Rising Waters: Flood Prediction Using Machine Learning

SMARTBRIDGE INTERNSHIP PROJECT REPORT

Project Title: Flood Risk Prediction Using Machine Learning

Technology: Python, Flask, HTML, CSS

Team Id: LTVIP2026TMIDS37461

Student Name: M.Guruchaithanya kumari

Roll Number: 22F11A05G0

College: Narayana Engineering College, Gudur

Year: 2026

TABLE OF CONTENTS

1. ABSTRACT

Floods are one of the most dangerous natural disasters that cause severe damage to human life, agriculture, and infrastructure. Early prediction of floods helps authorities take preventive measures and reduce losses. This project focuses on predicting flood occurrence using Machine Learning techniques.

The system analyzes environmental parameters such as temperature, humidity, cloud cover, rainfall distribution, and seasonal weather values. A Random Forest classification model is trained using historical flood data to determine whether flood risk exists or not.

A Flask-based web application is developed to allow users to enter weather parameters and obtain flood prediction results instantly. The application displays either **High Flood Risk** or **No Flood Risk** based on the trained model output.

This project demonstrates how machine learning can be applied to real-world disaster prediction problems.

2. INTRODUCTION

Flood prediction is a crucial task in disaster management systems. Traditional flood monitoring methods rely on manual observation and delayed reporting, which often leads to late warnings.

With the advancement of Machine Learning, it is possible to analyze large amounts of historical weather data and identify patterns that indicate flood risk.

This project builds a predictive model that:

- Learns from past flood data
- Identifies relationships between weather conditions
- Predicts flood chances automatically

A web interface is provided so that users can easily input data and view predictions.

3. PROBLEM STATEMENT

Rising Waters: A Machine Learning Approach to Flood Prediction Flood Prediction using Machine Learning is a vital application that aims to forecast and predict flood occurrences with high accuracy. By analyzing historical weather data, river levels, terrain information, and other relevant factors using machine learning algorithms, this project helps in early warning and mitigation of potential flood events. The goal is to provide timely alerts and actionable insights to authorities, communities, and individuals to minimize the impact of floods on lives and infrastructure.

Floods cause:

- Loss of human lives
- Damage to property
- Crop destruction
- Economic instability

Existing systems often:

- Provide late warnings

- Require manual calculations
- Lack automated prediction tools

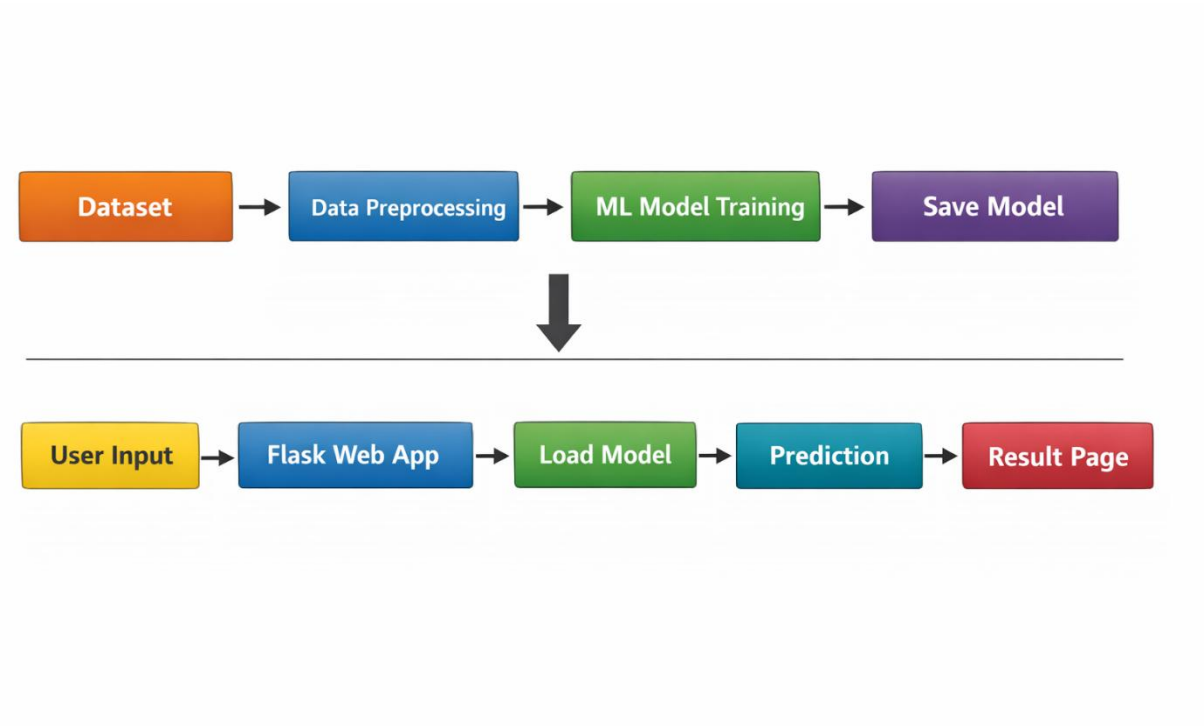
Therefore, there is a need for an automated flood prediction system using machine learning.

