Flood Prediction in South Asia: A Machine Learning Approach

How to Build and Run the Code

To reproduce the results, follow the steps below:

1. **Clone the Repository**: Download or clone the repository containing the code and dataset.

Install Dependencies: Ensure Python 3.x is installed on your system. Install all required libraries using the `Makefile` provided. Run the following command:

```
make install
```

2. This will install dependencies such as:
    - `pandas`
    - `scikit-learn`
    - `seaborn`
    - `matplotlib`
    - `xgboost`
    - `folium`

Run the Code: Use the following command to execute the script and generate results:

```
make run
```

3. This will:
    - Load the dataset
    - Process and visualize the data
    - Train models (Logistic Regression, Random Forest, XGBoost)
    - Fine-tune the Logistic Regression model
    - Generate evaluation metrics and visualizations.

4. **Dependencies**: The main libraries used in the project are listed in the `requirements.txt` file. To install them manually, run:

pip install -r requirements.txt

5. **Dataset**: Ensure the dataset (`flood_risk_dataset_india.csv`) is in the same directory as the code. This dataset contains environmental factors such as rainfall, water levels, humidity, land cover types, and flood occurrences.

Visualizations of Data

To better understand the dataset and relationships between features, the following visualizations were created:

1. **Feature Distributions**:
    - The distributions of `Water Level (m)` and `Rainfall (mm)` were plotted, showing their variation with respect to flood occurrences.
    - These visualizations provide insights into how these features influence flood risk.
2. **Heatmap of Flood Probability**:
    - A heatmap visualizes the combined effect of rainfall and water level bins on flood probability. It highlights critical ranges where flood risk increases.
3. **Flood Occurrence by Land Cover Type**:
    - A bar chart was generated to explore the relationship between land cover types (e.g., urban, forest, desert) and flood occurrences.
4. **Geospatial Visualization**:
    - An interactive map shows the geographical distribution of flood occurrences across South Asia, with clusters identifying high-risk areas.

Description of Data Processing and Modeling

1. **Data Preprocessing**:
    - Categorical features (`Land Cover` and `Soil Type`) were one-hot encoded to convert them into numerical format.
    - To address class imbalance, the dataset was balanced using RandomUnderSampler, ensuring equal representation of flood and no-flood cases.
    - Numerical features were standardized using StandardScaler to ensure all variables were on the same scale.
    - Recursive Feature Elimination (RFE) was applied to select the top 10 most predictive features for the model, reducing dimensionality.

2. **Modeling**:
    - Three machine learning models were trained: Logistic Regression, Random Forest, and XGBoost.
    - Logistic Regression performed best with an accuracy of **50.40%** and a ROC-AUC score of **51.08%**, making it the final model.
    - The Logistic Regression model was fine-tuned using GridSearchCV, optimizing hyperparameters (`C` for regularization strength and solver type). The best configuration achieved an accuracy of **50.27%** and a ROC-AUC score of **51.08%**.
3. **Reason for Choosing Logistic Regression**:
    - Despite the low accuracy, Logistic Regression showed the most consistent performance and was interpretable for this problem.
    - Fine-tuning focused on improving generalization without overfitting, which is critical for datasets with limited predictive signals.

Results

The following results summarize the model performance and insights gained:

1. **Model Evaluation**:
    - Logistic Regression: Accuracy = **50.40%**, ROC-AUC = **51.08%**
    - Random Forest: Accuracy = **48.58%**, ROC-AUC = **48.94%**
    - XGBoost: Accuracy = **50.22%**, ROC-AUC = **49.70%**
2. **Key Observations**:
    - Logistic Regression achieved the highest accuracy and ROC-AUC, making it the final model.
    - The dataset's low accuracy could be attributed to:
        - **Limited Predictive Power**: Environmental features may not fully capture the complexity of flood occurrence.
        - **Data Quality**: The dataset may contain noise or insufficient resolution for certain features.
        - **Model Assumptions**: Logistic Regression assumes linear relationships, which may not entirely align with real-world flood dynamics.
    - Why We Chose Logistic Regression:
        - Logistic Regression was selected because it achieved the highest accuracy (50.40%) and ROC-AUC score (51.08%) compared to Random Forest and XGBoost.
        - Logistic Regression is interpretable and suitable for binary classification tasks like this one. It also allows for easier fine-tuning and analysis of feature importance.

3. **Visualizations**:
    - The heatmap revealed that certain ranges of `Rainfall` and `Water Level` increase flood risk.
    - Geospatial analysis identified specific regions in South Asia prone to flooding, aligning with historical flood patterns.

Why Accuracy Was Low

1. **Dataset Limitations:**
    - The dataset may lack sufficient predictive power. For instance, some features like rainfall or water level alone may not fully explain flood occurrences.
    - Potential noise or inaccuracies in the dataset could be affecting model performance.
    - Missing important environmental factors, such as elevation, river discharge, or land slope, may hinder the model's ability to capture flood patterns.
2. **Linear Assumptions of Logistic Regression:**
    - Logistic Regression assumes a linear relationship between features and the target variable, which may not align with the complex and non-linear nature of flood dynamics.
3. **Small Sample Size:**
    - While undersampling balanced the classes, it also reduced the number of samples available for training, possibly leading to overfitting or underperformance.

Future Improvements

To improve model performance and better understand flood risks:

1. **Enhanced Data Collection**:
    - Include additional features such as elevation, river flow rates, or satellite imagery for better predictive power.
    - Ensure higher resolution and accuracy in collected environmental data.
    - Collect additional features like elevation, river discharge, satellite imagery, or seasonal data to capture more predictive signals.
    - Increase the dataset size or resolution to reduce noise and improve generalization.
    - Collaborate with experts in hydrology or climate science to identify and include key variables affecting flood risk.
2. **Modeling Techniques**:
   Experiment with non-linear models (e.g., Support Vector Machines or Neural Networks) to capture complex interactions between features.
    - Use ensemble methods like Gradient Boosting for improved performance.
3. **Temporal Analysis**:

    - Analyze trends over time by incorporating historical flood data and using time-series forecasting techniques.
4. **Explore Non-Linear Models:**
    - Use algorithms like Support Vector Machines, Gradient Boosting, or Neural Networks to model non-linear relationships between features and flood occurrences.
5. **Temporal Analysis:**
    - Incorporate time-series data to analyze trends and seasonal variations in floods, which could improve predictions.
6. **Feature Engineering:**
    - Create interaction terms (e.g., `Rainfall × Water Level`) or polynomial features to capture more complex relationships.

Conclusion

This project demonstrated the application of machine learning to predict flood risks in South Asia. Despite low accuracy, the findings highlight the potential for environmental data to inform flood prediction models. The interactive visualizations provide valuable insights for policymakers and researchers, and the framework can be further improved with additional data and modeling techniques.