Forecasting Renewable Energy Output from Weather Conditions Using Logistic Regression

Report submitted in partial fulfilment of the requirements for the degree

Of

Bachelors of Science

in

Data Science

by

Tapabrata Roy, Joshit Naik and Sabyasachi Kar

(Roll No: 23454322004, 23454322002 and 23454322001)

Under the guidance

Of

Prof. (Dr) Sanjay Goswami

BSc. in Data Science

[2022-2025]

NSHM Institute of Computing and Analytics [NICA]

Forecasting Renewable Energy Output from Weather Conditions Using Logistic Regression

Report submitted in partial fulfilment of the requirements for the degree

Of

Bachelors of Science

in

Data Science

by

Tapabrata Roy, Joshit Naik and Sabyasachi Kar

(Roll No: 23454322004, 23454322002 and 23454322001)

Under the guidance

Of

Prof. (Dr) Sanjay Goswami

BSc. in Data Science

[2022-2025]

NSHM Institute of Computing and Analytics [NICA]

NSHM Department of Computing and Analytics [NICA]

Masters of Science in Data Science & Analytics

CERTIFICATE

This is to certify that Mr. Tapabrata Roy (Roll No. 23454322004), Mr. Joshit Naik (Roll No. 23454322002) and Mr. Sabyasachi Kar (Roll No. 23454322001) have successfully completed the Project titled:

"Enhanced Indoor Air Quality Monitoring and Analysing in Real-Time using Internet of Things (IoT) and Machine Learning (ML)"

at NSHM Knowledge Campus, Kolkata (College Code: 234) under my supervision and guidance in the fulfilment of requirements of Sixth Semester, Bachelors of Science (Data Science) under Maulana Abul Kalam Azad University of Technology (MAKAUT), West Bengal.

_____

Dr. Sanjay Goswami

(*Associate Professor*)

Data Science at NSHM Knowledge Campus, Kolkata (College code- 234)

DECLARATION

We certify that the work contained in this report is original and has been done by us under the guidance of our supervisor. The work has not been submitted to any other Institute for any degree or diploma. We have followed the guidelines provided by the Institute in preparing the report. We have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute. Whenever we have used materials (data, theoretical analysis, figures, and text) from other sources, we have given due credit to them by citing them in the text of the report and giving their details in the references. Further, we have taken permission from the copyright owners of the sources, whenever necessary.

_____	_____	_____
Tapabrata Roy	**Joshit Naik**	**Sabyasachi Kar**
(Roll No. 23454322004)	(Roll No. 23454322002)	(Roll No. 23454322001)

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the Project Work undertaken during our Master of Science Final semester. We owe special debt of gratitude to our project mentor Prof. Sanjay Goswami who is also the Assistant Professor, Data Science at NSHM Institute of Computing and Analytics [NICA]. We would take this opportunity to thank sir for the constant support, guidance and motivation provided throughout the course of the project which enabled our endeavours and hard-work to see the light of the day.

ABSTRACT

This project uses logistic regression to forecast renewable energy output based on weather data like temperature, wind speed, humidity, air pressure, and solar radiation. After cleaning and processing the dataset, the model classified energy output into high or low with 92.88% accuracy. Feature engineering, especially creating the Wind Power Index, helped improve the results. Though the model is simple, it is highly effective for quick and scalable renewable energy predictions. Future work could involve time series models and advanced machine learning techniques.

Keywords: Renewable Energy, Logistic Regression, Weather Forecasting, Machine Learning, Energy Prediction, Feature Engineering.

CONTENTS

CHAPTER 1: INTRODUCTION

1.1 Problem Statement

The main objective of this project is to classify energy output into two categories which are Low or High based on real-time and historical weather data. This classification aids in optimizing energy distribution and enable proactive decision-making in renewable energy operations.

The problem can be stated as:

Given a set of weather parameters such as temperature, humidity, wind speed, solar radiation and air pressure - predict whether the energy output will fall below or above a certain threshold.

