# Study on factors affecting power output of a Combined Cycle Power Plant and prediction based on data

**A project report submitted for the partial fulfillment of the Bachelor of Technology**

**Degree in Electrical Engineering under**

**Maulana Abul Kalam Azad University of Technology**

BY

**Prateek Majumder**

(UNIVERSITY ROLL NO:10401618074, REGISTRATION NO:181040110553)

**Tannistha Chakraborty**

(UNIVERSITY ROLL NO:10401618025, REGISTRATION NO:181040110602)

**Subham Dey**

(UNIVERSITY ROLL NO:10401618035, REGISTRATION NO:181040110592)

**Mousumi Dasgupta**

(UNIVERSITY ROLL NO:10401618082, REGISTRATION NO:181040110545)

**Sayak Acharjee**

(UNIVERSITY ROLL NO:10401618052, REGISTRATION NO:181040110575)

Under the Guidances of:

**Prof. Madhumita Pal**

**Department of Electrical Engineering**

**Prof. Amartya Mukherjee**

**Department of Computer Science and Engineering (AIML)**

For the Academic Year 2021 - 22

Institute of Engineering & Management

Y-12, Salt Lake, Sector-V, Kolkata-700091

Affiliated to

Maulana Abul Kalam Azad University of Technology

BF-142, Salt Lake, Sector I, Kolkata-700064

INSTITUTE
OF ENGINEERING & MANAGEMENT
Salt Lake Electronics Complex, Kolkata- 700091, WB, INDIA

Phone : (033) 2357-2969/2059/2995
: (033) 2357-8189/8908/5389
Fax : 91-33-2357-8302
E-mail : director@iemcal.com
Website : www.iem.edu.in

# CERTIFICATE

## TO WHOM IT MAY CONCERN

This is to certify that the project report entitled "**Study on factors affecting power output of a Combined Cycle Power Plant and prediction based on data**", submitted by

1. **Prateek Majumder**
   (***Registration No.*** 181040110553 ***of 2018-2019 Roll no*** 10401618074***)***,

2. **Tannistha Chakraborty**
   (***Registration No.*** 181040110602 ***of 2018-2019 Roll no*** 10401618025***)***,

3. **Subham Dey**
   (***Registration No.*** 181040110592 ***of 2018-2019 Roll no*** 10401618035***)***,

4. **Mousumi Dasgupta**
   (***Registration No.*** 181040110545 ***of 2018-2019 Roll no*** 10401618082***)***,

5. **Sayak Acharjee**
   (***Registration No.*** 181040110575 ***of 2018-2019 Roll no*** 10401618052***)***

IEM
श्रद्धावान लभते ज्ञानम्
INSTITUTE OF ENGINEERING & MANAGEMENT
Good Education, Good Jobs

INSTITUTE
OF ENGINEERING & MANAGEMENT
Salt Lake Electronics Complex, Kolkata- 700091, WB, INDIA

Phone : (033) 2357-2969/2059/2995
: (033) 2357-8189/8908/5389
Fax : 91-33-2357-8302
E-mail : director@iemcal.com
Website : www.iem.edu.in

Students of **INSTITUTE OF ENGINEERING & MANAGEMENT,** in partial fulfilment of requirements for the award of the degree of **Bachelor of Technology in Electrical Engineering,** is a bona fide work carried out under the supervision and guidance of **Prof. Madhumita Pal & Prof. Amartya Mukherjee** during the final year of the academic session of 2018-2022. The content of this report has not been submitted to any other University or Institute for the award of any other degree.

It is further certified that work is entirely original and its performance has been found to be quite satisfactory.

Madhumita Pal

Prof. Madhumita Pal
Project Guide
Dept. of Electrical Engineering
Institute of Engineering & Management

Prof. Amartya Mukherjee
Project Guide
Dept of Computer Science and Engineering (AIML)
Institute of Engineering &Management

Prof. Tapas Kumar Dutta
H.O.D
Dept of Electrical Engineering
Institute of Engineering & Management

Institute of Engineering & Management
Sector-V, Salt Lake Electronics Complex, Kolkata-700091

# ACKNOWLEDGEMENT

We should like to take this opportunity to extend our gratitude to the following revered persons without whose immense support, completion of this project wouldn't have been possible.

We are sincerely grateful to our advisor and mentor Prof. Madhumita Pal and Prof. Amartya Mukherjee of the Electrical Engineering department, IEM Kolkata, for his/her constant support, significant insights and for generating in us a profound interest for this subject that kept us motivated during the entire duration of this project.

We would also like to express our sincere gratitude to **Prof. Dr. Satyajit Chakrabarti** (Director**,** IEM)**, Prof. Dr. Arun Kumar Bar (**Principal, IEM) and **Prof. Tapas Kumar Datta**, HOD of **Electrical Engineering Department** and other faculties of Institute of Engineering & Management, for their assistance and encouragement.

Last but not the least, we would like to extend our warm regards to our families and peers who have kept supporting us and always had faith in our work.

Prateek Majumder

**Prateek Majumder**
Reg. No: 181040110553
Dept. of Electrical Engineering
Institute of Engineering & Management, Kolkata

Subham Dey

**Subham Dey**
Reg. No: 181040110592
Dept. of Electrical Engineering
Institute of Engineering & Management, Kolkata

Tannistha Chakraborty

**Tannistha Chakraborty**
Reg. No: 181040110602
Dept. of Electrical Engineering
Institute of Engineering & Management, Kolkata

Mousumi Dasgupta

**Mousumi Dasgupta**
Reg. No: 181040110545
Dept. of Electrical Engineering
Institute of Engineering & Management, Kolkata

**Sayak Acharjee**
Reg. No: 181040110575
Dept. of Electrical Engineering
Institute of Engineering & Management, Kolkata

# ABSTRACT

A combined cycle power plant is one of the most efficient power plants that uses both gas and steam turbine to produce 50% more electricity than what a traditional simple cycle plant produces. The main aim of this paper is to predict the output of the combined cycle power plant using several machine learning algorithms. Predicting the full load electric power is extremely instrumental in order to maximize the profit from the available megawatt hours. The base load operation of this power plant is affected by four main factors- ambient temperature, atmospheric pressure, relative humidity and exhaust steam pressure. The power output which is influenced by these factors, is considered as the target variable. This paper consists of a detailed study of the data available and assess which machine learning algorithm works the best in examining the factors and predicting the output. The most accurate machine learning algorithm is found using various accuracy metrics.

# TABLE OF CONTENTS

**TOPIC** **PAGE NUMBER**

**CHAPTER 1. INTRODUCTION 1-2**

1.1. Motivation 1

1.2. Objective 1

1.3. Organization of the report 2

**CHAPTER 2. BACKGROUND AND LITERATURE REVIEW 3-5**

2.1. Combined Cycle Power Plant 3

2.2. Inner Workings of a Combined-Cycle Power Plant 3

2.3. How a Combined-Cycle Power Plant Produces Electricity 4

2.4 Advantages of Combined-Cycle Power Plant 4

2.5 Disadvantages of Combined-Cycle Power Plant 5

2.6 Literature Review 5

**CHAPTER 3. PROBLEM STATEMENT AND PROPOSED STRATEGY 6-12**

3.1. Machine Learning 6

3.2. Types of Machine Learning 6

3.2.1 Supervised Learning 6

3.2.2 Unsupervised Learning 7

3.2.3 Reinforcement Learning 7

3.3 Regression 7

3.3.1 Independent vs Dependent Variable 7

3.4 Linear Regression 8

3.5 Polynomial Regression 10

3.6 Decision Tree Regression 11

3.7 Random Forest Regression 12

**CHAPTER 4.**
**EXPERIMENTAL SETUP AND PRACTICAL IMPLEMENTATION 13-24**

4.1 About data set 13
4.2 Attribute information 13
4.3 Data preparation and Analysis 15
4.4 Regression Plots 17
4.5 Creating new Features in Data 20
4.6 Data Normalization 20
4.7 Application of Machine Learning 21-24

**CHAPTER 5. RESULTS 25-30**

5.1 Linear Regression Accuracy Metrics 25
5.2 Polynomial Regression Accuracy Metrics 26
5.3 Decision Tree Regression Accuracy Metrics 27
5.4 Random Forest Regression Accuracy Metrics 28
5.5 Comparative Study 29

**CHAPTER 6. CONCLUSION 30**
**REFERENCES 31**

**APPENDIX 32-35**

# CHAPTER 1
# INTRODUCTION

## 1.1 Motivation

Computer Science has its roots from Electrical Engineering because our computer hardware consists of a lot of elements which would not have been possible to make without the knowledge of Electrical concepts. As Electrical Engineering students, we decided to develop a project by combining concepts of Electrical Engineering and Computer Science, specifically, Machine Learning to diversify the spectrum of both the subjects and find out new possibilities and dimensions with this combination.

A combined-cycle power plant combines a gas and steam turbine to generate up to 50% more energy from the same fuel as a standard simple-cycle plant. The gas turbine's waste heat is sent to a neighboring steam turbine, which provides additional power.

Many real-life problems can be solved as regression problems, and evaluated using machine learning approaches to develop predictive models [1]. To reflect the system's true uncertainty, these models include a wide variety of assumptions and parameters. Machine learning predictions include a lot of mistakes, thus different methods are evaluated to see which one has the least. The main goal of this study is to look at the dataset using four different techniques and figure out which one has the best R2 score and the smallest mean absolute error. Also, to estimate the total electric power that a gas turbine could create.

## 1.2 Objective

The power output from the Gas Turbine is dependent on factors like Ambient Temperature, Atmospheric Pressure, Relative Humidity and the power output from the Steam Turbine is dependent on the vacuum output at exhaust. Power forecasting is critical for a combined cycle power plant's smooth and cost-effective operation. Furthermore, effective forecasting aids in the reduction of a variety of connected concerns such as power outages, among others. The cost of electric electricity per unit rises as a result of inaccurate forecasting.

The main objective behind our project is to:

- Studying these factors on which Power outputs of a Combined Cycle Power Plant are dependent.
- Analyzing the data distribution and correlations in data.
- Training the Machine Learning models with data and making predictions
- Finding out the accuracy metrics of all the models to deduce the best suited Machine Learning algorithm with best R2 score, least Root Mean Square Error (RMSE) and Mean Absolute Error.

## 1.3 Organization of Report

### Chapter 1. BACKGROUND

Understanding the overall nature of Combined Cycle Power Plant and figuring out the project roadmap.

### Chapter 2. PROBLEM STATEMENT AND PROPOSED STRATEGY

Then, in course of action, the report deals with the various algorithms. The accuracy metrics has been calculated and the results have also been focused on. Normalization, which has also been performed, has been mentioned in the report.

### Chapter 3. EXPERIMENTAL SETUP AND PRACTICAL IMPLEMENTATION

The data source has been provided and the complete Machine Learning pipeline has been explained with all the intricate details.

### Chapter 4. RESULTS

Finally, we will be able to see the details of various accuracy scores that has been evaluated with the help of various Machine Learning algorithms. This section will help the reader to intricately understand the solutions of the mentioned problem.

### Chapter 5: CONCLUSION AND REFERENCE

Finally, we have concluded the report and the references which have been used to complete the project successfully, have been mentioned.

# CHAPTER 2
# BACKGROUND AND LITERATURE REVIEW

## 2.1 Combined-Cycle Power Plant

A gas turbine drives an electrical generator in a combined-cycle power system, which then recovers waste heat from the turbine exhaust to make steam. To generate additional power, waste heat steam is passed through a steam turbine. A combined-cycle power system's overall electrical efficiency is generally in the region of 50–60 percent, which is a significant improvement over the 33 percent efficiency of a standard open-cycle application [2].

The phrase "combined cycle" refers to the power generated by merging different thermodynamic cycles. A heat recovery steam generator (HRSG) collects heat from high-temperature exhaust gases to make steam, which is subsequently delivered to a steam turbine to generate extra electric power in a combined cycle operation.

The functioning of a combined-cycle system is acceptable for applications with steady load profiles, while it is less suitable for applications with changing or falling load profiles.

## 2.2 Inner workings of a Combined-Cycle Power Plant

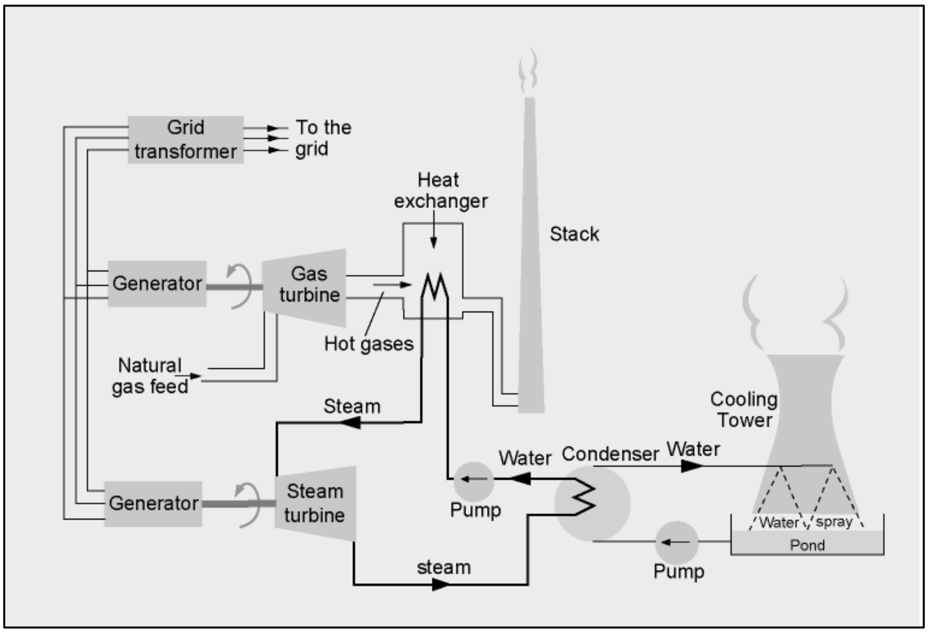


*Figure 1: Basic Layout of Combined Cycle Power Plant*

The HRSG is essentially a heat exchanger, or a set of heat exchangers. It's also known as a boiler because it generates steam by transferring hot exhaust gas from a gas turbine or combustion engine through banks of heat exchanger tubes. The heat lost from the gas turbine will be captured in the Heat Recovery Steam Generator (HRSG) [3]. Natural circulation or forced circulation utilizing pumps are also options for the HRSG. Heat is absorbed as hot exhaust gases pass through heat exchanger tubes in which hot water flows, resulting in the formation of steam in the tubes. The tubes are organized into parts, or modules, with each performing a specific purpose in the creation of dry superheated steam. Economizers, evaporators, superheaters/reheaters, and preheaters are the names for these modules.

