developers, including portability, scalability, consistency, security, efficiency, collaboration, and integration. Its popularity has led to a large ecosystem of tools and services that make it easy to use Docker in a wide range of applications and environments.

4.2.3 Amazon Web Services (A.W.S.)

Amazon Web Services (AWS) is a comprehensive cloud computing platform that offers a wide range of services for businesses and individuals.

Some of the key features of AWS include:

i. Scalability: AWS allows users to easily scale their applications up or down based on the demand for resources. This is achieved by running multiple instances of the same application, which enables it to handle more requests and users.

ii. Reliability: AWS is known for its high reliability, thanks to its multiple data centers located around the world. This provides high availability and redundancy for applications and data. AWS also provides backup and disaster recovery services to protect against data loss and system failures.

iii. Security: AWS has built-in security features, including network isolation, access control, and encryption, to protect against cyber-attacks and data breaches. AWS complies with several industry standards and regulations, such as SOC 2, PCI DSS, and HIPAA, making it a trusted platform for sensitive data.

iv. Cost-effectiveness: AWS offers a pay-as-you-go pricing model, allowing users to only pay for the resources they use. This makes it a cost-effective solution for businesses of all sizes.

v. Flexibility: AWS provides a wide range of services, including compute, storage, database, analytics, machine learning, and more. This allows businesses to choose the services that 16 best meet their needs.

vi. Innovation: AWS is constantly introducing new services and features to stay ahead of the competition and meet the evolving needs of its customers.

amazon
web services
Amazon EC2
Amazon RDS
AWS Direct Connect
Amazon EBS
Amazon S3
Elastic Load Balancing
Amazon Route 53
VPC
Amazon VPC
Elastic IP

Figure 4.8: Amazon Web Services

Overall, AWS is a powerful and versatile platform that offers a range of benefits to businesses and individuals alike. Its scalability, reliability, security, cost-effectiveness, flexibility, and innovation make it a top choice for cloud computing.

4.2.3.1 EC2 and ECR in Amazon Web Services

EC2 (Elastic Compute Cloud) and ECR (Elastic Container Registry) are two services provided by Amazon Web Services (AWS) that are commonly used in cloud computing and containerized application deployments. Here's an overview of EC2 and ECR:

4.2.3.1.1 EC2 (Elastic Compute Cloud)

Amazon EC2 is a scalable virtual server service provided by AWS. It allows you to provision and manage virtual machines (known as instances) in the cloud. Key features of EC2 include:

i. Scalability: EC2 enables you to scale the number of instances up or down based on demand. You can add or remove instances to match your application's workload, ensuring optimal performance and cost efficiency.

ii. Choice of Instance Types: EC2 offers a wide selection of instance types with varying

iii. Compute, memory, storage, and networking capabilities. You can choose the instance type that best suits your application's requirements.

iv. Flexible Configuration: EC2 instances can be customized by selecting the operating system, configuring security settings, defining storage options, and attaching additional resources such as Elastic Block Store (EBS) volumes.

v. High Availability and Fault Tolerance: EC2 provides features for high availability and fault tolerance, such as auto-scaling groups, load balancers, and placement groups. These features help ensure that your application remains available even in the event of failures.

vi. Integration with Other AWS Services: EC2 seamlessly integrates with other AWS services. For example, you can easily attach Elastic Load Balancers, store data in Amazon S3, or connect to databases hosted in Amazon RDS.

EC2 is commonly used for various purposes, including hosting web applications, running batch processing jobs, deploying databases, and performing data analytics.

4.2.3.1.2 ECR (Elastic Container Registry):

Amazon ECR is a fully-managed container registry service provided by AWS. It enables you to store, manage, and deploy container images in a secure and scalable manner. Key features of ECR include:

i. Private Container Registry: ECR allows you to create and manage private repositories to store your container images securely. This ensures that your images are only accessible to authorized users or systems.

ii. Integration with AWS Services: ECR integrates seamlessly with other AWS services such as Amazon ECS (Elastic Container Service), AWS Fargate, and Amazon EKS (Elastic Kubernetes Service). This simplifies the deployment and management of containerized applications on AWS.