4. OBJECTIVES

Main objectives of this project:

- Build a machine learning model for flood prediction
- Train the model using historical dataset
- Create a user-friendly web application
- Provide instant flood risk results
- Help in disaster preparedness

5. SYSTEM ARCHITECTURE



6. TECHNOLOGIES USED

1. Python

Python is the main programming language used to develop the flood prediction system. It is easy to use and supports many libraries for machine learning and web development.

2. Pandas

Pandas library is used for loading, reading, and processing the dataset. It helps in handling tabular data efficiently.

3. NumPy

NumPy is used for numerical operations and converting user input values into arrays for model prediction.

4. Scikit-learn

Scikit-learn is used to build and train the machine learning model. The Random Forest classifier from this library is used for flood prediction.

5. Pickle

Pickle module is used to save the trained machine learning model into a file so it can be reused later in the Flask application.

6. Flask

Flask is a lightweight web framework used to create the web application interface. It connects the trained model with HTML pages and processes user input.

7. HTML & CSS

HTML is used to design the structure of the web pages, while CSS is used to style the pages and improve the visual appearance.

8. VS Code

Visual Studio Code is used as the development environment for writing and running Python and Flask code.

7. DATASET DESCRIPTION

The dataset contains environmental parameters:

- Temperature
- Humidity
- Cloud Cover
- Annual rainfall
- Seasonal rainfall distribution
- Average June rainfall
- Sub-division rainfall

Target column:

- Flood (0 = No Flood, 1 = Flood)

This dataset is used to train the machine learning model.

8. DATA PREPROCESSING

Steps performed:

1. Dataset loaded using pandas
2. Features separated from target column
3. Training and testing data split
4. Missing values handled if present

9. MACHINE LEARNING MODEL

Random Forest Classifier is used.

Why Random Forest?

- High accuracy

- Works well with multiple features
- Handles large datasets
- Prevents overfitting

The model learns patterns from historical data and predicts flood probability.

10. MODEL TRAINING PROCESS

Steps:

1. Load dataset
2. Separate X and Y
3. Split into training/testing
4. Train RandomForestClassifier
5. Save model using pickle

Saved file:

model.pkl

11. WEB APPLICATION DEVELOPMENT

A Flask application is created.

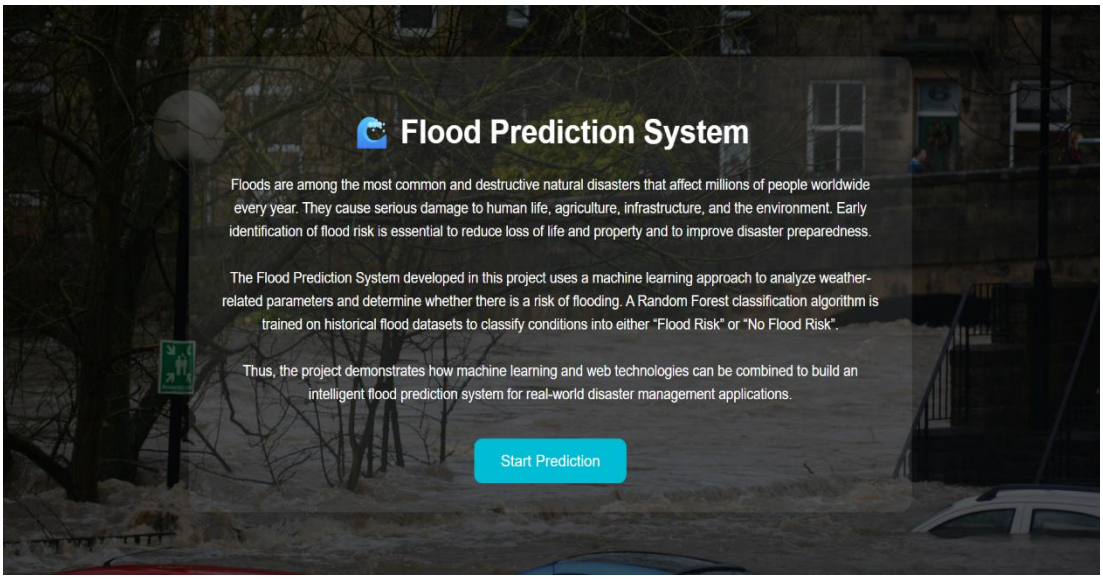
The web system includes:

- Home Page
- Input Page
- Flood Risk Page
- Safe Page

12. WORKING OF WEB APPLICATION

1. User opens home page

The user first accesses the flood prediction web application through the browser. The home page provides a brief introduction about the system and an option to start the prediction process.



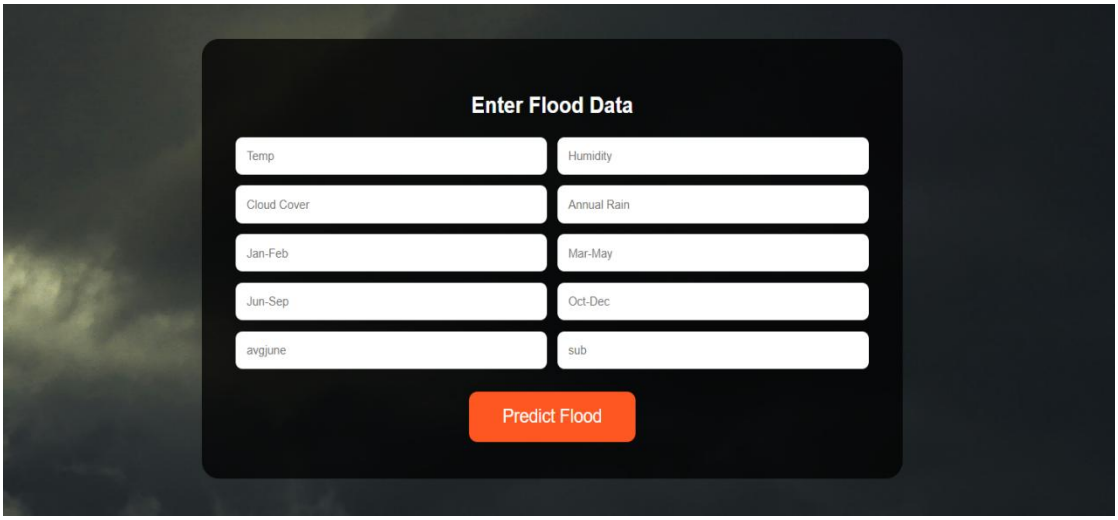
Home Page of Flood Prediction System

2. Clicks prediction option

The user clicks the prediction button or link available on the home page. This redirects the user to the input form page where weather parameters can be entered.

3. Enters weather values

The user enters required environmental values such as temperature, humidity, rainfall distribution, cloud cover, and other seasonal data into the input fields provided on the form.