In a combined cycle system, a gas turbine does not only produce electricity but also produces a lot of heat exhaust. Using a water-cooled heat exchanger to transport these gases create steam, which may be converted into electricity using a generator. Steam turbine and generator are connected. As a result, a gas turbine generator is used. creates power and utilizes the waste heat from exhaust gases used to create steam in order to generate additional electricity via a steam turbine. When the ambient temperature rises, the net power generated in the combined-cycle thermal-plant decreases [4].

The economizer is a heat exchanger that preheats water to near saturation temperature (boiling point) before supplying it to a steam drum with thick walls. The drum is near the finned evaporator tubes, which circulate hot water. Heat is absorbed as hot exhaust gases pass through the evaporator tubes, resulting in the formation of steam in the tubes. The steam-water mixture in the tubes enters the steam drum, where moisture separators and cyclones separate the steam from the hot water. The evaporator tubes are recirculated with the separated water. Steam drums are also used for water treatment and storage. A gas turbine generator generates electricity and waste heat of the exhaust gases is used to produce steam to generate additional electricity via a steam turbine [5].

The superheater uses saturated steam from the steam drums or once-through system to create dry steam, which is needed for the steam turbine. Preheaters are situated at the coldest end of the HRSG gas path and absorb energy to preheat heat exchanger liquids such as water/glycol mixes, allowing the highest economically possible quantity of heat to be extracted from exhaust gases.

The HRSG produces superheated steam, which is fed into the steam turbine, where it expands through the turbine blades, causing the turbine shaft to rotate. Electricity is generated from the energy provided to the generator driving shaft. After departing the steam turbine, the steam is delivered to a condenser which directs the condensed water back to the HRSG.

## 2.3. How a Combined-Cycle Power Plant Produces Electricity

- **Gas Turbine burns fuel**

The fast-spinning turbine drives a generator that conveys a portion of the spinning energy into electricity.

- **Heat recovery system captures exhaust**

The HRSG creates steam from the gas turbine exhaust heat and delivers it to the steam turbine

- **Steam turbine delivers additional electricity**

The steam turbine sends its energy to the generator drive shaft, where it is converted into additional electricity.

## 2.4 Advantages of Combined-Cycle Power Plant

- The efficiency of the combined cycle plant is better or higher than the turbine cycle or steam cycle plant. The efficiency of combined cycle power plant will be of the order of about 45 to 50%.
- The amount of cooling water required is just approximately 40% to 50% of what a steam plant requires.
- The simple steam cycle allows for speedy start-up and shut-down of the plants, which improves efficiency (reducing start-up losses).
- Because gas turbines can start up considerably faster than a steam plant, installation may be done in phases. While the steam plant is being built, the gas turbine plant may continue to provide electricity. This allows for the adjustment of energy demand increase in a grid. If the price of oil or gas rises too quickly, a coal gasification plant can be installed later.

- Fewer moving parts and less vibration than a reciprocating engine.
- Very low toxic emissions.
- Runs on a wide variety of fuels.
- High operating speed.

## 2.5 Disadvantages of Combined-Cycle Power Plant

- Higher costs of setting up and operating.
- This power plant is incredibly complicated to construct. That implies it must be properly created and tested.
- They have high maintenance costs. Every part should be checked on a regular basis.
- They are less responsive to changes in Power Demand.

## 2.6 Literature Review

The main motivation for this study is that there exist thermodynamical studies to predict the output of a CCPP. To check if these outliers are influential to the model fit, a linear model is considered excluding these data points. Power prediction is important not only for the smooth and economic operation of a combined cycle power plant (CCPP) but also to avoid technical issues such as power outages. In this work, we propose to utilize machine learning algorithms to predict the hourly-based electrical power generated by a CCPP. The utilized machine algorithms are -Linear Regression, Polynomial Regression, Decision Tree Regression, Random Forest Regression. We report state-of-the-art performance where GBRT outperforms not only the utilized algorithms but also all the previous methods on the given CCPP dataset. Thus, it is important to predict the power to increase and maximize profit. This paper compares four machine learning algorithms which are Linear Regression, Polynomial Regression, Decision Tree Regression, Random Forest Regression. The PE predictions derived by applying the TOB optimized data matching technique are more accurate than published predictions for the dataset from fifteen correlation-based, machine-learning algorithms.

# CHAPTER 3

# PROBLEM STATEMENT AND PROPOSED STRATEGY

## 3.1 Machine Learning

Machine Learning (ML) is the study of computer algorithms that can improve automatically through experience and by the use of data [6]. It is seen as a part of artificial intelligence.

A Machine Learning system learns from previous data, constructs prediction models, and predicts the result whenever fresh data is received. The amount of data helps to construct a better model that predicts the output more precisely, hence the accuracy of anticipated output is dependent on the amount of data. In our study, we have to deal with a lot of data to make predictions, hence in this scenario, Machine Learning is a good application.

If we have a complex situation for which we need to make predictions, rather than creating code for it, we may just input the data to generic algorithms, and the machine will develop the logic based on the data and forecast the outcome. Machine learning has shifted our perspective on the issue.

A block diagram of a Machine Learning problem is as follows:

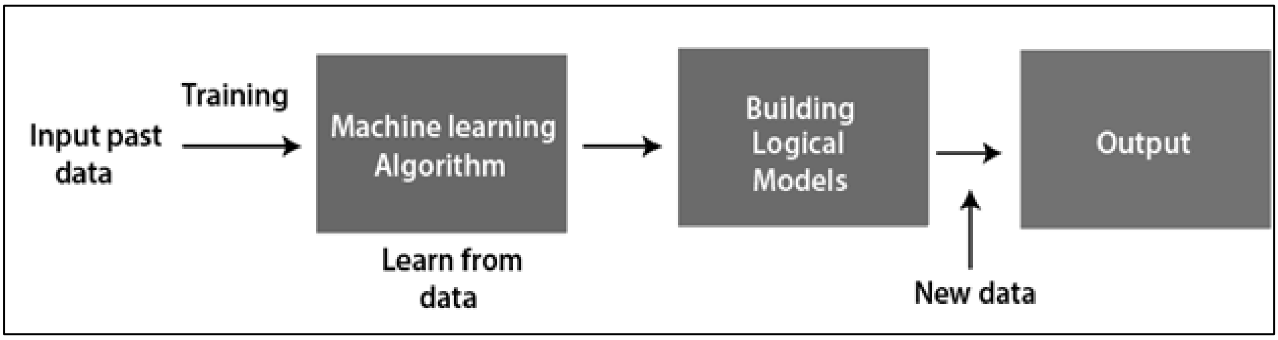


*Figure 2: Machine Learning Workflow*

We shall be using Machine Learning algorithms to perform the prediction task at hand.

We can train machine learning algorithms by feeding them massive amounts of data and allowing them to autonomously examine the data, build models, and anticipate the desired output. The cost function can determine the performance of the machine learning algorithm, which is dependent on the amount of data. We can save both time and money by using machine learning.

## 3.2 Types of Machine Learning

Machine learning may be divided into three categories on a general level:

- Supervised learning
- Unsupervised learning
- Reinforcement learning

### 3.2.1 Supervised Learning:

Supervised learning is a form of machine learning approach in which we provide the machine learning system sample labelled data to train it on, and it then predicts the result.

The system constructs a model using labelled data to interpret the datasets and learn about each data; once the training and processing are completed, we test the model by supplying sample data to see if it accurately predicts the output [7].

The purpose of supervised learning is to connect the input and output data. The basis of supervised learning is supervision, and it is similar to when a student learns under the observation of a teacher. Spam filtering is one example of supervised learning.

Supervised learning can be grouped further in two categories of algorithms:

- Classification
- Regression

### 3.2.2 Unsupervised Learning:

Unsupervised learning is a type of learning in which a machine learns without any human intervention. The machine is taught given a collection of data that hasn't been labelled, classified, or categorized, and the algorithm is expected to operate on it without supervision. Unsupervised learning aims to reorganize incoming data into new features or a collection of objects with similar patterns.[8]

### 3.2.3 Reinforcement Learning:

Reinforcement learning is a feedback-based learning strategy in which a learning agent is rewarded for correct actions and punished for incorrect ones. With these feedbacks, the agent learns automatically and improves its performance. The agent interacts with and investigates the environment in reinforcement learning. An agent's purpose is to earn the greatest reward points, so it enhances its performance.

This project will be regression challenge. Let us try to understand what is Regression.

## 3.3 Regression:

The supervised Machine Learning method regression may be used to predict continuous data. The technique of detecting relationships between independent and dependent data is known as regression. The independent variables (x) are mapped to their dependent variables in regression (y). Regression may be used to forecast rainfall, estimate property values, and estimate stock prices, among other things. When working with a data set of real numbers, regression procedures are commonly utilized.

### 3.3.1 Independent vs Dependent variable:

The term "independent variable" means precisely what it says. It's a stand-alone variable that isn't affected by the other variables you're attempting to track. A dependent variable is dependent on a variety of things. Let us say, we are trying to understand the salary of a person on the basis of his/her years of experience in a job. So, we shall be having data of salary and years of experience. In this case, the years of experience in the job will be independent variable, and the salary will be the dependent variable, which depends on the years of experience.

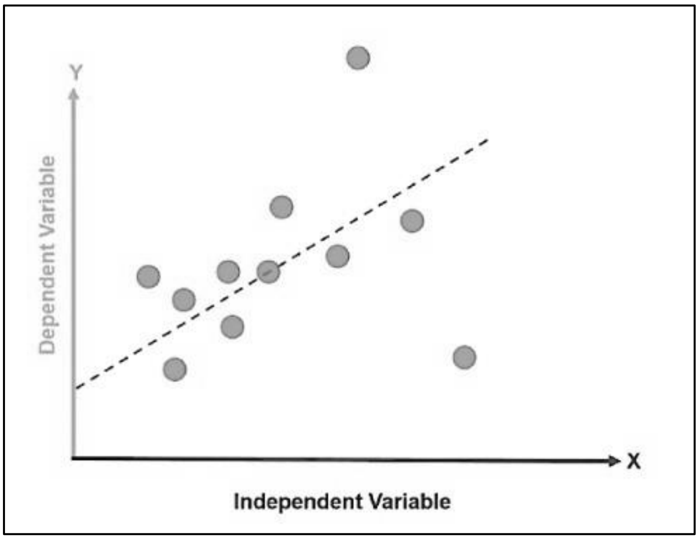


*Figure 3: Regression Overview*

We shall be working with four different regression algorithms for this project. They are as follows:

1. Linear Regression
2. Polynomial Regression
3. Decision Tree Regression
4. Random Forest Regression

## 3.4 Linear Regression

Linear regression is a supervised learning machine learning technique. It carries out a regression job. Based on independent variables, regression models a goal prediction value. It is mostly utilized in predicting and determining the link between variables. Different regression models differ in terms of the type of relationship they evaluate between dependent and independent variables, as well as the number of independent variables they employ [9].

The relationship between the variables may be represented in the following way since the model is used to predict the dependent variable.

$\boldsymbol{Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i}$

| Variable | Description |
|---|---|
| Yi | Dependent variable |
| β0 | Intercept |
| β1 | Slope Coefficient |
| Xi | Independent Variable |
| εi | Random Error Term |

Multiple Linear Regression is used when the association between independent and dependent variables is multiple in number.

The formula for Multiple Linear Regression is:

**$y= b_0+b_1x+ b_2x_2+ b_3x_3+.... + b_nx_n$**

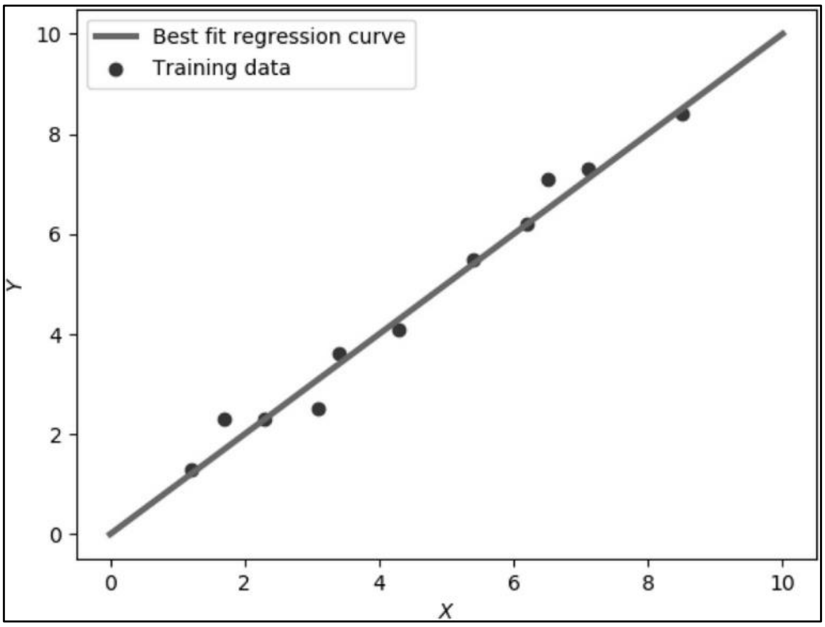


*Figure 4: Linear Regression*

When using linear regression, our major aim is to identify the best fit line, which implies that the difference between projected and actual values should be as little as possible. The line with the best fit will have the least amount of inaccuracy. Large mistakes are punished quadratically in linear regression because the least squares error of the model to the data is generally minimized.

The following is the loss function:

$$MSE = 1/N \sum_{i=1}^{N} (fi - yi)^2$$

Where:

- N is the number of datapoints
- fi is the value returned by the model
- yi is the actual value for the datapoint i

## 3.5 Polynomial Regression

Polynomial Regression is a regression approach that uses an nth degree polynomial to represent the connection between a dependent(y) and independent variable(x). In machine learning, it's also known as the specific case of Multiple Linear Regression. Because we turn the Multiple Linear regression equation into Polynomial Regression by adding certain polynomial terms [10].