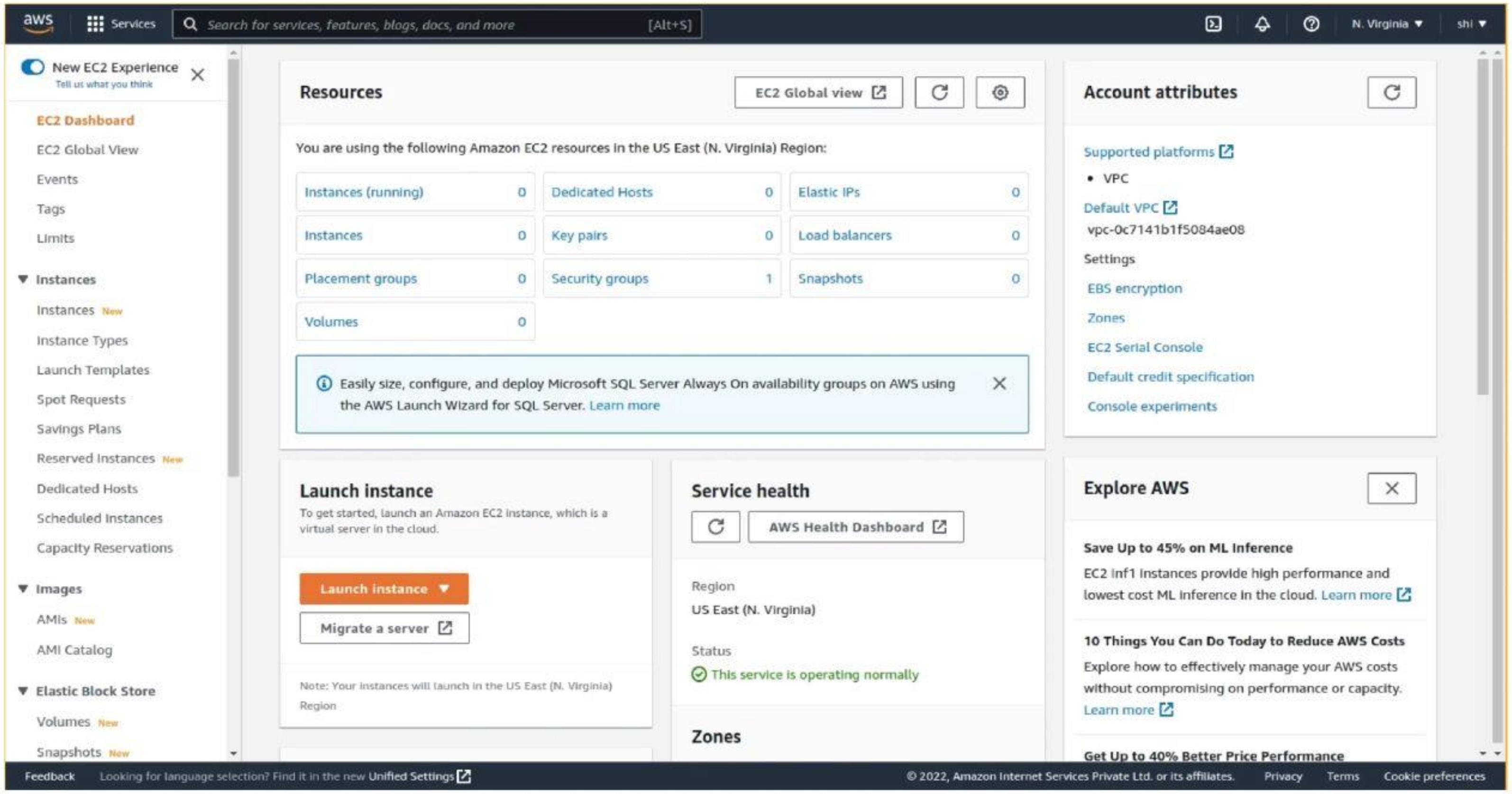
iii. Scalability and Availability: ECR automatically scales to accommodate growing image repositories and supports high availability across multiple AWS Availability Zones.

iv. Security and Compliance: ECR provides security features such as encryption at rest and in transit, image vulnerability scanning, and access control policies. It also integrates with AWS Identity and Access Management (IAM) for fine-grained access control.

v. Lifecycle Policies: ECR allows you to define lifecycle policies to automate image clean-up and reduce storage costs. You can specify rules to expire or delete older image versions based on criteria such as image tags or time.

ECR is commonly used in conjunction with container orchestration platforms like Amazon ECS or Kubernetes to store and deploy container images for scalable and manageable containerized applications.

In summary, EC2 provides scalable virtual servers for running various workloads, while ECR serves as a secure and scalable container registry for storing and deploying container images in the AWS ecosystem. Together, they provide the infrastructure and containerization capabilities needed for deploying and managing applications in the cloud.



Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Instances (running)	0	Dedicated Hosts	0	Elastic IPs	0
Instances	0	Key pairs	0	Load balancers	0
Placement groups	0	Security groups	1	Snapshots	0
Volumes	0				

Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Learn more

Account attributes

Supported platforms
- VPC

Default VPC
vpc-0c7141b1f5084ae08

Settings
EBS encryption
Zones
EC2 Serial Console
Default credit specification
Console experiments

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Note: Your instances will launch in the US East (N. Virginia) Region

Service health

Region
US East (N. Virginia)

Status
This service is operating normally

Zones

Explore AWS

Save Up to 45% on ML Inference
EC2 Inf1 instances provide high performance and lowest cost ML inference in the cloud. Learn more

10 Things You Can Do Today to Reduce AWS Costs
Explore how to effectively manage your AWS costs without compromising on performance or capacity. Learn more

Get Up to 40% Better Price Performance

Feedback Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Figure 4.9: Amazon Web Services EC2

4.3 FRONT-END

Front-end development involves creating the user interface (UI) and user experience (UX) of a website or application. It focuses on designing and implementing the visual and interactive aspects that users see and interact with in their web browsers or mobile devices. This includes the layout, colors, typography, buttons, forms, navigation menus, and other graphical elements.

