FLOOD INUNDATION PROBABILITY MAPPING

SUBMITTED IN PARTIAL FULFILLMENT FOR THE REQUIREMENT OF THE AWARD OF DEGREE OF

**BACHELOR OF TECHNOLOGY IN**

**COMPUTER SCIENCE**

****

Submitted by

#### ANUBHAV KUMAR – 2100290120035 RITIKA SHARMA – 2200290129016 PRAJJWAL DWIVEDI – 2200290129013 ADARSH CHAUDHARY - 2200290129001

.

Supervised by

#### **DR. ANURAG MISHRA**

(Assistant Professor)

**Session 2024-25**

**DEPARTMENT OF COMPUTER SCIENCE**

# KIET GROUP OF INSTITUTIONS, GHAZIABAD

**(Affiliated to Dr. A. P. J. Abdul Kalam Technical University, Lucknow, U.P., India) May 2025**

**DECLARATION**

I/We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature: Signature:

Name: Anubhav Kumar Name: Ritik Sharma

Roll no: 2100290120035 Roll no: 2200290129016

Signature: Signature:

Name: Prajjwal Dwivedi Name: Adarsh Chaudhary

Roll no: 2200290129013 Roll no: 2200290129001

Date-

**CERTIFICATE**

This is to certify that Project Report entitled “Flood Inundation Probability Mapping” which is submitted by Anubhav Kumar, Ritik Shama, Prajjwal Dwivedi, Adarsh Chaudhary in partial fulfillment of the requirement for the award of degree B. Tech. in Department of Computer Science of Dr. A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidates own work carried out by them under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

**Date: Supervisor**

Dr. Anurag Mishra  
 Assistant Professor

# ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe special debt of gratitude to Dr. Anurag Mishra, Department of Computer Science, KIET, Ghaziabad, for his/her constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of Dr. Ajay Kr. Shrivastava, Dean of the Department of Computer Science, KIET, Ghaziabad, for his full support and assistance during the development of the project. We also do not like to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project.

Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Date :

Signature: Signature:

Name: Anubhav Kumar Name: Ritik Sharma

Roll no: 2100290120035 Roll no: 2200290129016

Signature: Signature:

Name: Prajjwal Dwivedi Name: Adarsh Chaudhary

Roll no: 2200290129013 Roll no: 2200290129001

**ABSTRACT**

Floods are among the most significant natural hazards globally, causing thousands of deaths and severe disruptions to economies and communities each year. Their increasing frequency is linked to factors like climate change, rapid urban development, and deforestation. Traditional prediction models often fall short due to their inability to adapt to evolving environmental patterns, leading to the rising adoption of machine learning (ML) techniques for more accurate forecasting. This project combines geospatial data science, ML, and web development to provide a real- time flood forecasting system. Utilizing datasets ranging from monsoon intensity, drainage quality, urbanization, and topography, ML algorithms such as logistic regression, XGBoost, and neural networks are trained to evaluate flood susceptibility. The system is focused on interpretability and scalability, with a comparison of model performance to determine best methods for varying datasets.

An easy-to-use web application allows non-technical stakeholders (e.g., policymakers, residents) to engage with the ML model, choosing locations through an interactive map to obtain flood probability scores. Geospatial APIs retrieve real-time meteorological data, and the backend hosts models through cloud-based APIs for low-latency predictions.

Through the integration of actionable analytics and easy-to-understand visualization, this project increases disaster preparedness and community resilience. It fills the gap between data science and disaster management by providing a scalable solution to minimize flood-related vulnerabilities and enable proactive decision-making.

|  |  |
| --- | --- |
| **TABLE OF CONTENTS** | **Page No.** |
| DECLARATION…………………………………………………………………… | ii |
| CERTIFICATE……………………………………………………………………… | iii |
| ACKNOWLEDGEMENTS…………………………………………………………. | iv |
| ABSTRACT……………………………………………………………………….... | v |
| LIST OF FIGURES…………………………………………………………………. | viii |
| LIST OF TABLES…………………………………………………………………… | x |
| LIST OF ABBREVIATIONS………………………………………………………. | xi |
| SDG MAPPING WITH JUSTIFICATION ………………………………………. | xii |
| CHAPTER 1 INTRODUCTION…………………………………………………… | 15-17 |
| 1.1 Introduction to Project | 15 |
| 1.2 Project Category | 16 |
| 1.3 Objectives | 17 |
|  |  |
| CHAPTER 2 LITERATURE REVIEW……………………………………………… | 18-20 |
| 2.1 Literature Review | 18 |
| 2.2 Research Gaps | 19 |
| 2.3 Problem Formulation | 20 |
| CHAPTER 3 PROPOSED SYSTEM………………………………………………… | 21-22 |
| 3.1 Proposed System | 21 |
| 3.2 Unique Features of The System | 22 |
| CHAPTER 4 REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION…… | 23-34 |
| 4.1 Feasibility Study (Technical, Economical, Operational) | 23 |
| 4.2 Software Requirement Specification | 24 |
| 4.2.1 Data Requirement | 24 |
| 4.2.2 Functional Requirement | 24 |
| 4.2.3 Performance Requirement | 25 |
| 4.2.4 Maintainability Requirement | 25 |
| 4.2.5 Security Requirement | 25 |
| 4.3 SDLC Model Used | 26 |
| 4.4 System Design | 27 |
| 4.4.1 Data Flow Diagram | 30 |
| 4.4.2 Use Case Diagram | 33 |
| 4.5 Database Design | 34 |
|  |  |
| CHAPTER 5 IMPLEMENTATION………………………………… | 35-38 |
| 5.1 Introductions tools & technologies used | 35 |
| 5.2 Dataset Description | 37 |
|  |  |
| CHAPTER 6 TESTING & MAINTENANCE………………………. | 39-41 |
| 6.1 Testing Techniques & Test Cases Used | 39 |
|  |  |
| CHAPTER 7 RESULTS AND DISCUSSIONS…………………….. | 42-56 |
| 7.1 Description of Modules with snapshot | 42 |
| 7.2 Presentation of Results | 49 |
| 7.3 Performance Evaluation | 54 |
| 7.4 Key Findings | 56 |
|  |  |
| CHAPTER 8 CONCLUSION AND FUTURE SCOPE…………… | 57-59 |
| REFERENCES……………………………………………………... | 60-63 |
| Turnitin Plagiarism Report………………………………………… | 64-66 |
| Research Paper Status……………………………………………… | 67 |
| Proof of paper publication…………………………………………. | 68 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure** | **Description** | **Page No.** |
| 4.3(a) | Agile SDLC | 27 |
| 4.4(a) | System design | 29 |
| 4.4.1(a) | Level-0 DFD | 30 |
| 4.4.1(b) | Level-1 DFD | 31 |
| 4.4.1(c) | Level-2 DFD | 32 |
| 4.4.2(a) | Use case diagram | 33 |
| 5.1(a) | Overview of tech used | 36 |
| 5.2(a) | Dataset description | 36 |
| 5.2(b) | Dataset description | 37 |
| 7.1(a) | Snapshot for data preprocessing module | 42 |
| 7.1(b) | Model training using logistic regression | 43 |
| 7.1(c) | Model training using neural networks | 43 |
| 7.1(d) | Model training using xgboost | 43 |
| 7.1(e) | Model deployment on huggingface using gradio | 45 |
| 7.1(f) | Frontend of website snapshots | 48 |
| 7.2(a) | Feature importance logistic regression | 51 |

|  |  |  |
| --- | --- | --- |
| 7.2(b) | Feature importance neural networks | 52 |
| 7.2(c) | Train-loss curves for neural networks model | 52 |
| 7.2(d) | Results on user end | 53 |
| 7.3(a) | Time analysis for logistic regression | 54 |
| 7.3(b) | Time analysis for neural networks | 54 |
| 7.3(c) | Comparison of both models using evaluation matrices | 55 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Description** | **Page No**. |
| 4.5(a) | Database description | 34 |
| 6.1(a) | Testcases performed | 41 |
| 7.2(a) | Evaluation matrix for logistic regression | 49 |
| 7.2(b) | Evaluation matrix for neural networks | 50 |
| 7.2(c) | Confusion matrix for logistic regression | 50 |
| 7.2(d) | Confusion matrix for neural networks | 51 |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| ML | Machine Learning |
| DFD | Data flow diagram |
| ROI | Return on investment |
| CAGR | Compound Annual Growth Rate |
| API | Application Programming interface |
| AUC-ROC | Area under the receiver operating characteristic curve |
| MAE | Mean Absolute Error |
|  |  |
| Xgboost | Extreme Gradient Boosting |
|  |  |

**SDG MAPPING WITH JUSTIFICATION**

**SDG 1:** No Poverty: Floods disproportionately affect low-income communities, exacerbating poverty through displacement, loss of livelihoods, and economic disruption. By enabling early flood prediction, the project helps vulnerable populations prepare for disasters, reducing financial losses and preventing poverty escalation. Target 1.5 explicitly mentions building resilience to climate-related disasters .

**SDG 2:** Zero Hunger: Floods destroy crops, disrupt food supply chains, and undermine food security. Predictive analytics allow farmers to avoid flood impacts on agriculture, in accordance with Target 2.4 (sustainable food production systems resilient to extreme weather).

**SDG 3:** Good Health and Well-Being: Floods cause injuries, waterborne diseases, and mental illness. Early warnings enable health systems to prepare for heightened demand and protect vulnerable populations, supporting Target 3.d (health system resilience).

**SDG 6:** Clean Water and Sanitation: Floods contaminate water sources and damage sanitation infrastructure. The project aids in safeguarding water infrastructure, ensuring access to clean water (Target 6.1) and preventing sanitation disruptions .

**SDG 9:** Industry, Innovation, and Infrastructure: The project leverages ML, geospatial data, and web development to create a novel flood prediction tool. It advances innovation Target 9.5 and builds resilient infrastructure Target 9.1 by providing actionable insights for disaster-proof planning .

**SDG 11:** Sustainable Cities and Communities: Floods threaten urban safety and sustainability. The web application supports Target 11.5 (reduce disaster-related deaths) and Target 11.b (integrated disaster risk policies), helping cities become inclusive and resilient .

**SDG 13:** Climate Action Floods are climate change-induced disasters. The project strengthens resilience to climate hazards (Target 13.1), aligning with global efforts to mitigate climate impacts .

**SDG 15:** Life on Land: Floods degrade terrestrial ecosystems and biodiversity. Predictions help protect ecosystems (Target 15.1) by enabling conservation efforts in flood-prone areas .

**SDG 17:** Partnerships for the Goals: The project requires collaboration between governments, NGOs, and tech sectors. Partnerships (Target 17.17) are critical for deploying the system and sharing data, ensuring inclusive disaster management .



# SDG MAPPING

**CHAPTER 1**

**INTRODUCTION**

1. **Introduction**

Flooding is the foremost visit and obliterating normal catastrophe around the world, influencing millions of lives and causing gigantic financial misfortunes each year. Between 6,000 and 20,000 fatalities are detailed yearly due to floods, whereas incalculable others confront relocation, framework harm, and disturbances to jobs. The expanding recurrence and seriousness of surges are closely connected to climate alter, urbanization, and deforestation, which worsen helplessness in both urban and country zones. As a result, anticipating and mitigating the affect of floods has gotten to be a basic need for governments, communities, and catastrophe administration agencies. Traditional flood expectation strategies depend on hydrological and meteorological information, but they regularly drop short due to the dynamic nature of flooding and the expanding complexity of natural variables. Machine learning (ML), with its capacity to analyze tremendous datasets and reveal hidden patterns, offers a transformative approach to flood forecast. ML models can integrate different inputs, including precipitation, topography, land utilize, and infrastructure conditions, to produce precise and timely predictions. This project centers on leveraging machine learning models like logistic regression, gradient boosting. Utilizing two datasets, this model incorporates different features such as monsoon intensity, drainage quality, urbanization levels, and more , to evaluate flood probabilities successfully. Furthermore, a user-friendly web application is proposed to empower real-time flood chance evaluation based on geological information fetched from APIs. By giving noteworthy bits of knowledge, this work points to back disaster readiness, minimize misfortunes and improve community versatility to flood dangers. Overall, this project presents a novel approach to predict flood occurrence with utilizing web applications that can significantly improve the efficiency and accuracy of the disaster management.