The Polynomial Regression equation is as follows:

$$y= b_0+b_1x_1+ b_2x_1^2+ b_2x_1^3+\ldots\ldots\ b_nx_1^n$$

To fit the intricate and non-linear functions and datasets, it employs a linear regression model. As a result, in Polynomial regression, the original data are transformed into Polynomial features of the desired degree (2, 3 ,….n) and then modelled with a linear model.

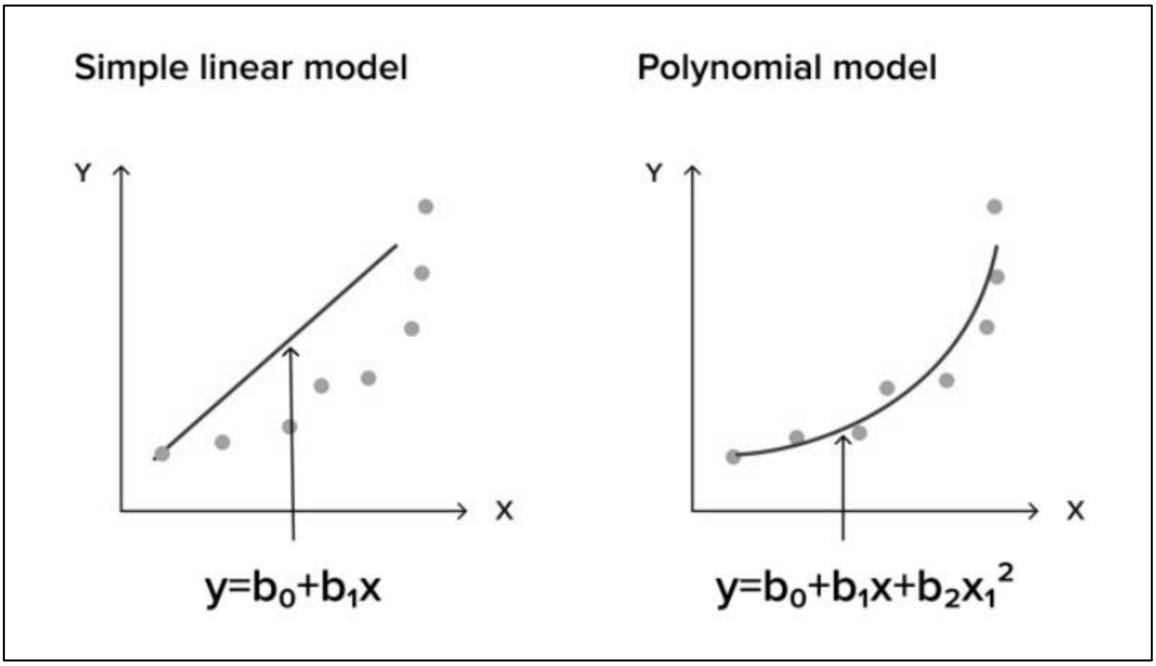


*Figure 5: Polynomial Regression*

Polynomial regression is beneficial in a variety of situations. Because a linear relationship between the independent and dependent variables isn't essential, you have more options for datasets and circumstances to deal with. When basic linear regression underfits the data, this approach might be used.

## 3.6 Decision Tree Regression

Decision Tree is a decision-making tool that employs a flowchart-like tree structure or is a model of decisions and all of their possible consequences, including outcomes, input costs, and utility. The decision-tree method is classified as a supervised learning algorithm. It works for both continuous and categorical output variables. So, decision trees are used for both, classification and regression tasks [11]. A regression tree is essentially a decision tree that is used for regression and may predict continuous valued outputs rather than discrete outputs.

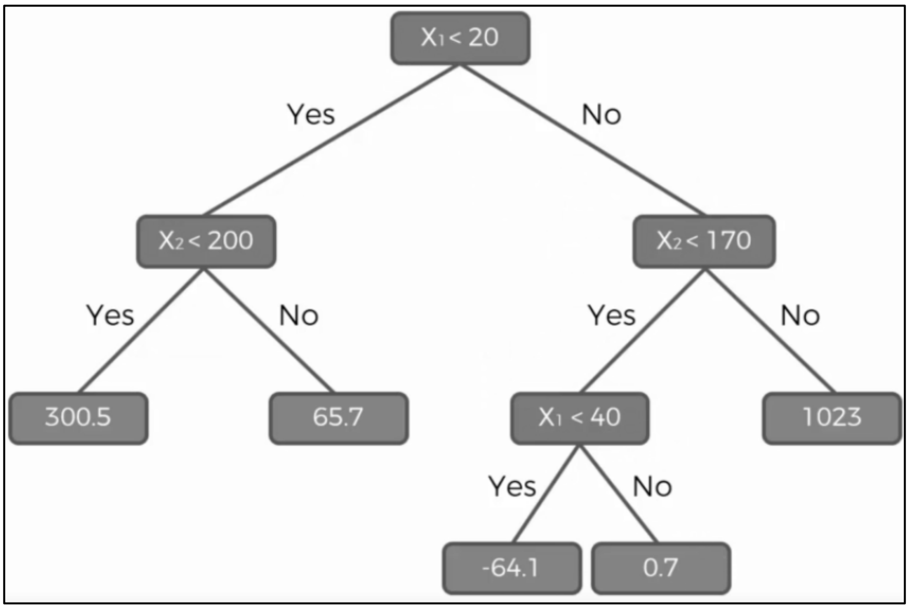


*Figure 6: Decision Tree Regression*

The branches/edges represent the result of the node and the nodes have either:

1. Conditions [Decision Nodes]
2. Result [End Nodes]

Decision tree regression evaluates an object's attributes and trains a model with a tree structure to forecast data in the future to create meaningful continuous output.
Following the training phase, the decision tree generates a tree similar to the one shown above, determining the best questions to ask as well as the sequence in which they should be asked in order to create the most accurate estimations possible. When we wish to create a forecast, we should supply the model with the same data structure. The forecast will be an estimate based on the train data used to train it.

The decision to make strategic splits has a significant impact on a tree's accuracy. The decision criteria for classification and regression trees differ. To decide whether to break a node into two or more sub-nodes, decision trees regression often uses mean squared error (MSE).
If we are doing a binary tree, the method will first select a value and divide the data into two subsets. It will compute the MSE independently for each subgroup. The tree selects the value that produces the least MSE value.

## 3.7 Random Forest Regression

Random Forest is a well-known machine learning algorithm from the supervised learning approach. It may be applied to both classification and regression issues in machine learning. It is built on the notion of ensemble learning, which is a method that involves integrating several classifiers to solve a complicated issue and enhance the model's performance [12].

Every decision tree has a significant variance, but when we mix all of them in parallel, the final variance is minimal since each decision tree is completely trained on that specific sample data, and so the outcome is dependent on numerous decision trees rather than one. In the event of a classification problem, the majority voting classifier is used to determine the final result. The ultimate result of a regression problem is the mean of all the outputs. This process is called Aggregation.

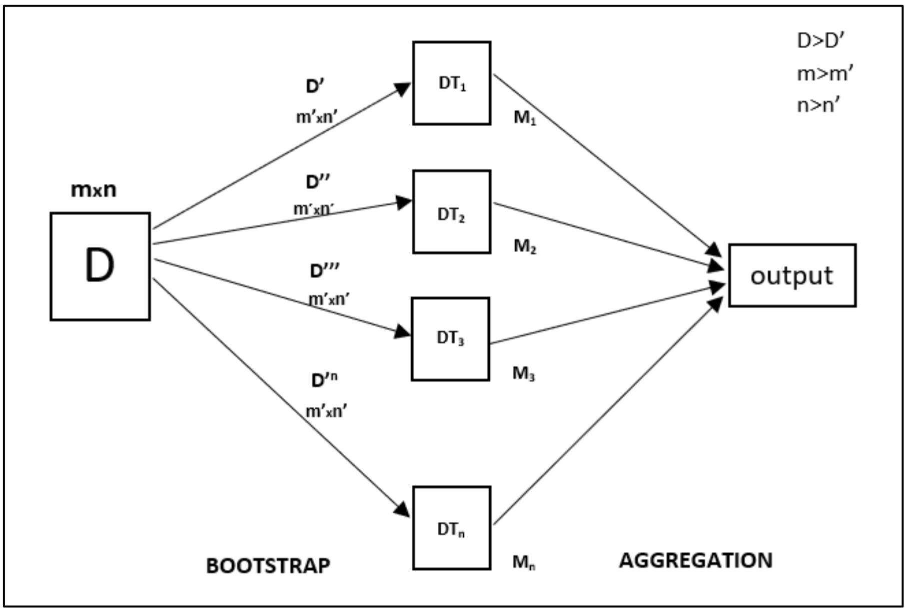


*Figure 7: Random Forest Regression*

Random Forest's foundation learning models are numerous decision trees. We randomly choose rows and features from the dataset to create sample datasets for each model. This section is known as Bootstrap.

# CHAPTER 4
# EXPERIMENTAL SETUP AND PRACTICAL IMPLEMENTATION

## 4.1 About data set

The dataset contains 9568 data points collected from a Combined Cycle Power Plant over 6 years (2006-2011), when the power plant was set to work with full load. Features consist of hourly average ambient variables Temperature(T), Ambient Pressure (AP), Relative Humidity (RH) and Exhaust Vacuum (V) to predict the net hourly electrical energy output (EP) of the plant [13].

These factors have an impact on the correct operation of a combined cycle power plant, causing power fluctuations. A gas turbine and a steam turbine make up a combined cycle power plant. The ambient conditions affect the gas turbine, whereas the exhaust steam pressure affects the steam turbine. The characteristics are measured and recorded using sensors positioned throughout the plant.

## 4.2 Attribute information

The combined cycle power plant (CCPP) is composed of gas turbines (GT), steam turbines (ST) and heat recovery steam generators. In a CCPP, the electricity is generated by gas and steam turbine, which are combined in one cycle, and is transferred from one turbine to another. While the vacuum is collected from and has effect on the Steam Turbine, the other three of the ambient variable effect the GT performance.

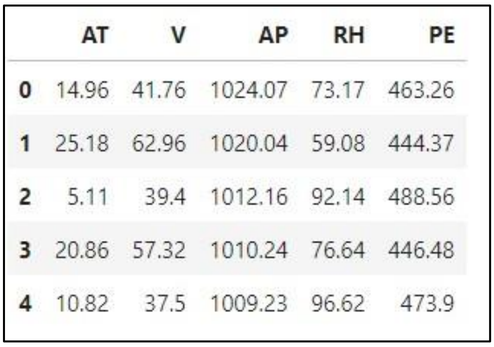
Features consist of hourly average ambient variables

- Ambient Temperature (T) in the range 1.81 C and 37.11 C
- Ambient Pressure (AP) in the range 992.89 – 1033.30 millibar,
- Relative Humidity (RH) in the range 25.56% to 100.16%
- Exhaust Vacuum (V) in the range 25.36-81.56 cm Hg
- Net hourly electrical energy output (EP) 420.26 – 495.76 MW

The averages are taken from various sensors located around the plant that record the ambient variables every second. The variables are given without normalization.

So, the temperature, ambient pressure, relative humidity, exhaust vacuum is going to be input features that is X, and the net hourly electrical energy are going to be the output features, y.

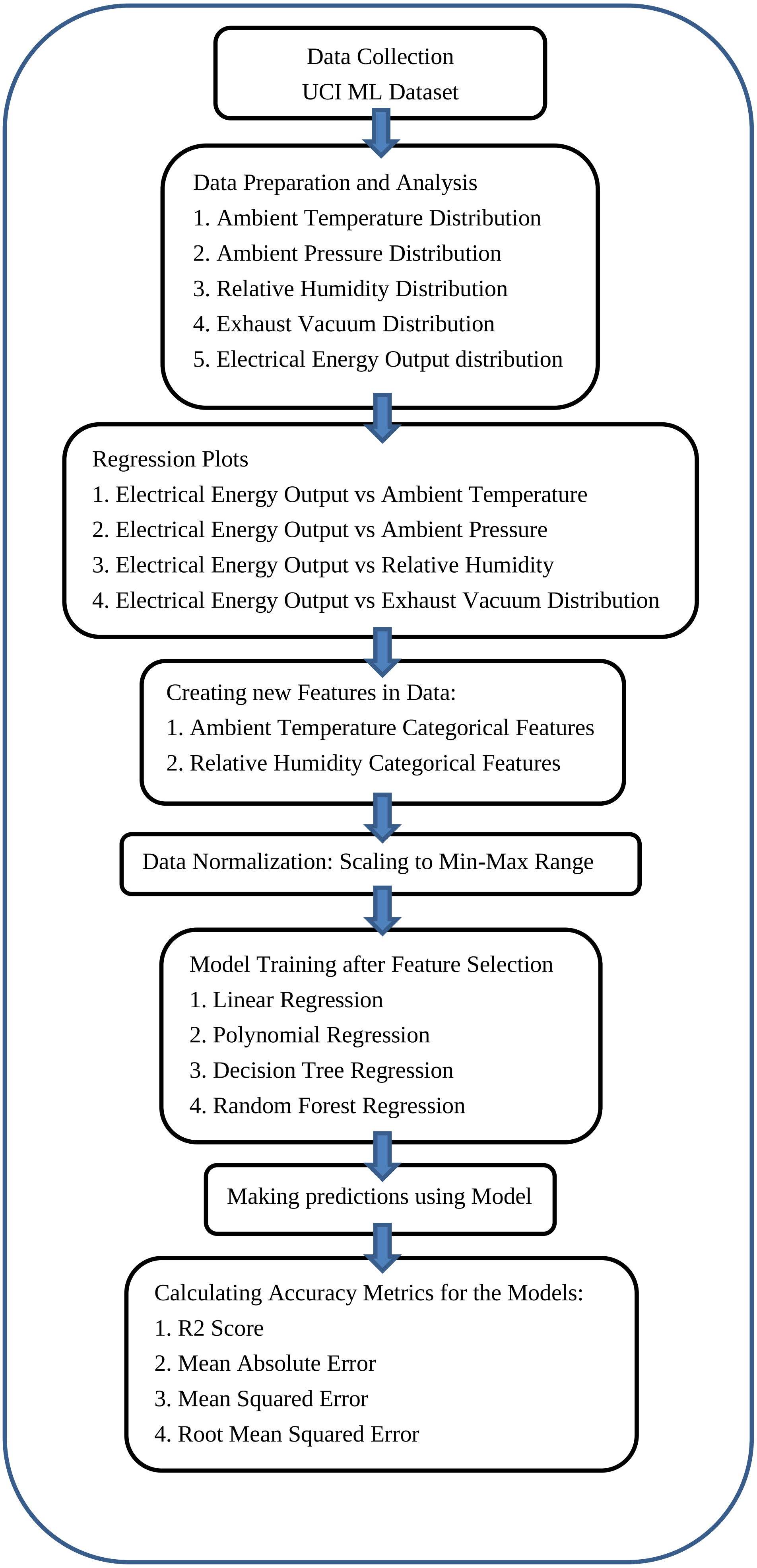
This is how a snippet of the data looks like:



| | AT | V | AP | RH | PE |
|---|---|---|---|---|---|
| **0** | 14.96 | 41.76 | 1024.07 | 73.17 | 463.26 |
| **1** | 25.18 | 62.96 | 1020.04 | 59.08 | 444.37 |
| **2** | 5.11 | 39.4 | 1012.16 | 92.14 | 488.56 |
| **3** | 20.86 | 57.32 | 1010.24 | 76.64 | 446.48 |
| **4** | 10.82 | 37.5 | 1009.23 | 96.62 | 473.9 |