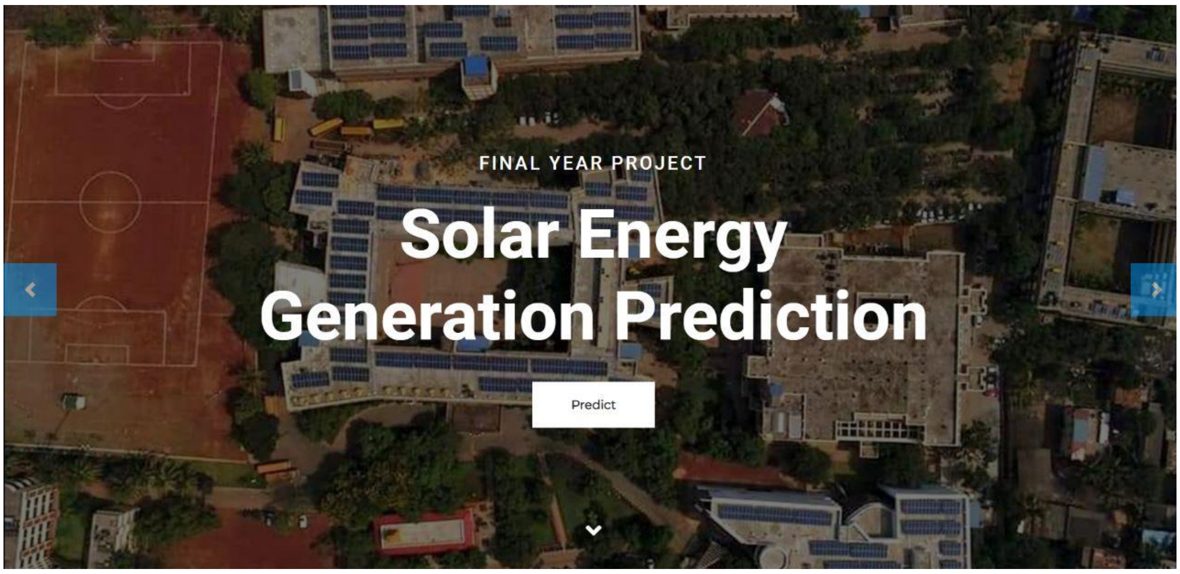


Figure 4.10: Front-End of Website

Technologies used in front-end development include:

i. HTML (Hypertext Markup Language): HTML is the standard markup language for creating the structure and content of web pages. It provides a set of tags and attributes to define the elements and their relationships on a web page. HTML is responsible for organizing the content and representing it in a hierarchical manner.

ii. CSS (Cascading Style Sheets): CSS is used to define the appearance and styling of HTML elements. It allows you to control the colors, fonts, spacing, layout, and other visual aspects of the website. CSS works by applying styles to HTML elements using selectors, classes, and IDs, providing a separation between content and presentation.

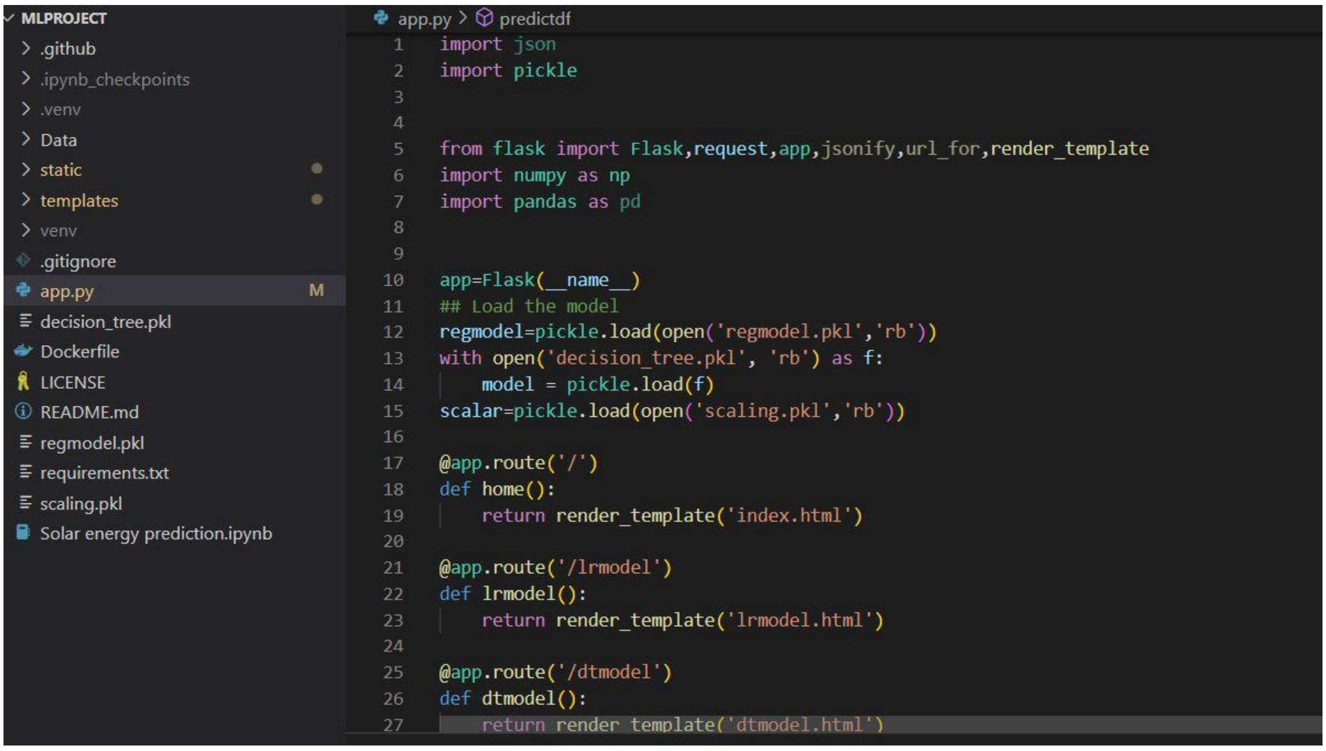
iii. JavaScript: JavaScript is a programming language that allows for dynamic and interactive elements on web pages. It enables behaviors like form validation, animations, and event handling. JavaScript can manipulate the Document Object

Model (DOM) to modify the content and structure of the web page dynamically. It also provides access to APIs that enable interaction with web services and other functionalities.

