IntelliHack 5.0

Team: CodeLabs

Task 01- Report 01

Initial round

Table of Content

Team: CodeLabs

Weather Forecasting using Machine Learning

1. Abstract

This project focuses on developing a machine-learning model to predict whether it will rain or not using historical weather data. The dataset consists of 300 days of observations with features such as average temperature, humidity, wind speed, and precipitation labels. Our objective is to build a model that not only classifies the occurrence of rain but also provides a probability of rain. Key steps include data cleaning, feature engineering, exploratory data analysis (EDA), model training using various algorithms, hyperparameter tuning, and generating the final output with predicted probabilities. This report documents the entire workflow, methodology, and results of the forecasting system.

2. Introduction

2.1 Background

Accurate weather forecasting is critical for smart agriculture. Farmers require hyper-local forecasts to make informed decisions about irrigation, planting, and harvesting. Traditional methods can struggle to capture localized nuances, which motivates the application of machine learning to improve forecast reliability.

2.2 Objectives

- **Data Cleaning and Preprocessing:** Address issues such as missing values and incorrect data types.
- **Feature Engineering:** Develop temporal, lagged, rolling, and interaction features.
- **Exploratory Data Analysis (EDA):** Understand relationships between the weather features and the target variable.
- **Model Development and Evaluation:** Train and compare multiple models (Logistic Regression, Random Forest, Gradient Boosting, SVM) using performance metrics.
- **Predict Rain Probability:** Provide the final output that includes the predicted probability of rain.

3. Data Description

3.1 Dataset Overview

The dataset comprises 300 days of daily weather observations with the following fields;

- **avg_temperature:** Average daily temperature (°C)
- **humidity:** Daily humidity (%)
- **avg_wind_speed:** Average wind speed (km/h)
- **rain_or_not:** Binary label (1 = Rain, 0 = No Rain)
- **date:** Date of observation

3.2 Data Quality Issues

Key challenges in the dataset include;

- **Missing Values:** Some records have missing entries.
- **Duplicates:** Duplicate records haven't existed.
- **Formatting Issues:** The date field needs to be correctly converted into a datetime format.

4. Data Preprocessing & Feature Engineering

4.1 Data Cleaning

- **Handling Duplicates:** Duplicate entries were not found.
- **Handling Missing Values:** Rows with missing values were dropped since each had four missing columns, though only three exist. This affected just 15 rows, accounting for a 4.82% data loss.
- **Data Type Conversion:** The date column was converted to a datetime object.

4.2 Feature Engineering

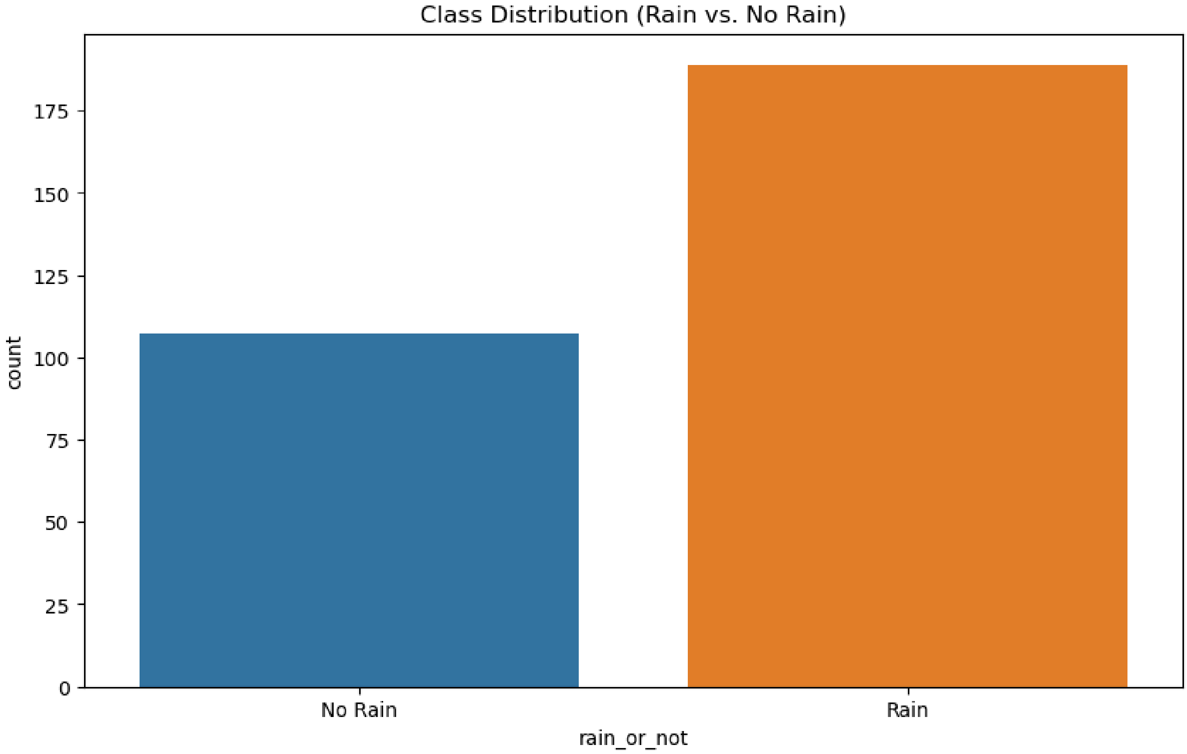
The following new features were created;