*Figure 8: Data Snippet*

## Work Flowchart:


Data Collection
UCI ML Dataset
Data Preparation and Analysis
1. Ambient Temperature Distribution
2. Ambient Pressure Distribution
3. Relative Humidity Distribution
4. Exhaust Vacuum Distribution
5. Electrical Energy Output distribution
Regression Plots
1. Electrical Energy Output vs Ambient Temperature
2. Electrical Energy Output vs Ambient Pressure
3. Electrical Energy Output vs Relative Humidity
4. Electrical Energy Output vs Exhaust Vacuum Distribution
Creating new Features in Data:
1. Ambient Temperature Categorical Features
2. Relative Humidity Categorical Features
Data Normalization: Scaling to Min-Max Range
Model Training after Feature Selection
1. Linear Regression
2. Polynomial Regression
3. Decision Tree Regression
4. Random Forest Regression
Making predictions using Model
Calculating Accuracy Metrics for the Models:
1. R2 Score
2. Mean Absolute Error
3. Mean Squared Error
4. Root Mean Squared Error

## 4.3 Data preparation and Analysis

First, the data is taken. Then, the null values of the data are removed. We have used Python programming language for the project. An important step here is understanding the data distribution. We shall be plotting some distribution plots to understand the data distribution.

### Ambient Temperature Distribution:

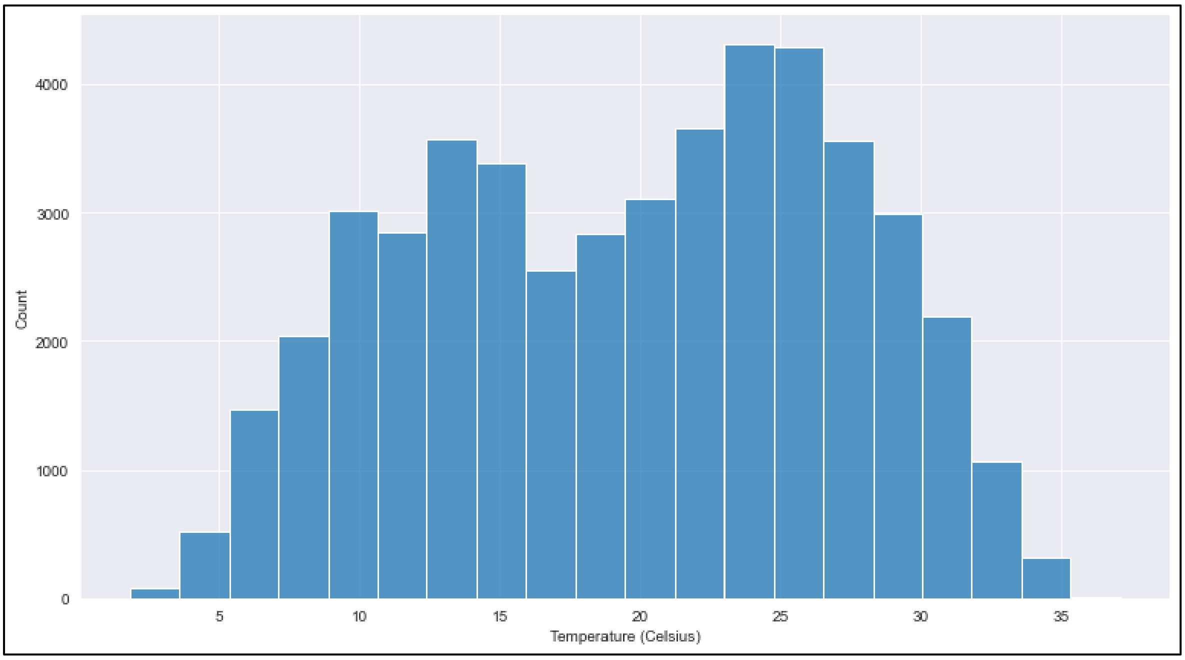


*Figure 9: Distribution of Ambient Temperature*

### Ambient Pressure (Millibar) Distribution:

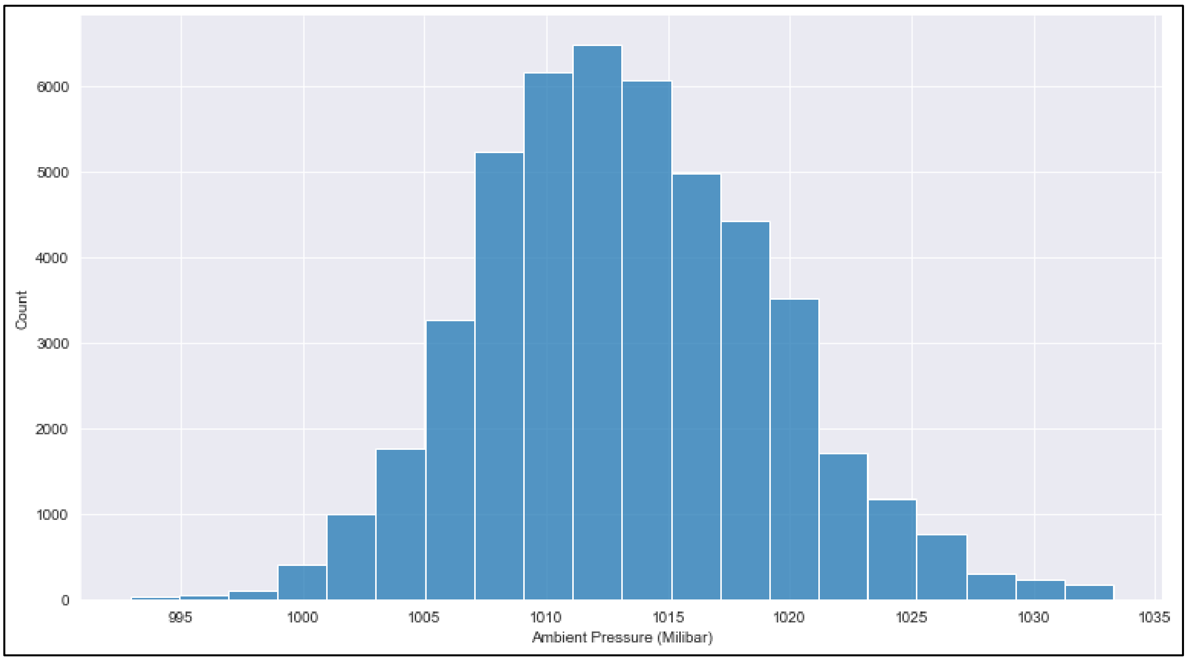


*Figure 10: Ambient Pressure (Millibar) Distribution*

## Relative Humidity (Percentage) Distribution:

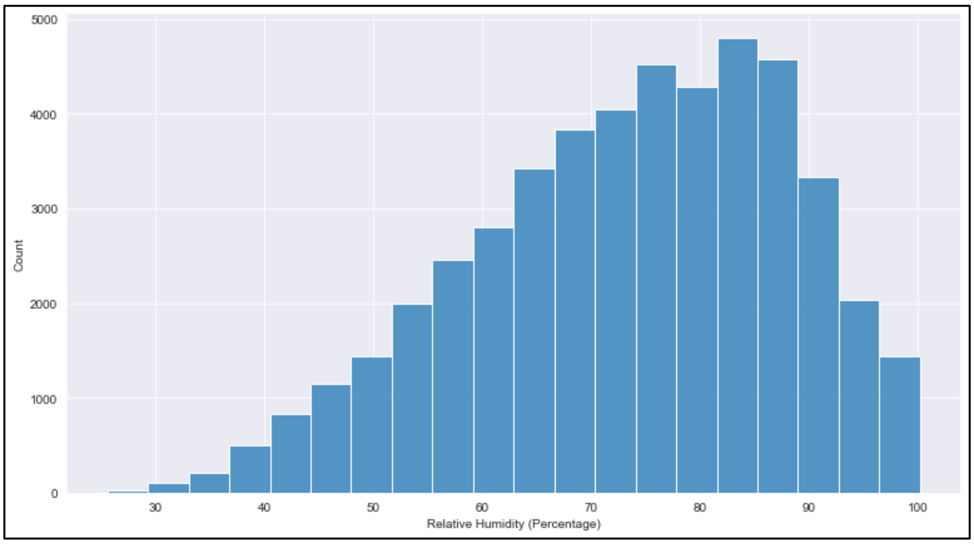


*Figure 11: Relative Humidity (Percentage) Distribution*

## Exhaust Vacuum (Cm of Hg) Distribution:

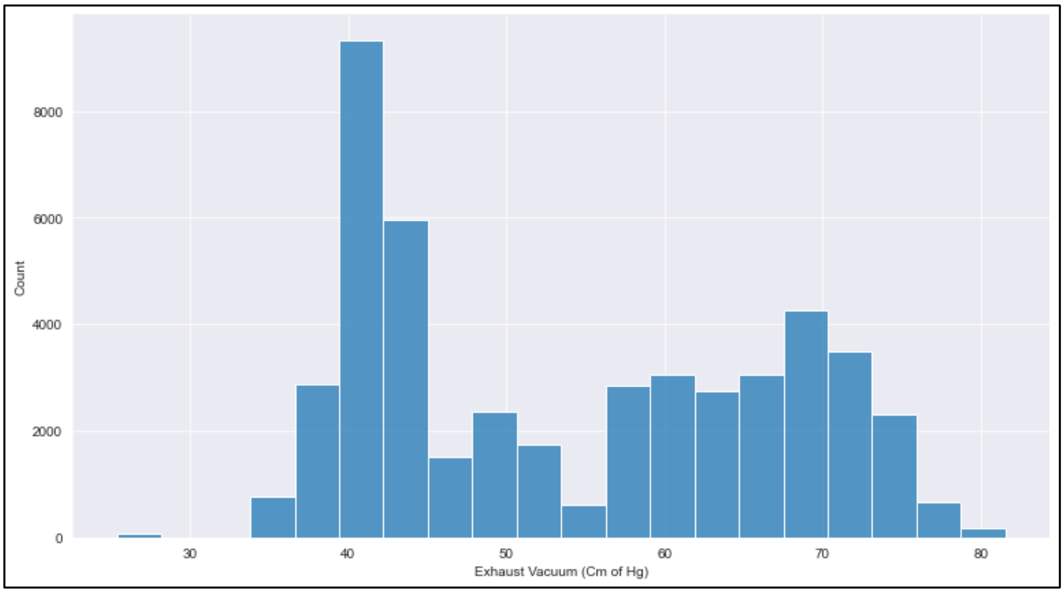


*Figure 12: Exhaust Vacuum (Cm of Hg) Distribution*

### Net hourly electrical energy output (MW) Distribution:

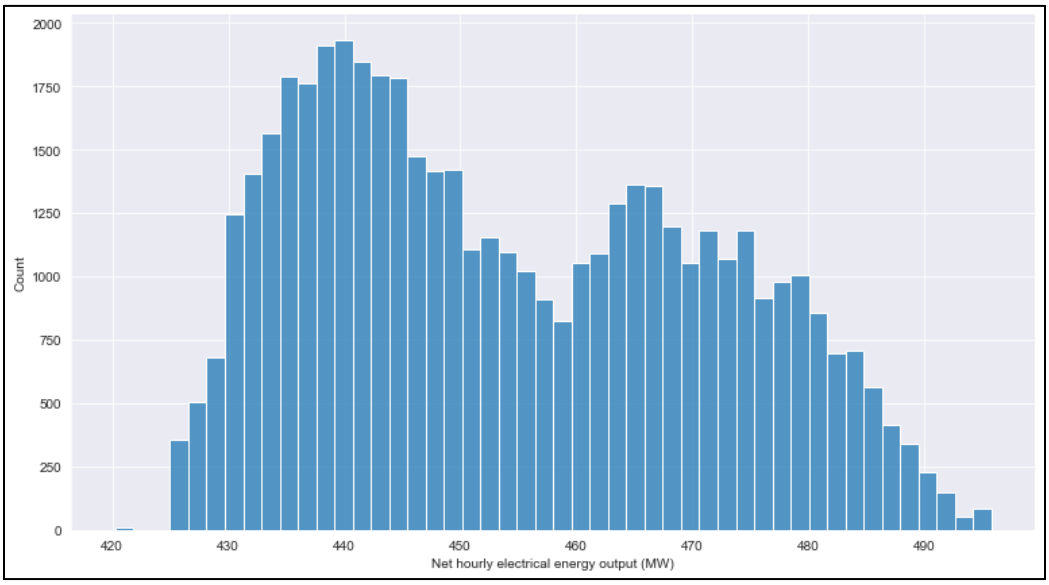


*Figure 13: Net hourly electrical energy output (MW) Distribution*

So, we can see that the data is widely distributed.

Let us plot a heatmap to check the correlation between the data features.