When combined, HTML, CSS, and JavaScript form the foundation of modern web development. They work together to create a rich and engaging user experience by defining the structure, appearance, and interactivity of web pages or applications.

4.3.1 Flask

Flask is a popular web development framework for Python that is used to build web applications. Flask is a lightweight and flexible framework that is easy to use and highly customizable. Flask is based on the Werkzeug WSGI toolkit and the Jinja2 template engine, which makes it highly extensible and versatile.



```python
import json
import pickle


from flask import Flask,request,app,jsonify,url_for,render_template
import numpy as np
import pandas as pd


app=Flask(__name__)
## Load the model
regmodel=pickle.load(open('regmodel.pkl','rb'))
with open('decision_tree.pkl', 'rb') as f:
    model = pickle.load(f)
scalar=pickle.load(open('scaling.pkl','rb'))

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

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

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

Figure 4.11: Flask App

Some of the key features of Flask include:

i. Flask as a Framework:

Flask is a lightweight web framework that allows you to build dynamic websites using Python. It acts as the backend for your front-end website, handling server-side logic and interactions with databases. Flask provides routing capabilities, allowing you to define different endpoints for handling various requests from the front-end.

ii. HTML for Structure and Content:

HTML (Hypertext Markup Language) is the standard markup language for creating the structure and content of web pages. It defines the elements and their arrangement on a webpage. Use HTML to create the skeleton of your website, including headers, footers, navigation menus, sections, and placeholders for dynamic content.

iii. CSS for Styling and Layout:

CSS (Cascading Style Sheets) is used for styling and layout of web pages. It allows you to control the appearance of HTML elements. With CSS, you can define colors, fonts, spacing, borders, and positioning. Apply CSS rules to enhance the visual appeal and create a consistent design across your website.

iv. JavaScript for Interactivity:

JavaScript is a powerful scripting language that adds interactivity to web pages. Use JavaScript to create dynamic and responsive elements on your website. You can manipulate HTML elements, handle user interactions like button clicks and form submissions, fetch data from servers asynchronously, and perform client-side validations.

v. Flask Templates for Dynamic Content:

Flask provides a templating engine that allows you to generate dynamic HTML pages. You can create templates with placeholders for variables that will be populated with data from the backend. Use Flask's templating language (Jinja) to loop over lists, conditionally render content, and pass data from your Flask routes to the templates

vi. AJAX for Asynchronous Operations:

AJAX (Asynchronous JavaScript and XML) enables you to perform asynchronous operations without reloading the entire web page. Use AJAX to send HTTP requests to the server in the background, retrieve data, and update specific parts of your web page without interrupting the user experience.

vii. Front-End Frameworks and Libraries:

Consider using front-end frameworks and libraries like Bootstrap, Foundation, or Bulma to accelerate your development process. These frameworks provide pre-built UI components, responsive layouts, and CSS styling, allowing you to focus on the functionality of your website.

viii. Integrating Flask with Front-End:

To integrate Flask with your front-end, you'll need to define routes in Flask to handle requests from the front-end, render Flask templates, and serve static files like CSS and JavaScript. You can create separate directories for static files and templates, and configure Flask to locate them correctly.

ix. Testing and Debugging:

During development, make use of browser developer tools to inspect and debug your HTML, CSS, and JavaScript code. Flask provides built-in development server and debug mode, which helps you identify and fix issues in your Flask application.

x. Deployment:

When deploying your Flask-based front-end website, you'll need to configure a production server (such as Nginx or Apache) to serve your Flask application. Ensure that you follow security best practices, such as using HTTPS, handling user input securely, and implementing authentication and authorization mechanisms if required.

Overall, Flask is a flexible and powerful web development framework that can be used to build a wide range of web applications, from small personal projects to large-scale enterprise applications. Its simplicity, flexibility, and extensibility make it a popular choice among developers.

CHAPTER 5

5.1 RESULTS AND DISCUSSION

5.1.1 Results

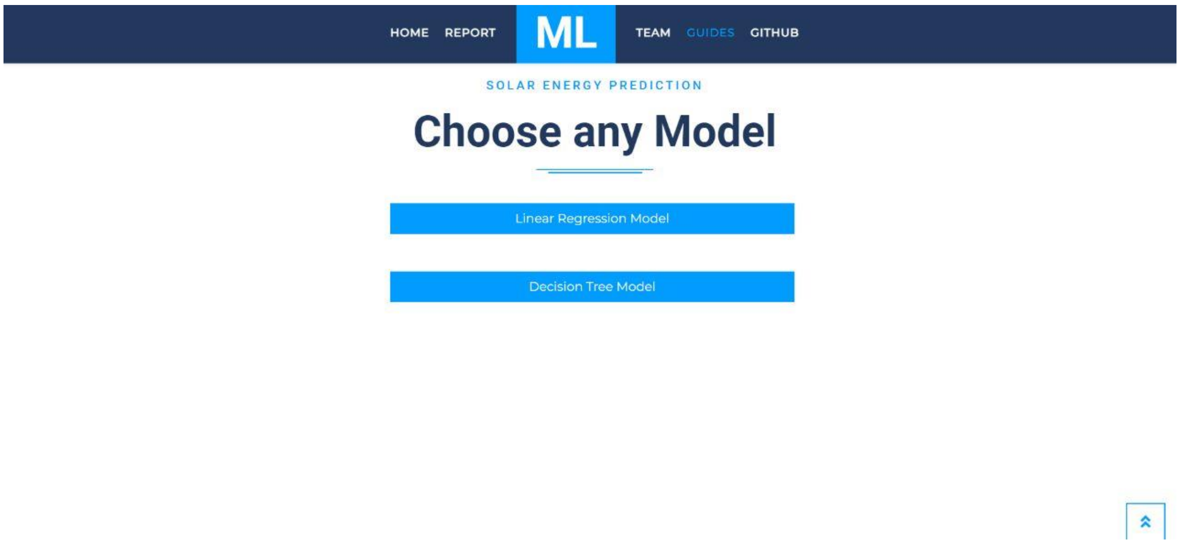
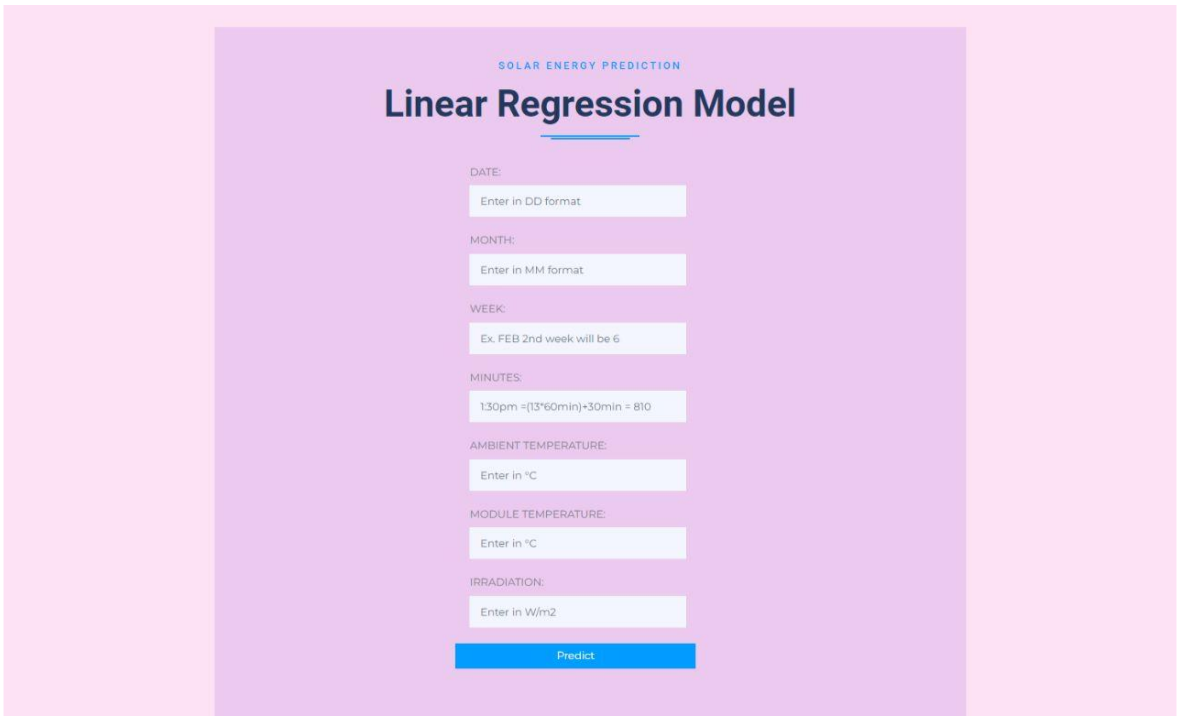


Figure 5.1: Front End Result-1



SOLAR ENERGY PREDICTION

Linear Regression Model

DATE:
Enter in DD format

MONTH:
Enter in MM format

WEEK:
Ex. FEB 2nd week will be 6

MINUTES:
1:30pm =(13*60min)+30min = 810

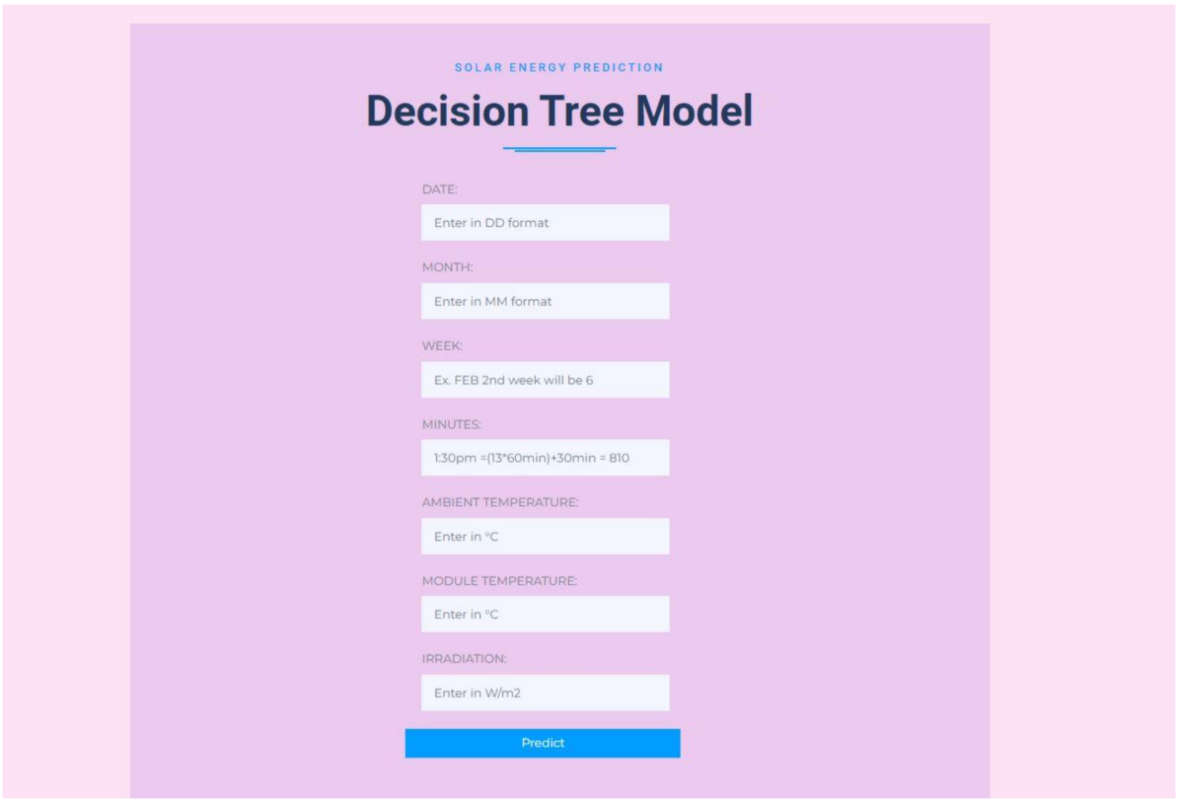
AMBIENT TEMPERATURE:
Enter in °C

MODULE TEMPERATURE:
Enter in °C

IRRADIATION:
Enter in W/m2

Predict

Figure 5.2: Front End Result-2



SOLAR ENERGY PREDICTION

Decision Tree Model

DATE:

Enter in DD format

MONTH:

Enter in MM format