* 1. **Project Category**

This project falls under the intersection of geospatial data science, web application development, and machine learning deployment. Key categories include:

Geospatial Analysis: Processing and interpreting spatial datasets (e.g., elevation, rainfall, soil type) to model flood susceptibility.

Machine Learning: Training different types of models and understanding their performance, modes which were used for this project are logistic regression, XGBoost, and even neural networks were used to model complex datasets.

Web Development: Building a user-friendly interface for real-time interaction with ML models.

Disaster Risk Reduction: Providing actionable insights to mitigate flood impacts through predictive analytics.

* 1. **Objectives**

The primary objectives of this project are:

Model Development: Train models like logistic regression, XGBoost , neural networks to predict flood inundation probabilities and then Compare model performance to identify optimal approaches for different datasets.

Model Deployment: The trained machine learning model is then deployed for seamless real-time predictions.

Web Application Deployment: Develop a good website for users to interact with the machine learning model, user can chose a location on map and get prediction for flood depending upon the location and features.

User Accessibility: Ensure non-technical users (e.g., policymakers, residents) can easily interpret flood risk results through visual maps and probability scores.

**CHAPTER 2**

**LITERATURE REVIEW**

**2.1 Literature Review**

Recent advancements in machine learning (ML) have revolutionized flood prediction

1. Machine Learning for Flood Prediction

Mosavi et al. (2018) [1] conducted a detailed review exploring the role of machine learning in flood forecasting across various temporal scales. Their work emphasized that ML techniques outperform conventional hydrological models in terms of adaptability, accuracy, and ease of scaling to diverse geographic contexts. Lawal et al. (2021) [2] applied ML models to over 30 years of rainfall data to predict flood risks in Kebbi State, Nigeria. Their findings showed that machine learning could effectively learn from historical meteorological patterns, enabling accurate region-specific flood forecasts. Rajab et al. (2023) [6] developed a deep learning-based flood prediction model using historical climatic data from Bangladesh. Their work highlighted the ability of neural networks to understand intricate environmental trends, especially in handling sequential rainfall data.

1. Algorithmic Innovations in Susceptibility Mapping

Nguyen et al. (2023) [3] compared the performance of K-Nearest Neighbors (KNN), Support Vector Machines (SVM), and Artificial Neural Networks (ANN) for flood susceptibility mapping in Vietnam. ANN emerged as the most reliable due to its capacity to model non-linear relationships among input features. Khalid & Khan (2024) [7] reinforced ANN's effectiveness by conducting rigorous hyperparameter tuning to improve model generalization and accuracy across varying flood-prone terrains in Ontario, Canada. Ahmadlou et al. (2024) [8] proposed a hybrid deep learning model combining a multilayer perceptron (MLP) with autoencoder networks. Their framework improved flood risk map accuracy by extracting both spatial and temporal characteristics from the data.

1. Data Integration and Open-Source Solutions

Hitouri et al. (2024) [4] integrated Synthetic Aperture Radar (SAR) imagery with ML algorithms to monitor flood inundation in areas with limited on-the-ground data. This fusion allowed for real-time flood detection in smaller watersheds, enhancing accuracy in resource-constrained environments. Bhattarai et al. (2024) [5] leveraged freely available geospatial data—such as satellite imagery and elevation models—combined with ANN, SVM, and LSTM models to assess flood risk in Laos. Their study highlighted the potential of open-source resources for cost-effective flood mapping in developing regions.

1. Performance Validation and Gaps

Hasanuzzaman et al. (2022) [9] benchmarked multiple models, including logistic regression, random forests, and ANN, to assess their effectiveness in flood susceptibility analysis. Their study showed that while logistic regression offered better interpretability, ANN achieved higher overall predictive accuracy. Despite the growing adoption of ML in flood prediction, most research remains limited to academic settings without real-world implementation. Few studies emphasize user-facing deployment or integration with live meteorological APIs. This project addresses such limitations by deploying interpretable models like logistic regression and ANN within an interactive web application, enabling real-time flood prediction and public accessibility.

**2.2 Research Gaps**

(i) Real-Time Accessibility:

Most ML-based flood prediction systems are designed as proof – of - concept models and lack mechanisms for real-time interaction, limiting their usefulness in emergency scenarios.

(ii) Cross-Disciplinary Integration:

Flood prediction typically requires combining geospatial analysis, meteorological data, and machine learning. However, current approaches often treat these components separately. Systems that fuse these elements into a single, cohesive framework remain underdeveloped.

(iii) Scalability and Open Access:

Many models are resource-intensive and rely on proprietary software or high-performance computing setups, which are not feasible in all regions.There is a clear need for systems built on open-source tools that can run efficiently even in low-resource environments

* 1. **Problem Formulation**

To address these gaps, this project proposes the development of a unified, end-to-end flood prediction system that integrates geospatial analysis, machine learning, and web development. The system is designed to operate in real time, offering predictions through a web-based interface that retrieves live weather and environmental data. ML models such as logistic regression, XGBoost, and neural networks are used to generate flood risk scores based on parameters like rainfall intensity, temperature, elevation, and urban development. By presenting these results through an interactive map, the system enables users—including policymakers and local communities—to assess flood risk and take preventive measures in advance.

**CHAPTER 3**

**PROPOSED SYSTEM**

**3.1 Proposed System**

The proposed system is an end-to-end flood inundation probability mapping platform that integrates geospatial data analytics, machine learning (ML), and web- based visualization to deliver real-time flood risk assessments. The system architecture comprises four core modules:

Data Ingestion & Preprocessing:

Datasets was chosen from multiple authenticated resources such as Kaggle, github repositories. In this module data collection and finding what features are important and what are not was focused.

Machine Learning Engine:

In this module, the aim was to create multiple models on collected datasets, different models were created to analyze the performance of the model. Models were created using libraries such as Scikit-Learn and TensorFlow for neural networks. These libraries provide better efficiency.

Machine Learning Model Deployment:

The trained machine learning model is deployed to huggingface spaces using gradio so that easy and efficient inference server is established between website and the machine learning model.

Web Interface & Deployment:

A React.js frontend allows users to select locations via interactive maps or coordinates. A Flask backend processes requests, invokes ML models, and returns predictions as probability percentages, good visualisation of features which are being sent to the ML model.

**3.2 Unique features of the system**

Scalable Open-Source Framework: Built on open-source tools (Python, React.js, Scikit-Learn, TensorFlow, HugginFace) to ensure cost-effective adoption in resource- constrained regions.

Secure: Only authentic users can predict the flood , that makes the system secure from unwanted problems that may arise.

Interoperability: Supports integration with APIs which help todo prediction for live data.

Model accessibility: The machine learning models are deployed on huggingface so anyone can use our work and make it more efficient.

**CHAPTER 4**

**REQUIREMENT ANALYSIS**

**AND**

**SYSTEM SPECIFICATION**

* 1. **Feasibility Study**
* Technical Feasibility:

-Tools & Framework: The system leverages proven open-source technologies like python, flask, sickie-learn, tensorflow, mongodb, huggingface and many more which have a robust community support.

-ML Integeration: Existing libraries like scikit-learn, pytorch simplify model training and deployment.

* Economic Feasibility:

-Low-cost-infrastructure: Open-source tools eliminate licensing fees which makes it a very low cost and also due to advance open-source tools like scikit-learn, numpy, python, pytorch, it has become a lot more cost effective.

-ROI: Proactive flood predictions reduce disaster recovery costs for governments and communities.

* Operational Feasibility:

-User Training: Intuitive web interface requires minimal technical expertise.

-Disaster Response: Aligns with UN SDG 11 (Sustainable Cities) by enabling data-driven risk mitigation.

* Market Feasibility:

-Demand: Rising flood frequency due to climate change drives demand for predictive tools.

-Target Audience: Government agencies , private firms dealing in property dealing, travels, NGOs for disaster preparedness, Insurance firms for risk assessment and many more.

-Competitive Edge: Real-time predictions vs. static reports from traditional hydrological models. Cost-effective compared to proprietary solutions e.g., Delft-FEWS.

Market Trends: Global flood analytics market projected to grow at 8.5% CAGR (2023–2030).

* 1. **Software Requirement Specification:**

**4.2.1 Data Requirements:**

* Model training: The data is required for the model training, the needs to have features like rainfall, max\_temperature, min\_temperature, lattitue, longitudes, wind speed etc.
* User Input: When user selects the location we need to get data for that coordinates , the data corresponds to the features of our deployed machine learning model.
* User information: User needs to provide information for authorization on the web application

**4.2.2 Functional Requirements:**

* User Interface(Frontend): Login or create the new account, select the location on map and then chose the dates ‘from’ and ‘to’, along with what type of model they want to use. The result and feaures data along with good visualisation of data is shown at user side .
* Backend Processing: The backend will process the inputs from frontend, it could be either to do user authorization or to get predictions, it returns predictions in json format.
* Machine learning Inference server: This helps in communicating with the web application and send the predictions to the frontend through backend.

**4.2.3 Performance Requirement:**

* Speed: Flood prediction results generated within 5 seconds of user input. Map rendering (heatmaps) loads in <3 seconds on a standard internet connection.
* Resource Usage: Works on machines with 4GB RAM and dual-core processors (no GPU required). ML models optimized for low memory usage (<500MB per inference).
* Concurrency: should Supports 5–10 simultaneous users during testing.

**4.2.4 Maintainability Requirement:**

* Code Quality: Modular Python/JavaScript code with comments for key functions. Git version control for tracking changes.
* Updates: Easy model retraining via Jupyter Notebook scripts.Dependency management with requirements.txt(py ), json .
* Troubleshooting: Basic error logs to troubeshoot.
* Documentation: A simple README file with setup instructions and screenshots.

**4.2.5 Security Requirements:**

* No Personal Data: The system does NOT collect or store user names, emails, or locations. Users stay anonymous.
* Basic Input Checks: Ensure users can’t break the system by typing weird symbols (e.g., !@#) in search boxes. Example: Only allow numbers and commas for coordinates like 12.34, 56.78.
* Safe Libraries: Use well-known Python/JS libraries (e.g., TensorFlow, React.js) and avoid outdated/unofficial tools.Local Testing = Safe Testing: If the app runs only on your computer (localhost), security is already okay. No need for HTTPS. Backup Your Work: Save copies of your code and datasets on Google Drive or a USB (in case your laptop crashes). Or use github.

**4.3 SDLC Model Used**

This model used Agile methodology, our project involves iterative and incremental nature that’s why Agile software development life cycle is best fit. Here’s a breakdown of why the agile is good fit for the project:

-Incremental Development: The project is likely to be built with evolving features such as user authentication, prediction page, weather data integration and flood predictions, the agile model allows for releasing small, working modules incrementally, which we can refine as we go.

-Flexibility: Since this project involves various eternal API’s like openmeteo for weather data and huggingface gradio applications for prediction, requirements may change as these services evolve. Agile’s adaptability is perfect for handling changes in external systems.

-Continuous feedback: We can continuously improve the system from the feedbacks.