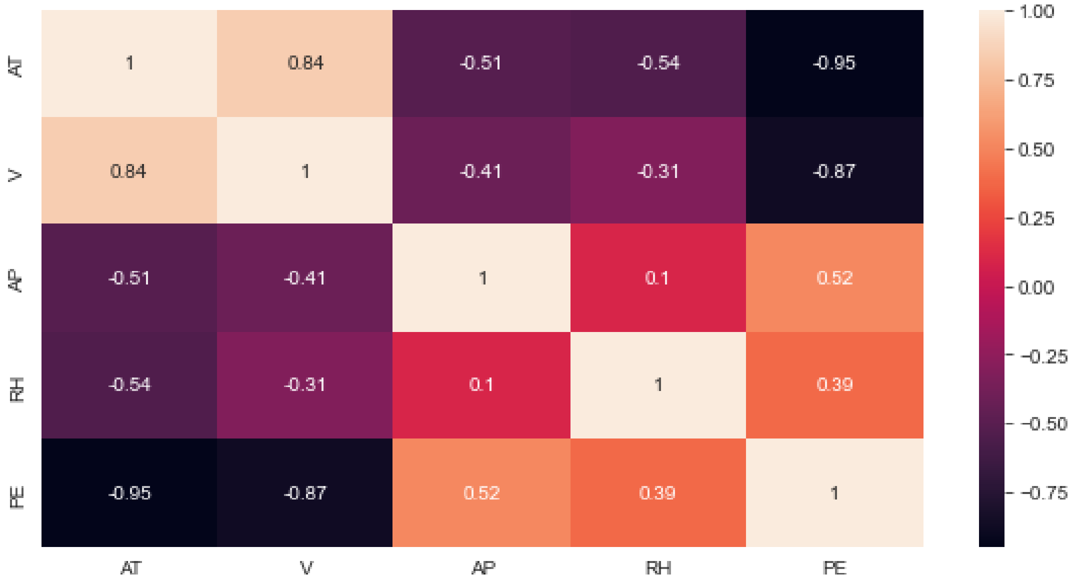


*Figure 14: Heatmap of the correlation between the data features*

We can check that a few features are highly negatively related to each other. Energy output and ambient temperature have high negative correlation. This means an increase in AT will cause a fall in PE and so on.

## 4.4. Regression Plots

As the problem at hand is a regression problem, we must understand the correlation of the input data with the output data. For this purpose, we will plot a scatterplot with all data points and plot a linear regression line. This way, we shall be able to see any patterns in the data.

## Net hourly electrical energy output (MW) vs Temperature (Celsius):

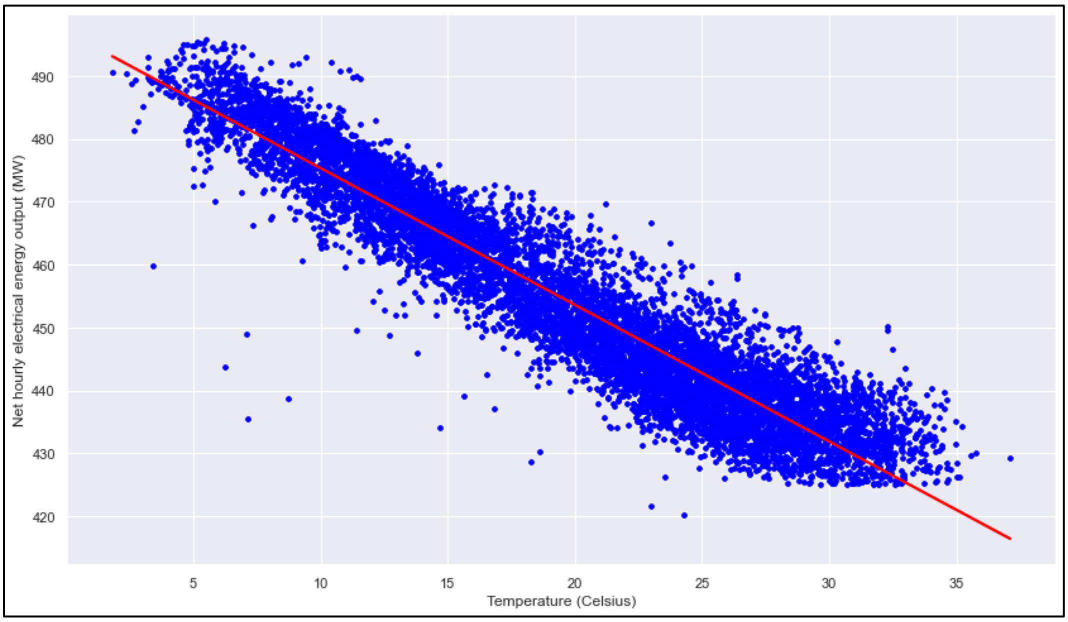


*Figure 15: Net hourly electrical energy output (MW) vs Temperature (Celsius) Regression Plot*

Relation of Energy Output compared to Temperature:

*y= -2.1713x + 497.03*

Ambient Temperature has the greatest influence on the operation of a Combined Cycle Power Plant. We can see that the increase in temperature causes electrical energy output to drop. Thus, they are negatively related with each other. They have a correlation of -0.95, which is a very high negative correlation.

## Net hourly electrical energy output (MW) vs Ambient Pressure (Millibar):

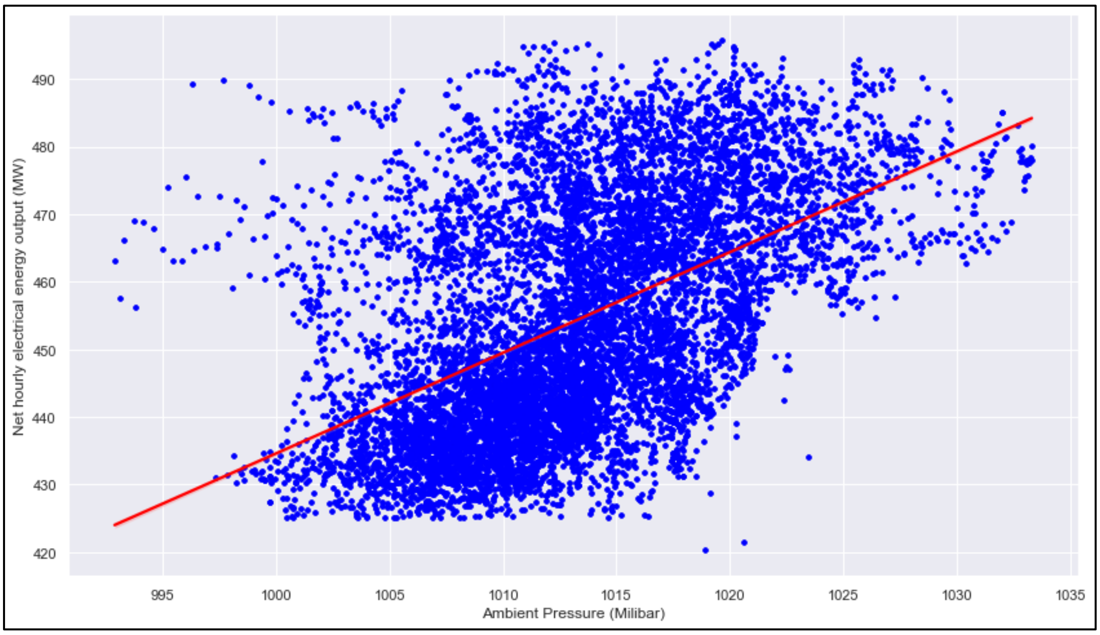


*Figure 16: Net hourly electrical energy output (MW) vs Ambient Pressure (Millibar)*

Relation of Energy Output compared to Ambient Pressure:

*y = 1:4335x – 998:78*

Ambient Pressure is the second most significant of the ambient variables. It does not, however, have a high enough correlation with the objective variable to make an individual forecast. We can see that, with the increase in Ambient Pressure, there is increase in Net hourly electrical energy output.

## Net hourly electrical energy output (MW) vs Relative Humidity (Percentage):

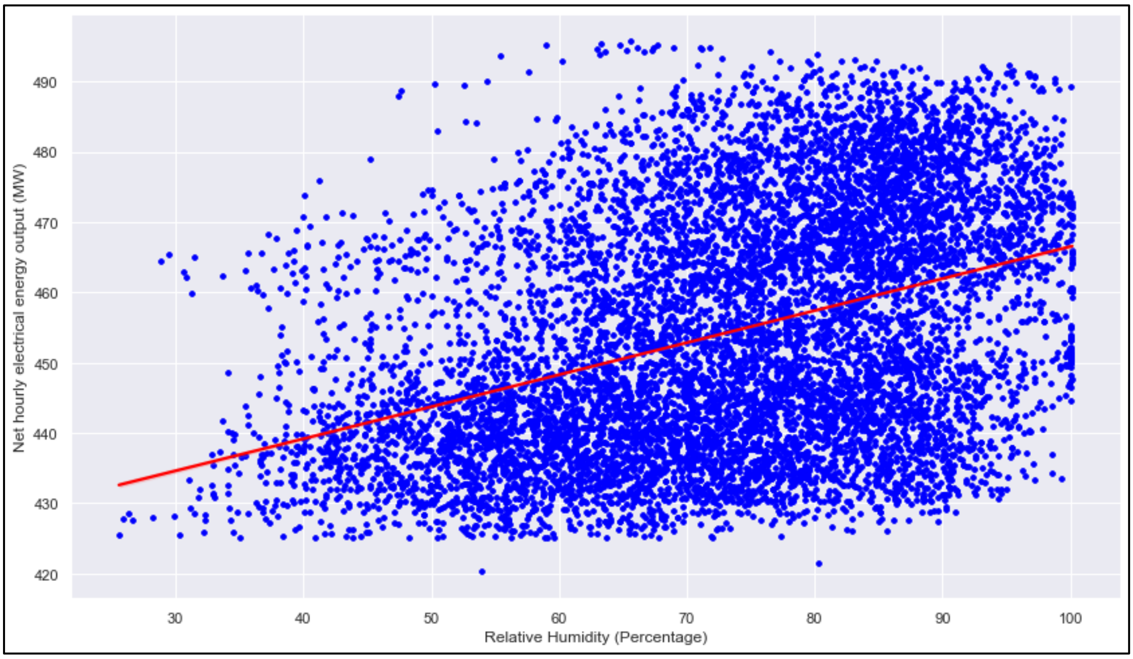


*Figure 17: Net hourly electrical energy output (MW) vs Relative Humidity (Percentage)*

Relation of Energy Output compared to Relative Humidity:

*y =0.4556x + 420:96*

Higher relative humidity raises the temperature of the gas turbine's exhaust gas, resulting in more power generated by the steam turbine. The data is widely scattered and no clear relation can be deduced from this chart.

## Net hourly electrical energy output (MW) vs Exhaust Vacuum (cm of Hg):

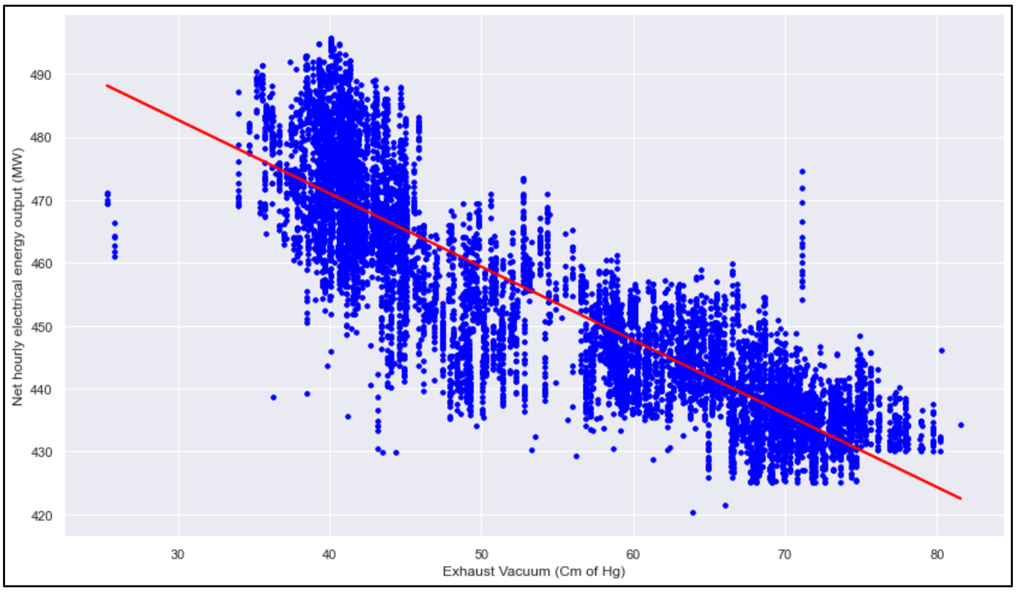


*Figure 18: Net hourly electrical energy output (MW) vs Exhaust Vacuum (Cm of Hg)*

Relation of Energy Output compared to Exhaust Vacuum (Cm of Hg):

*y = - 1.1681x + 517:8*

A steam turbine is also used in the facility, resulting in a significant boost in total electricity efficiency. The V variable (exhaust vacuum in cm Hg) is gathered from the steam turbine and discovered to have an impact on its performance. Exhaust-vacuum is known to have a detrimental impact on condensing-type turbine efficiency when all other factors are held constant. Vacuum is inversely proportional to PE.

The regression plots give us a basic idea about the relationship between the data features.

## 4.5. Creating new Features in Data

The ambient temperature and relative humidity are present in the data as continuous features. We can create some new categorical features from this numeric continuous data. We shall create three new classes of temperature and four new classes of relative humidity.

