Final Portfolio Assessment: End-to-End Machine Learning Project

Flood Risk Classification Using Environmental and Climatic Factors (CLASSIFICATION)

Course: Concepts and Technologies of AI (5CS037)

Student Name: Kuldeep Mandal

Student ID: 2505925

Tutor: Ayush Regmi

Abstract

Floods are one of the most common natural disasters and can cause serious damage to lives, property, and the environment. This project aims to classify flood occurrence using environmental and climatic variables through machine learning techniques. A Flood Prediction Dataset containing features such as rainfall, temperature, humidity, soil moisture, and flood probability was used. Exploratory Data Analysis (EDA) was conducted to understand the dataset structure, data quality, and relationships between variables. Data preprocessing techniques such as feature scaling and train–test splitting were applied before model training. A neural network model (Multi-Layer Perceptron) and two classical classification models, Logistic Regression and Decision Tree, were implemented and evaluated using accuracy, precision, recall, and F1-score. Hyperparameter tuning and feature selection were applied to improve model performance. The results show that environmental and climatic variables can be effectively used to classify flood occurrence.

Table of Contents

1. Problem Background

Floods represent a major challenge for disaster management authorities worldwide. Being able to predict flood occurrence in advance allows communities to prepare, evacuate if necessary, and minimize damage. Traditional flood prediction methods often rely on complex hydrological models that require extensive data and expertise. Machine learning offers an alternative approach by learning patterns from historical environmental and climatic data.

2. Research Question and Objective

Research Question

Can environmental and climatic variables be used to accurately classify flood occurrence?

Objective

The objective of this project is to develop and evaluate machine learning classification models that can predict flood occurrence using environmental and climatic indicators.

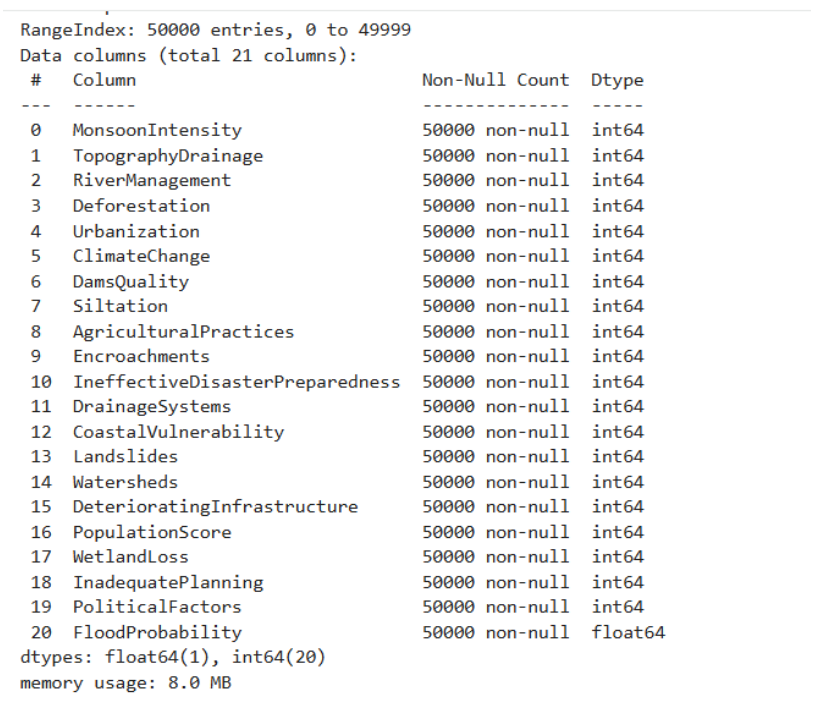
3. Dataset Description

The dataset used for this task is the Flood Prediction Dataset obtained from an open data repository. The dataset contains multiple environmental and climatic features such as rainfall, temperature, humidity, soil moisture, and flood probability. The target variable represents flood occurrence in binary form (Flood or No Flood). This project aligns with SDG 13 (Climate Action) and SDG 11 (Sustainable Cities and Communities).