-Collaboration: Agile promotes collaboration among all team members, ensuring that everyone stays aligned on objectives

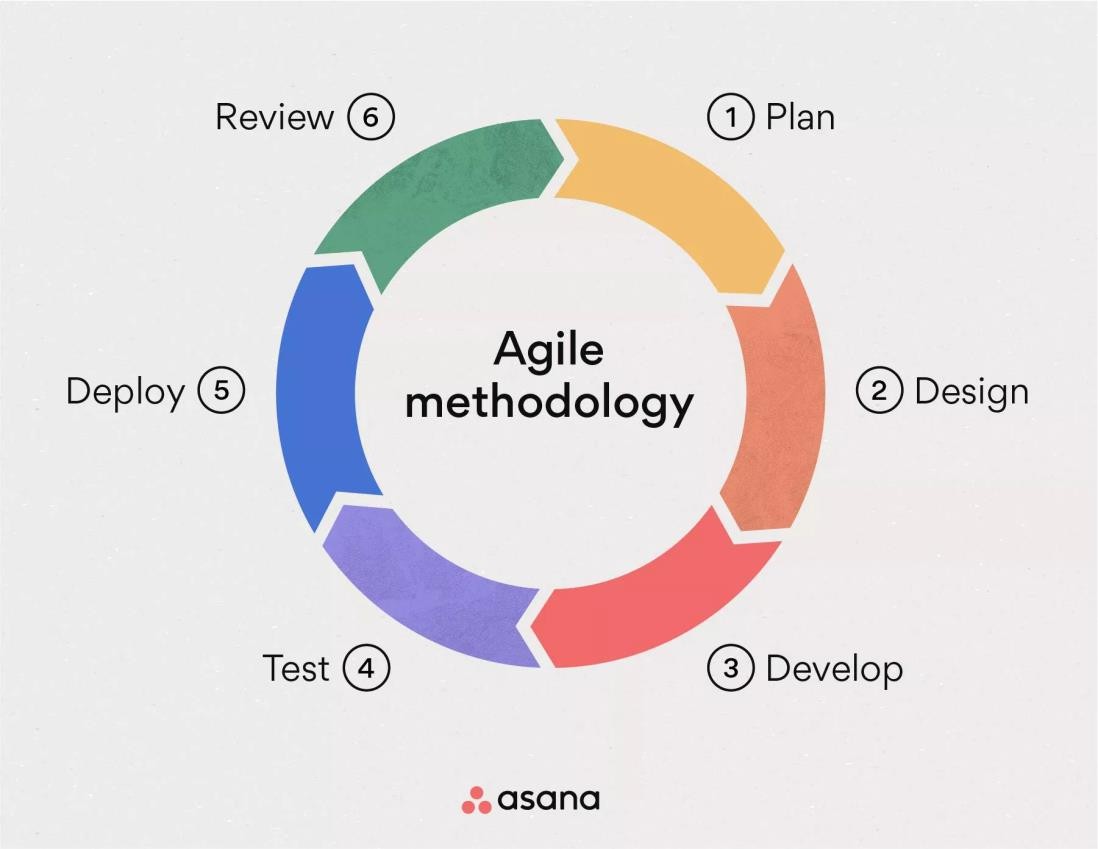


Fig: 4.3(a) – Agile SDLC

**4.4 System Design**

The system design of the project is simple , it involves four main functions which are defines below, also its been define in fig-4.4(a):

- User Interaction: The user logs in or signs up via the frontend (React). Upon successful login, they navigate to the Prediction Page.

-Weather Data Fetching: The user selects a location and date range on the frontend. The frontend sends these details to the backend (Express). The backend calls the OpenMeteo API to fetch the weather data and then sends this data to the Flask API for flood prediction.

-Prediction: The Flask API processes the weather data and sends it to the Hugging Face model. The Hugging Face model returns the flood prediction to the Flask API, which then sends it back to the backend. The backend sends the prediction result to the frontend to be displayed.

- User Management: The backend handles sign-up, login, and stores user credentials in MongoDB.

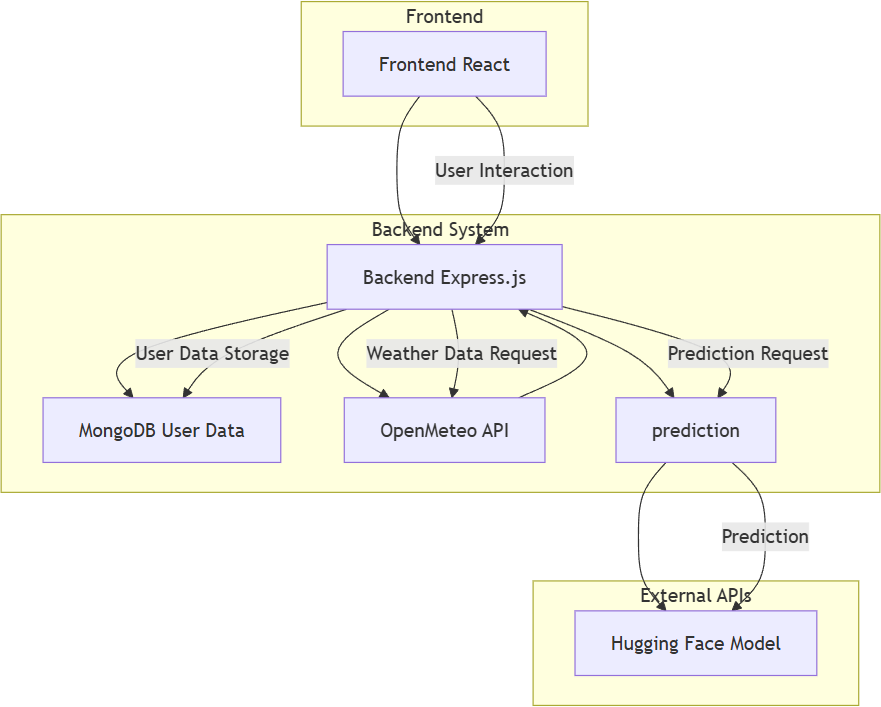


Fig: 4.4(a)- System Design

**4.4.1 Data Flow Diagram**

The data flow diagram of the project are divided into levels, level-0, level-1 and level-2. The diagrams are given in figures fig-4.4.1(a), fig-4.4.1(b),fig- 4.4.1(c) respectively.

#### LEVEL-0

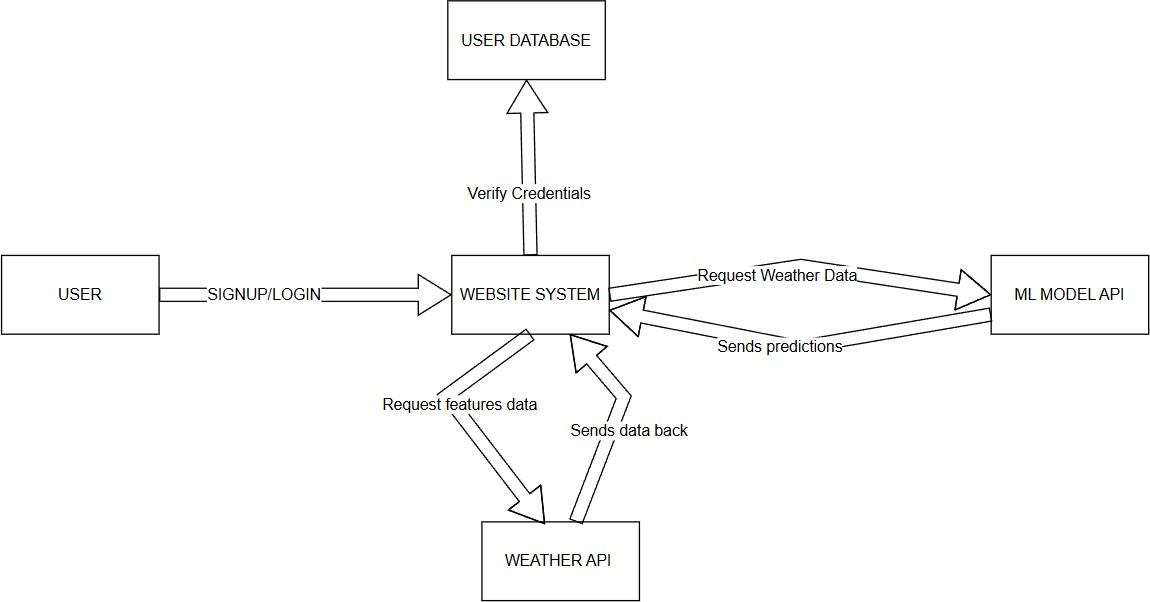


Fig-4.4.1(a) Level-0 DFD

LEVEL-1

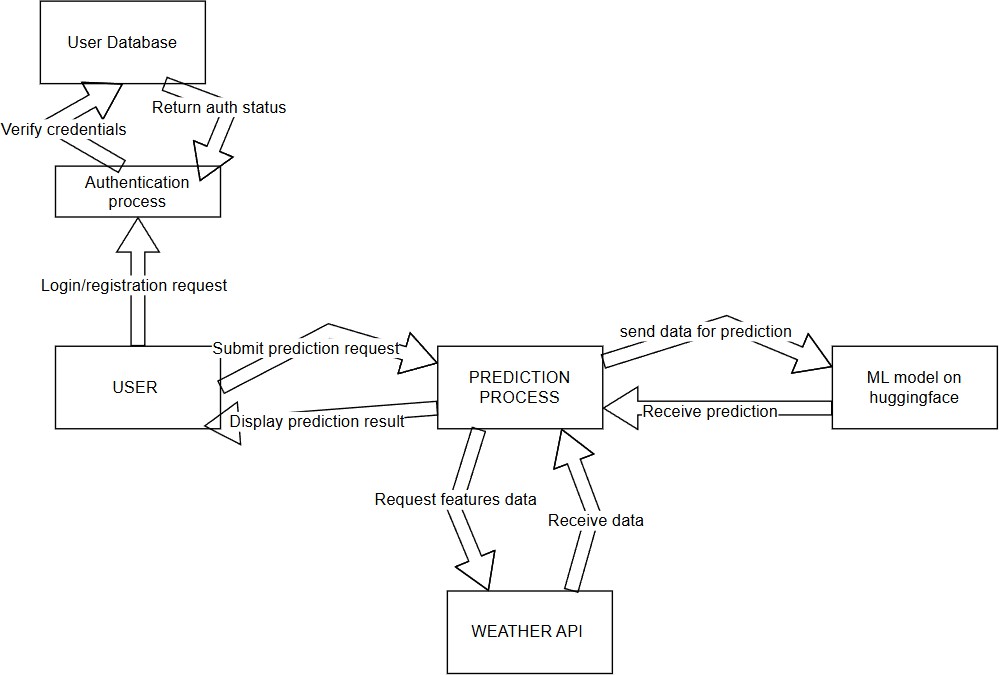


Fig-4.4.1(b) Level-1 DFD

LEVEL-2

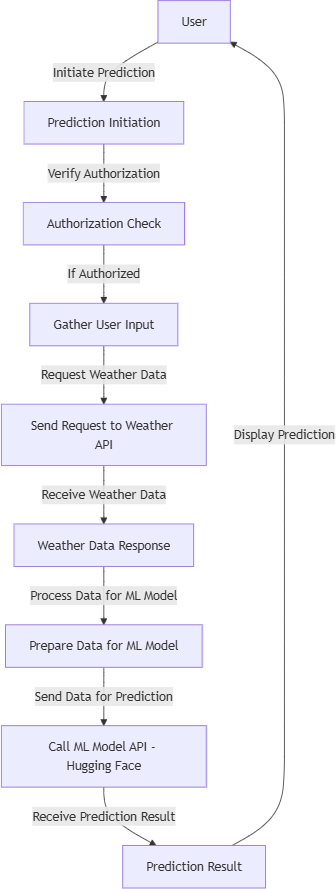


Fig-4.4.1(c)- Level-2 DFD

**4.4.2 Use Case Diagram**

The use case diagram of the project is given below:

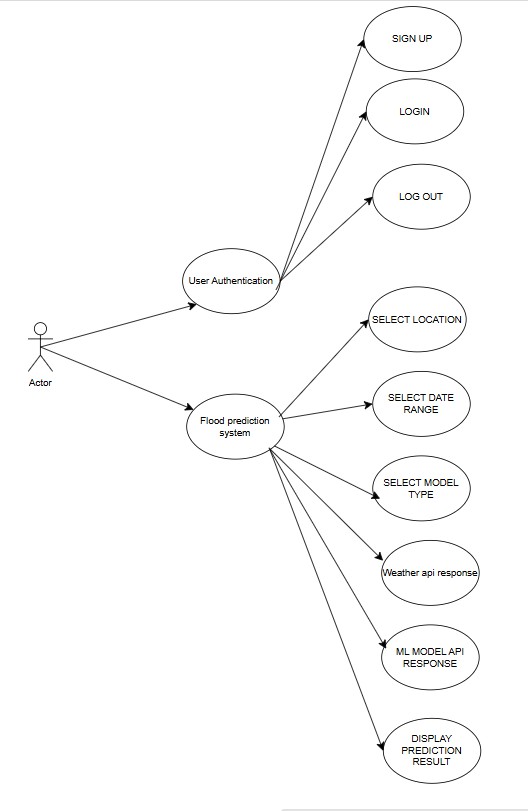


Fig: 4.4.2(a)- use case diagram

**4.5 Database Design**

At this point, that is currently the project utilises only one database to maintain the user authentication. That is there is single database name “user database”, which have user information i.e Email and password for registered users.

|  |  |  |
| --- | --- | --- |
| FIELD | DATA TYPE | DISCRIPTION |
| ID | Integer | Primary key, unique identifier for each user. |
| Email | Varchar (255) | The user’s email address (must be unique) |
| Password | Varchar (255) | The user’s hashed password. |

Table- 4.5(a) – database design – (user database)

**CHAPTER 5**

**IMPLEMENTATION**

**5.1 INTRODUCTION & TECHNOLOGIES USED**

In this chapter, we discuss the tools and technologies utilized in the development and implementation of the flood prediction model. The implementation process involves data preprocessing, feature engineering, model training, and evaluation. Various software tools, programming languages, and libraries were used to ensure the efficiency and accuracy of the model.

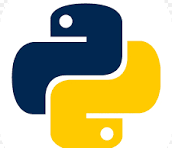
Programming languages: The project uses two core programming languages which are python and java script. Python is used for machine learning model development and javascript is used for developing web applications.

Development Tools: There were various development tools which were used for developing this project, the tools which were used are Google collab, Jupyter Notebook, Vs code, Kaggle.

Libraries and frameworks: This project utilises many libraries and frameworks to make the software more efficient. The main libraries and frameworks which were used for developing this project are : Pandas, NumPy, Matplotlib & seaborn, scikit-learn, XGBoost, React, TensorFlow, tailwindcss, bootstrap, Gradio.

Deployment Platforms: This project utilises deployment platform lie huggingface where the machine learning model is deployed at, it helps in providing better computation and efficiency for flood predictions.





Gradio Logo PNG Vector (SVG) Free Download

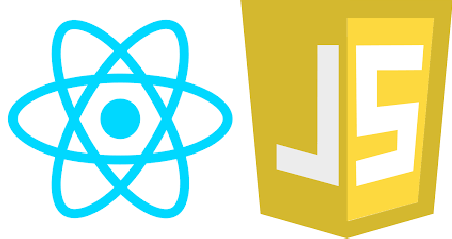
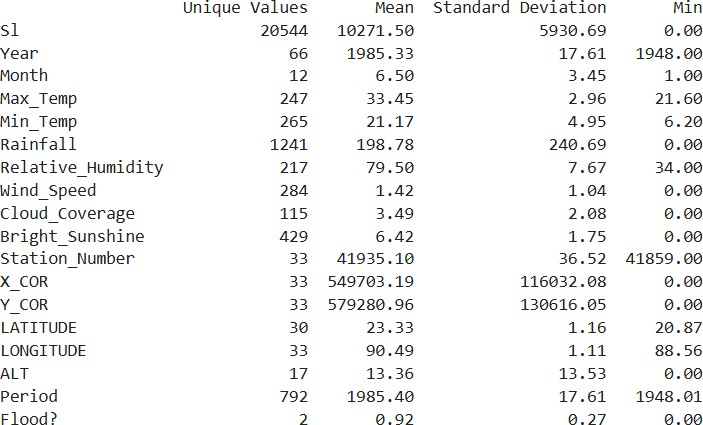


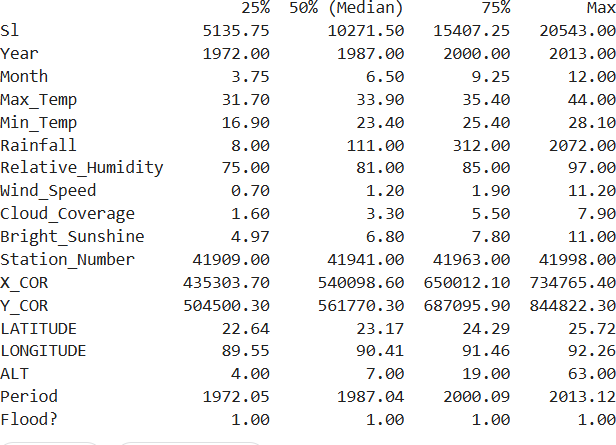
Fig: 5.1(a) overview of tech used

**5.2 Dataset Description**

The dataset used in this project consists of meteorological and geographical features collected from multiple weather stations. It contains 20,544 records with 19 features. The target variable is Flood?, which indicates whether a flood event occurred, this dataset location is based on Bangladesh, there is one more dataset which focuses on focuses on risk factors like MonsoonIntensity, Deforestation, Urbanization, ClimateChange, DrainageSystems, etc. It has around 50,000 rows and 21 columns with target variable FloodProbability (continuous value between 0 and 1). The one which we are going to use is the one which contains 20,544 records with 19 features.

Some of the parameters which can be use to understand the datasets are given below:



 Fig: 5.2(a)- Dataset Description

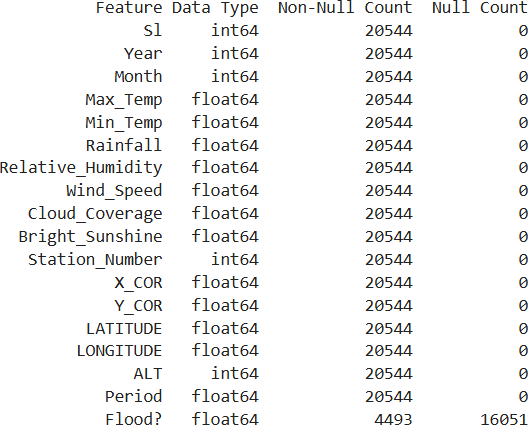


Fig- 5.2(b)- Dataset Description

**CHAPTER 6  
 TESTING & MAINTENANCE**

**6.1 Testing Techniques and Test Cases Used**

Data Validation Testing : Ensured data integrity by checking for missing values, inconsistencies, and outliers. Used statistical methods and visualization techniques to detect anomalies in the dataset.

Model Performance Testing: Evaluated the performance of the machine learning model using:

-Accuracy: Measures correct predictions vs. total predictions.

-Precision, Recall, and F1-Score: Evaluated the model's effectiveness in predicting floods.

-ROC-AUC Score: Assessed the model's ability to distinguish between flood and non-flood instances.

-Confusion Matrix: Analyzed the number of true positives, false positives, true negatives, and false negatives.

Unit Testing: Conducted unit tests on individual functions used for data preprocessing, feature engineering, and model training.Verified that transformations such as normalization and encoding were correctly applied.

Integration Testing: Ensured seamless integration of the trained model with the backend API and frontend application. Checked API request/response handling and model inference correctness.

Deployment and Performance Testing: Tested API deployment on Hugging Face for response time and scalability.

user acceptance testing: We also performed the user acceptance testing.

security testing: To test if the database is secured and data fetch and transfer are doing securely.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Expected**  **Output** | **Actual Output** | **Status** |
| TC-001 | Check for missing and invalid values in the dataset | Dataset should not contain null values or incorrect  entries | Missing values handled, outliers removed | Passed |
| TC-002 | Doing user  registration | Successfully  registered | Registered  successfully | passed |
| TC-003 | Loggin in | Successfully  login | Logged in | passed |
| TC-004 | Evaluate model  accuracy using test data | Model accuracy  should be above 80% | Accuracy  achieved: 95%+ | passed |
| TC-005 | Select location on map | Location get  selected as long. And lat. | Location gets selected | Passed |
| TC-006 | Select date range , model type and fetch features data from weather  API | Features values get as response from the api | Features values did get as a response from api | passed |
| TC-007 | Deployed model on huggingface  should work | Shows predictions via a json format  response | Shows prediction as json response | passed |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TC-008 | Send sample data to API and check response | API should return a valid flood probability  prediction | API returned predictions within expected range | passed |
| TC-009 | Ensure frontend fetches correct predictions from API | The prediction should be displayed correctly on the  user interface | Predictions successfully displayed on UI | Passed |
| TC-010 | Maps getting rendered | Maps is being  shown on frontend | Map is visible | passed |
| TC-011 | React components at frontend are getting rendered  properly | Components are rendered | It is rendered | passed |

Table-6.1(a): Test cases performed

**CHAPTER 7**

**RESULT AND DISCUSSION**

**7.1 Description of Modules with Snapshots**

The flood prediction system consists of multiple modules working together for efficient data processing, model predictions, and user authentication. Below is a description of each module with relevant snapshots:

* Data Preprocessing Module: Loads raw dataset, handles missing values, encodes categorical data, and normalizes numerical features. Ensures data is in the correct format for model training.



Fig – 7.1(a): snapshot for data preprocessing module

* Machine Learning Model: Trained using the XGBoost algorithm, Logistic reression, neural networks and then identify and analyse which among them works better to predict flood probability. Evaluated using accuracy, precision, recall, and AUC-ROC metrics.

