FSD Documentation for Rainfall Prediction Web App

1. Introduction

Project Title: Exploratory Analysis of Rain Fall Data in India for Agriculture

Team Members: B.Divya Reddy

K.Mounika

E.Madan Dilip

N. Sai Teja Surya Vaibhav

2. Project Overview

- **Purpose:** To predict rainfall based on weather parameters such as temperature, humidity, pressure, and wind speed using a machine learning model and provide actionable insights.
- **Features:**
  - User-friendly web interface with input fields for MinTemp, MaxTemp, Humidity, Pressure, WindSpeed.
  - Predict button returns chance of rain or no rain.
  - Error handling for invalid inputs.
  - Dynamic prediction using dataframe structure consistent with training data.
  - Updated predict() function handles only necessary fields, calculates derived features, and sets default values for Rainfall and RainToday.

3. Architecture

- **Frontend:** React not used; HTML forms (index.html) used for input.
- **Backend:** Flask handles routing, prediction logic, and rendering templates (chance.html, nochance.html).
- **Database:** Not required; uses pre-trained model and scaler (Rainfall.pkl and scale.pkl).

4. Setup Instructions

- **Prerequisites:** Python, Flask, pandas, numpy, scikit-learn, pickle-mixin
- **Installation:**

```
git clone https://github.com/divyabhuma/Exploratory-Analysis-of-Rain-Fall-Data-in-India-for-Agriculture
```

```
cd RainfallPrediction
python -m venv venv
# Windows
env\Scripts\activate
# Mac/Linux
source venv/bin/activate
pip install -r requirements.txt
```

5. Folder Structure

- **Client:** templates/index.html (home page), chance.html, nochance.html
- **Server:** app.py (Flask backend with updated predict() function)
- **Static:** static/style.css (CSS for styling web pages)
- **Model:** Rainfall.pkl (ML model), scale.pkl (feature scaler)

6. Running the Application

- **Frontend & Backend:** Run Flask app with:

```
python app.py
```

- Open browser at `http://127.0.0.1:5000/`
- Input weather parameters and click Predict.
    - Redirects to chance.html if rain is predicted.
    - Redirects to nochance.html if no rain.

7. API Documentation

- **Endpoint:** `/predict` (POST)
    - **Parameters:** MinTemp, MaxTemp, Humidity, Pressure, WindSpeed
    - **Response:** Renders chance.html or nochance.html based on prediction.

8. Authentication

- Not implemented; public access.

9. User Interface

- **index.html:** Input form for weather parameters.
- **chance.html:** Displays prediction if rain is expected.
- **nochance.html:** Displays prediction if no rain is expected.

10. Testing

- Tested multiple input combinations for rain and no rain.
- Performance verified: response <2 seconds.

11. Screenshots or Demo

- Repository link for demo: https://github.com/divyabhuma/Exploratory-Analysis-of-Rain-Fall-Data-in-India-for-Agriculture

12. Known Issues

- Prediction accuracy depends on dataset quality.
- Extreme weather events may not be accurately predicted.

13. Future Enhancements

- Add date and location input for localized predictions.
- Integrate interactive charts for weekly weather forecasts.
- Deploy on Heroku or AWS.
- Integrate real-time weather API.