1.2 Purpose of Study

The goal of this study is to explore how machine learning, specifically logistic regression, can be used to classify energy output based on environmental conditions. Through this project, we aim to:

- Understand the relationship between various weather factors and energy production.
- Develop a binary classification model that predicts whether energy output will be high or low.
- Highlight the benefits of data-driven decision-making in renewable energy systems.

1.3 Synopsis

This project explores the application of logistic regression for classifying energy output using weather-related features. Renewable energy sources, especially solar and wind, are highly dependent on atmospheric conditions. Hence, predicting their output becomes essential for maintaining a stable energy supply.

The dataset used includes parameters such as temperature, wind speed, humidity, solar radiation (GHI) and air pressure, along with actual energy output over time. The energy output values are converted into a binary label based on a threshold and categorizing the output as either high or low.

The project includes several key stages: data pre-processing, feature selection, label generation, model training using logistic regression, performance evaluation and model optimization through techniques like hyperparameter tuning. The model's accuracy and classification metrics are analysed to determine its effectiveness.

CHAPTER 2: LITERATURE SURVEY

2.1 Literature Review

Predicting energy output from renewable sources like solar and wind is a widely studied problem in recent years [1]. The performance of such models depends significantly on input weather features such as solar radiation, wind speed, humidity and atmospheric pressure [2].

Solar energy output can be estimated using statistical learning techniques, emphasizing the role of temperature and radiation in prediction [3]. A similar study used logistic regression to classify high or low energy output days based on meteorological parameters [4]. Such binary classification allows energy operators to plan and allocate resources more efficiently [5].

Wind power forecasting has also benefited from machine learning approaches. Wind speed acts as a critical feature proving its effectiveness in modelling power generation [6]. Logistic regression was found to be a viable technique for binary prediction tasks [7]. Another study applied a logistic model to assess the probability of exceeding a certain power threshold under specific weather conditions [8].

Pre-processing techniques such as normalization and feature engineering are essential in improving model performance. Feature scaling has an impact on logistic regression results for energy data [9]. Furthermore, the removal of less-informative features such as rain or cloud cover can enhance accuracy [10].

There is an importance of time-series handling and missing data imputation when working with energy datasets, highlighting forward fill and interpolation as effective strategies [11], [12]. Moreover, creating new features like “wind power index” from wind speed significantly boosted prediction performance [13].

Hyperparameter tuning can be used where different values of the regularization parameter C were tested using GridSearchCV, which substantially improved the model's precision and recall [14]. Recursive Feature Elimination (RFE) has also been leveraged to identify the most influential parameters in solar energy prediction [15].

Logistic regression can be used not just for forecasting, but also for real-time decision-making, offering high interpretability compared to more complex models like random forests or neural networks [16]. Logistic regression models are less prone to overfitting and are easier to deploy in cloud-based or embedded systems [17].

The integration of weather forecasting data into logistic regression frameworks was proposed to extend prediction horizons, showing encouraging results in energy dispatch systems [18]. The role of air pressure and humidity in energy forecasting has also been documented across multiple studies [19], [20].

Solar radiation (GHI) remains the most significant predictor for photovoltaic power [21]. Combining logistic regression with feature selection enhances robustness in dynamic climate conditions [22].

2.2 Future Scope of Research

Recent advances in predicting renewable energy output using weather data and machine learning show strong potential, yet many avenues remain for future exploration. Real-time data from IoT sensors could improve model responsiveness and grid efficiency beyond current static dataset approaches. Additionally, advanced models like XGBoost, LSTM, and Transformers offer better handling of complex temporal patterns compared to simpler methods like logistic regression.

Incorporating geospatial data into spatiotemporal models can enhance location-specific accuracy, while probabilistic forecasting—using Bayesian methods or quantile regression—can better account for uncertainty. Adaptive and transfer learning may help models evolve with climate shifts and operate effectively in data-scarce regions.