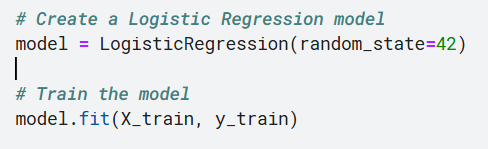


Fig- 7.1(b): model training using logistic regression

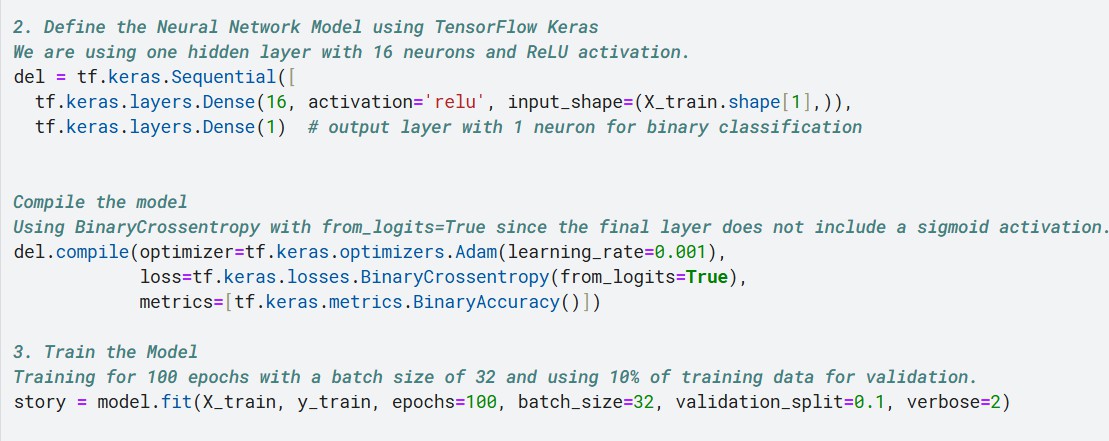


Fig- 7.1(c): Model training using Neural Networks

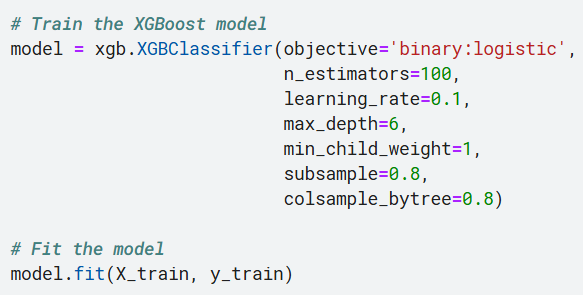
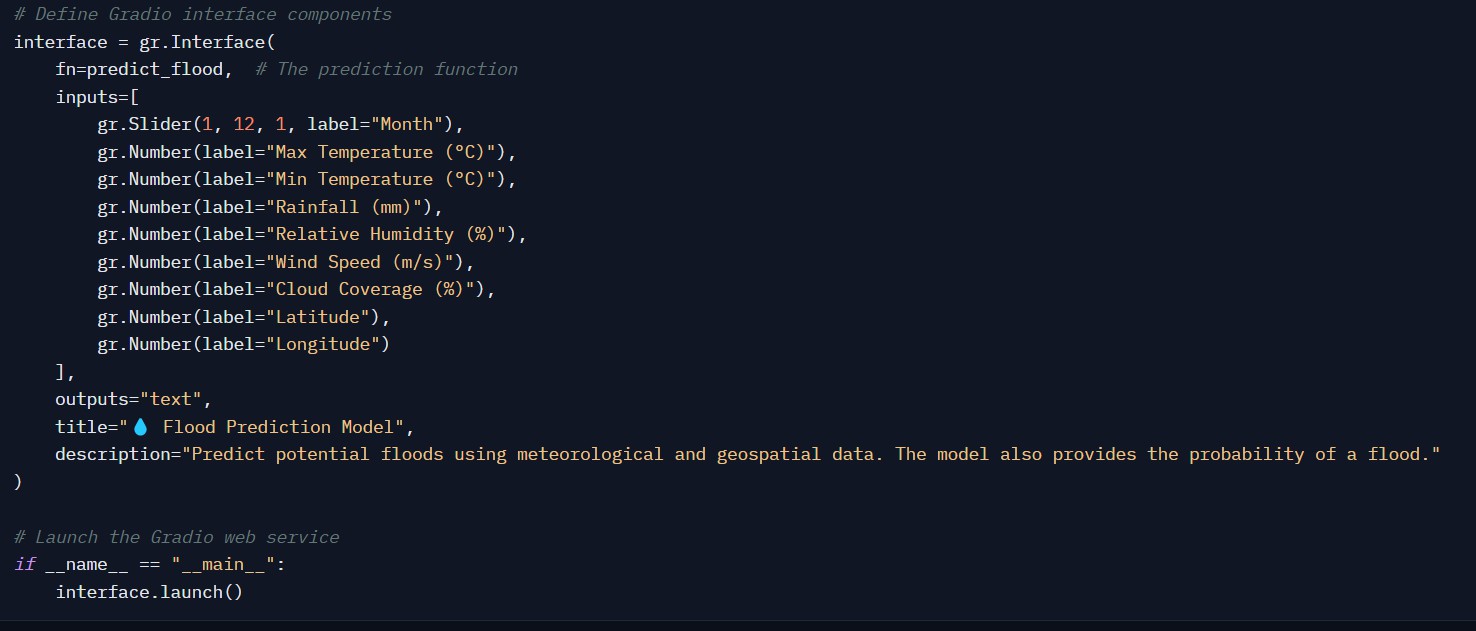
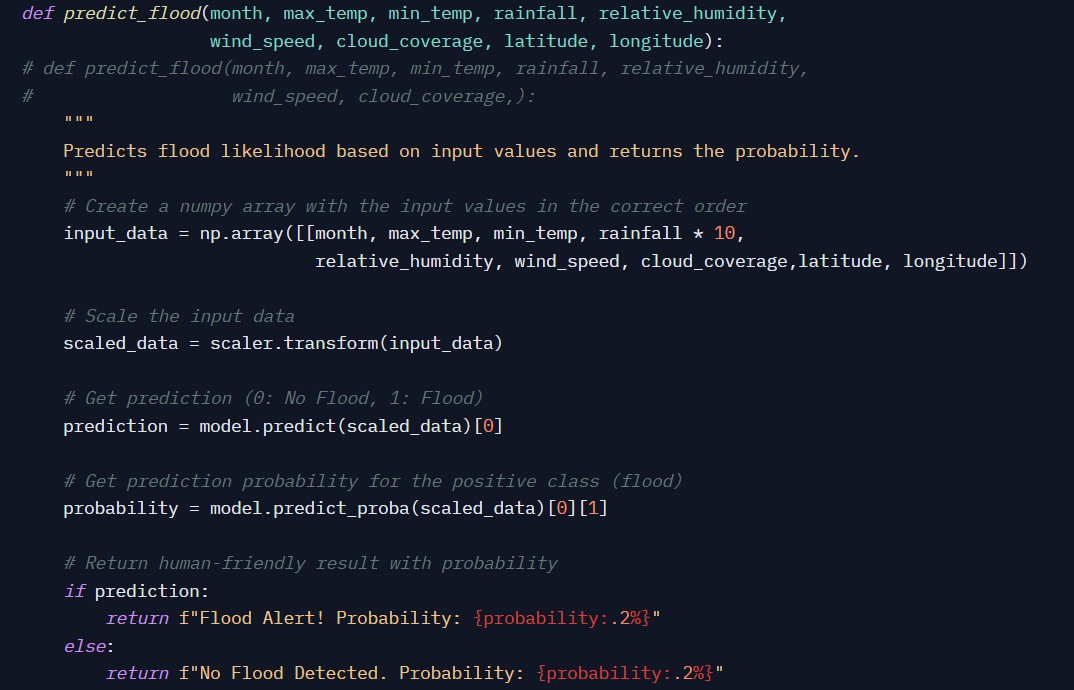


Fig:- 7.1(d): Model training using XGBoost

* API Deployment Module: Model deployed using gradio on huggingface spaces and then Integrated with the frontend for real-time predictions using flask.





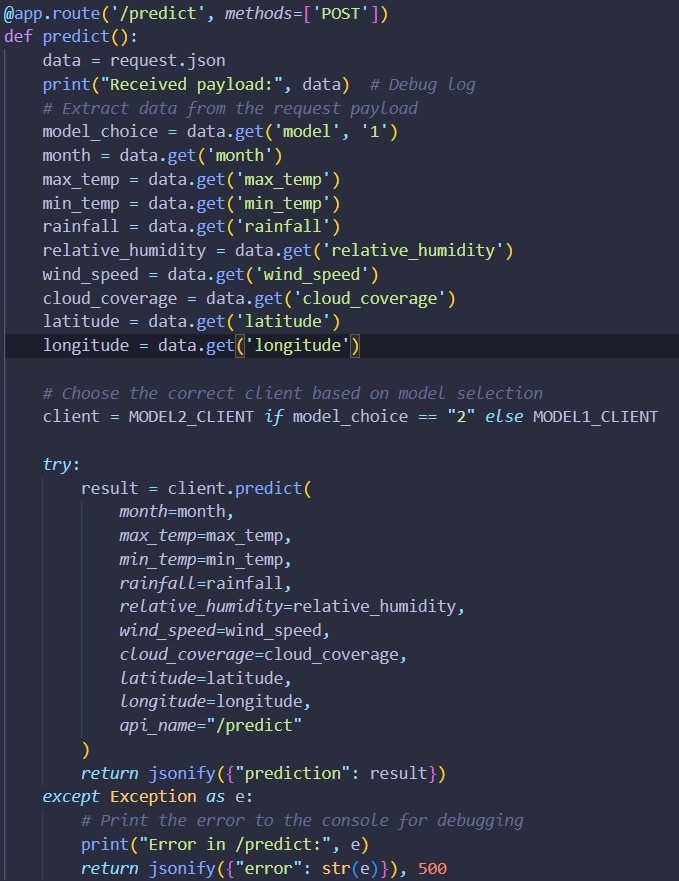
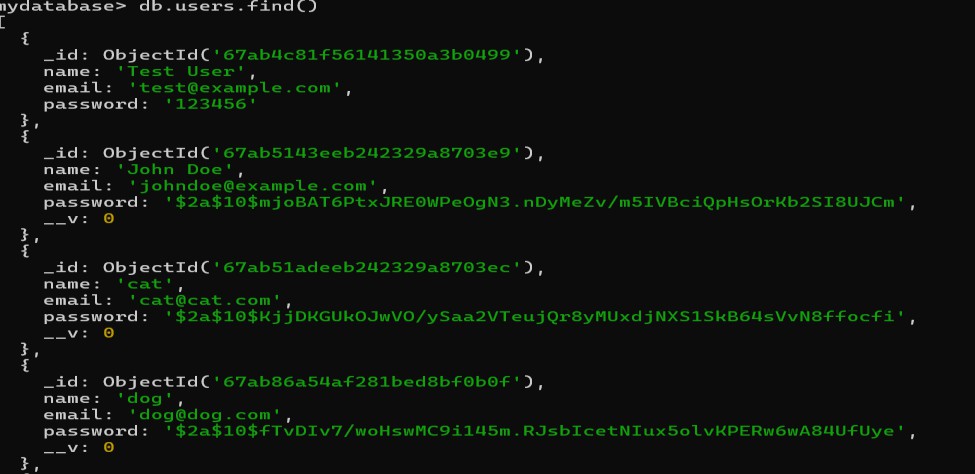
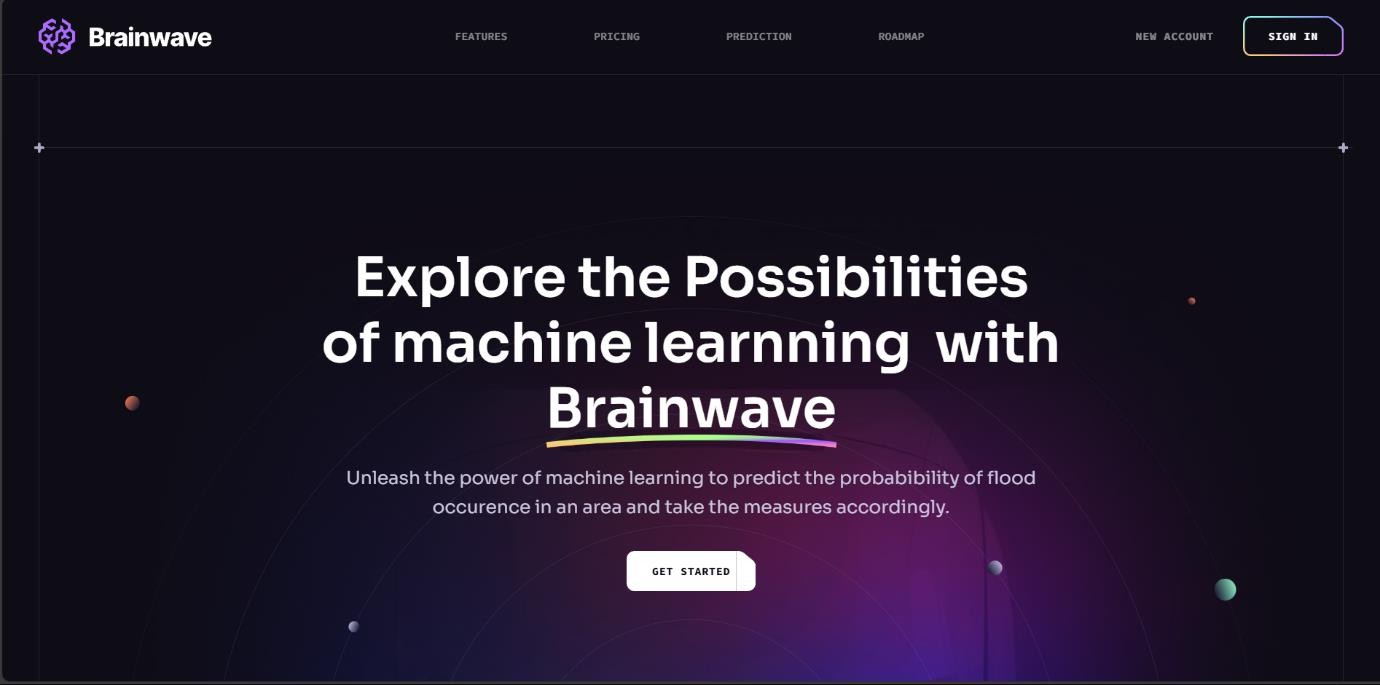