- **Temporal Features:** Extracted month, day of the year, and day of the week from the date column.
- **Lagged Variables:** For each primary feature (e.g., avg_temperature, humidity), lag1 features were generated to capture the previous day's values.
- **Rolling Variables:** Calculated rolling averages and standard deviations (e.g., 3-day rolling average for temperature, 7-day rolling standard deviation for humidity) to capture short-term trends.
- **Interaction Variables:** Created new features by multiplying key variables (e.g., temperature and wind speed, pressure and humidity) to capture interaction effects.

5. Exploratory Data Analysis (EDA)

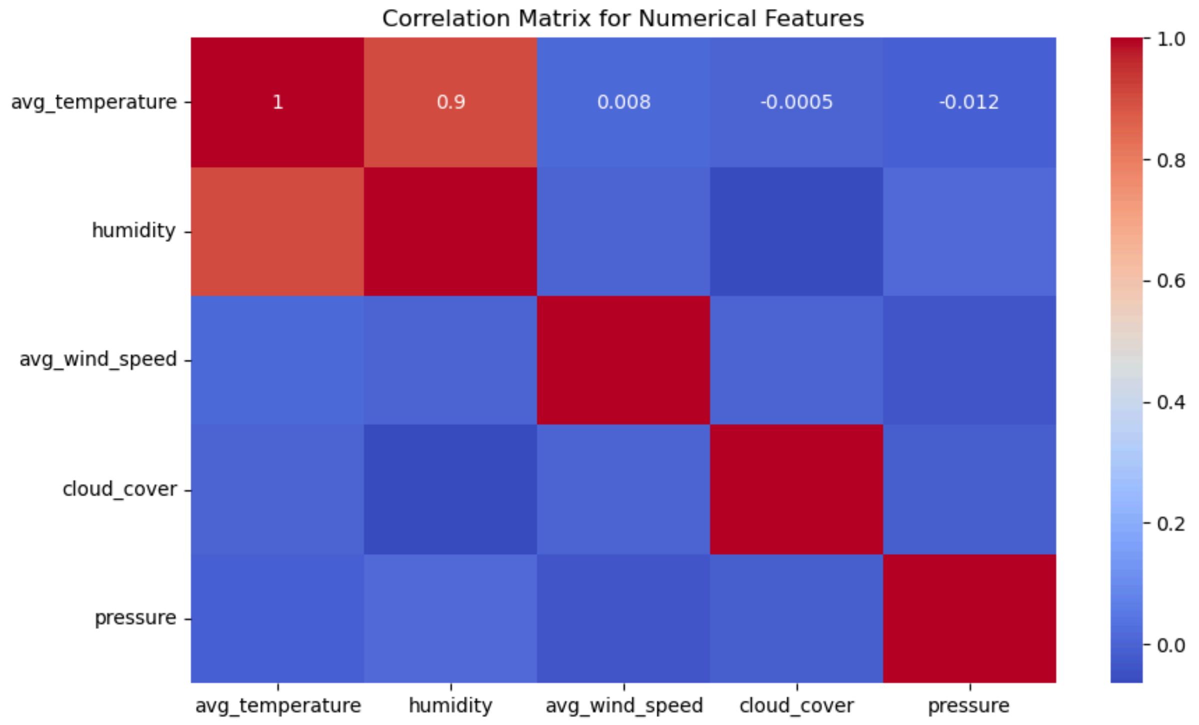
5.1 Visualizing the Target Variable

- **Count Plot:** A countplot was used to visualize the distribution of the target variable (rain_or_not), ensuring that class imbalances could be identified.