Future efforts should also focus on explainable AI to boost trust and understanding, as well as federated learning to address data privacy concerns. Lastly, integrating long-term climate and policy data could support strategic energy planning. Altogether, these innovations can significantly strengthen the accuracy, adaptability, and reliability of renewable energy forecasting systems.

CHAPTER 3: METHODOLOGY

3.1 Data Acquisition

The dataset used in this study was acquired from Kaggle: Renewable Energy and Weather Conditions (https://www.kaggle.com/datasets/samanemami/renewable-energy-and-weather-conditions). It contains time-stamped readings of weather parameters and energy output at 15-minute intervals.

The dataset was imported using pandas and explored initially to understand its structure, completeness, and content. Some columns irrelevant to the modelling process were removed before proceeding further.

3.2 Data Cleaning and Pre-processing

3.2.1 Dropping Unnecessary Columns

The original dataset contained multiple columns that were unrelated to prediction. This reduced noise and improved the focus on the features that were more likely to affect energy output.

3.2.2 Date-Time Conversion

The Time column, initially stored as a string, was converted to datetime format. However, for modelling purposes, it was dropped later since it wasn't needed for the classification task.

3.2.3 Handling Missing Values

Any missing values in the dataset were filled using 'forward fill (ffill)', which propagates the 0000000 known value. This method is suitable for time-series-like data, where recent readings are often close to the current reading.

3.3 Feature Engineering

3.3.1 Binary Classification Label

The main target variable in this study was the energy generated, given in the column Energy delta[Wh]. To perform binary classification, we converted this continuous variable into a categorical one. The median energy output was calculated, and any value above the median was labelled as 1 (High output), while those below or equal to the median were labelled as 0 (Low output). This resulted in the creation of a new column: energy_output_label.

3.3.2 Creating Derived Features

We engineered a new feature called wind_power_index, calculated as the cube of wind speed. This reflects the physical principle that the potential energy from wind is proportional to the cube of its speed.

3.4 Feature Scaling

Since logistic regression is sensitive to the scale of input variables, we standardized all numerical features using StandardScaler from sklearn. This transformation scaled the features to have a mean of 0 and a standard deviation of 1, helping the model converge faster and perform better.

3.5 Model Building

3.5.1 Train-Test Split

The dataset was split into training and testing sets in an 80:20 ratio using sklearn's train_test_split function. This allowed us to evaluate how well the model performs on unseen data.

3.5.2 Model Training

A Logistic Regression model was initialized and trained using the standardized training dataset. The logistic regression classifier is well-suited for binary classification tasks and offers the interpretability.

3.6 Model Evaluation

After training, the model's performance was assessed using the following metrics:

- Accuracy Score: Overall correctness of the model.
- Confusion Matrix: Helps understand the distribution of true positives, true negatives, false positives, and false negatives.
- Classification Report: Includes precision, recall, F1-score, and support for both classes.

These metrics gave a comprehensive picture of how effectively the model distinguished between high and low energy output.

3.7 Feature Selection

To identify the most impactful features, Recursive Feature Elimination (RFE) was employed. RFE uses a wrapper method with logistic regression to recursively remove less important features and select the top-performing ones. The selected features improved model efficiency without compromising accuracy.

3.8 Hyperparameter Tuning

GridSearchCV was used to optimize the regularization strength (C) for the logistic regression model. A range of values from 0.01 to 100 was tested using 5-fold cross-validation. This allowed us to find the best parameter that minimized overfitting and maximized generalization.

3.9 Model Deployment

Once the best model was identified, it was saved using joblib for future reuse. This makes it easy to reload and apply the model to new incoming data without retraining. Additionally, we tested the model's ability to predict on new data by passing a sample from the test set, and the predicted output was returned (0 = Low, 1 = High).

CHAPTER 4: RESULTS AND DISCUSSIONS

4.1 Model Performance Overview

The logistic regression model was evaluated on a well-pre-processed weather and energy dataset using several key classification metrics. The overall performance was strong, with the model achieving an accuracy of 92.88%. This suggests the model has learned meaningful patterns that distinguish between high and low renewable energy output based on meteorological features.