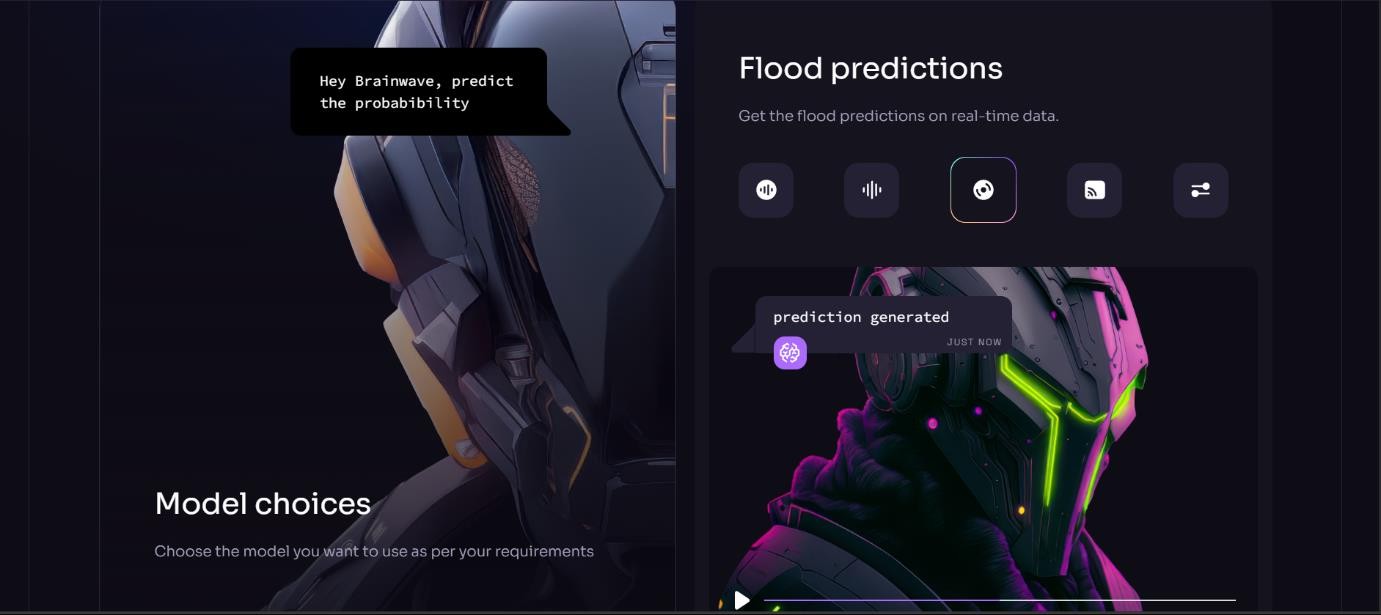
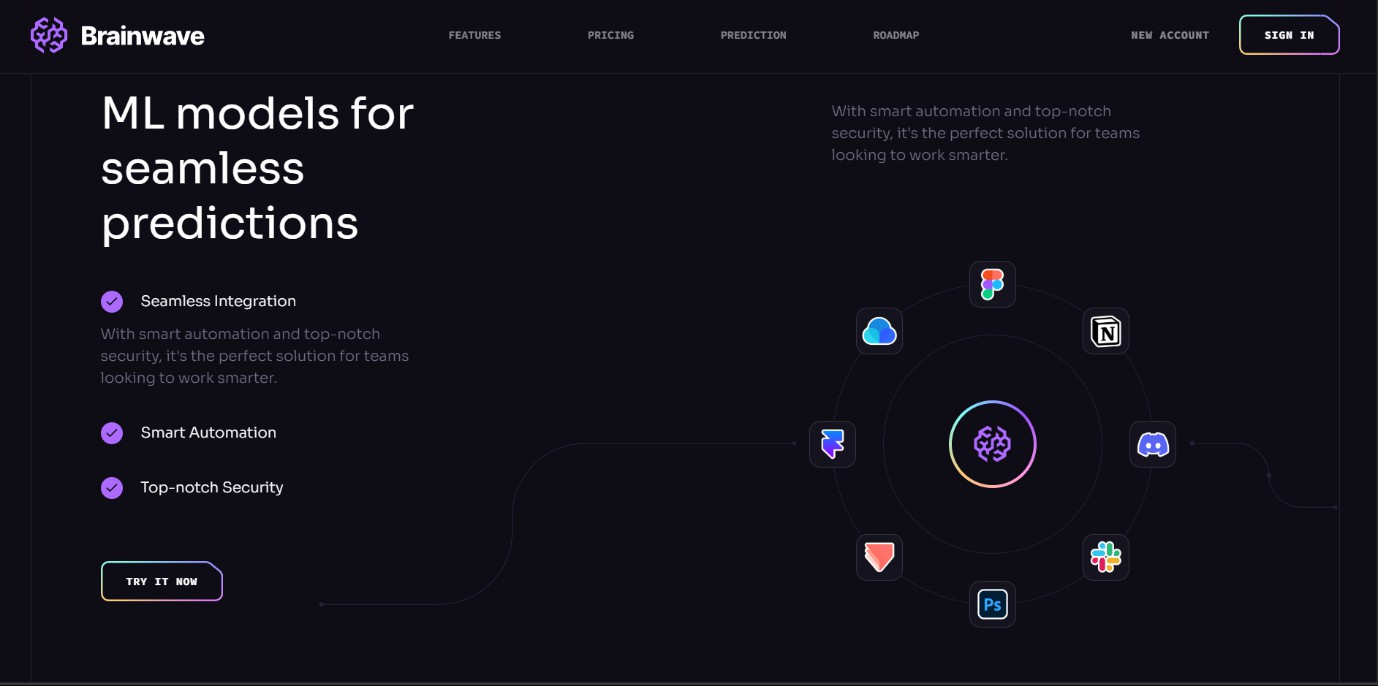
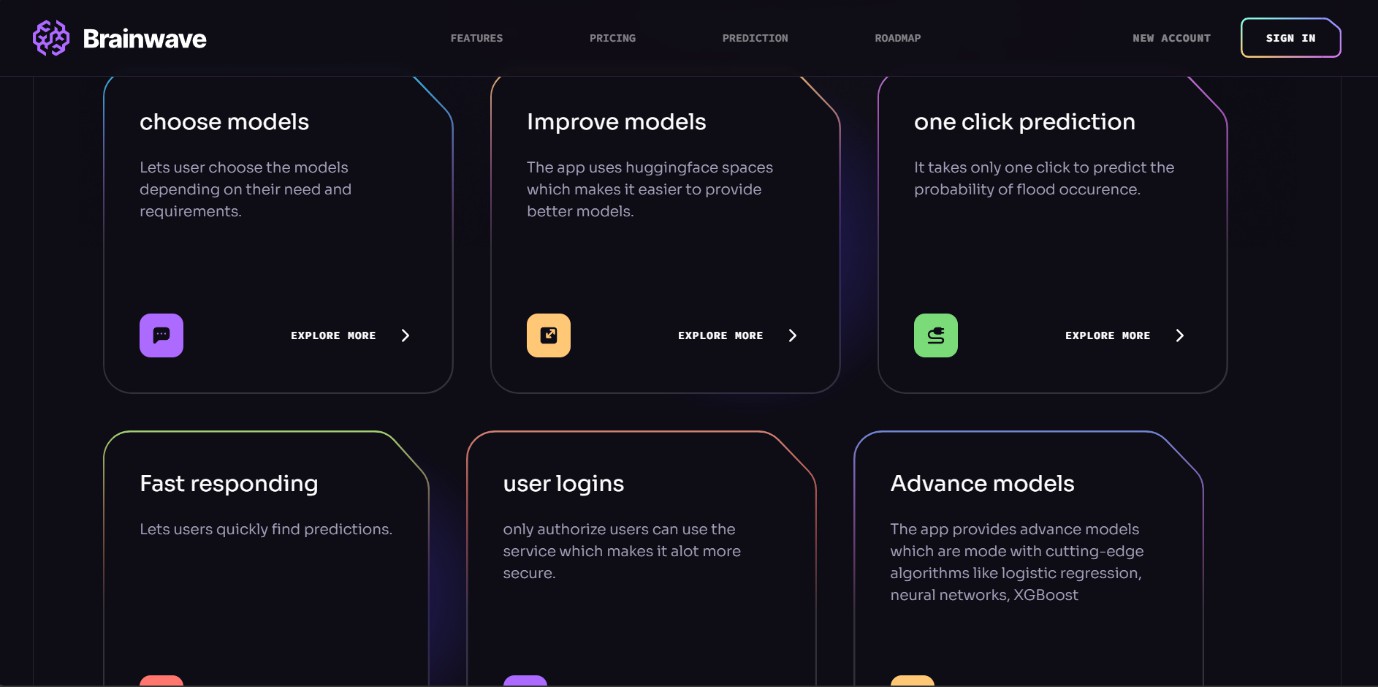
Fig: 7.1(e)- deployment of machine learning model on huggingface using gradio

* User Authentication Module (MongoDB): Users can sign up and log in using email and password. Authentication is handled via Express backend with MongoDB as local storage for development and mongo atlas for production.



* Frontend: This module deals with designing and making a good frontend that follows good user experience and user interface norms.





A screenshot of a weather forecast system

AI-generated content may be incorrect.

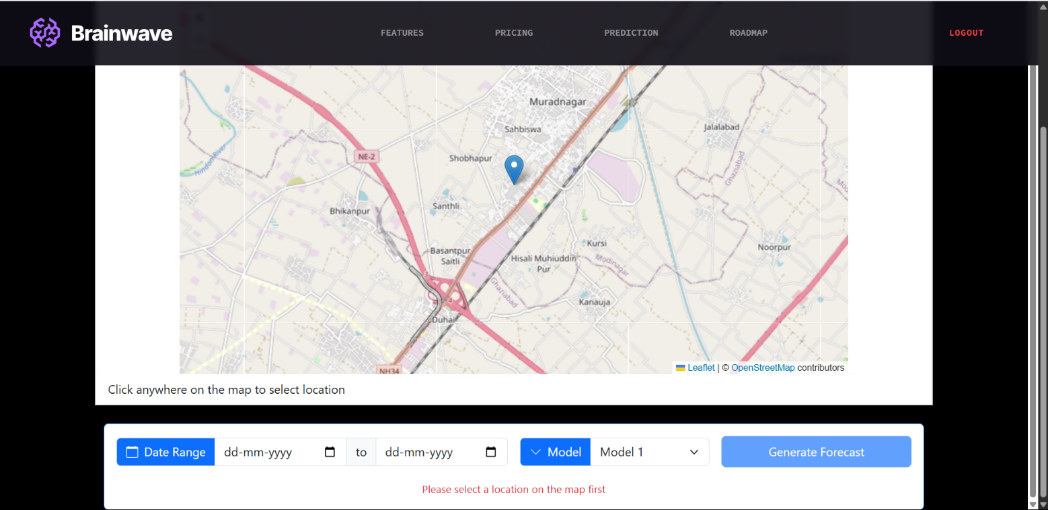


Fig-7.1(f)- frontend of website

**7.2 Presentation of Results**

The system's performance was assessed using various evaluation metrics. The metrices that were used to evaluate the performance of model are Accuracy, Precision, Recall, F1 Score, AUC – ROC for logistic regression and neural networks, apart from these we also used MAE, R-squared score for XGBoost.