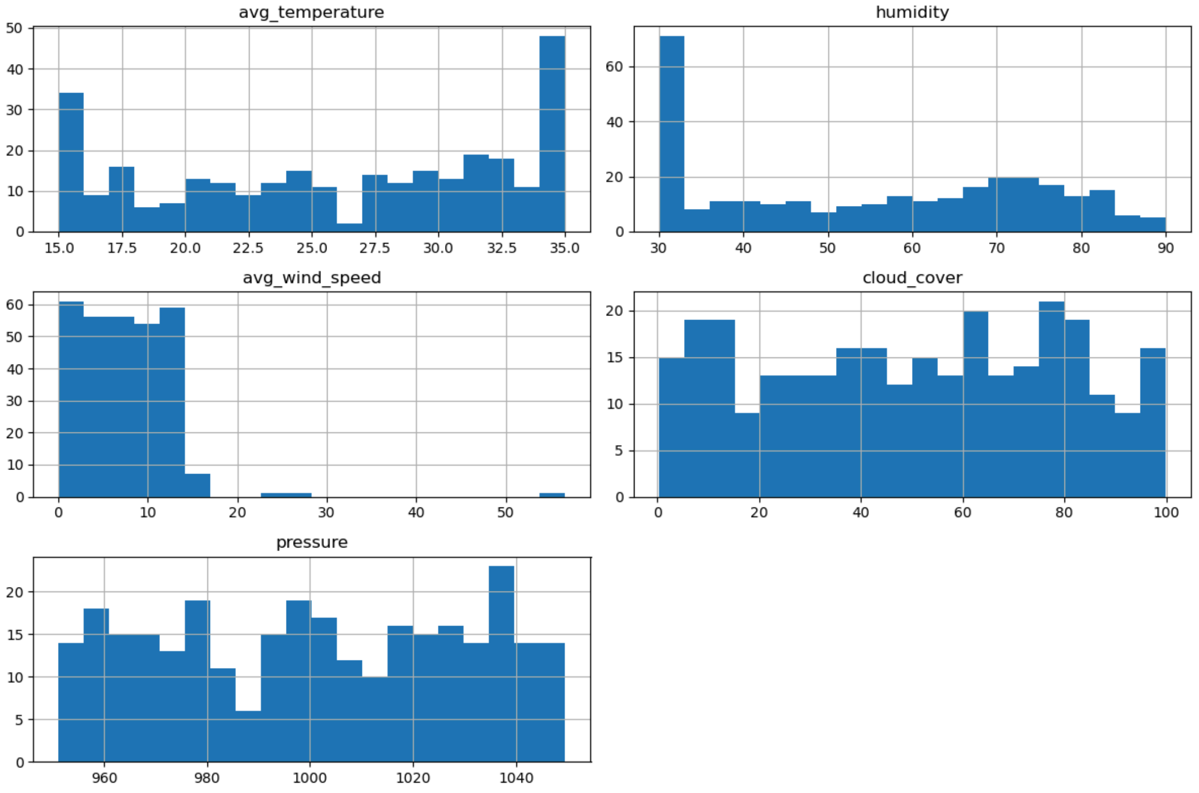
5.2 Correlation Analysis

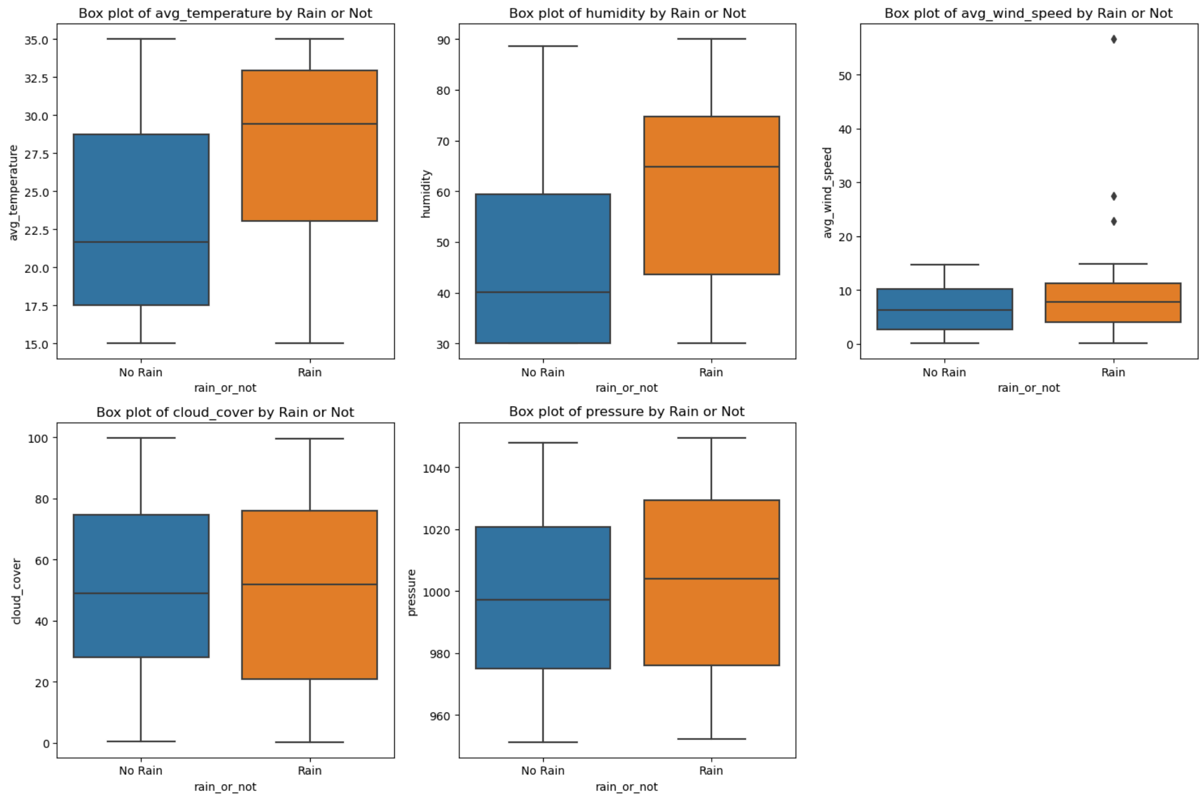
- **Heatmap:** A correlation matrix was plotted to examine relationships between numerical features and identify potential multicollinearity issues.



