# Weather Prediction using Machine Learning

Author: Diksha kumari

Date: October 15, 2025

Github link:-[Diksha-kumari-singh/ERA5-Rainfall-Prediction-Model](https://github.com/Diksha-kumari-singh/ERA5-Rainfall-Prediction-Model)

## Table of Contents

1. Introduction
2. Methodology
3. Data Exploration and Preprocessing
4. Machine Learning Algorithms and Tuning
5. Model Evaluation and Comparative Analysis
6. Conclusion and Discussion

## 1. Introduction

Weather prediction is a critical application of data science that impacts various sectors, including agriculture, transportation, disaster management, and daily life. Accurate forecasting can save lives, optimize resources, and improve economic outcomes. This project focuses on a multi-class classification task to predict the type of weather (e.g., sun, rain, snow) based on meteorological variables, providing a practical demonstration of machine learning's capabilities.

### Objective

The primary objective of this project is to build and evaluate a robust machine learning model for multi-class weather classification. We aimed to:

* Explore and visualize the Seattle weather dataset.
* Preprocess the data and handle severe class imbalances using SMOTEENN.
* Implement and compare a suite of individual classification algorithms.
* Enhance model performance through hyperparameter tuning and advanced ensemble techniques.
* Conduct a comprehensive comparative analysis to identify the most effective predictive model.

## 2. Methodology

Our approach follows a structured machine learning workflow, starting from data exploration and ending with a comparative analysis of advanced ensemble models.

### Project Workflow

The overall process followed a standard data science pipeline: Data Loading EDA Preprocessing Imbalance Handling (SMOTEENN) Train-Test Split Model Training (Individual and Ensemble) Evaluation and Comparison.

### Evaluation Methods and Equations

To evaluate our classification models, we used standard metrics including Accuracy, Precision, Recall, and the F1-Score.

* Accuracy: The ratio of correctly predicted instances to the total instances.
* Precision: The ratio of correctly predicted positive instances to the total predicted positive instances.
* Recall (Sensitivity): The ratio of correctly predicted positive instances to all instances in the actual class.
* F1-Score: The harmonic mean of Precision and Recall, which is crucial for evaluating models on imbalanced multi-class datasets.

## 3. Data Exploration and Preprocessing

### Data Characteristics

We utilized the Seattle Weather dataset, which was found to be complete with no missing values.

* **Columns:** date, precipitation, temp\_max, temp\_min, wind, weather (Target).
* **Target Classes:** The five unique weather classes were: 'drizzle', 'rain', 'sun', 'snow', and 'fog'.

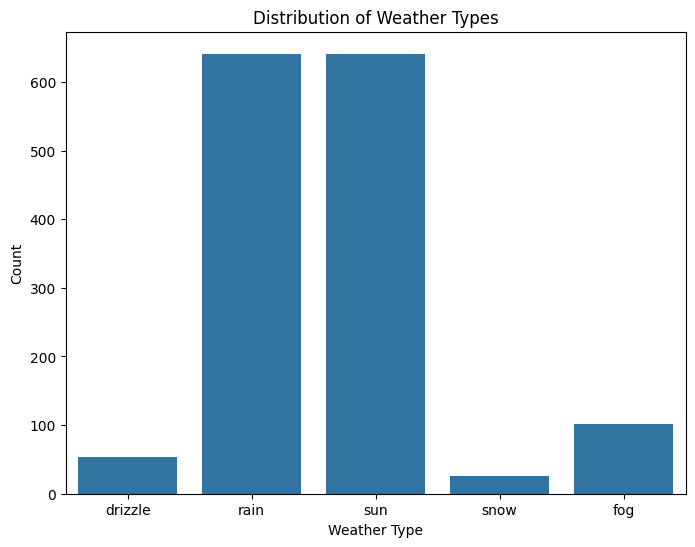
### Exploratory Data Analysis (EDA)

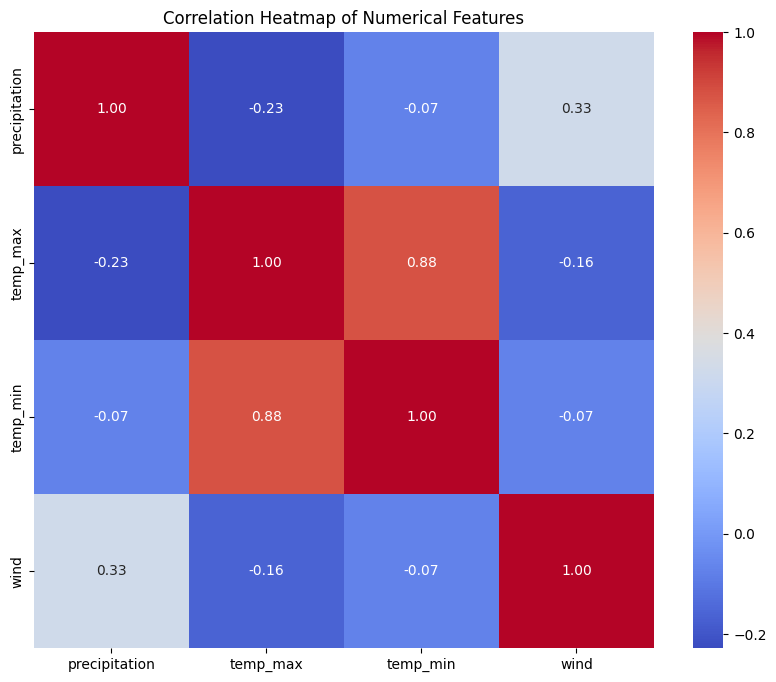
The initial EDA revealed a highly imbalanced class distribution, with 'rain' and 'sun' dominating the original dataset.