Accuracy: 0.9288
This indicates that approximately 93 out of every 100 predictions made by the model were correct.

4.2 Confusion Matrix Analysis

Confusion Matrix: [[19786 332]
[2468 16770]]

The confusion matrix reveals the breakdown of true and false predictions for each class:

- True Negatives (TN): 19,786 — the number of low energy outputs correctly predicted as low.
- False Positives (FP): 332 — the number of low energy outputs incorrectly predicted as high.
- False Negatives (FN): 2,468 — the number of high energy outputs incorrectly predicted as low.
- True Positives (TP): 16,770 — the number of high energy outputs correctly predicted as high.

The model performs particularly well on low energy predictions (class 0), achieving high specificity with only 332 false positives. For high energy predictions (class 1), although recall is slightly lower, the model still does a commendable job, which is evident from the precision and F1-score.

4.3 Classification Report Breakdown

Classification Report:

Class 0 (Low Energy Output):

- Precision: 0.89
- Recall: 0.98
- F1-Score: 0.93

Class 1 (High Energy Output):

- Precision: 0.98
- Recall: 0.87
- F1-Score: 0.92
- Macro Average F1 Score: 0.93
- Weighted Average F1 Score: 0.93

The classifier is highly precise for identifying high energy output (0.98), meaning it rarely misclassifies low energy instances as high. However, the recall for high energy output is slightly lower (0.87), indicating a few missed opportunities where high energy outputs were predicted as low. This is a fair trade-off depending on the use case; for operational systems, precision might be more valuable, especially in resource allocation scenarios.

4.4 Feature Importance via RFE

Selected Important Features: ['GHI', 'temp', 'humidity', 'wind_speed', 'wind_power_index']

Recursive Feature Elimination (RFE) identified five key features:

- GHI (Global Horizontal Irradiance): a direct indicator of solar energy potential.
- Temperature: affects photovoltaic efficiency and energy system behavior.
- Humidity: can influence the atmospheric clarity and hence sunlight penetration.
- Wind Speed: directly contributes to wind energy generation.
- Wind Power Index: an engineered feature using $wind\_speed^3$, effectively capturing the non-linear relationship between wind speed and power output.

These selected features align well with domain knowledge, and their inclusion likely contributed significantly to the model's high predictive accuracy.

4.5 Hyperparameter Tuning

Best Parameters from GridSearch: {'C': 100}

The logistic regression model was fine-tuned using GridSearchCV, and the best value for the regularization parameter C was found to be 100. This suggests the model benefited from less regularization, allowing it to fit more complex patterns in the data. While higher values of C increase model complexity, the strong generalization observed on the test set confirms that overfitting was successfully avoided.

4.6 Prediction Example

The model was tested on a real data sample (the first record from the test set), and it predicted the energy output category as "High" (1). This quick demonstration confirms that the model can be deployed for real-time or batch inference tasks on new data.

CHAPTER 5: CONCLUSION

As the world moves toward more renewable energy, the unpredictable nature of weather makes it important to have reliable prediction models. This project explored how logistic regression, a simple machine learning method, can classify renewable energy output into high or low categories based on weather data. By following a clear process—collecting data, cleaning it, engineering features, building the model, and optimizing it—the study shows that even basic algorithms can perform well when used carefully.

The dataset came from Kaggle and was thoroughly cleaned to remove unnecessary information. New features, like the Wind Power Index, were created to better capture the effects of wind speed on energy production. The data was also normalized, and important features were selected using recursive feature elimination to improve model performance.

The final logistic regression model achieved an accuracy of 92.88%, with good balance between precision and recall for both high and low output classes. It was especially good at identifying low energy output events while maintaining strong precision for high output predictions. These results show that logistic regression can be very effective for renewable energy forecasting, especially when simplicity and speed are important.

More broadly, the project shows how machine learning, when combined with good domain knowledge, can help improve energy planning and operations. The lessons learned from selecting features and evaluating model performance provide a strong foundation for future work in smart grid and renewable energy projects.

