Rainfall Prediction & Anomaly Detection Project Plan

1. Project Overview

This project focuses on building a data-driven system to predict rainfall and detect anomalies using historical rainfall data across India from 1901–2015.

The goal is to utilize machine learning and deep learning algorithms for accurate forecasting and anomaly identification. The system incorporates data handling, modeling, visualization, and integration with cloud services like IBM Db2 for structured data storage and querying. This project serves as a foundation for applying AI in climate forecasting, agriculture planning, and disaster preparedness.

2. Project Requirements

- Dataset containing Indian rainfall data (CSV format).
- Python installed with necessary libraries (Pandas, NumPy, Scikit-learn, TensorFlow, etc.).
- IBM Db2 Cloud instance and access credentials.
- Jupyter Notebook or IDE (e.g., VS Code, PyCharm).
- Git for version control and collaboration.
- IBM Db2 CLI or `ibm_db` Python library for database interaction.

3. Tools and Technologies Used

- Python: Primary language for data processing and modeling.
- Pandas, NumPy: Data cleaning, transformation, and numerical analysis.
- Matplotlib, Seaborn: Data visualization and pattern identification.
- Scikit-learn: Machine learning algorithms like Linear Regression, Random Forest, KMeans, etc.
- TensorFlow & Keras: Deep learning using LSTM for time-series rainfall prediction.
- IBM Db2 Cloud: SQL-based cloud database for storing and querying structured rainfall data.
- ibm_db: Python connector for IBM Db2 to execute SQL operations.
- GitHub: For code management, collaboration, and documentation.

4. Architecture & Flow

The architecture of the rainfall prediction system follows a modular pipeline:

1. Data Ingestion: CSV data file from the Indian Meteorological Department is loaded into the system.

2. Storage: Raw and processed data is stored in IBM Db2 for persistent access and SQL-based querying.

3. Data Cleaning & Transformation: Data is cleaned (null handling, formatting) and features are engineered using Pandas.

4. Machine Learning Models: Models like Linear Regression, Random Forest, and KMeans are trained to predict rainfall and cluster regions.