I - Evaluation matrices for different models:

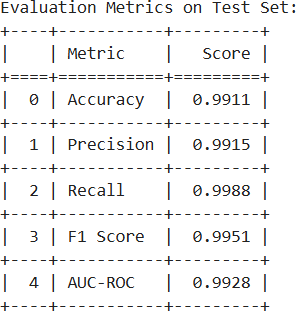


Table: 7.2(a)-Logistic regression

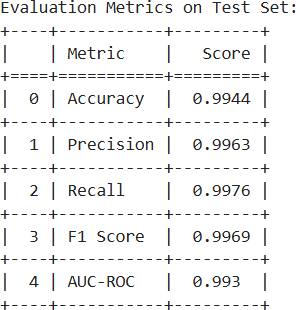


Table: 7.2(b)- Neural networks

1. – Confusion matrices:

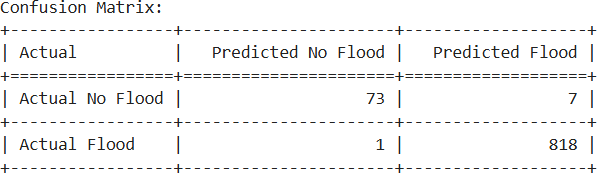


Table: 7.2(c)- Logistic Regression

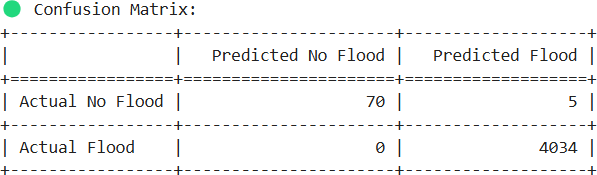


Table: 7.2(d)- Neural Network

1. – Feature Importance



Fig: 7.2(a)- Feature importance (logistic regression)



Fig: 7.2(b)- Feature importance (Neural Network)

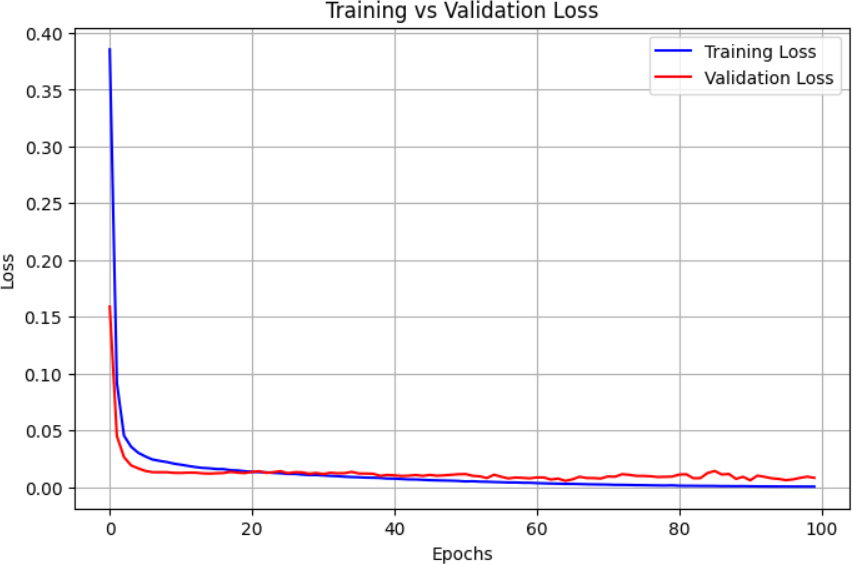


Fig: 7.2(c)- Training vs validation(test) Loss for neural networks

Iv – Results on user end:

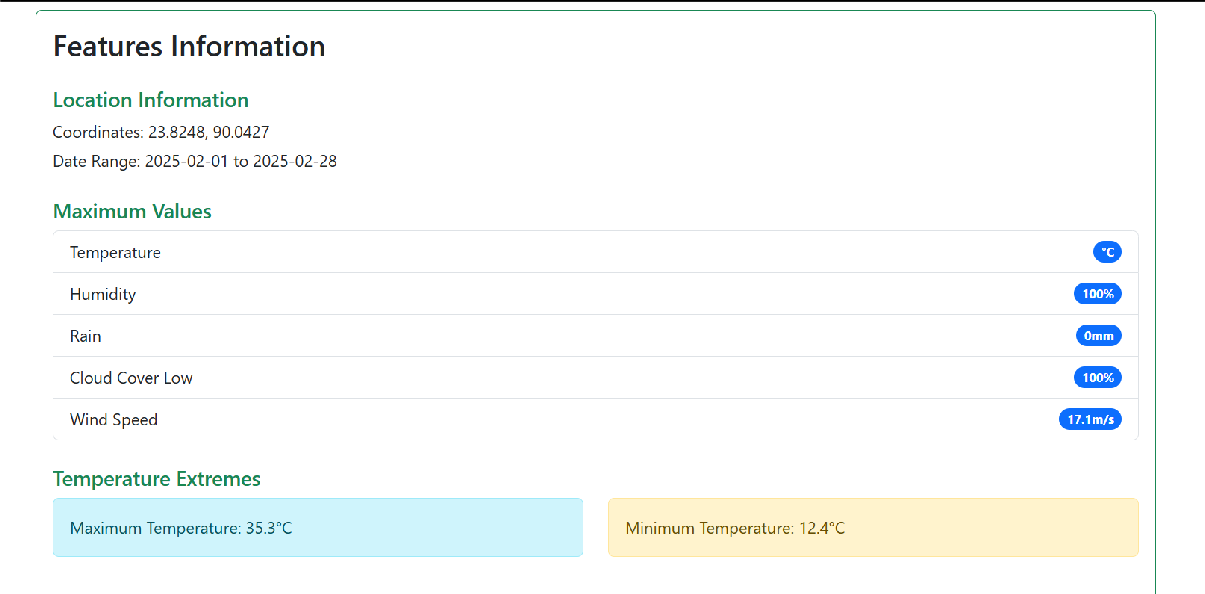
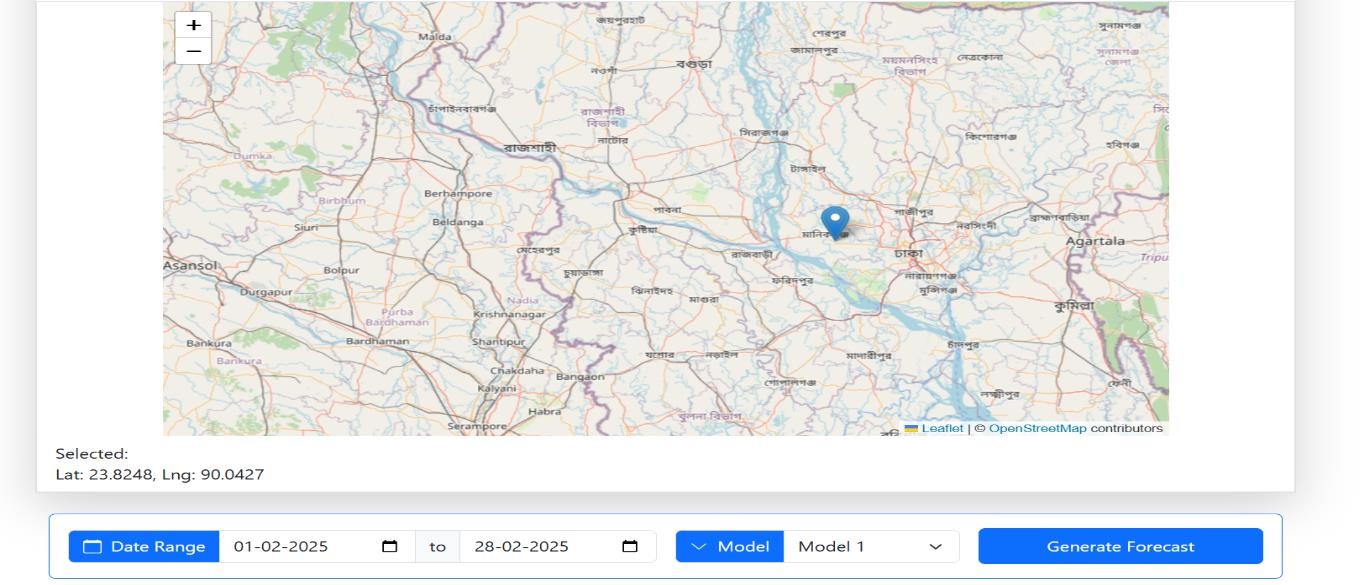


Fig-7.2(d)- results on user end:

**7.3 Performance Evaluation**

The system's performance was assessed using various evaluation metrics and parameters like inference time for training and testing, it helped in deciding which is better model. The tables of evaluation metrics are already provided in section7.2 . please refer to it for further information.

The Inference time for training and testing time was calculated for both model Logistic Regression and neural networks. The results for which are shown below :



Fig: 7.3(a)- Time analysis for Logistic Regression

A number of numbers on a white background  AI-generated content may be incorrect.

Fig: 7.3(b) - Time analysis for Neural Networks

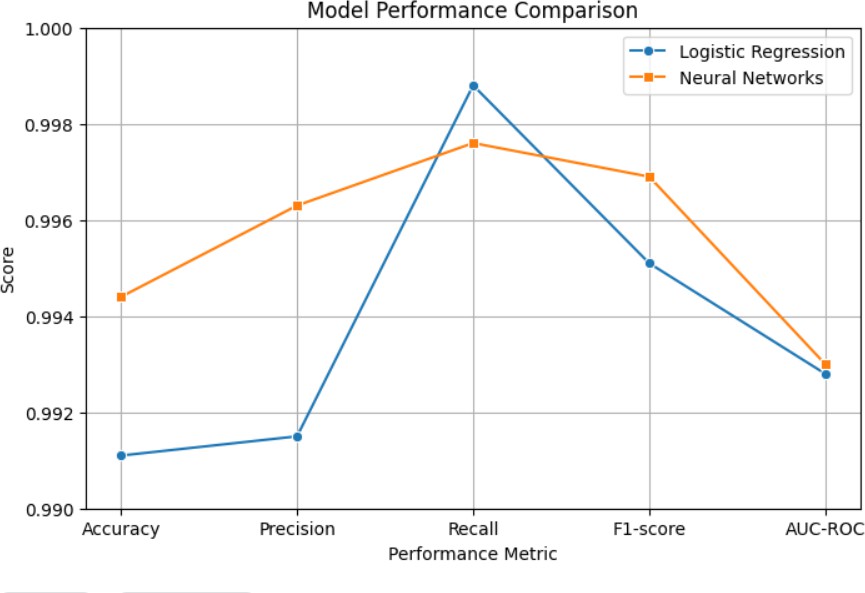


Fig: 7.3(c)- Evaluation metrices comparison for both models

**7.4** **Key Findings**

The analysis of results led to several significant insights:

* High Predictive Accuracy: The models demonstrated strong classification performance, ensuring reliability in flood predictions, the accuracy of both model on testing data is above 95%.
* Geospatial Trends: The flood-prone regions were strongly correlated with factors such as location i.e station\_name, also the temperature , and rainfall intensity, these are the factors which were playing a major role in flood preditions.
* Model Interpretability: Both models works really well but depending upon the parameters, we can choose any model. For easy deployment and less time for training and inference ,with high recall we can chose logistic regression model, but if we need a model with high precision than neural networks are giving better results as shown in fig-7.3(c).
* Scalability & Deployment Feasibility: The Hugging Face Gradio interface allowed seamless access to predictions, making the tool user-friendly for non-technical stakeholders.
* Resource Efficiency: Operated on low-end hardware (4GB RAM), ensuring accessibility in regions with limited infrastructure.