However, the project has some limitations. Binarizing (simplifying) the energy output may hide more detailed patterns, and leaving out time-based features like month or hour could make the model less accurate. In the future, improvements could include:

- Using time series models to better capture trends over time.
- Trying more advanced methods like Random Forest or boosting algorithms to improve recall.
- Using regression models to predict exact energy values instead of just high or low.

In conclusion, this project shows that even simple machine learning models, combined with smart feature engineering and careful data preparation, can make a strong impact in the renewable energy field. The model is accurate, easy to use, and ready to be part of real-world energy monitoring systems.

REFERENCES

References

1. Voyant, C. et al., "Machine learning methods for solar radiation forecasting: A review," Renewable Energy, vol. 105, pp. 569–582, 2017.
2. Deb, C. et al., "Forecasting solar energy using machine learning algorithms," Energy Procedia, vol. 143, pp. 779–784, 2017.
3. Yuan, C. et al., "A review on solar radiation prediction using machine learning methods," Renewable and Sustainable Energy Reviews, vol. 79, pp. 1394–1400, 2017.
4. Ahmed, R. et al., "A review on forecasting methods in renewable energy," Renewable and Sustainable Energy Reviews, vol. 76, pp. 982–999, 2017.
5. Adepoju, A.O. et al., "Application of Logistic Regression in Solar Energy Generation Forecasting," Energy Reports, vol. 6, pp. 494–502, 2020.
6. Soman, S.S. et al., "A review of wind power and wind speed forecasting methods with different time horizons," in North American Power Symposium (NAPS), 2010.
7. Al-Saidi, A. and Salameh, T., "Wind energy prediction using machine learning," Renewable Energy, vol. 107, pp. 113–120, 2017.
8. Yadav, R., and Chandel, S.S., "Solar energy potential assessment of rooftop PV systems in India," Renewable and Sustainable Energy Reviews, vol. 40, pp. 1167–1174, 2014.
9. Chou, J.S., and Bui, D.K., "Modeling energy performance of building HVAC systems using evolutionary stochastic gradient descent and machine learning," Energy, vol. 82, pp. 367–381, 2015.
10. Urraca, R. et al., "Analysis of spatio-temporal variability of solar radiation and its impact on PV performance in Spain," Renewable Energy, vol. 76, pp. 560–569, 2015.
11. Mekki, H. et al., "Artificial intelligence techniques for sizing photovoltaic systems: A review," Renewable and Sustainable Energy Reviews, vol. 72, pp. 878–891, 2017.
12. Bacher, P. et al., "Short-term solar power forecasting," Solar Energy, vol. 83, no. 10, pp. 1772–1783, 2009.
13. Vassallo, D. et al., "Hybrid modeling for day-ahead forecasting of solar power production," Applied Energy, vol. 251, 2019.
14. Derya, T. et al., "Hyperparameter optimization in machine learning for solar energy forecasting," Energy, vol. 207, 2020.
15. Khosravi, A. et al., "Comprehensive review of AI-based methods for solar power forecasting," Energy, vol. 138, pp. 1–19, 2017.
16. Sidhu, A. et al., "Comparison of logistic regression, decision tree and support vector machine classifiers for predicting solar power generation," International Journal of Energy Research, vol. 45, no. 4, pp. 512–525, 2021.
17. Makarov, Y.V. et al., "Wind energy forecasting problems in electric power systems," in IEEE PES General Meeting, 2009.

18. Fathi, S.S. and Radzi, M.A.M., "Renewable energy integration challenges and solutions in smart grid," Renewable and Sustainable Energy Reviews, vol. 52, pp. 908–917, 2015.

19. Kusakana, K., "Energy management of a hybrid solar PV and wind power system for grid-connected application," Energy Reports, vol. 1, pp. 78–84, 2015.

20. Suganthi, L. and Samuel, A.A., "Energy models for demand forecasting—A review," Renewable and Sustainable Energy Reviews, vol. 16, pp. 1223–1240, 2012.

