**4.1 Model Performance Overview**

**4.1.1 Mathematical Foundation of Logistic Regression**

Logistic Regression is a supervised classification algorithm used to predict a binary outcome (0 or 1) based on independent variables. Unlike linear regression, logistic regression maps predicted values to probabilities using the sigmoid (logistic) function:

The output represents the probability that the instance belongs to class 1. The decision rule is:

**4.1.2 Logistic Regression with Graphical Representation**

The model essentially draws a decision boundary in the feature space that best separates the two classes.

* The sigmoid curve flattens values between 0 and 1
* As features increase or decrease, the probability of class 1 changes smoothly

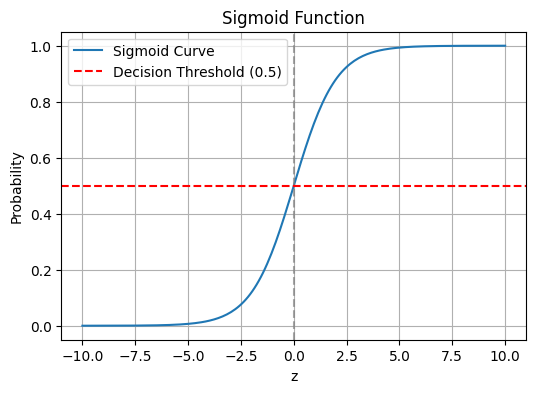


Fig 4.1 Sigmoid Function

**4.1.3 Decision Boundary**

To visually understand how the logistic regression model separates classes, we can use a 2D plot using two features (e.g., GHI and wind\_speed ):

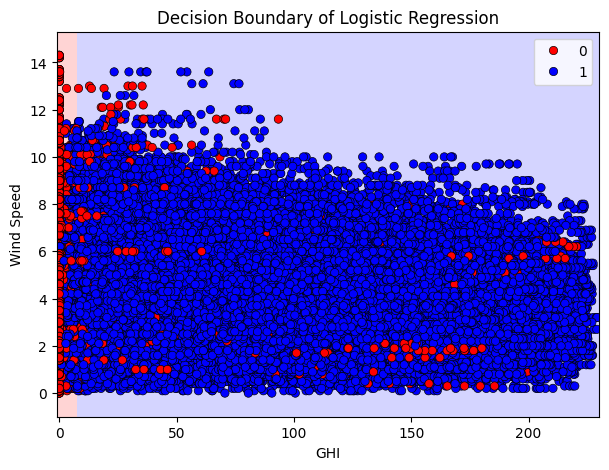


Fig 4.2 Decision Boundary of Logistic Regression

**4.1.4 Model Accuracy and Interpretation**

The logistic regression model was trained on a normalized weather-energy dataset. The model demonstrated excellent performance:

* **Accuracy**: 92.88%  
  This indicates that approximately 93 out of every 100 predictions were correct. The high accuracy reflects that the logistic regression model captured strong correlations between meteorological features and energy output classes.

**4.2 Confusion Matrix Analysis**

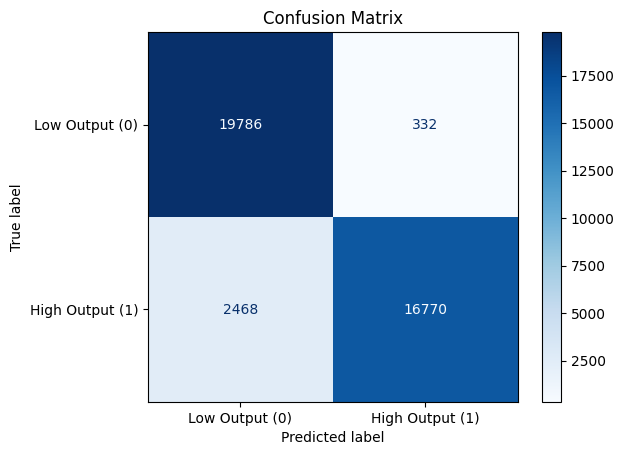
****

Fig 4.3 Confusion Matrix

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**

**4.3.1 Metric Formulas**

* Accuracy
* Precision
* Recall
* F-1 Score

**4.3.2 Tabular Summary of Classification Report**

|  |  |  |  |
| --- | --- | --- | --- |
| **Class** | **Precision** | **Recall** | **F1 Score** |
| **Low Energy Output (0)** | 0.89 | 0.98 | 0.93 |
| **High Energy Output (1)** | 0.98 | 0.87 | 0.92 |
| **Macro Avg** | 0.93 | 0.92 | 0.93 |
| **Weighted Avg** | 0.93 | 0.93 | 0.93 |

Fig 4.4Tabular Summary of Classification Report

**4.3.3 Interpretation of Results**

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 behaviour.
* **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\_speed³, effectively capturing the non-linear relationship between wind speed and power output.

**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 that is the first 10 samples from test set.

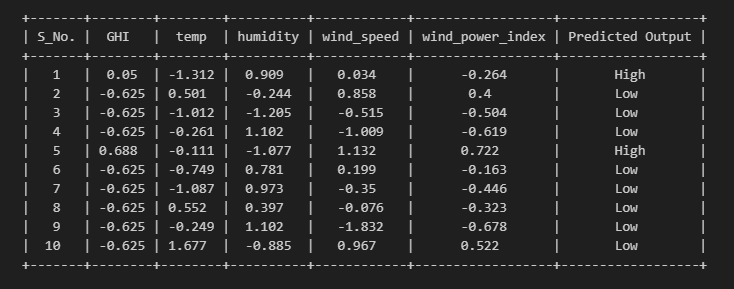


Fig 4.5 Table Showing Predicted Outputs of the Model

This quick demonstration confirms that the model can be deployed for real-time or batch inference tasks on new data.