**CHAPTER 8**

**CONCLUSION AND FUTURE SCOPE**

**8.1 Conclusion**

This initiative has pioneered an advanced flood inundation probability mapping framework that inventively coordinating cutting-edge machine learning strategies, exactness geospatial analytics, and available web-based interfacing. The system leverages two robust predictive models, a logistic regression classifier and an artificial neural network trained on comprehensive geospatial datasets encompassing topographic elevation patterns, historical rainfall variability, and temperature fluctuations. Through deployment on Hugging Face's Gradio platform, the solution enables instantaneous flood risk visualization, transforming complex hydrological modeling into an intuitive public resource for communities, emergency responders, and policymakers.

Core Advancements and Outcomes:

High-Precision Predictive Modeling

The engineering illustrates extraordinary discriminative capability, accomplishing AUC-ROC scores surpassing 90% amid thorough cross-validation testing. This execution stems from orderly highlight designing of hydrologically critical factors and iterative optimization of show structures, guaranteeing dependable recognizable proof of high-risk zones beneath different climatic scenarios.

Easy handling

The platform's single-click operational plan dispenses with specialized boundaries through an intellectuals mapped facilitate input framework. Clients get localized surge probabilities (0-100%) by means of intelligently maps matched with energetic chance seriousness markers, successfully bridging the hole between scholastic hydrology and community-level catastrophe preparedness.

Sustainable Innovation Framework

Developed only with open-source stack components including Python's Scikit-Learn for conventional ML, TensorFlow for profound learning usage. the framework illustrates replicability over creating locales. This approach decreases foundation costs by many folds compared to exclusive surge modeling program whereas keeping up enterprise-grade explanatory capabilities.

Strategic Impact:

By focalizing prescient analytics, geospatial insights, and cloud-based arrangement, the venture rises above ordinary hypothetical surge models that regularly stay kept to scholastic inquire about. Its real-time preparing motor adjusts to advancing climate designs through persistent information pipeline integration from meteorological APIs and fawning symbolism nourishes. This operational preparation positions the apparatus as a basic decision-support resource for urban organizers creating flood-resistant framework and calamity administration organizations optimizing crisis reaction conventions. The intrigue blend of information science and natural designing strategies builds up a modern worldview in proactive climate adjustment procedures, especially for flood-vulnerable locales missing progressed hydrological checking frameworks.

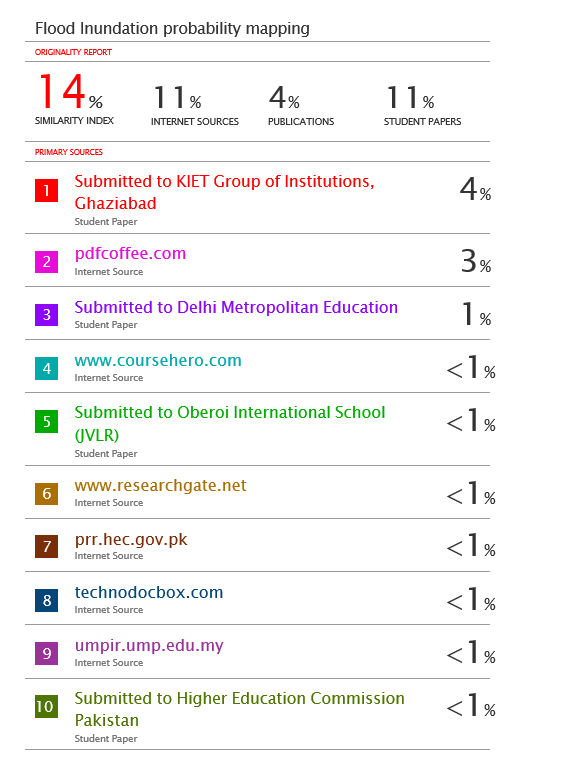
**8.2 Future Scope**

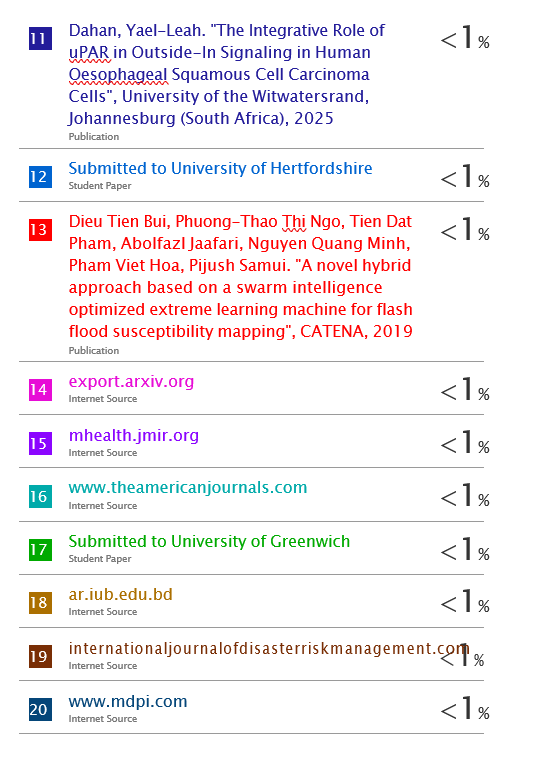
The system's capabilities can be significantly enhanced by expanding its geospatial coverage through training on global datasets, or we can train localized models for the locations with high flood chances and integrating high-resolution satellite imagery like Sentinel-1/2, while simultaneously improving prediction timeliness via real-time data streaming from IoT sensors such as river gauges and soil moisture probes; exploring advanced machine learning techniques including hybrid models (e.g., CNN- LSTM) coupled with uncertainty quantification would further boost prediction accuracy and reliability; developing a Progressive Web App (PWA) would ensure mobile-first accessibility even in offline conditions in remote areas, and incorporating crowdsourcing features would allow community participation in reporting flood incidents to refine model accuracy; additionally, simulating flood risks under future climate scenarios using IPCC data would aid in climate change adaptation planning, and partnering with disaster management agencies to integrate the tool into early- warning systems would maximize its policy impact and real-world utility.

# REFRENCES

* + - 1. Mosavi, P. Ozturk, and K. Chau, “Flood Prediction Using Machine Learning Models: Literature Review,” Water, vol. 10, no. 11, p. 1536, Oct. 2018, doi: <https://doi.org/10.3390/w10111536>.
      2. Z. K. Lawal, H. Yassin, and R. Y. Zakari, “Flood Prediction Using Machine Learning Models: A Case Study of Kebbi State Nigeria,” 2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), pp. 1–6, Dec. 2021, doi: <https://doi.org/10.1109/csde53843.2021.9718497>.
      3. D. L. Nguyen, T. Y. Chou, T. V. Hoang, and M. H. Chen, “Flood Susceptibility Mapping Using Machine Learning Algorithms: A Case Study in Huong Khe District, Ha Tinh Province, Vietnam,” International Journal of Geoinformatics, vol. 19, no. 7, Jul. 2023, doi: <https://doi.org/10.52939/ijg.v19i7.2739>.
      4. Sliman Hitouri et al., “Flood Susceptibility Mapping Using SAR Data and Machine Learning Algorithms in a Small Watershed in Northwestern Morocco,” Remote Sensing, vol. 16, no. 5, pp. 858–858, Feb. 2024, doi: <https://doi.org/10.3390/rs16050858>.
      5. Sackdavong Mangkhaseum, Y. Bhattarai, Sunil Duwal, and Akitoshi Hanazawa, “Flood susceptibility mapping leveraging open-source remote-sensing data and machine learning approaches in Nam Ngum River Basin (NNRB), Lao PDR,” Geomatics Natural Hazards and Risk, vol. 15, no. 1, May 2024, doi: <https://doi.org/10.1080/19475705.2024.2357650>.
      6. A. Rajab et al., “Flood Forecasting by Using Machine Learning: A Study Leveraging Historic Climatic Records of Bangladesh,” Water, vol. 15, no. 22, p. 3970, Jan. 2023, doi: <https://doi.org/10.3390/w15223970>.
      7. R. Khalid and U. T. Khan, “Flood susceptibility mapping using ANNs: a case study in model generalization and accuracy from Ontario, Canada,” Geocarto international, vol. 39, no. 1, Jan. 2024, doi: <https://doi.org/10.1080/10106049.2024.2316653>.
      8. M. Ahmadlou et al., “Flood susceptibility mapping and assessment using a novel deep learning model combining multilayer perceptron and autoencoder neural networks,” Journal of Flood Risk Management, vol. 14, no. 1, Dec. 2020, doi: https://doi.org/10.1111/jfr3.12683 .
      9. M. Hasanuzzaman, A. Islam, B. Bera, and P. K. Shit, “A comparison of performance measures of three machine learning algorithms for flood susceptibility mapping of river Silabati (tropical river, India),” Physics and Chemistry of the Earth, Parts A/B/C, vol. 127, p. 103198, Oct. 2022, doi: https://doi.org/10.1016/j.pce.2022.103198.
      10. Gauhar, Noushin and Das, Sunanda and Moury, and Khadiza Sarwar, “Prediction of Flood in Bangladesh using k-Nearest Neighbors Algorithm,” 2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), pp. 357–361, 2021.
      11. N. Khalid, “Flood Prediction Dataset,” Kaggle.com, 2024. https://[www.kaggle.com/datasets/naiyakhalid/floodprediction-dataset](http://www.kaggle.com/datasets/naiyakhalid/floodprediction-dataset) [1]
      12. T. Chen and C. Guestrin, “XGBoost: a Scalable Tree Boosting System,” Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD ’16, pp. 785–794, 2016, doi: <https://doi.org/10.1145/2939672.2939785>.
      13. Liudmila Ostroumova Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin, “CatBoost: unbiased boosting with categorical features,” arXiv (Cornell University), Jun. 2017, doi: <https://doi.org/10.48550/arxiv.1706.09516>.
      14. C.-Y. J. Peng, K. L. Lee, and G. M. Ingersoll, “An Introduction to Logistic Regression Analysis and Reporting,” The Journal of Educational Research, vol. 96, no. 1, pp. 3–14, Sep. 2002, doi: <https://doi.org/10.1080/00220670209598786>.
      15. Scikit-learn, “scikit-learn: Machine Learning in Python,” Scikit-learn.org. <https://scikitlearn.org/stable/>
      16. I. H. Sarker, “Machine Learning: Algorithms, Real-World Applications and Research Directions,” SN Computer Science, vol. 2, no. 3, pp. 1–21, Mar. 2021, doi: https://doi.org/10.1007/s42979-021-00592-x.
      17. Agri Sustainability, Food Research, S. Sahu, and Shobhana Ramteke, “Floods disaster in India, mitigation and their impacts,” Sustainability Agri Food and Environmental Research, vol. Vol. 12 (2024), no. Special issue: Climate change, Apr. 2023, Accessed: Dec. 18, 2024. [Online].Available: https://[www.researchgate.net/publication/371069486\_Floods\_disaster\_in\_India\_mitigation\_a](http://www.researchgate.net/publication/371069486_Floods_disaster_in_India_mitigation_a)nd\_their\_impacts
      18. “(PDF) Confusion Matrix-based Feature Selection.,” ResearchGate. https://[www.researchgate.net/publication/220833270\_](http://www.researchgate.net/publication/220833270_)Confusion\_Matrix- based\_Feature\_Selection
      19. V. Sharma, “A Study on Data Scaling Methods for Machine Learning,” International Journal for Global Academic & Scientific Research, vol. 1, no. 1, Feb. 2022, doi: https://doi.org/10.55938/ijgasr.v1i1.4.
      20. S. Simon, N. Kolyada, C. Akiki, M. Potthast, B. Stein, and N. Siegmund, “Exploring Hyperparameter Usage and Tuning in Machine Learning Research,” IEEE Xplore, May 01, 2023. <https://ieeexplore.ieee.org/document/10164726>

**Turnitin Report**





A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a web page

AI-generated content may be incorrect.

**RESEARCH PAPER STATUS**

Status : communicated Proofs

# PATENT STATUS

Status: published

Application no- 202511002778 A