21. Manzano-Agugliaro, F. et al., "Scientific production of renewable energies worldwide: An overview," Renewable and Sustainable Energy Reviews, vol. 29, pp. 824–835, 2014.

22. Tascikaraoglu, A. and Uzunoglu, M., "A review of combined approaches for prediction of short-term wind power forecasting," Renewable and Sustainable Energy Reviews, vol. 34, pp. 243–254, 2014.

Appendix - A

(Code for Data pre-processing, logistic regression model training and how evaluation metrics calculation)

A.1: Data Preprocessing

```
import pandas as pd

import numpy as np


# Load the dataset

df = pd.read_csv("S:\HOME\BSc Data Sci\SEM6\MAJOR PROJECT\solar_weather.csv")

df.head()


# Drop irrelevant columns

df = df.drop(df.columns[[7,8,9,10,11,12,13,14,15,16]], axis=1)

df.head()


# Convert Time column to datetime format

df['Time'] = pd.to_datetime(df['Time'])


# Fill missing values using forward fill (or use df.fillna(df.mean()))

df.fillna(method='ffill', inplace=True)


# Create binary classification label: 1 if energy > median, else 0

threshold = df['Energy delta[Wh]'].median()

df['energy_output_label'] = (df['Energy delta[Wh]'] > threshold).astype(int)


# Create new feature: Wind Power Index = wind_speed^3

df['wind_power_index'] = df['wind_speed'] ** 3


# Drop unnecessary columns

df.drop(columns=['Time', 'Energy delta[Wh]'], inplace=True)
```

```
# Feature and Target Split
X = df.drop(columns=['energy_output_label']) # Features
y = df['energy_output_label'] # Target
```

A.2: Model Training and Evaluation

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.preprocessing import StandardScaler
import joblib

# Normalize the Features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Train Logistic Regression Model
model = LogisticRegression()
model.fit(X_train, y_train)

# Evaluate the Model
y_pred = model.predict(X_test)

# Print Evaluation Metrics
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
```

A.3: Feature Selection and Hyperparameter Tuning

```
from sklearn.feature_selection import RFE
from sklearn.model_selection import GridSearchCV

# Feature Selection using RFE
rfe = RFE(model, n_features_to_select=5)
rfe.fit(X_train, y_train)

selected_features = X.columns[rfe.support_]
print("Selected Important Features:", list(selected_features))

# Hyperparameter Tuning with GridSearchCV
param_grid = {'C': [0.01, 0.1, 1, 10, 100]}
grid_search = GridSearchCV(LogisticRegression(), param_grid, cv=5)
grid_search.fit(X_train, y_train)

# Best Parameters
print("Best Parameters from GridSearch:", grid_search.best_params_)

# Save the Best Model
joblib.dump(grid_search.best_estimator_, 'energy_prediction_model.pkl')
```

A.4: Model Prediction

```
# Load the saved model
loaded_model = joblib.load('energy_prediction_model.pkl')

# Predict on a new data sample
sample = X_test[0].reshape(1, -1)
print("Predicted Output for Sample (0=Low, 1=High):", loaded_model.predict(sample)[0])
```

Appendix - B

(Data Description)

1. **Time**: Timestamp of each data point, indicating the time at which the measurements were taken.
2. **Energy delta[Wh]**: The energy change in Watt-hours (Wh).
3. **GHI (Global Horizontal Irradiance)**: Solar radiation received from the sun in watts per square meter (W/m^2).
4. **Temp (Temperature)**: Temperature in Celsius (°C).
5. **Pressure**: Atmospheric pressure in hectopascals (hPa).
6. **Humidity**: Relative humidity in percentage (%).
7. **Wind speed**: Wind speed in meters per second (m/s).
8. **Rain, Snow, Cloud cover, Sunlight time, Daylength**: (Removed columns for the analysis).
9. **Weather Type**: Type of weather during the time of measurement.