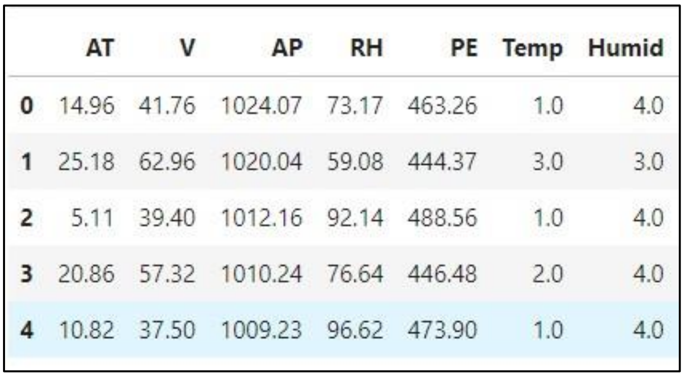
### Class Demarcations (Temperature):

| Temperature | Class |
|---|---|
| Temp<15 C | Class 1- Cool Temperature |
| Temp>=15 C and Temp< 25 C | Class 2- Normal Temperature |
| Temp>=25 C | Class 3- Warm Temperature |

### Class Demarcations (Relative Humidity):

| Relative Humidity (%) | Class |
|---|---|
| RH <30 | Class 1- Dry |
| RH >=30 and RH <50 | Class 2- Less Dry |
| RH >=50 and RH <70 | Class 3- Moderate Humidity |
| RH >=70 | Class 4- High Humidity |

So, after addition of the new features, a snippet of the data looks like this:



| | AT | V | AP | RH | PE | Temp | Humid |
|---|---|---|---|---|---|---|---|
| 0 | 14.96 | 41.76 | 1024.07 | 73.17 | 463.26 | 1.0 | 4.0 |
| 1 | 25.18 | 62.96 | 1020.04 | 59.08 | 444.37 | 3.0 | 3.0 |
| 2 | 5.11 | 39.40 | 1012.16 | 92.14 | 488.56 | 1.0 | 4.0 |
| 3 | 20.86 | 57.32 | 1010.24 | 76.64 | 446.48 | 2.0 | 4.0 |
| 4 | 10.82 | 37.50 | 1009.23 | 96.62 | 473.90 | 1.0 | 4.0 |

*Figure 19: Data with New Features*

## 4.6. Data Normalization

Normalization is a typical data preparation technique that allows us to adjust the values of numeric columns in the dataset to use a common scale [14]. The purpose of normalization is to convert the values of numeric columns in a dataset to a common scale while preserving disparities in value ranges. Every dataset does not require normalization for machine learning. It is only necessary when the ranges of characteristics differ.

The benefits of Normalization are:

1. Transforms features to a similar scale.
2. Improves performance of the data pipeline.
3. Training process of model is also improved.
4. Model becomes more stable.

We shall undergo the process of Scaling to a range. The formula is:

*Xn = (X - Xminimum) / ( Xmaximum - Xminimum)*

Where,

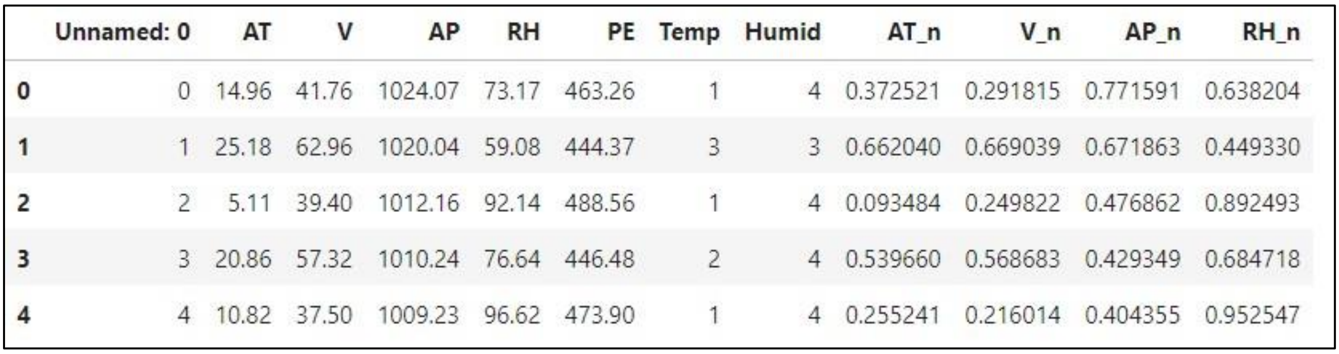
*Xn = Value of Normalization*

*Xmaximum = Maximum value of a feature*

*Xminimum = Minimum value of a feature*

Scaling to a range is a good method for this dataset, as we know the upper and lower bounds of the dataset. There are very low or no outliers and the data has sufficient units throughout.

After the application of normalisation, the data looks like this:



| | Unnamed: 0 | AT | V | AP | RH | PE | Temp | Humid | AT_n | V_n | AP_n | RH_n |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 14.96 | 41.76 | 1024.07 | 73.17 | 463.26 | 1 | 4 | 0.372521 | 0.291815 | 0.771591 | 0.638204 |
| **1** | 1 | 25.18 | 62.96 | 1020.04 | 59.08 | 444.37 | 3 | 3 | 0.662040 | 0.669039 | 0.671863 | 0.449330 |
| **2** | 2 | 5.11 | 39.40 | 1012.16 | 92.14 | 488.56 | 1 | 4 | 0.093484 | 0.249822 | 0.476862 | 0.892493 |
| **3** | 3 | 20.86 | 57.32 | 1010.24 | 76.64 | 446.48 | 2 | 4 | 0.539660 | 0.568683 | 0.429349 | 0.684718 |
| **4** | 4 | 10.82 | 37.50 | 1009.23 | 96.62 | 473.90 | 1 | 4 | 0.255241 | 0.216014 | 0.404355 | 0.952547 |

*Figure 20: Data with Normalized Features*

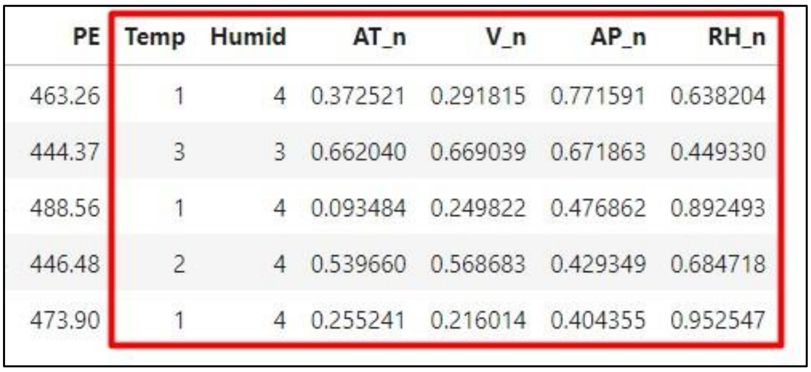
Due to normalization, the data is now scaled into a range of 0 to 1.

## 4.7. Application of Machine Learning

Now that we have prepared the data, we shall be implementing the machine learning process.

### a) Selecting the input and output features.

We take the categorical features made for the Temperature and humidity along with the normalised values as input features. The net hourly energy output will be the output feature.



| PE | Temp | Humid | AT_n | V_n | AP_n | RH_n |
|---|---|---|---|---|---|---|
| 463.26 | 1 | 4 | 0.372521 | 0.291815 | 0.771591 | 0.638204 |
| 444.37 | 3 | 3 | 0.662040 | 0.669039 | 0.671863 | 0.449330 |
| 488.56 | 1 | 4 | 0.093484 | 0.249822 | 0.476862 | 0.892493 |
| 446.48 | 2 | 4 | 0.539660 | 0.568683 | 0.429349 | 0.684718 |
| 473.90 | 1 | 4 | 0.255241 | 0.216014 | 0.404355 | 0.952547 |

*Figure 21: Data Features to be taken as Input*

So, basically, these features will be used, in which the features marked are the input features and PE is the output feature.

## b) Doing the Training and Testing split

In Machine Learning, majority of the data is used for training and the remaining part of the data is used for testing. We shall be using 75% of the data for training and 25% of the data for testing.

## c) Training the model

Model training is the process by which the data science team tries to match the optimal weights and biases to an algorithm in order to minimise the loss function throughout the prediction range. Loss functions specify how the ML algorithms should be optimised. Depending on the project objectives, the type of data utilised, and the type of methodology, a data science team may employ several loss functions.

Model training generates a mathematical representation of the connection between data characteristics and a target label when supervised learning is utilised. It establishes a mathematical representation among the data characteristics itself in unsupervised learning.

We used four models: Linear Regression, Polynomial Regression, Decision Tree Regression and Random Forest Regression.

### Model Training: Linear Regression

```
# importing module
from sklearn.linear_model import LinearRegression
# creating an object of LinearRegression class
LR = LinearRegression()
# fitting the training data
LR.fit(X_train,y_train)
```

### Model Training: Polynomial Regression

```
# importing module
from sklearn.linear_model import LinearRegression

polynomial_features = PolynomialFeatures(degree=3)
x_train_poly = polynomial_features.fit_transform(X_train)
x_test_poly = polynomial_features.fit_transform(X_test)
pr = LinearRegression()
pr.fit(x_train_poly, y_train)
```

### Model Training: Decision Tree

```
from sklearn.tree import DecisionTreeRegressor
regr= DecisionTreeRegressor(max_depth=2, min_samples_split=5,
min_samples_leaf=2)
regr.fit(X_train,y_train)
```

### Model Training: Random Forest

```
from sklearn.ensemble import RandomForestRegressor
rf = RandomForestRegressor(random_state=8, max_depth=5, n_estimators=100)
```

```
# fitting the training data
rf.fit(X_train,y_train)
```

## d) Making Predictions:

Now, that the model has been trained, we shall be making predictions. We shall be comparing the predicted values, with the actual test values to check some accuracy metrics and judge the model performance. We can compare model performance to find the best model.

## e) Accuracy Metrics:

## i) R2 Score:

For regression-based machine learning models, the R2 score is a performance evaluation metric. It’s another name is the coefficient of determination.

The R2 score is a critical indicator for assessing the effectiveness of a regression-based machine learning model. It's also known as the coefficient of determination and is pronounced R squared. It operates by calculating the amount of variation in the dataset-explained predictions. Simply expressed, it is the difference between the dataset's samples and the model's predictions.

Mathematical Formula:

$$\mathbf{R^2 = 1- SS_{res} / SS_{tot}}$$

Where,

- SSres is the sum of squares of the residual errors.
- SStot is the total sum of the errors

How should we understand R2 score. Let us assume, a model gives R2 score=75.

It is estimated that the model can account for 75% of the changeability of the dependent output attribute, while the remaining 25% of the variability remains unaccounted for.

R2 denotes the percentage of data points that fall within the regression equation's line. A greater R2 number suggests better outcomes, which is ideal.

## ii) Mean Absolute Error:

The average difference between estimated and real data is computed using Mean Absolute Error. Because it assesses inaccuracy in observations taken on the same scale, it's also known as scale-dependent accuracy. It's a statistic for evaluating regression models in machine learning. It estimates the differences between actual and model-predicted values. It is used to forecast the machine learning model's accuracy.

Mathematical Formula:

**Mean Absolute Error = (1/n) * $\sum|y_i – x_i|$**

Where ,

- Σ: Greek symbol for summation
- $y_i$: Actual value for the ith observation
- $x_i$: Calculated value for the ith observation
- n: Total number of observations

## iii) Mean Squared Error:

The Mean Squared Error is a metric that quantifies how near a regression line is to a set of data points. It's a risk function that corresponds to the squared error loss's predicted value. The average, especially the mean, of errors squared from data as it pertains to a function is used to determine mean square error.

Mathematical Formula:

**MSE = (1/n) * Σ (actual – forecast)2**

where:

- Σ – a symbol that means “sum”
- n – sample size
- actual – the actual data value
- forecast – the predicted data value

A higher MSE shows that the data points are widely spread around the centre moment (mean), whereas a lower MSE indicates the reverse. A smaller MSE is preferable since it implies that your data points are evenly distributed around the centre moment (mean). It shows your data's concentrated distribution, the fact that it is not skewed, and, most crucially, that it has fewer mistakes (errors measured by the dispersion of the data points from its mean).

## iv) Root Mean Squared Error/Deviation:

The root mean square error (RMSE) is the residuals' standard deviation (prediction errors). Residuals are a measure of how distant the data points are from the regression line; RMSE is a measure of how spread out these residuals are. In other words, it indicates how tightly the data is clustered around the line of best fit. To check experimental data, root mean square error is extensively utilized in climatology, finance, forecasting, and regression analysis.

Mathematical Formula:

$$RMSE = \sqrt{[\sum (Pi - Oi)^2/n]}$$

where:

- $P_i$ is the predicted value for the $i^{th}$ observation in the dataset
- $O_i$ is the observed value for the $i^{th}$ observation in the dataset
- n is number of data points

The root mean square error may be determined for any model that generates projected values that can then be compared to the dataset's observed values.

Root Mean Squared Error (RMSE) is an ordinary way to figure out how much inaccuracy exists between two training datasets. It is also known as the standard deviation of the residuals. Specifically, these residuals estimate how remote the regression line data areas are in the plotting graph; RMSE analyzes to what extent these residuary are spread out. That is to say, RMSE describes in what way the information is condensed all over the best-fitted line [15].

The validity of the learned model can be understood using a line plot where the first line would be the actual output test data and the other line would be the predicted data from the input test data. The higher the similarity of the lines, the better the model is performing. This indicates that the data points coincide with each other and make it look cohesive.