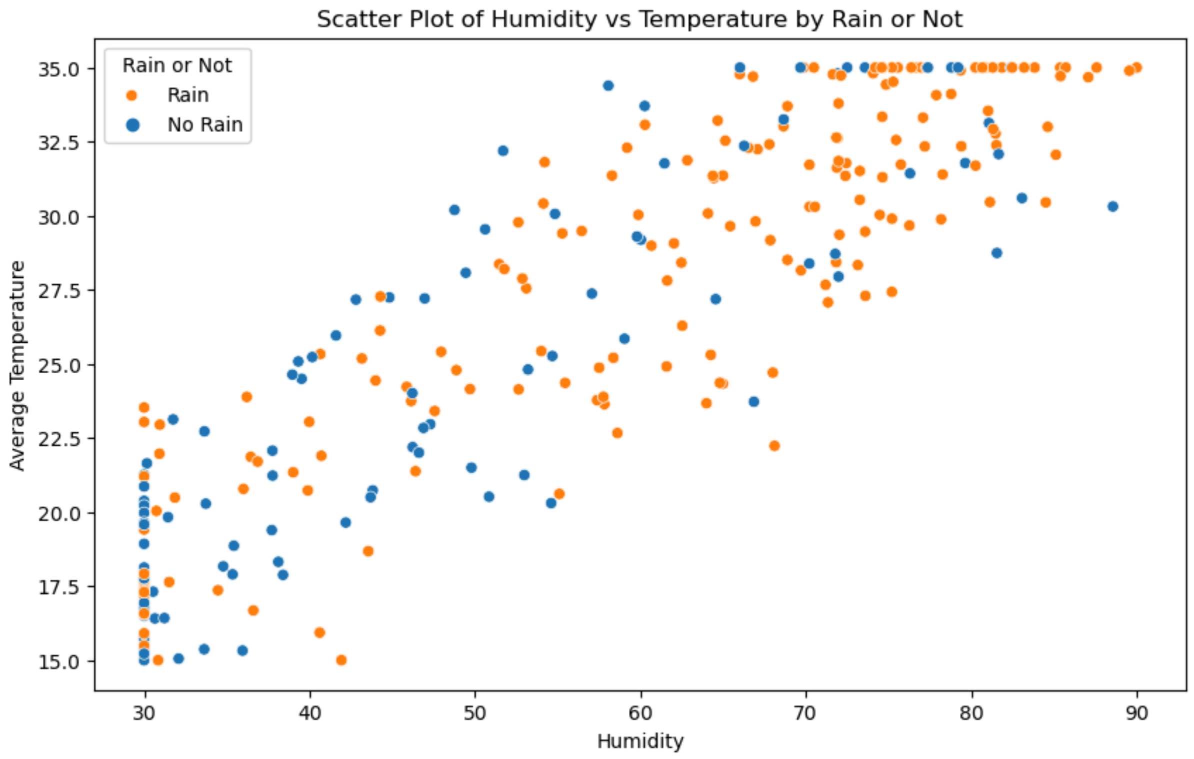
5.3 Distribution Analysis

- **Histograms and Box Plots:** Histograms and box plots were used to visualize feature distributions and to inspect the presence of outliers.





- **Scatter Plots:** Scatter plots were employed to analyze the relationships between pairs of features (e.g., humidity vs. temperature).



6. Model Development and Evaluation

6.1 Data Splitting

- **Train-Test Split:** The dataset was split into an 80% training set and a 20% testing set using stratified sampling to maintain the distribution of the target variable.

6.2 Evaluation Metrics

Models were evaluated using the following metrics;

