Exploratory Analysis and Rainfall Prediction for Agriculture

---

1. Abstract

This project presents an end-to-end machine learning system for rainfall prediction based on historical meteorological data. The objective is to assist agricultural planning by predicting whether rainfall will occur the next day using supervised learning techniques. The system performs exploratory data analysis, data preprocessing, model training, evaluation, and deployment through a Flask-based web application. A Random Forest classifier was selected as the final model based on performance metrics such as accuracy and ROC–AUC score.

---

2. Introduction

Rainfall plays a crucial role in agricultural productivity. Accurate rainfall prediction helps farmers make informed decisions regarding crop planning, irrigation scheduling, and risk management. This project aims to build a machine learning-based predictive system that classifies whether it will rain tomorrow based on current weather parameters.

The system integrates:

- Data analysis
- Machine learning model development
- Model evaluation
- Web application deployment

---

3. Problem Statement

To develop a classification model that predicts whether rainfall will occur the next day based on meteorological features such as temperature, humidity, pressure, cloud cover, and wind speed.

---

4. Objectives

- Perform exploratory data analysis (EDA)
- Clean and preprocess real-world weather data
- Train multiple classification models
- Evaluate model performance
- Deploy the final model using Flask
- Provide an interactive user interface

---

5. Dataset Description

- Dataset: Weather historical dataset
- Total Records: 142,193
- Features: 24 original columns
- Target Variable: RainTomorrow
- Type: Binary Classification (Yes / No)

Selected Features for Final Model:

- MinTemp
- MaxTemp
- Rainfall
- Evaporation
- Sunshine
- WindGustSpeed
- WindSpeed9am
- WindSpeed3pm
- Humidity9am
- Humidity3pm
- Pressure9am
- Pressure3pm
- Cloud9am
- Cloud3pm
- Temp9am
- Temp3pm

Categorical features were removed during final deployment to ensure feature consistency.

---

6. System Architecture

Flow of the System:

User Input (Web Form)
↓
Flask Backend
↓
Data Preprocessing (Imputer + Scaler)
↓
Random Forest Model
↓
Prediction Output (Rain / No Rain Page)

---

7. Methodology

Step 1: Data Preprocessing

- Removed unnecessary and leakage features (e.g., RISK_MM)
- Handled missing values using SimpleImputer (mean strategy)
- Standardized numerical features using StandardScaler

Step 2: Train-Test Split

- 80% Training
- 20% Testing
- Stratified sampling applied

Step 3: Model Training

Three models were tested:

- Logistic Regression
- Decision Tree
- Random Forest

Step 4: Model Selection

Random Forest was selected due to:

- Higher accuracy
- Better ROC–AUC score
- Reduced overfitting compared to Decision Tree

---

8. Model Evaluation

Final Model: Random Forest Classifier

- Accuracy: 85.44%
- ROC–AUC Score: 0.882
- Confusion Matrix:

[[21116 948]

 [3192 3183]]

This indicates good classification performance with balanced precision and recall.

---

9. Deployment

The trained model was serialized using pickle and integrated into a Flask web application.

Deployment Components:

- rainfall_model.pkl
- scaler.pkl
- imputer.pkl
- Flask backend (app.py)
- HTML frontend (templates)

Users enter weather parameters and receive real-time rainfall predictions.

🔟 Technologies Used

- Python
- Pandas
- NumPy
- Scikit-learn
- Flask
- HTML & CSS
- Git & GitHub

1️⃣1️⃣ Advantages

- Helps in agricultural planning
- Reduces manual weather dependency
- Provides real-time prediction
- Easy to use web interface

1️⃣2️⃣ Limitations

- Model depends on historical dataset
- Accuracy may vary across geographical regions
- Does not use real-time weather API

1️⃣3️⃣ Future Enhancements

- Integration with live weather APIs
- Deployment to cloud platform
- Addition of probability/confidence score

- Mobile responsive UI
- Deep learning model experimentation

1️⃣4️⃣ Conclusion

The project successfully demonstrates the application of machine learning in predicting rainfall for agricultural purposes. Through proper data preprocessing, model evaluation, and deployment, an efficient and practical rainfall prediction system was developed. The final Random Forest model achieved satisfactory performance and was successfully integrated into a web application for real-time usage.