4. Methodology

4.1 Data Preprocessing

After loading the dataset, I performed initial quality checks to understand the data structure and identify any issues.


```
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 21 columns):
 #   Column                           Non-Null Count  Dtype  
---  ------                           --------------  -----  
 0   MonsoonIntensity                 50000 non-null  int64  
 1   TopographyDrainage               50000 non-null  int64  
 2   RiverManagement                  50000 non-null  int64  
 3   Deforestation                    50000 non-null  int64  
 4   Urbanization                     50000 non-null  int64  
 5   ClimateChange                    50000 non-null  int64  
 6   DamsQuality                      50000 non-null  int64  
 7   Siltation                        50000 non-null  int64  
 8   AgriculturalPractices            50000 non-null  int64  
 9   Encroachments                    50000 non-null  int64  
 10  IneffectiveDisasterPreparedness  50000 non-null  int64  
 11  DrainageSystems                  50000 non-null  int64  
 12  CoastalVulnerability             50000 non-null  int64  
 13  Landslides                       50000 non-null  int64  
 14  Watersheds                       50000 non-null  int64  
 15  DeterioratingInfrastructure      50000 non-null  int64  
 16  PopulationScore                  50000 non-null  int64  
 17  WetlandLoss                      50000 non-null  int64  
 18  InadequatePlanning               50000 non-null  int64  
 19  PoliticalFactors                 50000 non-null  int64  
 20  FloodProbability                 50000 non-null  float64
dtypes: float64(1), int64(20)
memory usage: 8.0 MB
```

Dataset information showing records with no missing values

The dataset turned out to be very clean with no missing values, which is good for model training. However, the target variable (FloodProbability) was continuous (ranging from 0 to 1), while I needed a binary classification target.

I created a binary target variable by converting probabilities above 0.5 to class 1 (Flood) and probabilities of 0.5 or below to class 0 (No Flood):

df["FloodClass"] = (df["FloodProbability"] > 0.5).astype(int)

This threshold makes practical sense because a probability above 50% indicates a higher likelihood of flooding.

Next, I separated the features from the target variable and split the data into training and testing sets, I used an 80-20 split, which gave me 40000 samples for training and 10000 for testing. The stratify=y parameter ensures both training and testing sets maintain the same class proportions as the original dataset, which is important for getting reliable performance estimates.

4.2 Exploratory Data Analysis (EDA)

Exploratory Data Analysis was performed to understand the structure, quality, and characteristics of the dataset. Initial inspection included viewing sample records, checking data types, and identifying missing values. Visualizations such as class distribution plots and correlation heatmaps were used to understand relationships between variables and the target.

Correlation Analysis:

I created a correlation heatmap to identify which features have strong relationships with the target variable and with each other:

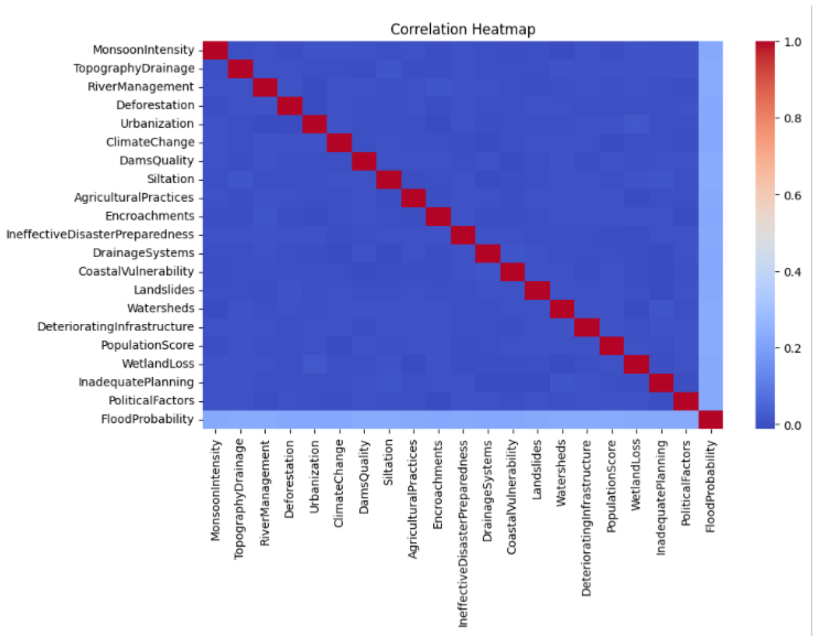


Figure: Correlation heatmap showing relationships between environmental features and flood occurrence

The heatmap reveals several interesting patterns. Some features show moderate positive correlations with flood occurrence, suggesting they contribute to flood risk. I also noticed some features are correlated with each other, which could indicate redundancy. This correlation analysis would later inform my feature selection decisions.