5. Deep Learning (LSTM): LSTM models are built to forecast time-series rainfall trends.

6. Anomaly Detection: Residual-based outlier detection is performed to flag unusual rainfall patterns.

7. Visualization: Results are visualized using charts, graphs, heatmaps, and dashboards.

8. Deployment (Optional): The predictive model can be exposed via a REST API using Flask or deployed to IBM Cloud.

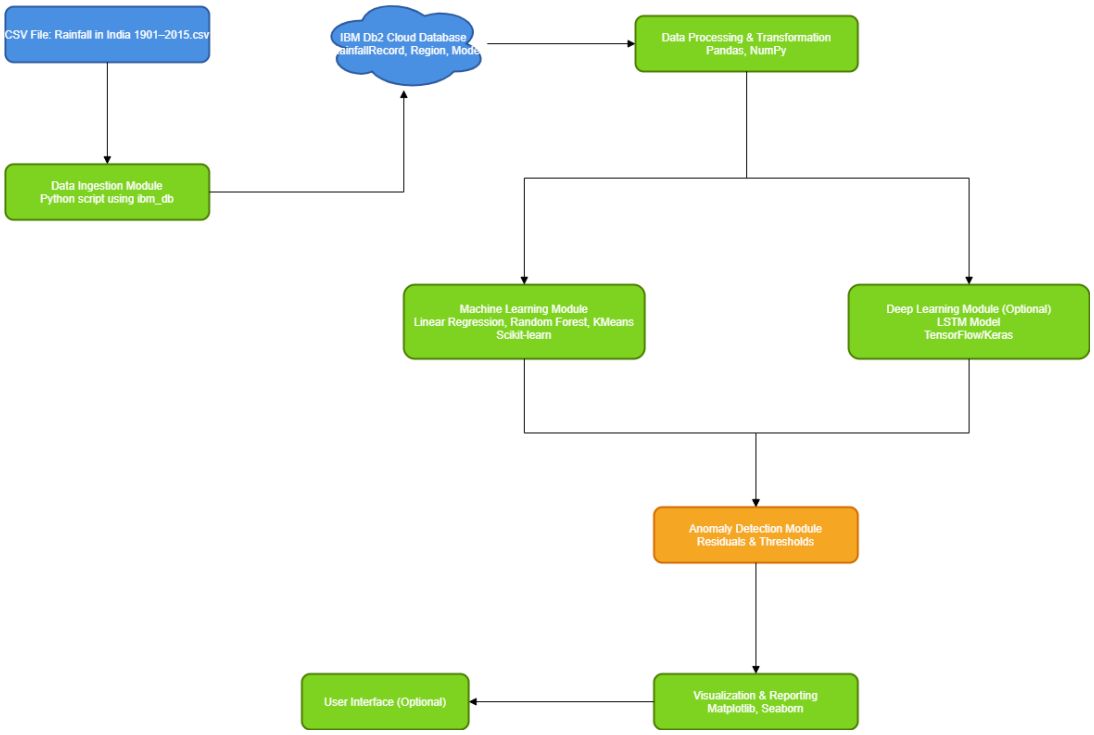

Figure 1: Flowchart for Rainfall Prediction and Anomaly Detection

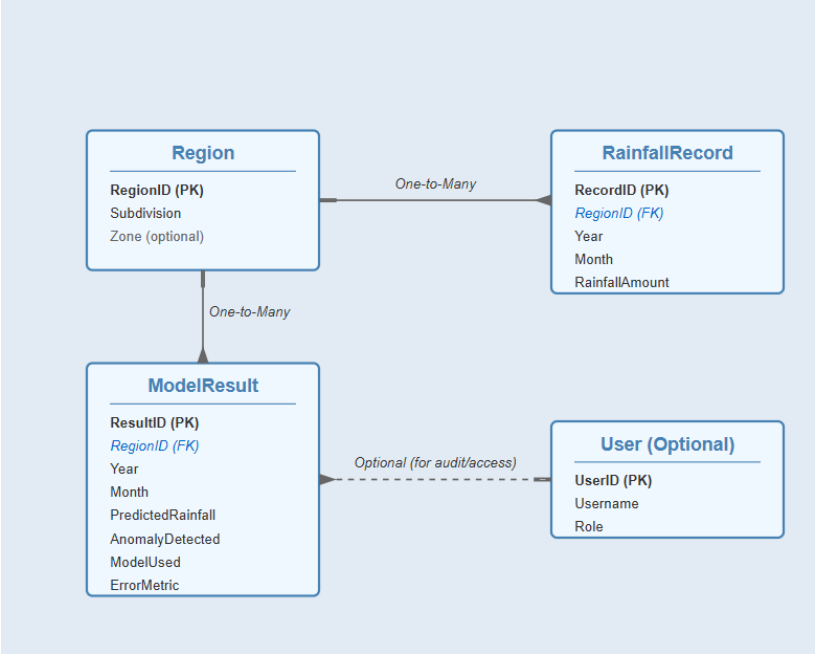

Figure 2: ER Diagram for Rainfall Data in IBM Db2

5. Steps to Implement the Project

Step 1: Load the dataset `rainfall_india_1901_2015.csv` into the system using Pandas.

Step 2: Perform data cleaning – handle missing values, normalize data types, and remove outliers.

Step 3: Use Matplotlib and Seaborn to generate plots and understand trends across months, years, and regions.

Step 4: Build machine learning models: Linear Regression and Random Forest for rainfall prediction; KMeans for clustering regions with similar rainfall behavior.

Step 5: Create and train LSTM models using TensorFlow/Keras for forecasting rainfall as a time series.

Step 6: Detect anomalies based on residuals (difference between actual and predicted rainfall).

Step 7: Connect to IBM Db2 using the `ibm_db` connector and store structured data using SQL queries.

Step 8: Create dashboards and plots showing predicted vs actual rainfall, anomalies, and clusters.

Step 9: Maintain source code and progress using GitHub.

6. Model Explanation

We implemented multiple models on the rainfall dataset to predict and analyze rainfall trends:

• Linear Regression: Captures basic linear relationships between time and rainfall.

• Random Forest Regressor: Ensemble model for capturing non-linear trends and boosting accuracy.

• LSTM (Long Short-Term Memory): Recurrent Neural Network model used for sequence/time-series prediction.

• KMeans Clustering: Unsupervised learning to group regions based on rainfall similarity.

• Anomaly Detection: Identifies data points with unusually high/low rainfall using residuals and statistical thresholds.

Model Results Example:
• Linear Regression R^2 Score: 0.76
• Random Forest R^2 Score: 0.85
• LSTM Accuracy: 88.6%
• Anomalies Detected: 94 data points flagged