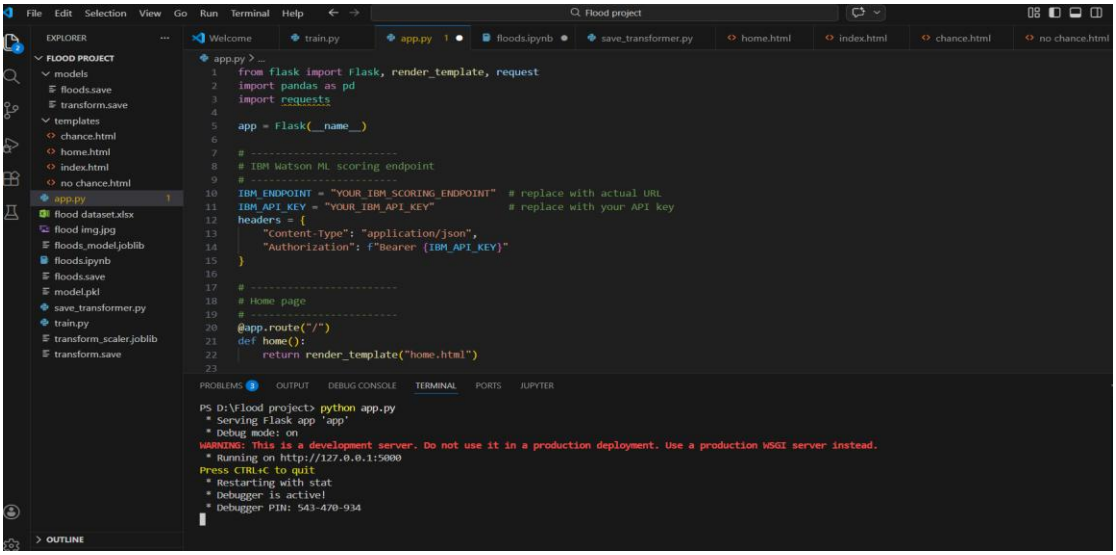
User Input Page

4. Flask sends data to model

After submitting the form, the Flask backend collects the entered values, converts them into numerical format, and sends them to the trained machine learning model for analysis.

5. Model predicts result

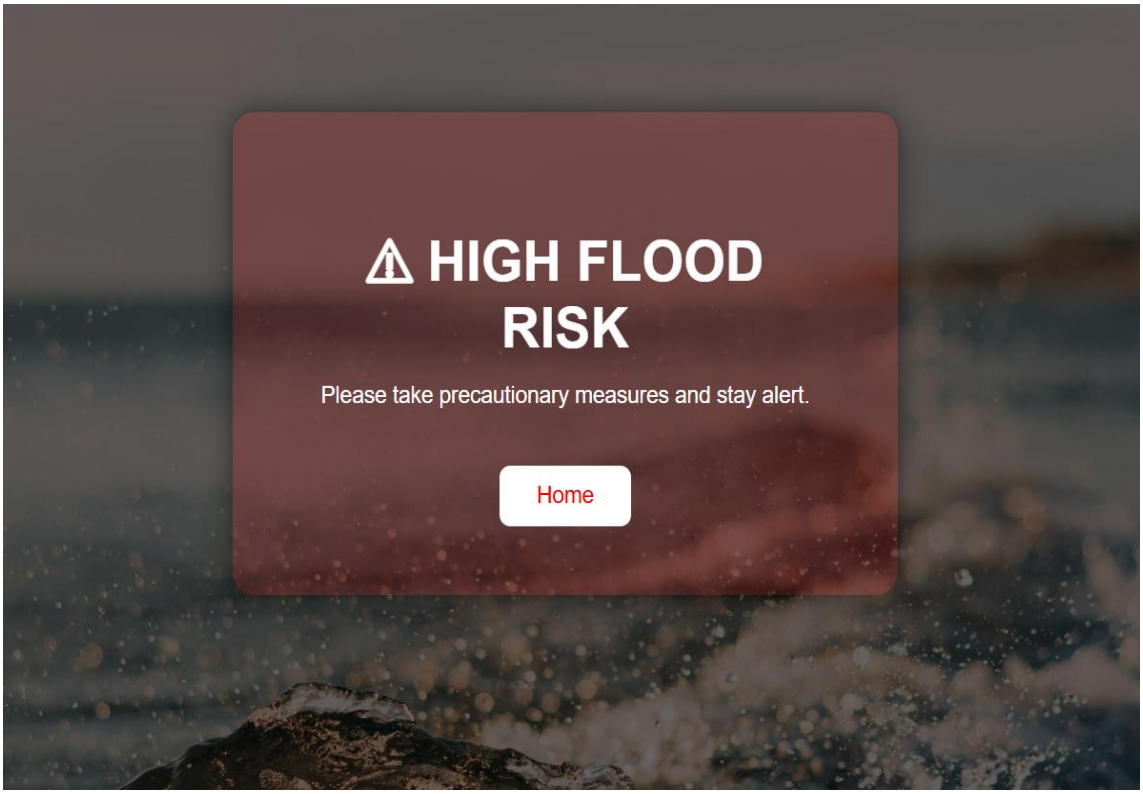
The trained Random Forest model processes the input values and compares them with patterns learned from historical flood data to determine whether there is a flood risk.