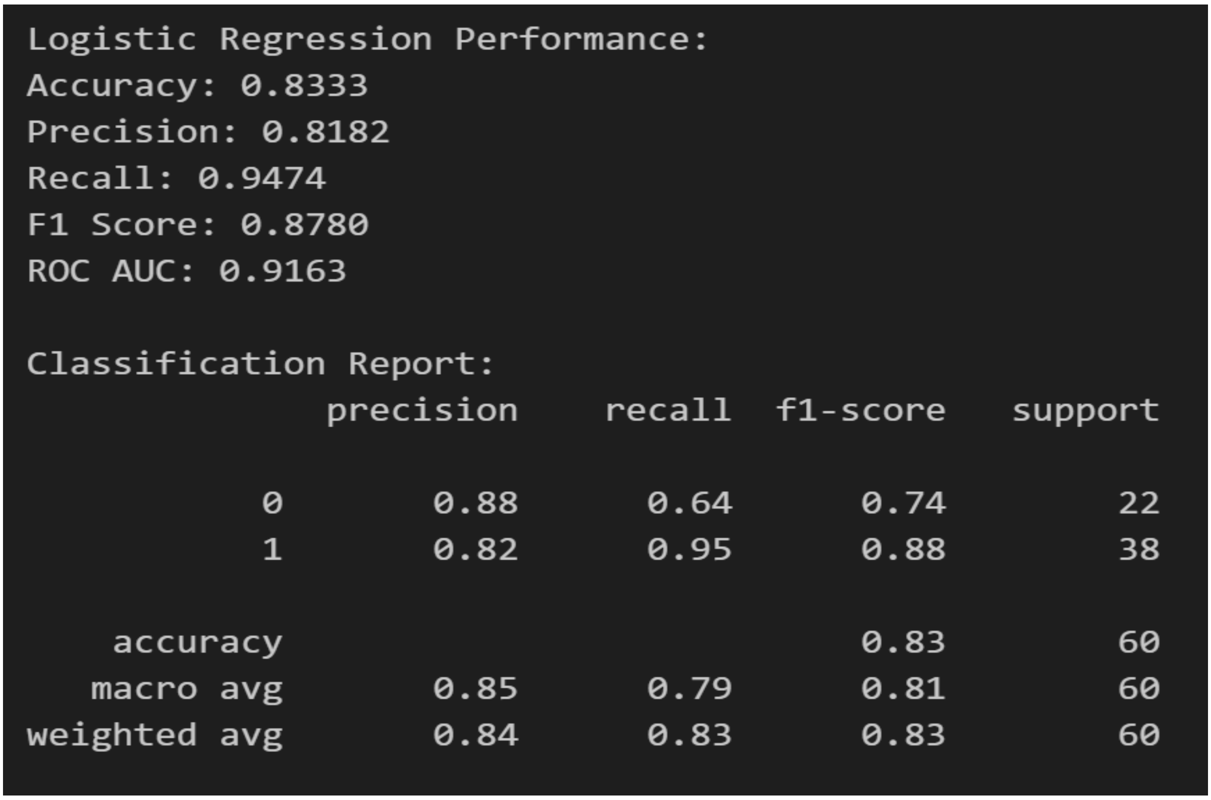
- Accuracy
- Precision
- Recall
- F1 Score
- ROC AUC

The evaluation results were used to compare model performance and select the best-performing model.

6.3 Implemented Models

Four machine learning models were developed;

- **Logistic Regression:** Baseline model using a pipeline with data standardization.



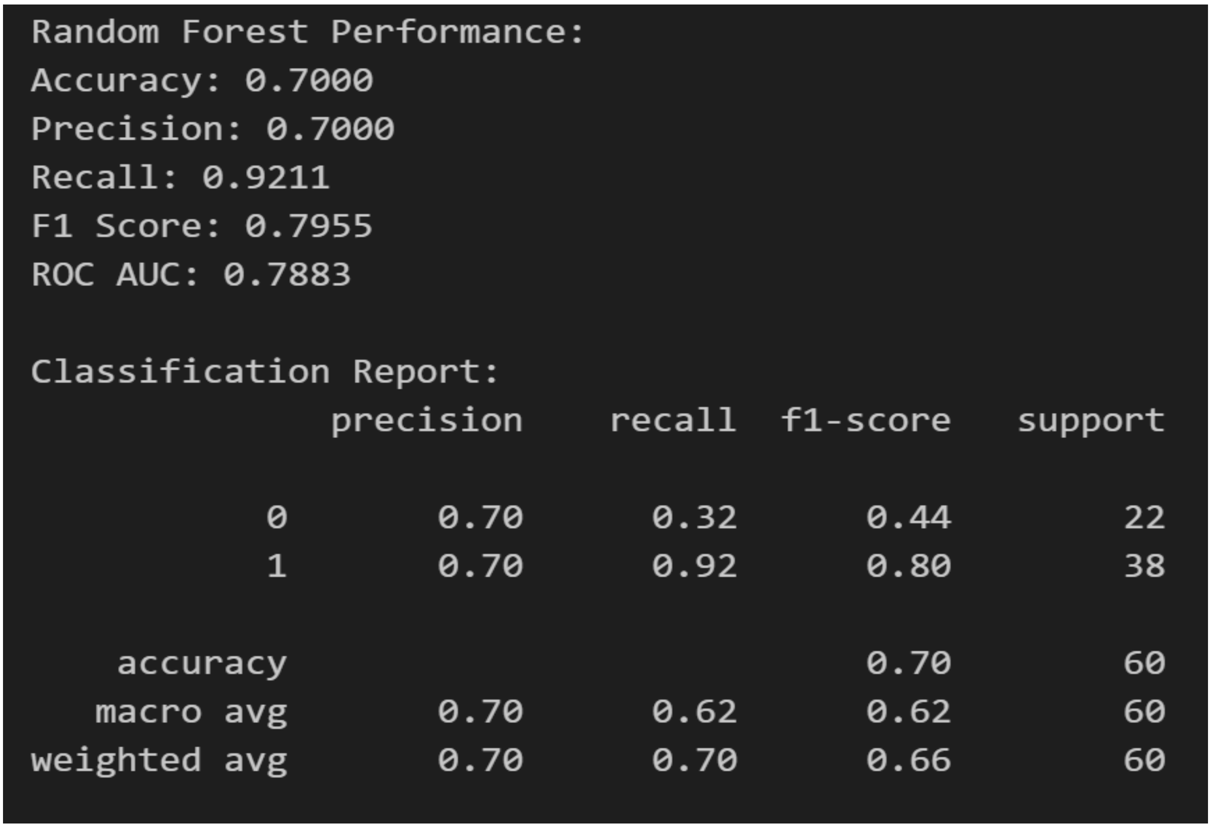
```
Logistic Regression Performance:
Accuracy: 0.8333
Precision: 0.8182
Recall: 0.9474
F1 Score: 0.8780
ROC AUC: 0.9163

Classification Report:
              precision    recall  f1-score   support

           0       0.88      0.64      0.74        22
           1       0.82      0.95      0.88        38

    accuracy                           0.83        60
   macro avg       0.85      0.79      0.81        60
weighted avg       0.84      0.83      0.83        60
```

