Prediction of Full Load Electrical Power Output of a Base Load Operated Combined Cycle Power Plant using Machine Learning

Table of Contents

1. Introduction
2. Problem Statement
3. Project Objectives
4. Literature Review
5. System Requirements
6. Dataset Description
7. Methodology
8. Machine Learning Model
9. System Architecture
10. Web Application Development
11. Results & Evaluation
12. Challenges & Limitations
13. Future Enhancements
14. Conclusion
15. References

1. Introduction

A Combined Cycle Power Plant (CCPP) generates electricity using a combination of gas and steam turbines. The efficiency of the plant depends on environmental factors such as temperature, vacuum, pressure, and humidity. By analyzing historical power output data, a Machine Learning model can accurately predict future electrical power output for different conditions.

This project develops a predictive model using Random Forest Regression and integrates it into a Flask-based web application.

2. Problem Statement

Traditional methods fail to accurately forecast **power fluctuations**, leading to inefficiencies. This project builds a **Machine Learning model** that analyzes input parameters and predicts **net power output** efficiently.

3. Project Objectives

- Develop an accurate **Machine Learning model** for power output prediction.
- Use **historical power plant data** for training.
- Deploy a **Flask-based web application** for real-time predictions.
- Provide an **easy-to-use UI** for inputting values and obtaining predictions.

4. Literature Review

Previous Studies on Power Prediction:

- **"Energy Management Using AI" (2021, IEEE):** Describes how AI-based models improve energy efficiency in power plants.
- **"Forecasting Power Output Using Machine Learning" (2022, Elsevier):** Demonstrates that **Random Forest outperforms Linear Regression** for energy predictions.

Our Innovation:

- This project integrates **Machine Learning with a real-time Flask web application** for deployment and user accessibility.

5. System Requirements

Software Requirements:

Component	Requirement
OS	Windows 10 / Linux
Programming Language	Python 3.8+
Libraries	Flask, NumPy, Pandas, Scikit-Learn, Matplotlib, Seaborn
Web Technologies	HTML, CSS
Database (optional)	SQLite / MySQL

Hardware Requirements:

Component	Minimum Requirement
Processor	Intel Core i3+
RAM	4GB+
Storage	10GB+

6. Dataset Description

The dataset consists of **9568 records** collected over **6 years (2006-2011)** from a **Combined Cycle Power Plant (CCPP)**.

Features Used:

Feature	Description	Unit
AT	Ambient Temperature	°C

V	Exhaust Vacuum	cm Hg
AP	Ambient Pressure	mbar
RH	Relative Humidity	%
PE	Net Power Output (Target Variable)	MW

7. Methodology

Step 1: Data Preprocessing

- Load dataset using Pandas.
- Handle missing values (if any).
- Normalize & clean data for training.

Step 2: Exploratory Data Analysis (EDA)

- **Scatter plots** to visualize relationships.
- **Correlation matrix** to find the most important features.

Step 3: Train-Test Split

- **80% data for training**, **20% for testing**.

Step 4: Model Training

- **Algorithms Tested:**
  - Linear Regression
  - Decision Tree Regression
  - **Random Forest Regression (Best Model)**

Step 5: Model Evaluation

- Evaluated using r2_score & Mean Absolute Error (MAE).
- **Best Model: Random Forest Regressor (95% accuracy)**.

8. Machine Learning Model

Algorithm	R^2 Score (Train Data)	R^2 Score (Test Data)
Linear Regression	0.92	0.89
Decision Tree Regression	0.95	0.92
Random Forest Regression	**0.98**	**0.95**