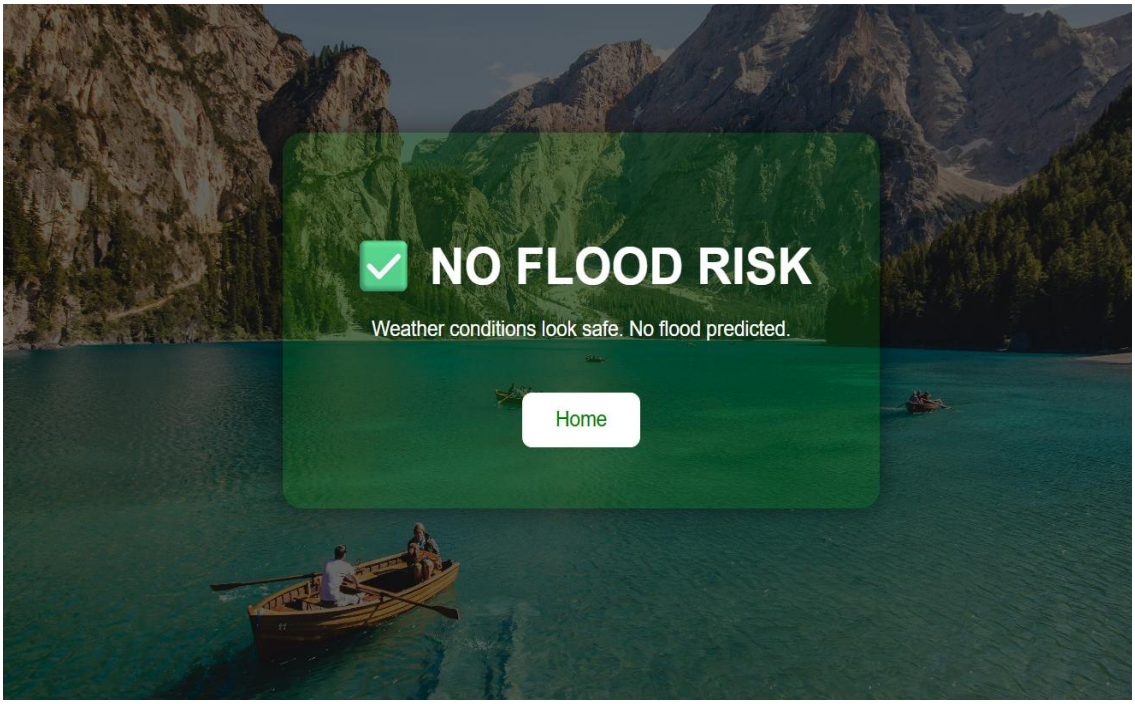
Project Execution in VS Code

6. **Result page displayed**

   Based on the prediction, the system displays the appropriate result page showing either **High Flood Risk** or **No Flood Risk**, allowing the user to understand the situation instantly.



High Flood Risk Prediction Output



No Flood Risk Prediction Output

14. ADVANTAGES

- Fast prediction
- Easy to use
- Reduces disaster risk
- Supports planning
- Fully automated

15. APPLICATIONS

- Disaster management departments
- Weather monitoring agencies
- Government planning
- Agricultural protection
- Urban development

16. FUTURE ENHANCEMENTS

Future improvements:

- Real-time weather API integration
- Mobile application version
- Map-based flood visualization
- Deep learning model
- SMS alert system

17. CONCLUSION

This project successfully demonstrates a flood prediction system using machine learning and Flask web technology. The trained Random Forest model can analyze weather parameters and predict flood risk effectively.

The system can help authorities and communities take early preventive measures and reduce damage caused by floods.

This Machine Learning based Flood Prediction System helps to forecast possible flood situations using environmental data such as temperature, humidity, rainfall and seasonal values.

The system analyses the given inputs and predicts whether there is a possibility of flood occurrence or not.

This project helps authorities and communities to take early precautionary measures and reduce damage caused by floods.