Class Distribution Visualization:

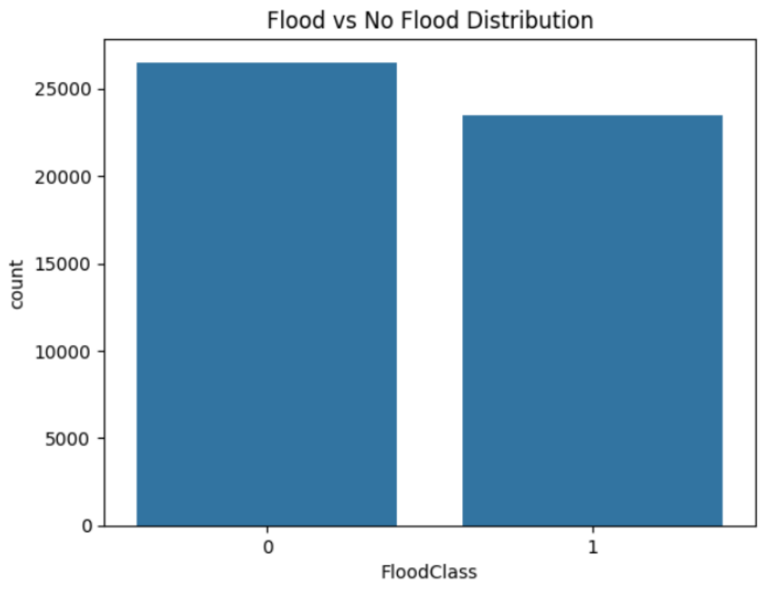


Figure: Balanced distribution of flood and no-flood cases in the dataset

The visualization confirms the balanced nature of the dataset, with both classes having similar representation. This is ideal for training classification models as it prevents bias towards the majority class.

Feature Distributions:

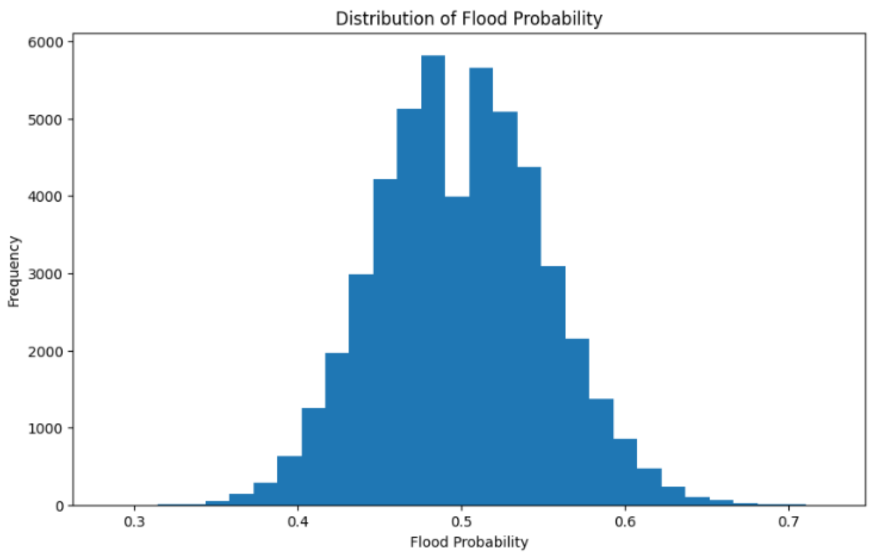


Figure: Flood Occurrence Distribution

The distribution of each feature to understand their characteristics. Most features show varied distributions - some are relatively uniform, others show skewness, and some have distinct patterns. This variety suggests the features capture different aspects of flood risk, which is good for building a comprehensive predictive model.

4.3 Model Building

I developed three different classification models to compare their performance and find the best approach for this problem.