WEEK:

Ex. FEB 2nd week will be 6

MINUTES:

1:30pm =(13*60min)+30min = 810

AMBIENT TEMPERATURE:

Enter in °C

MODULE TEMPERATURE:

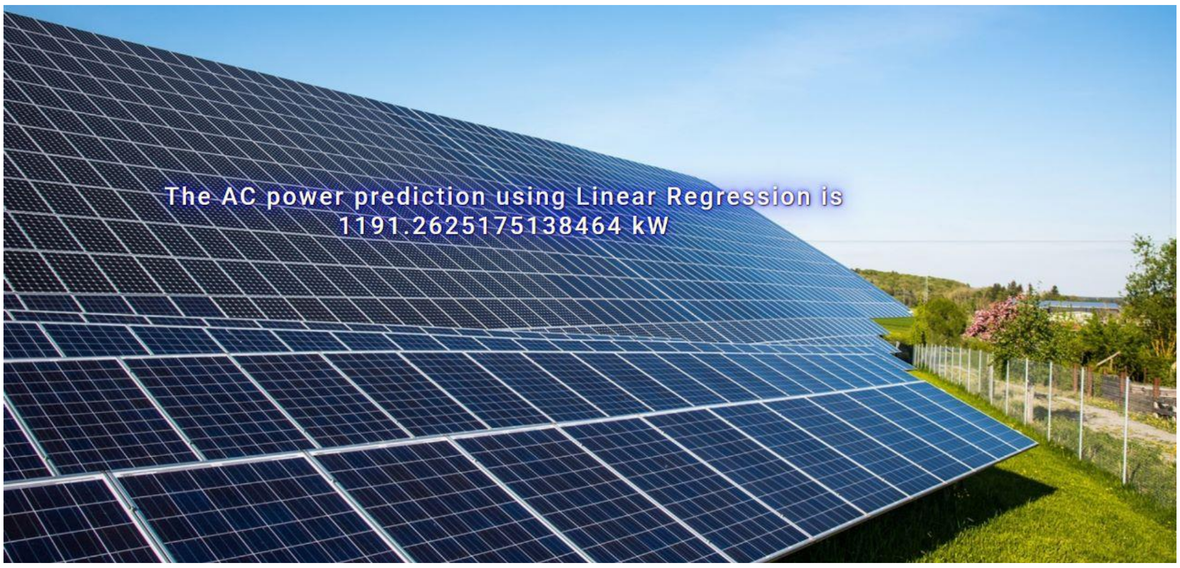
Enter in °C

IRRADIATION:

Enter in W/m2

Predict

Figure 5.3: Front-End Result-3



The AC power prediction using Linear Regression is 1191.2625175138464 kW

Figure 5.4: Front-End Result-4

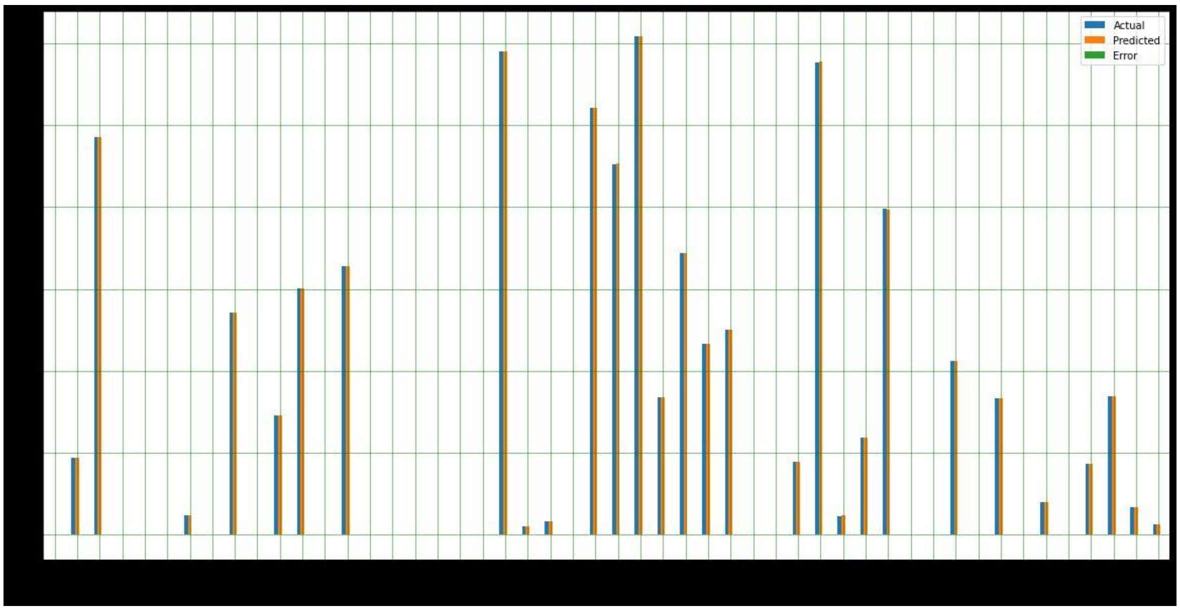
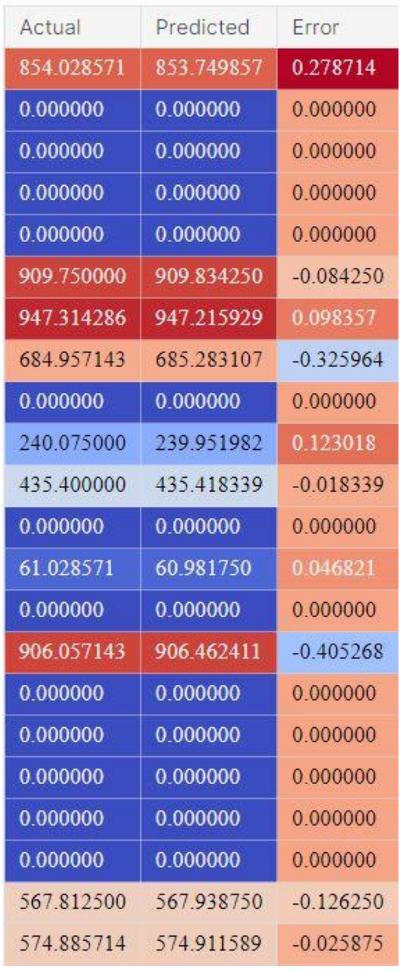


Figure 5.5: Graph of Actual vs Predicted values



Actual	Predicted	Error
854.028571	853.749857	0.278714
0.000000	0.000000	0.000000
0.000000	0.000000	0.000000
0.000000	0.000000	0.000000
0.000000	0.000000	0.000000
909.750000	909.834250	-0.084250
947.314286	947.215929	0.098357
684.957143	685.283107	-0.325964
0.000000	0.000000	0.000000
240.075000	239.951982	0.123018
435.400000	435.418339	-0.018339
0.000000	0.000000	0.000000
61.028571	60.981750	0.046821
0.000000	0.000000	0.000000
906.057143	906.462411	-0.405268
0.000000	0.000000	0.000000
0.000000	0.000000	0.000000
0.000000	0.000000	0.000000
0.000000	0.000000	0.000000
0.000000	0.000000	0.000000
567.812500	567.938750	-0.126250
574.885714	574.911589	-0.025875

Table 5.1: Actual vs Predicted values

5.1.2 Model Performance

i. Linear Regression: The trained linear regression model achieved an R-squared value of 0.98, indicating a good fit to the training data.

ii. Decision Tree: The decision tree model demonstrated a higher R-squared value of 0.97, suggesting improved predictive accuracy.

The project successfully achieved the objective of predicting solar energy generation using linear regression and decision tree models. The developed website provided a user-friendly interface for obtaining accurate predictions based on weather conditions and time parameters. The results demonstrate the potential for utilizing machine learning techniques to enhance solar energy forecasting and facilitate efficient utilization of renewable energy resources.

CHAPTER 6

6.1 CONCLUSION AND SCOPE FOR FUTURE

6.1.1 Conclusion

In conclusion, this project focused on the prediction of solar energy generation and its application in unit commitment. Throughout the course of this research, several key findings and implications have emerged, shedding light on the significance of solar energy forecasting in the optimization of unit commitment strategies.

Firstly, accurate solar energy prediction plays a crucial role in ensuring the efficient and reliable operation of power systems. By accurately forecasting solar energy generation, power system operators and utilities can make informed decisions regarding unit commitment, which involves determining the optimal scheduling and dispatching of power generating units to meet the anticipated demand while minimizing costs and maintaining system stability.