* **Initial Class Distribution:** rain (641), sun (640), fog (101), drizzle (53), snow (26).
* **Correlation:** A strong positive correlation was observed between temp\_max and temp\_min.

### Data Balancing and Splitting

* **Handling Imbalance:** The **SMOTEENN** (Synthetic Minority Over-sampling Technique combined with Edited Nearest Neighbors) method was applied. This corrected the distribution, leading to a much more balanced training set (e.g., snow at 635, sun at 185) for effective model learning.
* **Splitting:** The balanced dataset was split into an **80% training set** and a **20% testing set**.





## 4. Machine Learning Algorithms and Tuning

### Individual ML Algorithms

Seven individual classification models were implemented and evaluated on the test set:

|  |  |
| --- | --- |
| **Model** | **Initial Test Accuracy (%)** |
| **XGBoost** | 96.09% |
| **Random Forest** | 95.86% |
| KNN | 92.87% |
| Decision Tree | 92.18% |
| SVM | 77.01% |
| Logistic Regression | 75.86% |
| AdaBoost | 47.36% |

XGBoost and Random Forest showed the highest initial performance, while AdaBoost performed poorly, demonstrating its incompatibility with the dataset's characteristics.

### Hyperparameter Tuning

Hyperparameter tuning was performed on the best individual model, **XGBoost**, using GridSearchCV.

* **Best Parameters:** The tuning process yielded the best parameters as: colsample\_bytree: 0.9, learning\_rate: 0.1, max\_depth: 7, n\_estimators: 200, and subsample: 0.7.
* **Tuned Accuracy:** The tuned XGBoost model achieved a test accuracy of **96.09%**, confirming the model's robustness and optimal configuration.

### Ensemble Techniques

Two ensemble techniques were implemented to push the predictive performance boundary further.

* **Voting Classifier (Soft Voting):** A soft voting ensemble was constructed using Random Forest, KNN, and the Tuned XGBoost model. By averaging the predicted probabilities, this model successfully leveraged the complementary strengths of the base learners, achieving a test accuracy of **97.47%**.
* **Stacking Classifier:** The Stacking Classifier, using a Logistic Regression meta-model over a diverse set of base models, learned the optimal way to combine their outputs. This technique proved to be the most effective strategy.

## 5. Model Evaluation and Comparative Analysis

### Comparative Analysis Table

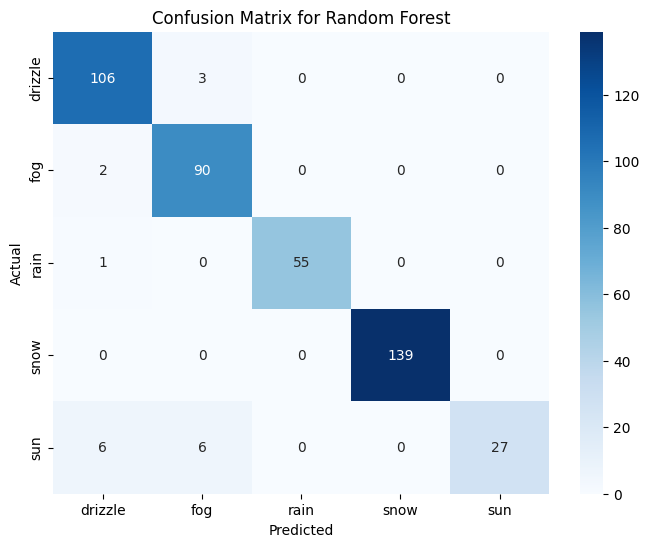
The final comparative analysis is presented below, confirming the superior performance of the ensemble methods.