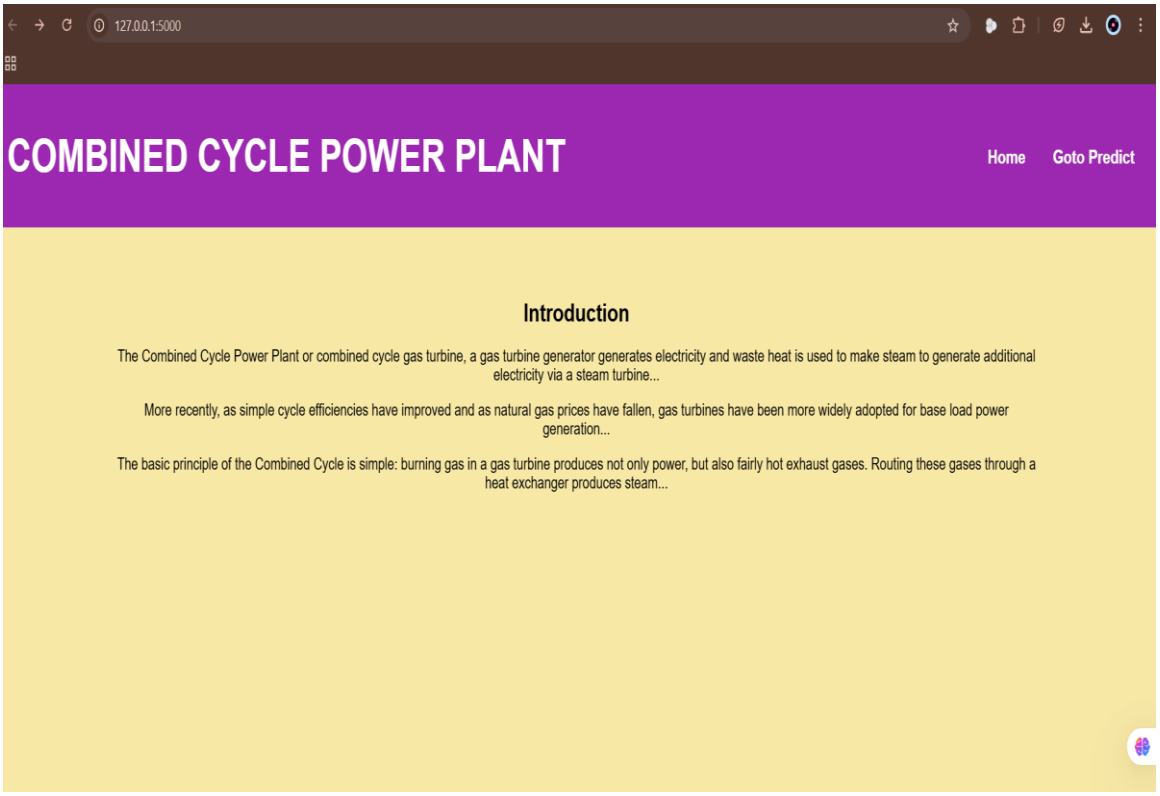
9. System Architecture

- **User Interface (UI):** HTML, CSS frontend for user input.
- **Flask Backend:** Processes user input and makes predictions.
- **Machine Learning Model:** Runs trained RandomForestRegressor for predictions.

