

# Rainfall Prediction System

- SAMIKSHA PATIL

## 1. Introduction :

Rainfall plays a crucial role in agriculture, water resource management, and disaster preparedness. Predicting rainfall accurately can aid farmers, meteorological departments, and policymakers in effective planning. This project presents a **machine learning-based Rainfall Prediction System** integrated with a **web-based interface** to provide easy and real-time access to predictions.

## 2. Objective:

The primary goal is to **predict the expected rainfall** based on multiple weather-related input features using machine learning models and provide a **user-friendly web interface** for predictions.

- Train a machine learning model using real-world weather data.
- Develop a web interface using Flask for prediction.
- Implement form-based user input for rainfall features.
- Return real-time prediction on the web.

## 3. Dataset

The dataset used is **rainfall dataset.csv**, which includes multiple environmental features.

### Sample Features:

- Temperature
- Humidity
- Wind Speed
- Pressure
- Cloud Cover
- Dew Point
- Visibility
- Solar Radiation
- **Target Variable: Rainfall** (in mm or a binary class indicating rainfall occurrence)

## 4. Model Used:

The primary model used for rainfall prediction is:

- **XGBoost Classifier**
- Logistic Regression
- Support Vector Machine (SVM)
- Random Forest

## 5. Evaluation Metrics

To assess the model's performance, the following metrics were used:

- **Accuracy:** Overall correctness of the prediction
- **Confusion Matrix:** To understand the true/false positives and negatives
- **Precision / Recall** (for classification tasks)
- **Mean Squared Error (MSE)** (for rainfall amount prediction)

## 6. System Architecture:

### Backend:

- Developed in Python using Flask framework
- Trained model and data scaler saved using pickle
- Accepts user inputs and returns prediction result

### Frontend:

- Web-based interface using HTML, CSS, and Jinja2 templating
- User enters feature values through a form → Result is displayed instantly after model

inference

## 7. Screenshots:

The image displays two screenshots of a web application titled "Rainfall Prediction".

The left screenshot shows the input form with the following fields and values:

- Day: 4
- Pressure: 1003
- Maxtemp: 27
- Temperature: 25
- Mintemp: 23
- Dewpoint: 22
- Humidity: 95
- Cloud: 8
- Sunshine: (empty)

A large black arrow points from the left form to the right form.

The right screenshot shows the same form with the following values entered:

- Day: 23
- Dewpoint: 22
- Humidity: 95
- Cloud: 8
- Sunshine: 0.5
- Winddirection: 190
- Windspeed: 6

Below the "Windspeed" field is a green "Predict" button. Below the button, the output is displayed: "Will it rain? No".

## 8. Conclusion:

The **Rainfall Prediction System** successfully predicts rainfall based on meteorological features using a machine learning model (XGBoost). The system is deployed using Flask and offers a clean, functional interface for real-time rainfall prediction. With future improvements such as integration of real-time weather APIs or sensor data, the system can be extended for use in precision farming, flood alerts, and climate monitoring.