Final Report: Indian Rainfall Analysis & Prediction Dashboard

Author: Sahil Bopche **Date:** May 28, 2025 **Contact:** [sahilbopche3@gmail.com]

1. Abstract

This report details the "Indian Rainfall Analysis & Prediction Dashboard," an interactive Streamlit application designed to provide comprehensive insights into India's rainfall patterns. The project leverages historical rainfall data from 1901 to 2021 (the Streamlit app mentions 1901-2020, while the notebook uses data up to 2021) across various meteorological subdivisions of India. The primary objectives include analyzing historical trends, comparing regional variations, performing seasonal analysis, and predicting future rainfall using machine learning. A Random Forest Regressor model was developed for predictions, demonstrating high accuracy. The dashboard features interactive visualizations, allowing users to explore rainfall data, understand trends, and assess potential risks associated with rainfall variability.

2. Introduction

Rainfall is a critical climatic factor for India, significantly impacting its agriculture, economy, and water resources. Understanding its patterns, variability, and future trends is crucial for effective planning and mitigation strategies. The "Indian Rainfall Analysis & Prediction Dashboard" project aims to address this need by providing an accessible platform for exploring and analyzing rainfall data.

Project Objectives:

- Analyze historical rainfall patterns in India from 1901 onwards.
- Compare rainfall data across different meteorological subdivisions of India.
- Develop a machine learning model to predict future rainfall trends.
- Provide interactive visualizations for dynamic exploration of data, including heatmaps and charts.
- Conduct seasonal analysis to understand monthly and seasonal rainfall patterns.
- Offer risk assessment capabilities based on rainfall variability and trend analysis.

The project utilizes data primarily from the RainFall_Data.csv file (referred to as rainfaLLIndia.csv in the notebook), which is sourced from the Indian Meteorological Department.

3. Methodology

The project methodology encompasses data acquisition, preprocessing, exploratory data analysis (EDA), feature engineering, and predictive modelling .

3.1. Data Description

The primary dataset (rainfaLLIndia.csv) contains monthly rainfall data for 36 meteorological subdivisions of India from 1901 to 2021. Key columns include Sub_Division, YEAR, and monthly rainfall values (e.g., JUN, JUL, AUG, SEP), as well as an aggregated JUN-SEP column representing the primary monsoon season rainfall.

3.2. Data Preprocessing & Feature Engineering

The following preprocessing and feature engineering steps were performed as detailed in the RainFall_ML.ipynb notebook:

- **Renaming Columns:** The 'subdivision' column was renamed to 'Sub_Division' for consistency.
- **Duplicate Handling:** The data was checked for exact duplicate rows and duplicates based on 'Sub_Division' and 'YEAR', with no such duplicates found in the provided notebook snippets.
- **Feature Creation:**
    - AVG_RAINFALL: Calculated as the mean of "JUN", "JUL", "AUG", "SEP" rainfall for each record.
    - YOY_CHANGE: Year-on-Year change in JUN-SEP rainfall, calculated per subdivision.
    - LAG_1: JUN-SEP rainfall from the previous year for the same subdivision.
    - LAG_2: JUN-SEP rainfall from two years prior for the same subdivision.
    - RAINFALL_CATEGORY_MM: Categorical feature based on JUN-SEP rainfall (Low: &lt;500mm, Normal: 500-1000mm, High: >1000mm).
- **Encoding:** The categorical Sub_Division column was converted into numerical representation using LabelEncoder.
- **Handling Missing Values:** Rows with NaN values in the engineered features (AVG_RAINFALL, YOY_CHANGE, LAG_1, LAG_2) were dropped. This resulted in a dataset of 4260 rows for model training.

3.3. Exploratory Data Analysis (EDA)

EDA was performed to uncover trends and patterns in the rainfall data. Key visualizations and insights from RainFall_ML.ipynb and the Streamlit app (RainFall.py) include:

- **National Annual Rainfall Trend:**
    - A line plot of year-wise average JUN-SEP rainfall in India (1903-2021) shows no clear long-term trend but high year-to-year variation. More extreme highs and lows are noticeable post-2000, with recent dry years around 2002 and 2015-2019.
    - The Streamlit app also presents this national average monsoon rainfall (1901-2020).
- **Regional Rainfall Variation:**
    - The Streamlit app displays bar charts for the top 5 and bottom 5 regions by average rainfall, highlighting significant geographical disparities.
    - The notebook includes boxplots for JUN-SEP rainfall distribution by the top 10 subdivisions.
- **Monthly Rainfall Patterns:**
    - The Streamlit app features a bar chart for average monthly monsoon rainfall (June-September) with standard deviation at a national level.
- **Long-Term Trends for Subdivisions:**
    - The notebook visualizes year-wise JUN-SEP rainfall trends for the top 6 subdivisions (by data count). Key observations include Arunachal Pradesh having the highest rainfall with a rising trend post-2000, while Madhya Maharashtra and Marathwada show the lowest rainfall with frequent drought-like conditions.

- **Rainfall Distribution:**
  - A histogram of JUN-SEP rainfall across India (all years) indicates that the most common rainfall is between 500-1000mm, with the distribution being right-skewed.
- **Correlation Analysis:**
  - A heatmap of the correlation matrix for rainfall months and the JUN-SEP total shows that July rainfall is highly correlated with the total JUN-SEP rainfall.
- **Rainfall Anomalies (Z-Score Based):**
  - Analysis reveals frequent deficits post-1970s, high excess periods around the 1950s-60s and late 1980s, and an increase in extreme anomalies (both deficits and excesses) in recent decades (post-2000), suggesting greater climate volatility.
- **Clustering Analysis:**
  - KMeans and DBSCAN clustering were used to group subdivisions based on their JUN-SEP and July rainfall patterns, identifying regions with similar climatic characteristics.