- **Random Forest:** Ensemble method to capture non-linear relationships.



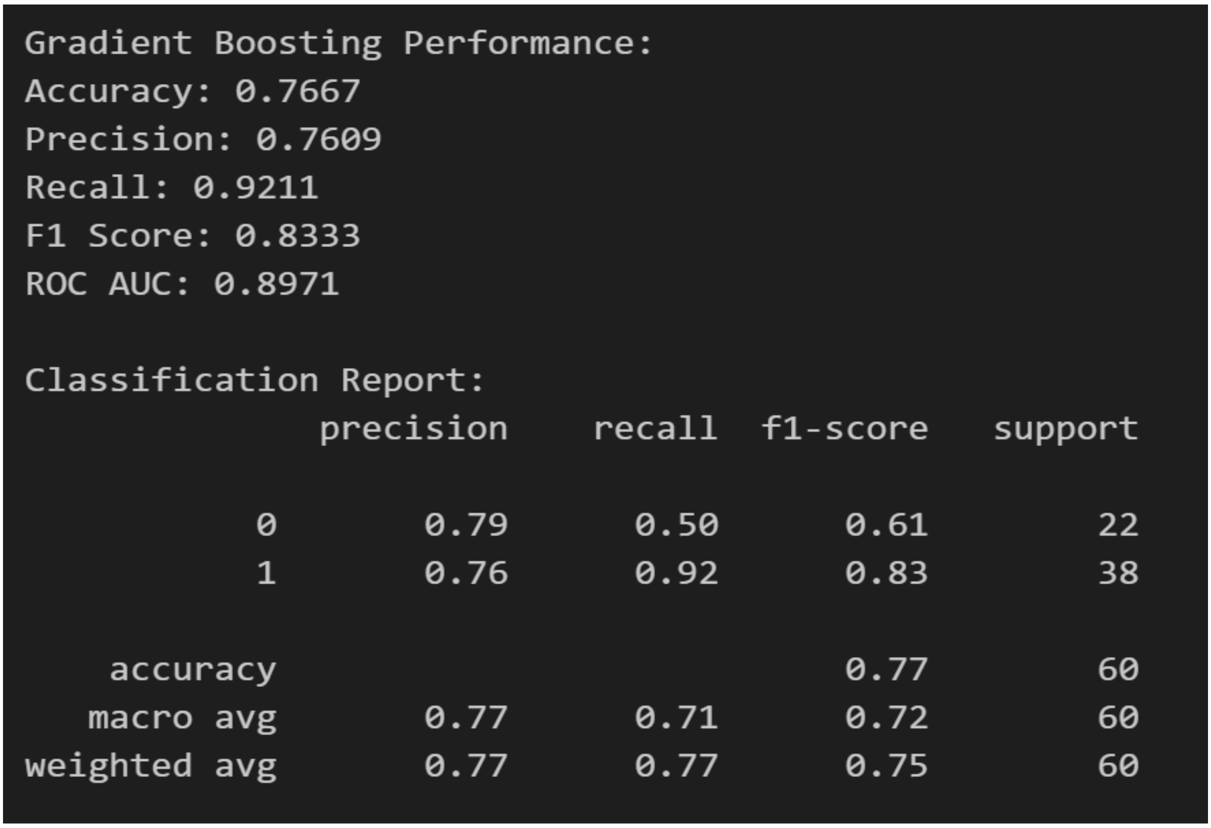
```
Random Forest Performance:
Accuracy: 0.7000
Precision: 0.7000
Recall: 0.9211
F1 Score: 0.7955
ROC AUC: 0.7883

Classification Report:
              precision    recall  f1-score   support

           0       0.70      0.32      0.44        22
           1       0.70      0.92      0.80        38

    accuracy                           0.70        60
   macro avg       0.70      0.62      0.62        60
weighted avg       0.70      0.70      0.66        60
```

- **Gradient Boosting:** Ensemble weak learners tuned via GridSearchCV.



