A

Project Report on

**FLOOD FORECASTER -FLOOD PREDICTION USING MACHINE LEARNING ALGORITHM**

Submitted in partial fulfillment of completion of the course

Advanced Diploma in IT, Networking and Cloud

Year 2022 -2024

Submitted by:

**Manjima Bhattacharyya**

**Anoushka Das**

**Archana Kumari**

**Aashika Raj**

**Roshni Verma**

Under Guidance of:

**Arpita Roy (Edunet)**

|  |  |  |
| --- | --- | --- |
|  |  |  |

Abstract

Acknowledgement

Team Composition and Workload Division

Table of Contents

1. Introduction to Problem

2. Proposed Solution

3. Requirements

* Technology Stack
* Hardware
* Software
* Deployment Environment

4.User Requirements

5. Design Documentation

6. Implementation Details

7.Testing

8.Deployment

9.Future Scope

10. Conclusion

Appendix A Screenshot of Project

References

**Abstract**

Floods are one of the most devastating natural disasters worldwide, causing significant loss of life, damage to infrastructure, and economic disruption. Early flood prediction is crucial for timely evacuation, damage mitigation, and resource allocation. In this paper, we present Flood Forecaster, a machine learning-based flood prediction system that leverages advanced algorithms and data sources to forecast flood events with high accuracy. Our system incorporates historical weather data, precipitation patterns, terrain information, and real-time sensor readings to develop robust predictive models. We employed ensemble methods and feature engineering techniques to improve model performance and address class imbalance issues. Flood Forecaster’s user-friendly interface enables stakeholders to visualize predicted flood zones, receive real-time alerts, and access critical information for informed decision-making. Our system has the potential to revolutionize flood risk management by providing early warning systems, reducing damages, and saving lives.

**Acknowledgment**

Team ‘Tech Titans’ (comprising 5 members, namely Manjima Bhattacharyya, Anoushka Das, Archana Kumari, Aashika Raj & Roshni Verma) are thankful to our teacher, Ms. Arpita Roy for her guidance and supervision which has provided a lot of resources needed in completing our project.

We are also thankful for the efforts put in by our team members and their contributions to the preparation of this project.

**Team Composition and Workload Division**

Our team comprises of 3 members:

Bhattacharyya, Anoushka Das, Archana Kumari, Aashika Raj & Roshni Verma.

**Table of Contents**

1. **Introduction to Problem**

Flood forecasting is needed for developing appropriate measures to control flood risk, mitigate flood hazard, evacuate people from flood hazard areas, determine insurance premiums, and manage environmental and water resources systems.

1. **Proposed Solution**

Hence, it is important that we develop a flood prediction model using machine learning techniques and Python programming language. The model should be able to analyse various environmental factors, such as rainfall, temperature, humidity, and water levels, and use these factors to predict the likelihood of a flood occurring in a given area. The goal of the project is to build an accurate and reliable flood prediction system that can provide warnings and help in emergency preparedness and response.

**3.Requirements**

* **Technology Stack:**

Technologies and Requirements

1**. Python:**

Purpose:Core programming language for developing ML and DL Models

**Frameworks:**

**Flask**: Lightweight web framework for serving ML models.

**Django**: More feature-rich web framework for building the backend.

**2. Machine Learning and Deep Learning Libraries:**

Scikit-learn: For traditional ML algorithms.

TensorFlow: For building and training DL models.

Keras: High-level neural networks API, running on top of TensorFlow.

PyTorch: Alternative to TensorFlow, popular for research and production.

**3. Data Processing and Analysis:**

* Pandas: Data manipulation and analysis.
* NumPy: Numerical computing.
* Matplotlib / Seaborn: Data visualization.

**4. Databases:**

* + - Purpose: Store historical data, user data, and prediction results.
    - Options:
    - PostgreSQL: Open-source relational database.
    - MySQL: Another popular relational database.
    - MongoDB: NoSQL database for more flexible data storage.
    - Tools: SQLAlchemy (for ORM with SQL databases).