3.4. Modeling Approach

- **Model Selection:** A RandomForestRegressor from scikit-learn was chosen for predicting JUN-SEP rainfall.
- **Feature Selection:** The features used for training the model included YEAR, AVG_RAINFALL (mean JUN-SEP rainfall for the subdivision), YOY_CHANGE, LAG_1, LAG_2, and the label-encoded Sub_Division.
- **Data Splitting:** The dataset was split into training and testing sets. The notebook specifies test_size=0.1 and random_state=42.
- **Training:** The Random Forest model was trained on the prepared training dataset.

4. Results and Discussion

4.1. Model Performance

The Random Forest Regressor demonstrated high performance on the test set, as indicated by the metrics in RainFall_ML.ipynb:

- **R^2 Score:** Approximately 0.9998 (specifically, the value 0.999815678237736 is shown in the notebook image output for cell 22)
- **Mean Absolute Error (MAE):** Approximately 1.53 mm (specifically, 1.5316596244131466 mm)
- **Root Mean Squared Error (RMSE):** Approximately 9.55 mm (specifically, 9.55182357997242 mm) These metrics suggest that the model can predict JUN-SEP rainfall with a high degree of accuracy for the given dataset. The Streamlit application's "About" page also mentions that the prediction model is regularly updated and validated.

4.2. Key Visualizations and Insights

The project provides numerous visualizations through the Streamlit dashboard and the Jupyter notebook, offering key insights:

- **National Trends:** While there's no distinct long-term increase or decrease in national average rainfall, there's significant year-to-year variability and a tendency towards more extreme events in recent decades.
- **Regional Disparities:** Coastal and northeastern regions like Arunachal Pradesh and Konkan & Goa experience the highest rainfall, whereas northwestern and some interior regions like West Rajasthan and Marathwada are drier and more prone to drought.
- **Seasonal Dominance:** July and August are typically the peak monsoon months nationally. The high correlation of July rainfall with the total monsoon rainfall underscores its importance.
- **Subdivision-Specific Trends:** Different subdivisions exhibit unique long-term rainfall trends and seasonal patterns, highlighting the need for localized analysis. For instance, Arunachal Pradesh shows a rising trend post-2000.
- **Increased Volatility:** Z-score analysis points towards increased rainfall variability and more frequent dry spells post-1970s, with a rise in extreme anomalies in recent decades.

4.3. Prediction and Risk Assessment

The Streamlit application provides a "Rainfall Prediction" section where users can select a subdivision and a future year (e.g., up to 2040) to get rainfall predictions.

- **Future Trends:** The dashboard displays predicted JUN-SEP rainfall trends along with confidence intervals.
- **Seasonal Pattern Comparison:** It allows comparison of historical seasonal patterns with predicted patterns.
- **Distribution Analysis:** Histograms compare historical and predicted rainfall distributions.
- **Risk Assessment:** A rainfall percentile comparison table (10th, 25th, 50th, 75th, 90th percentiles) for historical vs. predicted rainfall helps in assessing the risk of low or high rainfall events.

5. Conclusion

The "Indian Rainfall Analysis & Prediction Dashboard" successfully meets its objectives by providing a robust platform for analyzing historical rainfall data and predicting future trends. The Random Forest Regressor model achieves high accuracy in its predictions. The interactive visualizations and detailed analyses offered by the dashboard, covering national, regional, and seasonal aspects, as well as risk assessments, make it a valuable tool for understanding India's complex rainfall patterns. Key findings indicate significant year-to-year and regional variability in rainfall, with recent decades showing a tendency towards more extreme events and increased climate volatility.

6. Future Work and Recommendations

- **Model Enhancement:** Explore other machine learning models or ensemble techniques (e.g., Gradient Boosting, LSTM for time series) to potentially improve prediction accuracy or capture more complex temporal dependencies.
- **Incorporate More Features:** Include additional meteorological parameters (e.g., temperature, humidity, El Niño/La Niña indices) or agricultural data to enhance predictive power and contextual understanding.

- **Finer Granularity:** If data is available, extend the analysis and predictions to a more granular level (e.g., district-level).
- **Climate Change Scenarios:** Integrate different climate change scenarios (e.g., IPCC RCPs) to assess their potential impact on future rainfall patterns.
- **User Feedback Integration:** Implement a mechanism for users to provide feedback for continuous improvement of the dashboard.
- **Deployment & Scalability:** Ensure the Streamlit application is robustly deployed and can scale to handle more users or larger datasets if required.

7. Tools and Technologies Used

The project utilized the following key technologies and Python libraries:

- **Python 3.8+**
- **Streamlit:** For building the interactive web dashboard.
- **Pandas:** For data manipulation and analysis.
- **NumPy:** For numerical operations.
- **Scikit-learn:** For machine learning, including RandomForestRegressor, LabelEncoder, train_test_split, and metrics.
- **Plotly (Plotly Express & Graph Objects):** For creating interactive visualizations.
- **Seaborn & Matplotlib:** For static visualizations in the EDA phase.
- **Streamlit-Lottie:** For integrating Lottie animations in the dashboard.
- **Requests:** For making HTTP requests (used for Lottie animations).
- **Jupyter Notebook:** For ML model development and EDA.

8. Contact

Sahil Bopche

- Email: [sahilbopche3@gmail.com]
- Project Link: [https://github.com/sahil2k4/RainFall_Prediction]