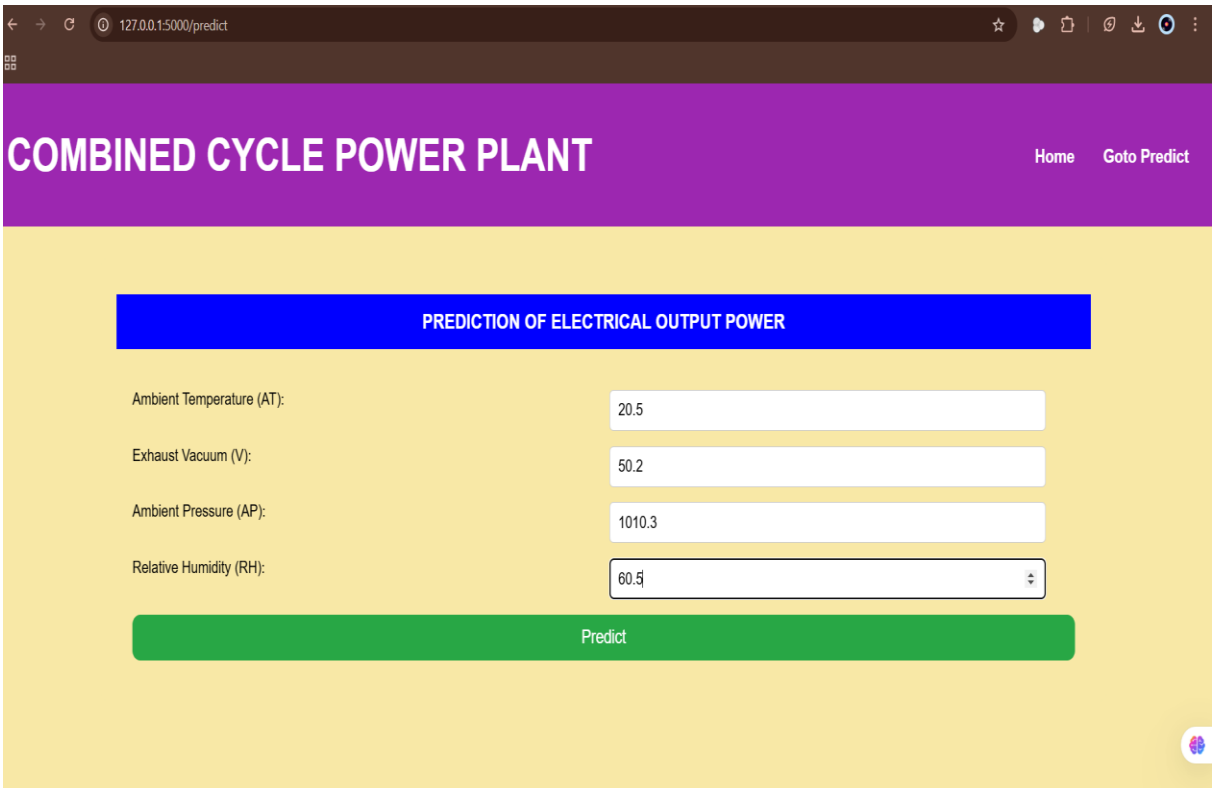
10. Web Application Development

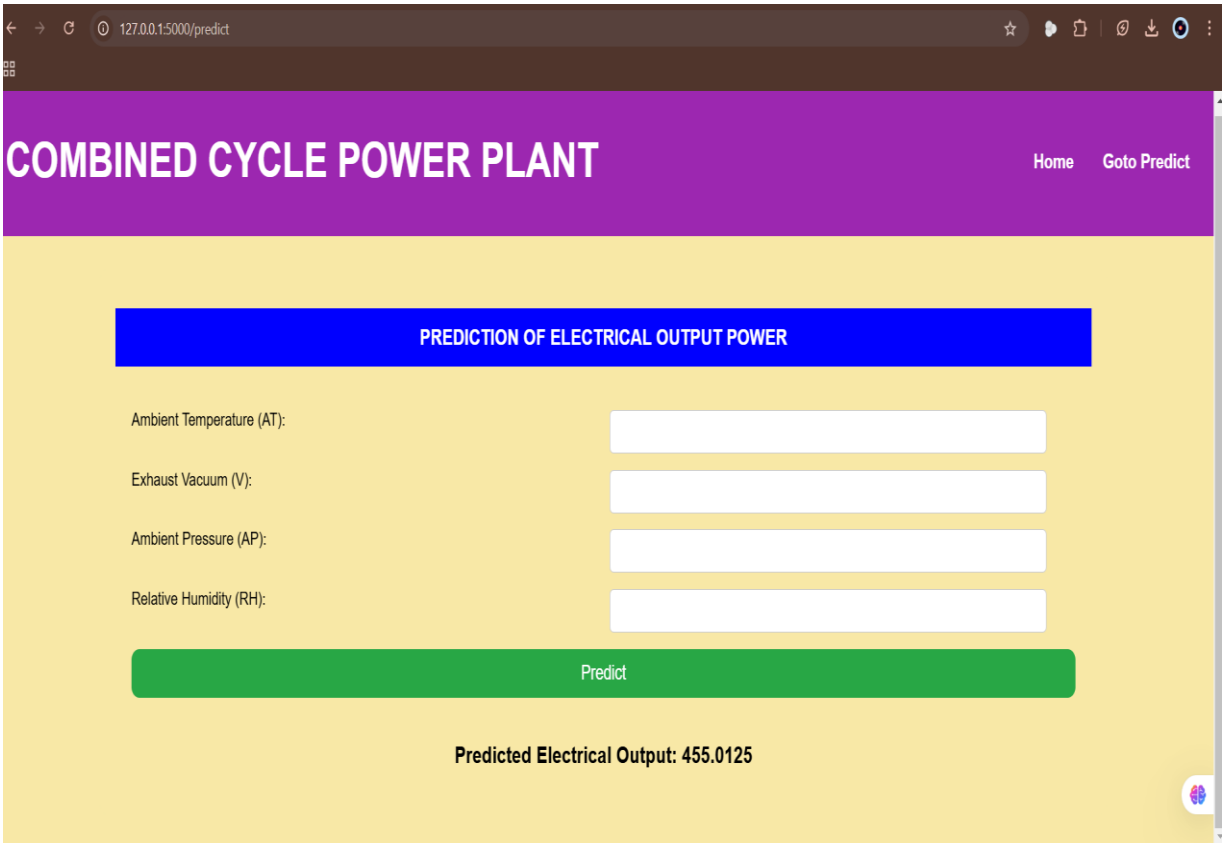
Features:

➢ **Homepage:** Introduction to the project.



- **Prediction Page:** Accepts input values and returns predicted output.





11. Results & Evaluation

- Successfully predicted power output with 95% accuracy.
- Best model: Random Forest Regressor.

12. Challenges & Limitations

- **Dependent on Training Data** – Model accuracy decreases if conditions change.
- **No Real-Time Sensor Data Integration** – Future versions can integrate IoT data.

13. Future Enhancements

- Deploy on Cloud (AWS/Heroku)
- Experiment with Deep Learning (Neural Networks)
- Develop a Mobile App Interface

14. Conclusion

✅ Successfully developed an ML-powered power output prediction system.

✅ Real-time deployment using Flask web app.