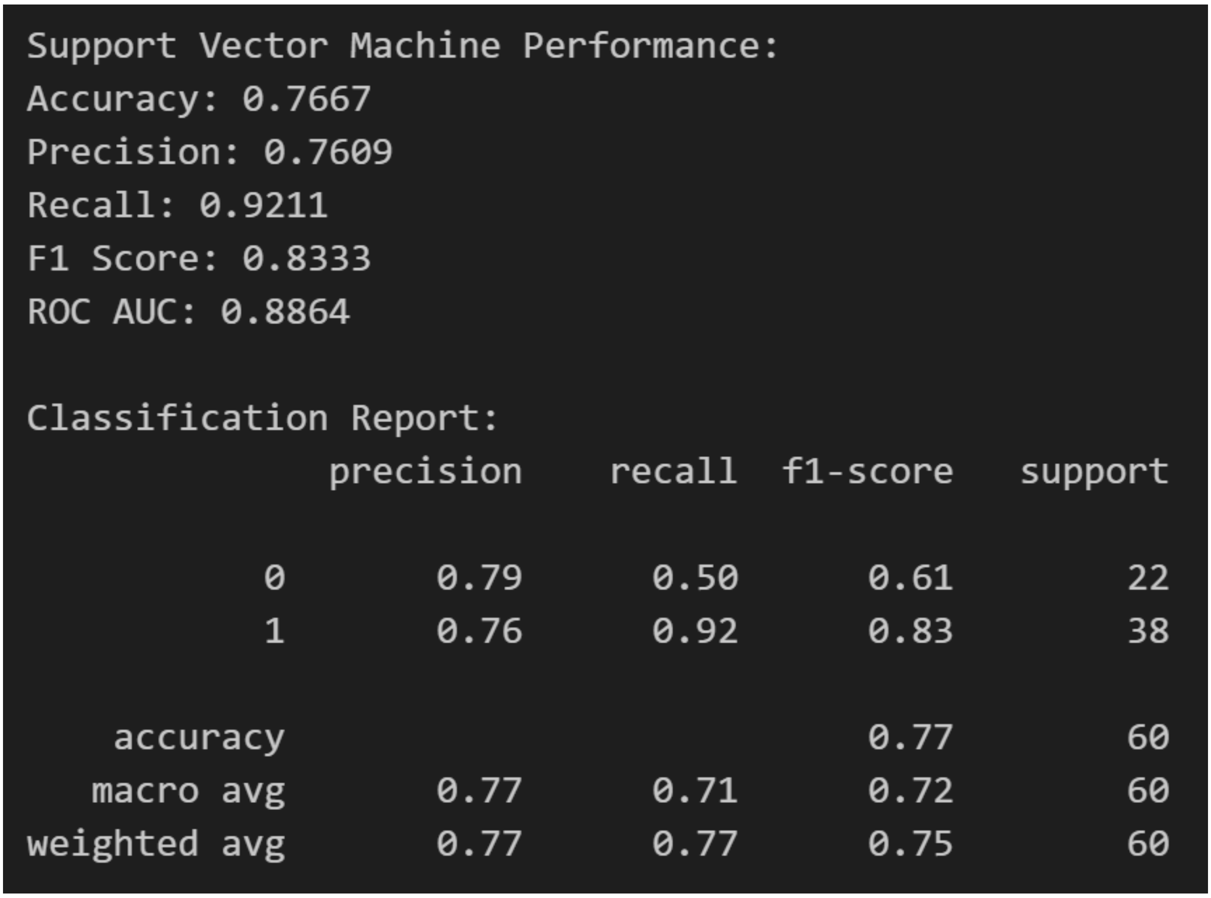
```
Gradient Boosting Performance:
Accuracy: 0.7667
Precision: 0.7609
Recall: 0.9211
F1 Score: 0.8333
ROC AUC: 0.8971

Classification Report:
              precision    recall  f1-score   support

           0       0.79      0.50      0.61        22
           1       0.76      0.92      0.83        38

    accuracy                           0.77        60
   macro avg       0.77      0.71      0.72        60
weighted avg       0.77      0.77      0.75        60
```

- **Support Vector Machine (SVM):** SVM with probability estimates for binary classification.



```
Support Vector Machine Performance:
Accuracy: 0.7667
Precision: 0.7609
Recall: 0.9211
F1 Score: 0.8333
ROC AUC: 0.8864

Classification Report:
              precision    recall  f1-score   support

           0       0.79      0.50      0.61        22
           1       0.76      0.92      0.83        38

    accuracy                           0.77        60
   macro avg       0.77      0.71      0.72        60
weighted avg       0.77      0.77      0.75        60
```

7. Hyperparameter Tuning & Final Model Selection

7.1 Hyperparameter Tuning

A GridSearchCV was used to tune the hyperparameters of the Gradient Boosting classifier. The parameters tuned included:

- Number of estimators
- Maximum depth
- Learning rate
- Minimum samples required to split an internal node
- Minimum samples required at a leaf node
- Subsample ratio
- Maximum features

Best Parameters: {'learning_rate': 0.01, 'max_depth': 3, 'max_features': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200, 'subsample': 1.0}