**5. Cloud Services**:

* Purpose: Hosting the web application, storing large datasets, deploying ML models.
* Providers:
* AWS: Amazon Web Services.
* Google Cloud Platform: Offers services like BigQuery, AI Platform.
* Microsoft Azure: Another major cloud provider.
* Tools: Docker (for containerization), Kubernetes (for orchestration).
  1. **Hardware:**

Device name DESKTOP-VDV25SG

Processor Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz 3.19 GHz

Installed RAM 16.0 GB (15.8 GB usable)

Device ID F15CC3EA-F8F5-4986-A23B-2D4800C276E9

Product ID 00330-51616-71380-AAOEM

System type 64-bit operating system, x64-based processor

* 1. **Deployment Environment:**

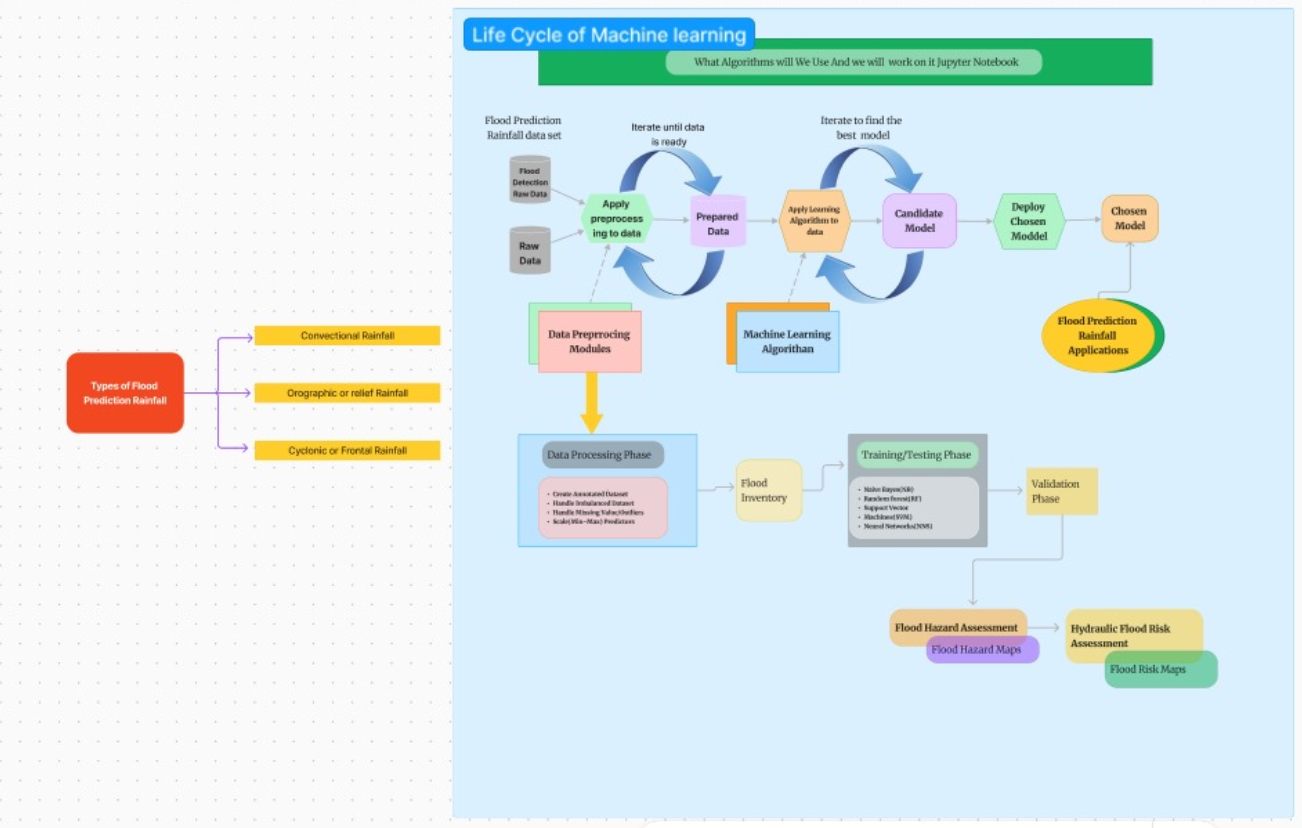
**4. User Requirements:**

A flood forecasting website can benefit a wide range of users, including:

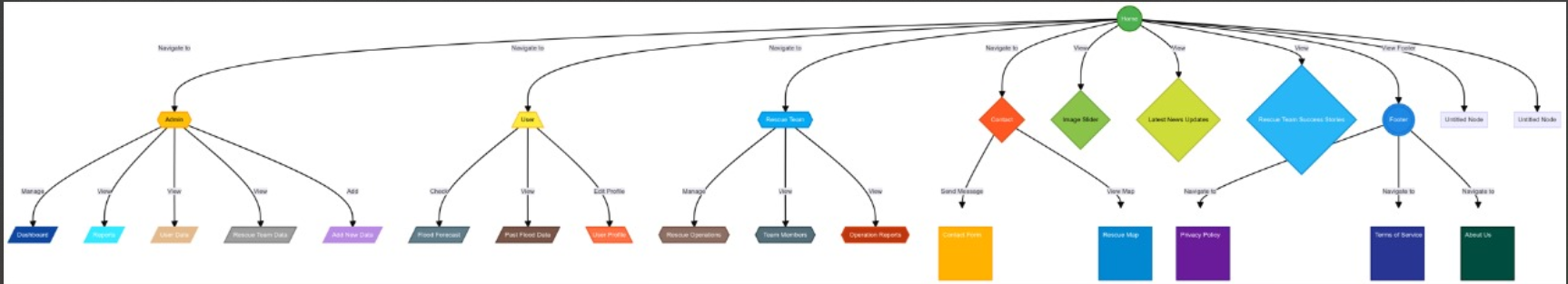
* Homeowners and property owners in flood-prone areas
* Farmers and agricultural producers
* Public health officials
* Environmental organizations
* Local governments
* Researchers and academics
* Tourism and travel industry professionals
* Media outlets

These users can access timely and accurate information on flood risks, warnings, and evacuation instructions to help them prepare for and respond to flood events.

* 1. **Design Documentation**



**Flowchart**



6. **Implementation Details**

**STEP1**- Firstly, we have to download csv file from Kaggle(‘<https://www.kaggle.com/datasets/shiratorizawa/ncarcsv2’>).

**STEP 2-** Secondly, we have to exact file.

**STEP 3-** We then have to open Jupiter notebook.

**STEP 4 -** We need to import all libraries like NumPy, pandas, math plot, seaborn etc.

**STEP 5 -** Then, we have to read the whole csv file by this command [df = pd. read\_csv ('floodprediction.csv')

print(df)]

**STEP 6 -** Then we have to give full information about csv file by using this command (df.info ()).

**STEP 7 –** After that, we have to take the backup file from my original dataset and describe it also. [ water \_backup = df1.copy()] [ missing\_water\_data = water\_backup [water\_backup. is null (). any(axis=1)]

missing\_water\_data] [water\_backup. describe ()]

**STEP 8 –** We need to show different charts and graph for understanding that in my csv file the parameter of water is high or low than normal water parameter and is also save for drinking, or some necessary work.

**STEP 9 -** Correlation matrix heat map. So, we have to check the quality of water for everything. It shows the relation with all.

**STEP 10 –** We have to update and improve your platform based on user because day by day the quality of water is getting bad/poor.

**STEP 11-** From there we implement dashboard editor

**7. Deployment**

**8. Conclusion**

Flood Forecaster's flood prediction system showcases the power of machine learning in tackling real-world challenges. By combining various data sources, including historical data, real-time information, and advanced algorithms, Flood Forecaster provides a cutting-edge solution for predicting flood events. This innovative system enables places to improve their preparedness and better manage flood-related challenges, ultimately protecting lives and property. With its global reach, Flood Forecaster sets a precedent for innovative disaster management solutions worldwide, highlighting the potential for machine learning to make a significant impact on disaster response and recovery efforts.

**9. Future Scope**

* Increased Prediction Accuracy: Enhanced models analyze extensive datasets for more precise flood forecasts.
* Real-Time Monitoring: Integration with IoT devices provides immediate flood predictions and alerts.
* Climate Change Insights: Predictive models adapt to changing weather patterns, improving future flood resilience.
* Risk Assessment Enhancement: Identifies high-risk areas for targeted disaster preparedness and resource allocation.
* Cost Efficiency: Automates analysis, reducing the need for manual intervention and lowering operational costs.

**Appendix A : Screenshot of Project**

**References**