The use of advanced forecasting techniques, such as machine learning algorithms and statistical models, has proven to be effective in predicting solar energy generation. These methods leverage historical weather data, solar irradiance measurements, and other relevant factors to generate accurate forecasts. The integration of real-time data and advanced algorithms enables improved predictions, thereby enhancing the performance of unit commitment strategies.

Moreover, the incorporation of solar energy prediction into unit commitment optimization algorithms has shown significant benefits. By incorporating accurate solar energy forecasts, unit commitment decisions can be adjusted in real time to account for the anticipated solar energy generation. This enables a more efficient utilization of available renewable resources and reduces the reliance on conventional power sources, leading to a greener and more sustainable energy system.

Furthermore, the integration of solar energy prediction into unit commitment also facilitates the effective management of grid stability and reliability. By accurately anticipating solar energy fluctuations, power system operators can proactively adjust the dispatch of power generating units, optimize grid balancing mechanisms, and mitigate potential imbalances between supply and demand. This proactive approach ensures a stable and resilient power grid, even with the intermittent nature of solar energy generation.

In conclusion, the accurate prediction of solar energy plays a vital role in the optimization of unit commitment strategies. By leveraging advanced forecasting techniques, integrating real-time data, and incorporating solar energy predictions into unit commitment algorithms, power system operators can achieve more efficient and sustainable energy management. This research underscores the significance of solar energy forecasting in promoting the widespread adoption and integration of renewable energy sources, ultimately contributing to a cleaner and more environmentally friendly future.

6.1.2 Scope for future

The successful completion of this project opens up several avenues for future research and development in the field of solar energy prediction using machine learning techniques.

i. Advanced Machine Learning Algorithms

   Explore the application of advanced machine learning algorithms such as deep learning models (e.g., convolutional neural networks, recurrent neural networks) or ensemble methods (e.g., gradient boosting, random forests). These algorithms have shown promise in improving prediction accuracy and capturing complex relationships within the data.