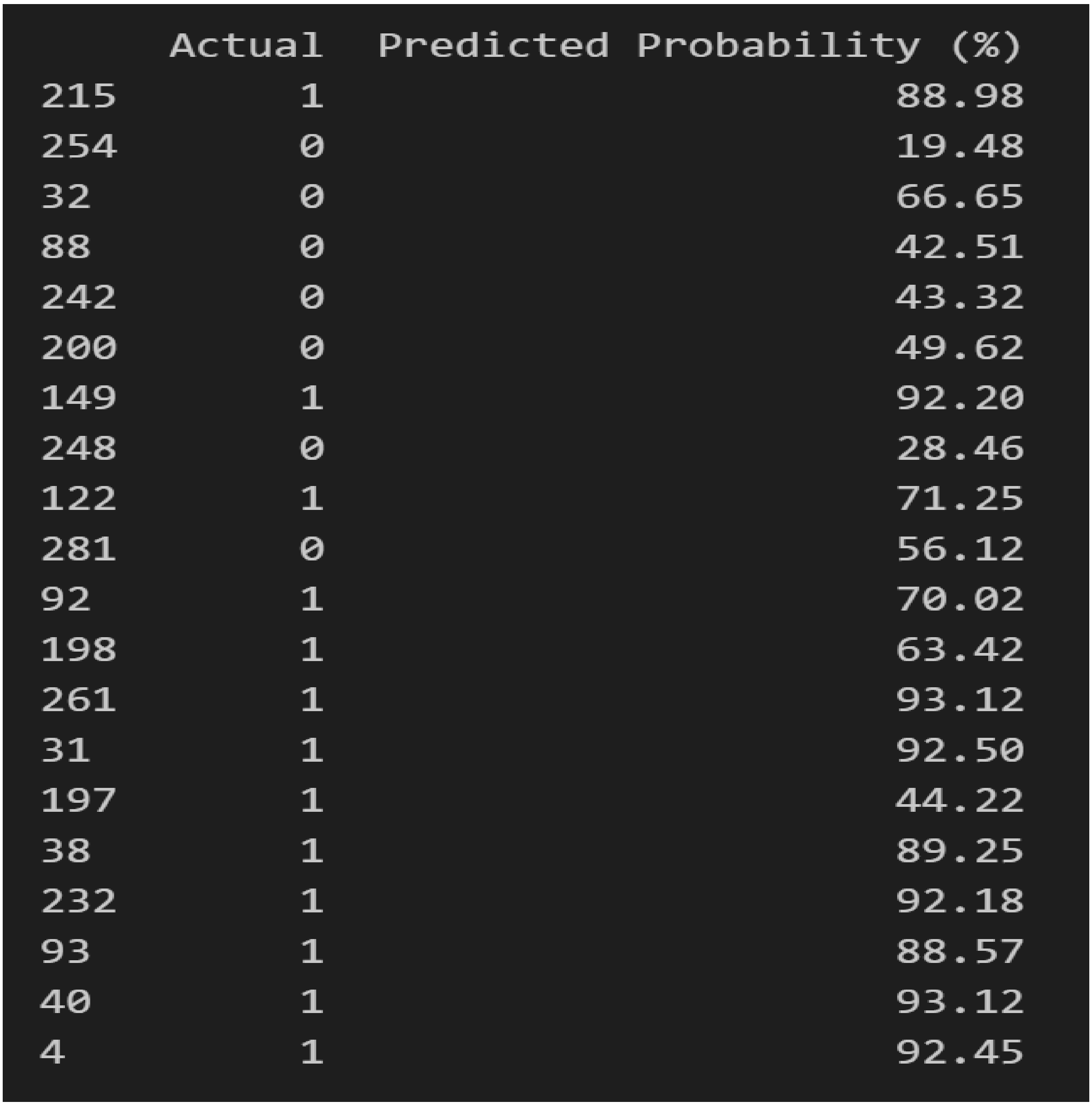
7.2 Final Model Selection

The best estimator was selected based on ROC AUC score and saved as ***rain_prediction_model.pkl*** for future inference.

ROC-AUC Score: 0.9198564593301435

8. Predict Rain Probability

The final output of the system is to provide the probability of rain. This section outlines how the model is used to generate probability predictions on the test set.



```
     Actual  Predicted Probability (%)
215       1                      88.98
254       0                      19.48
32        0                      66.65
88        0                      42.51
242       0                      43.32
200       0                      49.62
149       1                      92.20
248       0                      28.46
122       1                      71.25
281       0                      56.12
92        1                      70.02
198       1                      63.42
261       1                      93.12
31        1                      92.50
197       1                      44.22
38        1                      89.25
232       1                      92.18
93        1                      88.57
40        1                      93.12
4         1                      92.45
```

9. Conclusion

This project demonstrates the use of machine learning techniques to predict rain events based on historical weather data. The systematic approach—including data preprocessing, feature engineering, model training, and hyperparameter tuning—has resulted in a robust model capable of providing rain probabilities. These predictions offer valuable insights for agricultural decision-making and highlight areas for future research and model enhancement.