# Exploratory Analysis of Rainfall Data in India for Agriculture

## 1. Requirements Analysis

The objective of this system is to analyze historical and real-time rainfall data across India to support agricultural decision-making. The system should help farmers, agricultural planners, and policymakers understand rainfall patterns, variability, and trends to improve crop planning and risk management.

**Functional Requirements:**

- Upload rainfall datasets (CSV/Excel) from sources like IMD.
- Clean, preprocess, and validate rainfall data.
- Perform exploratory data analysis (EDA) by region, season, and year.
- Generate visualizations such as monthly/seasonal rainfall trends.
- Export analysis results and reports.

**Non-Functional Requirements:**

- Scalability to handle large historical datasets.
- Accuracy and reliability of data processing.
- User-friendly interface for non-technical users.
- Secure data storage and access control.

## 2. Customer Journey Analysis

- User logs into the system (Farmer/Analyst/Policy Maker).
- User uploads rainfall dataset or selects predefined data.
- System processes and cleans the data automatically.
- User explores dashboards and rainfall insights.
- User downloads reports or uses insights for crop planning.

## 3. Data Flow Diagram (DFD) – Conceptual Description

- External Entity: User (Farmer/Analyst).
- Process 1: Data Upload & Validation.
- Process 2: Data Cleaning & Transformation.
- Process 3: Rainfall Analysis & Aggregation.
- Data Store: Rainfall Database.
- Output: Reports, Dashboards, Visualizations.

## 4. Solution Requirements

- Reliable rainfall data sources (IMD, government open data portals).
- Automated ETL pipeline for data ingestion.
- Analytical engine for statistical and trend analysis.

- Visualization layer for charts and maps.
- Report generation and export functionality.

## 5. Technology Stack

- Data Collection: CSV, Excel, APIs (IMD datasets).
- Backend / Analysis: Python, Pandas, NumPy.
- Visualization: Matplotlib, Seaborn, Plotly.
- Database: MySQL / PostgreSQL.
- Frontend: Streamlit / Flask / Dash.
- Deployment: Cloud (AWS/GCP/Azure).