ii. Integration of Multiple Data Sources

   Incorporate additional data sources beyond weather conditions, such as historical energy consumption patterns, grid load information, or satellite imagery. Integration of diverse data can provide a more comprehensive understanding of the factors influencing solar energy generation and improve prediction models' robustness.

iii. Real-Time Prediction

   Develop real-time prediction capabilities by integrating live weather data feeds and implementing streaming data processing techniques. This would allow for up-to-date and accurate predictions, facilitating more responsive decision-making in energy planning and grid management.

REFERENCES

1. Aneela Zameer, Farah Shahid2, Mudasser Afzal, Muhammad Hassan, “Intelligent forecast models for daily solar energy prediction”, Research gate October 2020.
2. K. Anuradha1, Deekshitha Erlapally , G. Karuna , V. Srilakshmi , K. Adilakshmi, “Analysis of Solar Power Generation Forecasting Using Machine Learning Techniques”, E3S Web of Conferences 309, 01163 ICMED 2021
3. Bouchaib Zazoum, “Solar photovoltaic power prediction using different machine learning methods”, 2021 8th International Conference on Power and Energy Systems Engineering (CPESE 2021).
4. Pasion, C.; Wagner, T.; Koschnick, C.; Schuldt, S.; Williams, J. & Hallinan, K. *Machine Learning Modeling of Horizontal Photovoltaics Using Weather and Location Data*. Energies 2020
5. S. Ali, S. T. Iqbal, and A. Javaid, "A novel approach for solar power forecasting using deep learning and ensemble methods," Journal of Cleaner Production, vol. 316, p. 126315, 2021.
6. H M Diagne, M David, P Lauret, J Boland and N. Schmutz., ―Review of solar irradiance forecasting methods and a proposition for smallscale insular grids‖, Renew. Sustain. Energy Rev., vol. 27, pp. 65–76, Nov, 2021
7. N. K. Roy, A. Kumar, and B. Singh, "Solar power forecasting using hybrid LSTM-SVR model with feature engineering techniques," Energy, vol. 216, p. 119117, 2021.
8. A. Ahmed and M. Khalid, “A review on the selected applications of forecasting models in renewable power systems,” Renewable and Sustainable Energy Reviews, vol. 100, pp. 9–21, 2021
9. A. Zendehboudi, M. A. Baseer, and R. Saidur, “Application of support vector machine models for forecasting solar and wind energy resources: a review,” Journal of Cleaner Production, vol. 199, pp. 272–285, 2022
10. C. Voyant, G. Notton, S. Kalogirou et al., “Machine learning methods for solar radiation forecasting: a review,” Renewable Energy, vol. 105, pp. 569–582, 2019.
11. https://www.analyticssteps.com/blogs/introduction-decision-tree-algorithm-machine-learning (06/04/2023)
12. https://www.javatpoint.com/linear-regression-in-machine-learning (27/04/2023)