# CHAPTER 5
# RESULTS

## 5.1 Linear Regression Accuracy Metrics

| Metric | Value |
|---|---|
| R2 Score | 0.9278034642596562 |
| Mean Absolute Error | 3.6374586763847 |
| Mean Squared Error | 20.95100810799155 |
| Root Mean Squared Error | 4.577227119992141 |

## Actual vs Predicted Values:

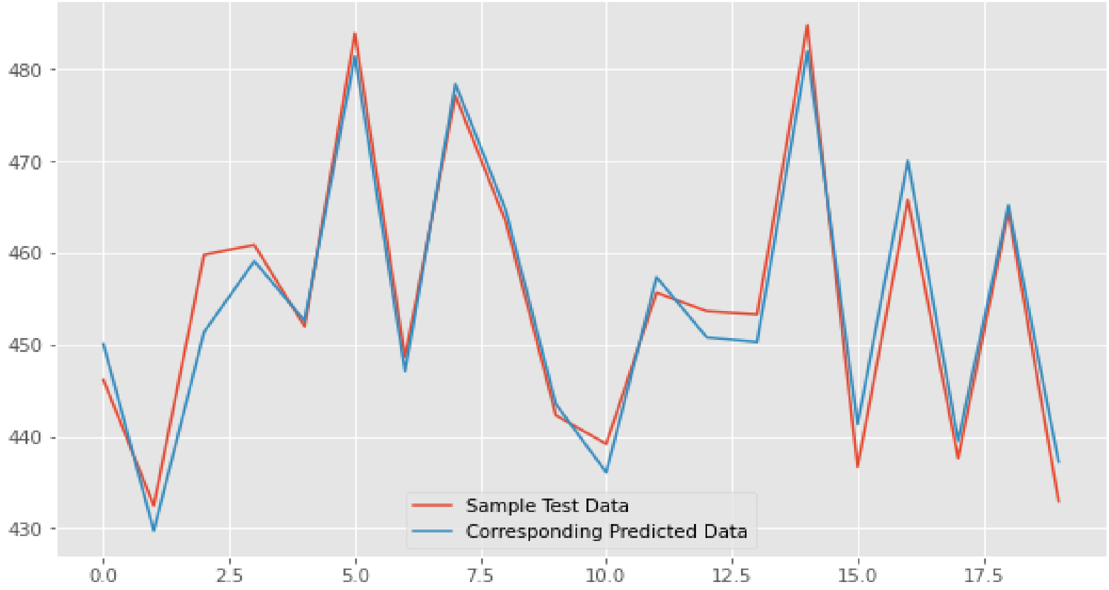


*Figure 22a: Actual vs Predicted Values (Linear Regression)*

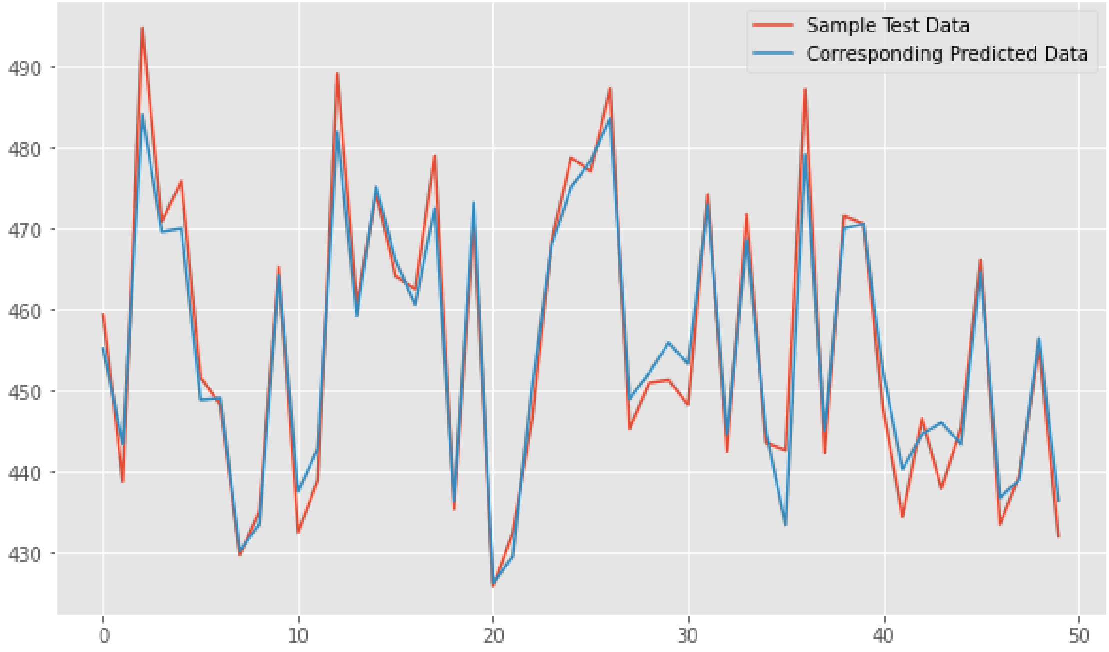


*Figure 22b: Actual vs Predicted Values (Linear Regression)*

In case of linear regression, the lines are matching at various points indicating that the model is able to predict properly.

## 5.2 Polynomial Regression Accuracy Metrics

| Metric | Value |
|---|---|
| R2 Score | 0.9416201631798347 |
| Mean Absolute Error | 3.2051238883549265 |
| Mean Squared Error | 16.941483715527134 |
| Root Mean Squared Error | 4.116003366802211 |

## Actual vs Predicted Values:

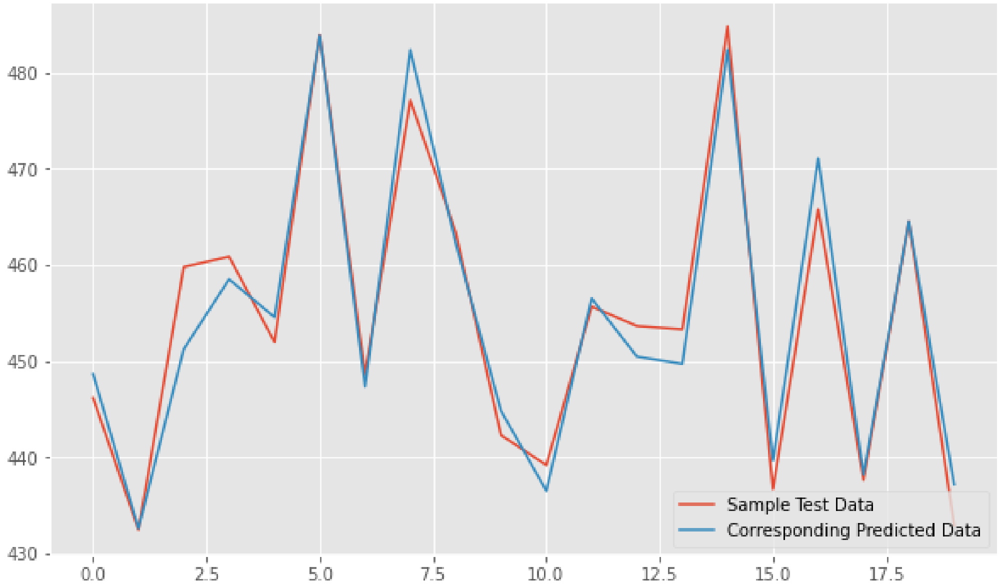


*Figure 23a: Actual vs Predicted Values (Polynomial Regression)*

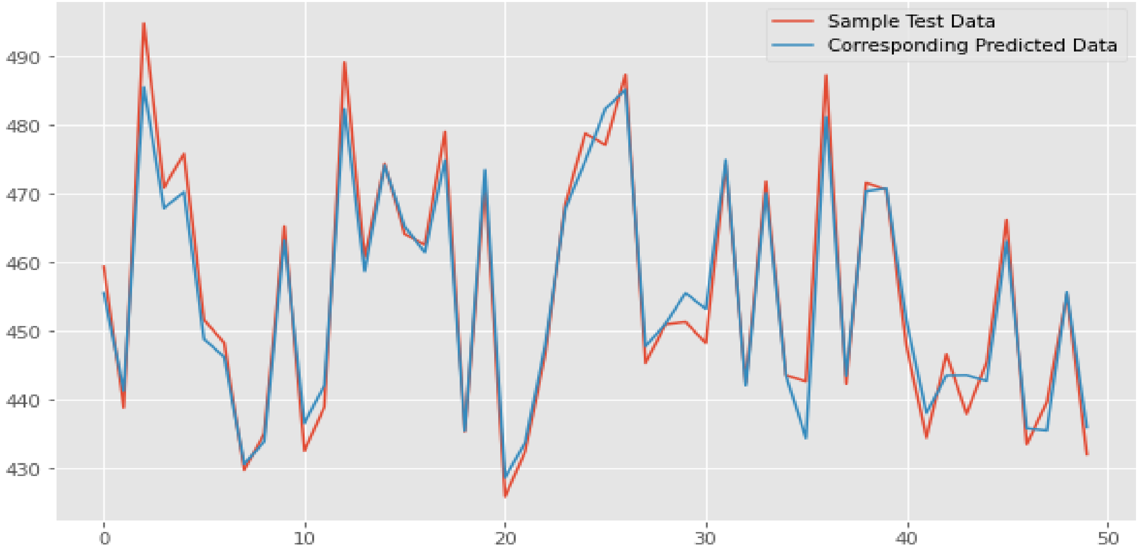


*Figure 23b: Actual vs Predicted Values (Polynomial Regression)*

The polynomial model performed better than the linear model. The both lines are very close to each other, which indicates the fact that the polynomial model has a really good R2 score and the model is able to make predictions very accurately.

### 5.3 Decision Tree Regression Accuracy Metrics

| Metric | Value |
|---|---|
| R2 Score | 0.8593733120875664 |
| Mean Absolute Error | 5.106282749531314 |
| Mean Squared Error | 40.80903395766401 |
| Root Mean Squared Error | 6.388194890394627 |

### Actual vs Predicted Values:

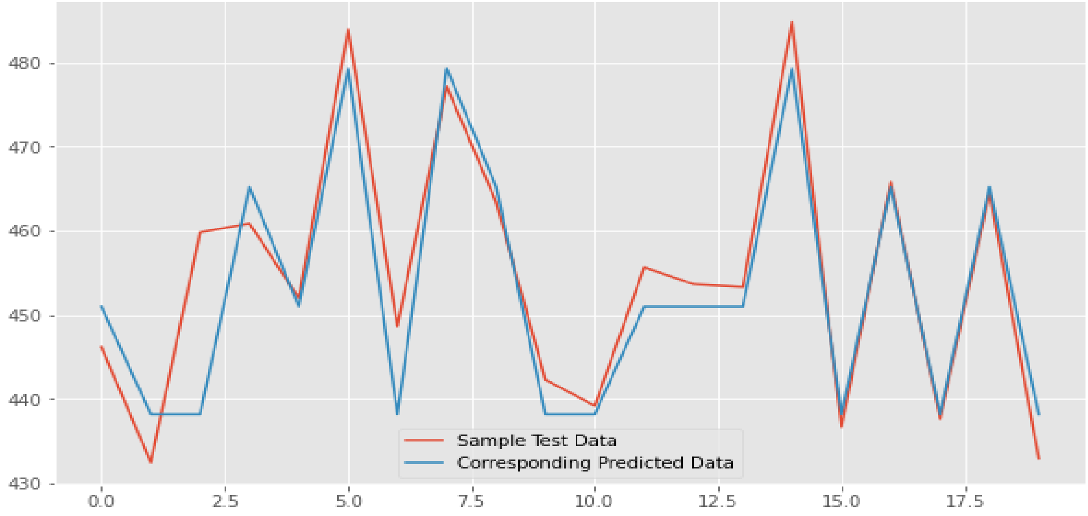


*Figure 24a: Actual vs Predicted Values (Decision Tree Regression)*

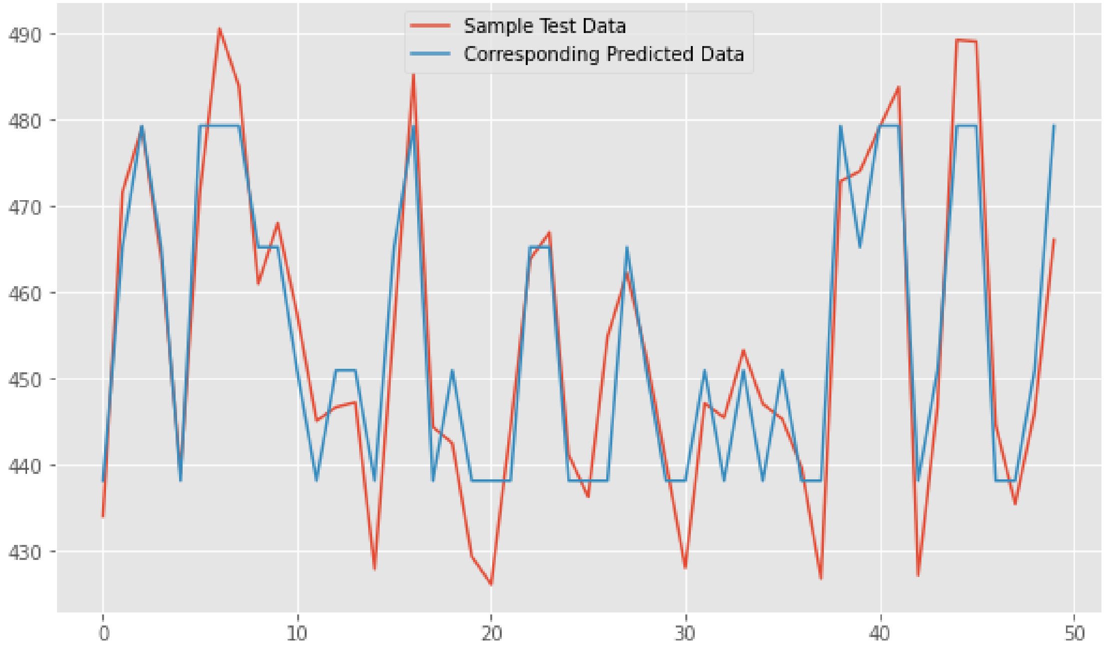


*Figure 24b: Actual vs Predicted Values (Decision Tree Regression)*

The performance of the Decision Tree Regression Model is not great, compared to the Linear and Polynomial Regression models. There are significant differences in the line and the decision tree regression model is not making predictions adequately correct, which is also reflected in its low R2 score.