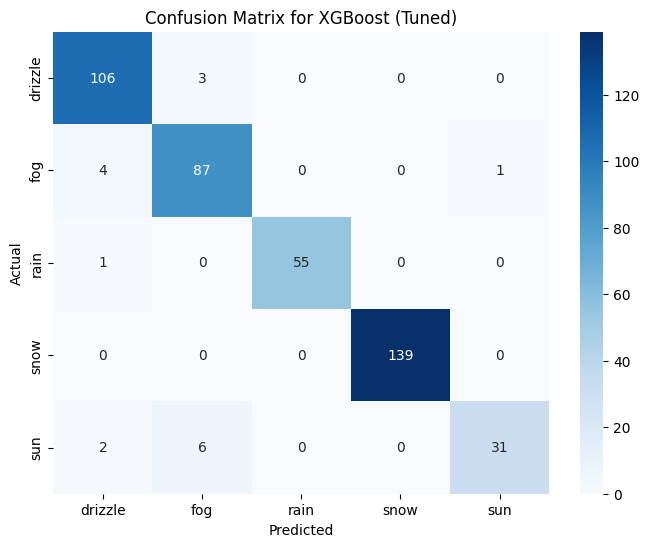
**Table 1: Final Comparative Analysis of Model Performance**

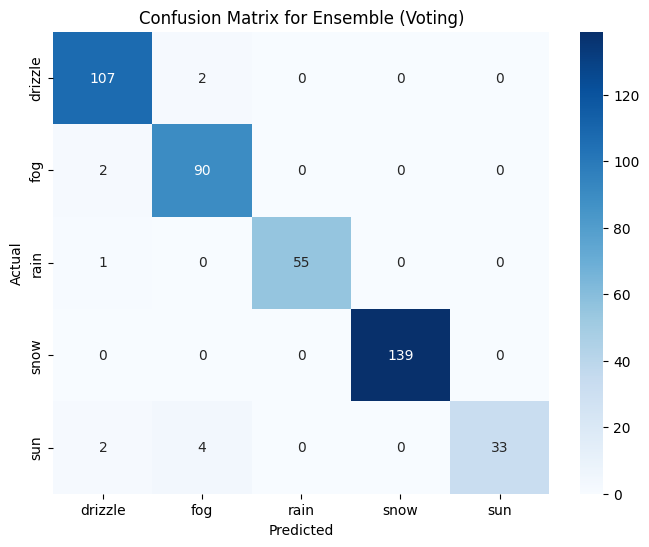
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
| **Ensemble (Stacking)** | **0.9862** | **0.9864** | **0.9862** | **0.9862** |
| **Ensemble (Voting)** | 0.9747 | 0.9756 | 0.9747 | 0.9744 |
| XGBoost (Tuned) | 0.9609 | 0.9618 | 0.9609 | 0.9604 |
| Random Forest | 0.9586 | 0.9612 | 0.9586 | 0.9569 |
| KNN | 0.9287 | 0.9326 | 0.9287 | 0.9275 |
| Decision Tree | 0.9218 | 0.9261 | 0.9218 | 0.9192 |
| SVM | 0.7701 | 0.8087 | 0.7701 | 0.7614 |
| Logistic Regression | 0.7586 | 0.7661 | 0.7586 | 0.7584 |
| AdaBoost | 0.4736 | 0.5475 | 0.4736 | 0.4692 |

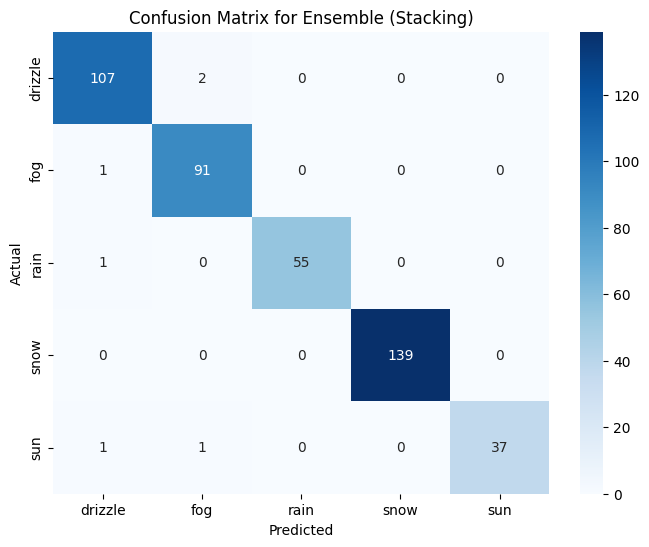
### Confusion Matrices

Visual inspection of the confusion matrices for the top models shows that the Stacking Classifier exhibits the highest concentration of true positives and the fewest misclassifications, indicating its superior generalization ability across all five weather classes.









**Figure: Confusion Matrices**

## 6. Conclusion and Discussion

The project successfully developed and evaluated a range of machine learning models for weather prediction on the Seattle Weather dataset.

* **Preprocessing is Crucial:** The application of **SMOTEENN** to handle the severe class imbalance was the critical foundation that enabled all models to achieve meaningful performance metrics.
* **Ensemble Models Excel:** The ensemble techniques provided a clear and substantial performance boost over the best individual models. The **Stacking Classifier** achieved the highest accuracy of **98.62%**, making it the final, most reliable model for this prediction task.
* **Model Selection Matters:** The contrast between the high performance of tree-based and ensemble models and the poor performance of models like AdaBoost and basic Linear/SVM models underscores the importance of selecting algorithms suited to the data characteristics.

The final Stacking model is highly accurate and robust, demonstrating that a systematic machine learning approach—combining careful data preprocessing with advanced ensemble methods—can yield state-of-the-art results in multi-class classification.