Planning logic

Date	10 February 2026
Team ID	LTVIP2026TMIDS90283
Project Name	rising waters: a machine learning approach to flood prediction

Data Collection and Data Cleaning

Flood Prediction Using Machine Learning

Data Collection

Data collection is a crucial step in building a reliable machine learning model. For this project, a flood-related dataset containing environmental and rainfall parameters was collected. The dataset includes features that significantly influence flood occurrence.

The primary input features used in this project are:

Cloud Cover

Annual Rainfall

Jan–Feb Rainfall

March–May Rainfall

June–September Rainfall

The target variable indicates flood occurrence:

1 → Possibility of Severe Flood

0 → No Possibility of Severe Flood

The dataset was stored in CSV format and imported into the Python environment using the Pandas library. Initial exploration of the dataset was performed using functions such as `head()`, `info()`, and `describe()` to understand the structure, data types, and statistical distribution of the features.

Proper understanding of the dataset ensured that relevant features were selected for model training and unnecessary attributes were removed.

Data Cleaning and Preparation

Planning logic

Raw data is often inconsistent or unstructured; therefore, data cleaning is essential before model training.

The following preprocessing steps were performed:

1. Handling Missing Values
The dataset was checked for null or missing values using appropriate Pandas functions. If missing values were present, numerical values were filled using mean or median methods to maintain data consistency.

2. Data Type Verification
All feature columns were examined to ensure correct data types (numerical format). This step ensures compatibility with machine learning algorithms.

3. Feature Selection
Only relevant environmental and rainfall features influencing flood prediction were selected. Irrelevant or redundant attributes were excluded to improve model efficiency.

4. Feature Scaling
Since machine learning models perform better when features are on a similar scale, StandardScaler was applied to normalize the input data. Scaling helps improve model accuracy and ensures better convergence during training.

5. Train–Test Splitting
The dataset was divided into training and testing sets using the `train_test_split()` function. This ensures that the model is trained on one portion of the data and evaluated on unseen data to measure performance accurately.

Conclusion

Effective data collection and cleaning ensured that the dataset was accurate, consistent, and suitable for machine learning model training. Proper preprocessing significantly improved prediction reliability and strengthened the overall performance of the Flood Prediction System.