## 5.4 Random Forest Regression Accuracy Metrics

| Metric | Value |
|---|---|
| R2 Score | 0.9395235633502752 |
| Mean Absolute Error | 3.2388779030504384 |
| Mean Squared Error | 17.549904598577477 |
| Root Mean Squared Error | 4.189260626718929 |

## Actual vs Predicted Values:

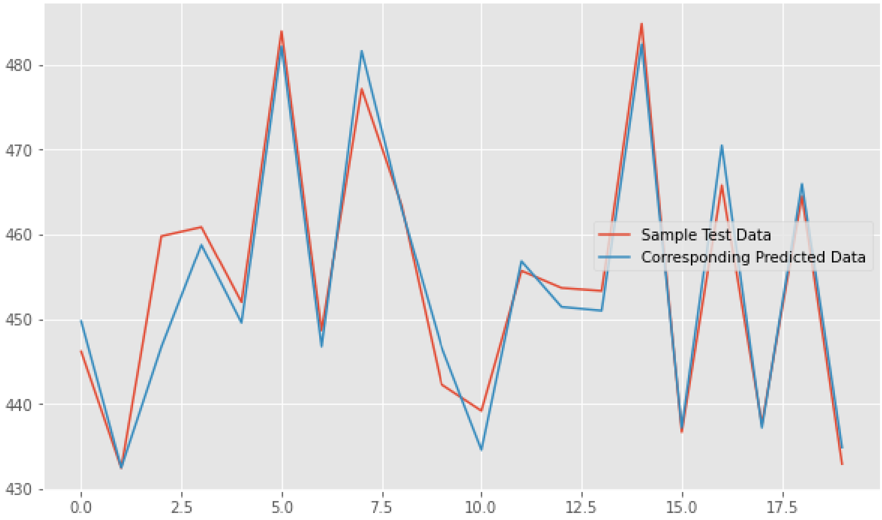


*Figure 25a: Actual vs Predicted Values (Random Forest Regression)*

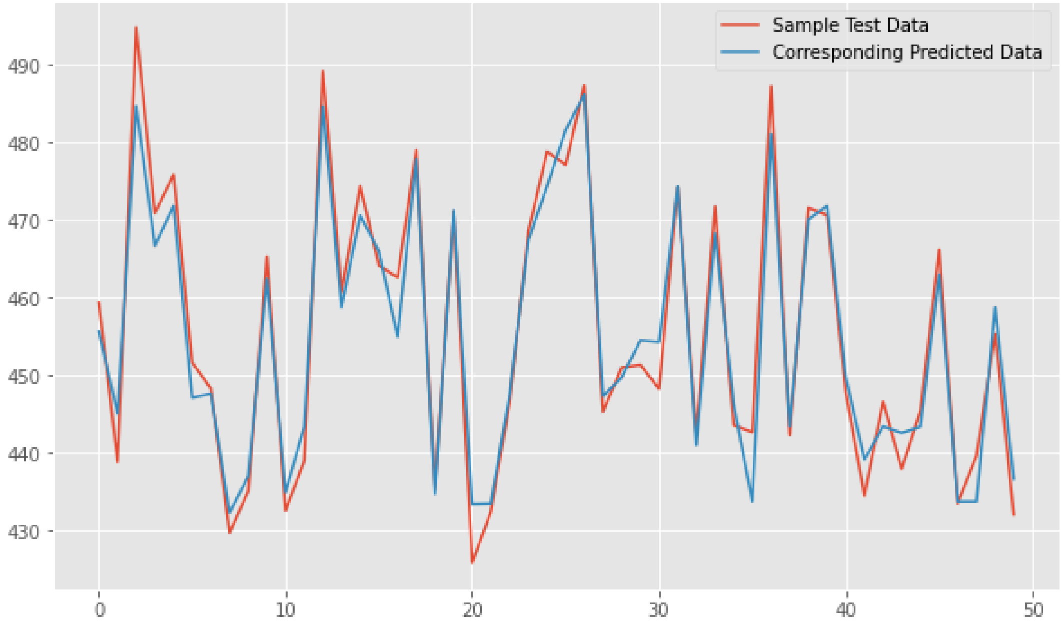


*Figure 25b: Actual vs Predicted Values (Random Forest Regression)*

The combination of multiple decision trees makes the model perform better than the plain decision tree regressor, but it is not having as good as R2 score, compared to the Polynomial Regression model.

- We took a small sample of data to visualize the results. In most of the data points, the models predict values close to the true values, but the prediction is off in many data points. For thar reason the line plots of the actual vs predicted data have some resemblance to each other, but are not exactly the same.
- The lower the resemblance between the two lines, the lower is the accuracy of the regression model.
- The random forest model performed better than the decision tree model, as it is an aggregation of multiple decision trees, but it is not as good as the polynomial regression model.
- The polynomial regression model had the highest R2 score, lowest Mean Absolute Error, lowest Mean Squared error and lowest Root Mean Squared error. So, we can say that the Polynomial regression model performed the best in this study.

# Comparative Study:

In the paper, Predicting the power of a combined cycle power plant using machine learning methods [3], they found out that the best method was Bagging method with REPTree algorithm with a mean absolute error of 2.818 and a Root Mean Squared Error of 3.787. The R2 score of the model is 0.9298. So we can compare:

| Score | REPTree Algorithm | Polynomial Regression |
|---|---|---|
| MAE | 2.818 | 3.2051 |
| RMSE | 3.787 | 4.1160 |
| R2 Score | 0.9298 | 0.9416 |

The polynomial regression model has higher MAE and RMSE, but has a better R2 score as well. So we can assume that our model will perform better in real life scenarios.

# CHAPTER 6
# CONCLUSION

This work offered a new solution model for predicting the electrical power output of a base load operated CCPP at full load. Machine learning techniques were selected for accurate prediction over thermodynamical approaches, which need some assumptions and a large number of nonlinear equations in a real-world application of a system. The study of a system utilizing thermodynamical methodologies requires too much computing time and effort, and the results of this analysis are sometimes unsatisfying and inaccurate due to the many assumptions made and nonlinear equations used.

The examination of multiple machine learning regression approaches for estimating output of a thermodynamic system, which is a CCPP with two gas turbines, one steam turbine, and two heating systems, was offered as an alternate analysis to address this problem.

This research has two key objectives. The initial step was to choose the best subset of our dataset from all of the other subset configurations tested. For this, we looked at which parameters or combinations of parameters had the largest impact on the target parameter's prediction. Second, we wanted to see which machine learning regression technique performed the best in terms of predicting full load electrical power production.

The Polynomial regression model and Random Forest Regression models performed the best. So, these models can be used to make the predictions of Net hourly electrical energy output. This study was important as it can be used to predict full load electrical power output in comparison to base load. It will help in improving efficiency of the power plant.

Comparison has been done with reference to a paper where the REPTree Algorithm gave a R2 score of 0.92, where as the Polynomial Regression Model gave an R2 Score of 0.94. So, we can conclude that the model has a better performance, which we have established with the help of the R2 Score metric.

# REFERENCES

1. Guvenir HA. Regression on feature projections. Knowl-Based Syst 2000; 13:207–14.

2. A. L. Polyzakis, C. Koroneos, and G. Xydis, “Optimum gas turbine cycle for combined cycle power plant,” Energy Convers. Manag., vol. 49, no. 4, pp. 551–563, 2008.

3. S. Alketbi, A. B. Nassif, M. A. Eddin, I. Shahin and A. Elnagar, "Predicting the power of a combined cycle power plant using machine learning methods," 2020 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI), 2020, pp. 1-5, doi: 10.1109/CCCI49893.2020.9256742.

4. Arrieta, F.R., & Lora, E.E. (2005). Influence of ambient temperature on combined-cycle power-plant performance. Applied Energy, 80, 261-272.

5. Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods Pınar Tüfekci ⇑

Department of Computer Engineering, Faculty of Çorlu Engineering, Namık Kemal University, TR-59860 Çorlu, Tekirdagˇ, Turkey

6. El Naqa, I., Murphy, M.J. (2015). What Is Machine Learning?. In: El Naqa, I., Li, R., Murphy, M. (eds) Machine Learning in Radiation Oncology. Springer, Cham. https://doi.org/10.1007/978-3-319-18305-3_1

7. Cunningham, P., Cord, M., Delany, S.J. (2008). Supervised Learning. In: Cord, M., Cunningham, P. (eds) Machine Learning Techniques for Multimedia. Cognitive Technologies. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-75171-7_2

8. Ghahramani, Z. (2004). Unsupervised Learning. In: Bousquet, O., von Luxburg, U., Rätsch, G. (eds) Advanced Lectures on Machine Learning. ML 2003. Lecture Notes in Computer Science(), vol 3176. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-28650-9_5

9. Gülden Kaya Uyanık, Neşe Güler, A Study on Multiple Linear Regression Analysis, Procedia - Social and Behavioral Sciences, Volume 106, 2013, Pages 234-240, ISSN 1877-0428, https://doi.org/10.1016/j.sbspro.2013.12.027.

10. Eva Ostertagová, Modelling using Polynomial Regression, Procedia Engineering, Volume 48, 2012, Pages 500-506, ISSN 1877-7058, https://doi.org/10.1016/j.proeng.2012.09.545.

11. Santosh Singh Rathore and Sandeep Kumar. 2016. A Decision Tree Regression based Approach for the Number of Software Faults Prediction. SIGSOFT Softw. Eng. Notes 41, 1 (January 2016), 1–6. https://doi.org/10.1145/2853073.2853083

12. Ulrike Grömping (2009) Variable Importance Assessment in Regression: Linear Regression versus Random Forest, The American Statistician, 63:4, 308-319, DOI: 10.1198/tast.2009.08199

13. UCI ML Combined Cycle Power Plant Data Set https://archive.ics.uci.edu/ml/datasets/combined+cycle+power+plant

14. Arthur A. Hancock, Eugene N. Bush, Dusanka Stanisic, John J. Kyncl, C.Thomas Lin, Data normalization before statistical analysis: keeping the horse before the cart, Trends in Pharmacological Sciences, Volume 9, Issue 1, 1988, Pages 29-32, ISSN 0165-6147, https://doi.org/10.1016/0165-6147(88)90239-8.

15. S. Glen, “RMSE: Root Mean Square Error,” StatisticsHowTo, [Online]. Available: https://www.statisticshowto.com/probability-andstatistics/regression-analysis/rmse-root-mean-square-error/

# Appendix

## Code Samples:

For full code, please visit: https://github.com/prateekmaj21/Project-Repo

Some important parts of the entire python code are added here.

### i) Data Normalisation:

```
for i in range(len(data)):

        data["AT_n"][i]=( data["AT"][i]- data["AT"].min()  )/ (data["AT"].max()-
data["AT"].min())

for i in range(len(data)):

        data["V_n"][i]=( data["V"][i]- data["V"].min()  )/ (data["V"].max()-
data["V"].min())

for i in range(len(data)):

        data["AP_n"][i]=( data["AP"][i]- data["AP"].min()  )/ (data["AP"].max()-
data["AP"].min())

for i in range(len(data)):

        data["RH_n"][i]=( data["RH"][i]- data["RH"].min()  )/ (data["RH"].max()-
data["RH"].min())
```

### ii) Linear Regression:

### a) Model Training:

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
dfs = pd.read_excel('final_prepared.xlsx')


X= dfs[["Temp","Humid","AT_n","V_n","AP_n","RH_n"]].values
y= dfs["PE"].values
X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.25,
random_state=7)
# importing module
from sklearn.linear_model import LinearRegression
# creating an object of LinearRegression class
LR = LinearRegression()
# fitting the training data
LR.fit(X_train,y_train)
y_prediction =  LR.predict(X_test)
```

### b) Accuracy Metrics:

```
from sklearn.metrics import r2_score
print("R2 Score:")
r2_score(y_test, y_prediction)
```

```
from sklearn.metrics import mean_absolute_error
print("Mean Absolute Error:")
mean_absolute_error(y_test, y_prediction)
from sklearn.metrics import mean_squared_error
print("Mean Squared Error:")
mean_squared_error(y_test, y_prediction)
mport math

from sklearn.metrics import mean_squared_error
print("Root Mean Squared Error:")
print (math.sqrt(mean_squared_error(y_test, y_prediction)))
```

## iii) Polynomial Regression:

## a) Model Training:

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import PolynomialFeatures
dfs = pd.read_excel('final_prepared.xlsx')


X= dfs[["Temp","Humid","AT_n","V_n","AP_n","RH_n"]].values
y= dfs["PE"].values


X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.25,
random_state=7)
# importing module
from sklearn.linear_model import LinearRegression

polynomial_features = PolynomialFeatures(degree=3)
x_train_poly = polynomial_features.fit_transform(X_train)
x_test_poly = polynomial_features.fit_transform(X_test)

pr = LinearRegression()
pr.fit(x_train_poly, y_train)
y_poly_test = pr.predict(x_test_poly)
```

## b) Accuracy Metrics:

```
from sklearn.metrics import r2_score

print("R2 Score:")
r2_score(y_test, y_poly_test)

from sklearn.metrics import mean_absolute_error
print("Mean Absolute Error:")
mean_absolute_error(y_test, y_poly_test)

from sklearn.metrics import mean_squared_error
print("Mean Squared Error:")
mean_squared_error(y_test, y_poly_test)

import math
from sklearn.metrics import mean_squared_error
print("Root Mean Squared Error:")
```

```
print (math.sqrt(mean_squared_error(y_test, y_poly_test)))
```

## iv) Decision Tree Regression:

## a) Model Training:

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
dfs = pd.read_excel('final_prepared.xlsx')



X= dfs[["Temp","Humid","AT_n","V_n","AP_n","RH_n"]].values
y= dfs["PE"].values
X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.25,
random_state=7)


from sklearn.tree import DecisionTreeRegressor
regr= DecisionTreeRegressor(max_depth=2, min_samples_split=5, min_samples_leaf=2)
regr.fit(X_train,y_train)
y_prediction =  regr.predict(X_test)
```

## b) Accuracy Metrics:

```
from sklearn.metrics import r2_score
print("R2 Score:")
r2_score(y_test, y_prediction)
from sklearn.metrics import mean_absolute_error
print("Mean Absolute Error:")
mean_absolute_error(y_test, y_prediction)


from sklearn.metrics import mean_squared_error
print("Mean Squared Error:")
mean_squared_error(y_test, y_prediction)


import math

from sklearn.metrics import mean_squared_error
print("Root Mean Squared Error:")
print (math.sqrt(mean_squared_error(y_test, y_prediction)))
```

## v) Random Forest Regression:

## a) Model Training:

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
dfs = pd.read_excel('final_prepared.xlsx')
X= dfs[["Temp","Humid","AT_n","V_n","AP_n","RH_n"]].values
y= dfs["PE"].values
```

```
X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.25,
random_state=7)


from sklearn.ensemble import RandomForestRegressor

rf = RandomForestRegressor(random_state=8, max_depth=5,n_estimators=100)
# fitting the training data
rf.fit(X_train,y_train)
y_prediction =  rf.predict(X_test)
```

## b) Accuracy Metrics:

```
from sklearn.metrics import r2_score
print("R2 Score:")
r2_score(y_test, y_prediction)

from sklearn.metrics import mean_absolute_error
print("Mean Absolute Error:")
mean_absolute_error(y_test, y_prediction)


from sklearn.metrics import mean_squared_error
print("Mean Squared Error:")
mean_squared_error(y_test, y_prediction)


import math
from sklearn.metrics import mean_squared_error
print("Root Mean Squared Error:")
print (math.sqrt(mean_squared_error(y_test, y_prediction)))
```