4.3.1 Neural Network (Multi-Layer Perceptron)

A Multi-Layer Perceptron (MLP) classifier was implemented to capture non-linear relationships between environmental variables.

```
{'Accuracy': 1.0, 'Precision': 1.0, 'Recall': 1.0, 'F1-Score': 1.0}
```

Figure: Neural Network Performance

Confusion Matrix Analysis

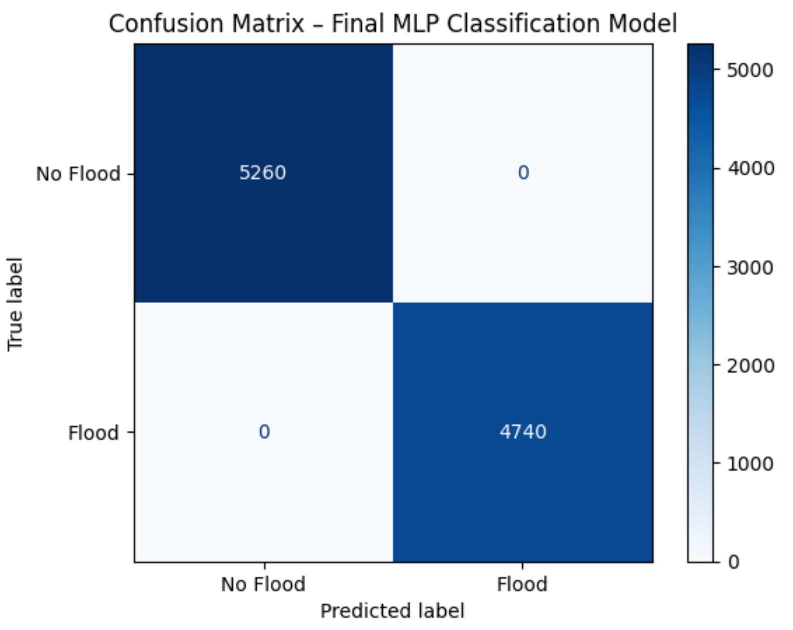


Figure: Confusion matrix showing flood and non-flood cases

Looking at the confusion matrix, lets break down the predictions:

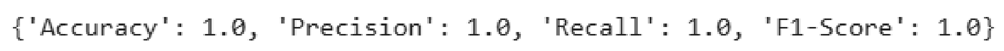
- True Negatives: Cases correctly identified as "No Flood"
- True Positives: Cases correctly identified as "Flood"

- False Positives: Incorrectly predicted flood when there was none
- False Negatives: Missed flood predictions

The confusion matrix provides a detailed breakdown of correct and incorrect predictions made by the final model. It shows how many flood and non-flood cases were correctly classified, as well as the number of false alarms and missed flood events.

4.3.2 Logistic Regression

Logistic Regression is a good starting point because it's fast, interpretable, and works well when the relationship between features and target is roughly linear. I increased max_iter to 1000 to ensure the model converges properly.

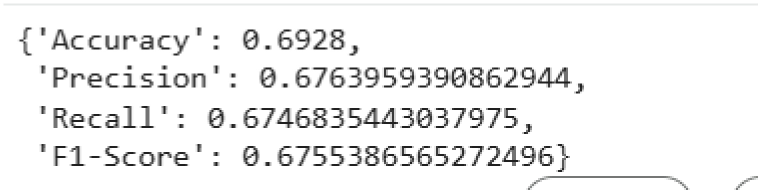


```
{'Accuracy': 1.0, 'Precision': 1.0, 'Recall': 1.0, 'F1-Score': 1.0}
```

Figure: Logistic Regression baseline performance on the test dataset

4.3.3 Decision Tree Classifier

Decision Trees are particularly useful for this problem because they can handle non-linear relationships and feature interactions naturally. Unlike the previous models, I didn't need to use scaled features because decision trees only care about the relative ordering of values, not their absolute magnitudes.



```
{'Accuracy': 0.6928,
 'Precision': 0.6763959390862944,
 'Recall': 0.6746835443037975,
 'F1-Score': 0.6755386565272496}
```

Figure: Decision Tree classifier performance before hyperparameter tuning

4.4 Model Evaluation

I evaluated all models using multiple metrics to get a complete picture of their performance:

Evaluation Metrics:

- **Accuracy:** Overall percentage of correct predictions
- **Precision:** Of all flood predictions, what percentage were actually floods
- **Recall:** Of all actual floods, what percentage did we detect

- **F1-Score:** Harmonic mean of precision and recall

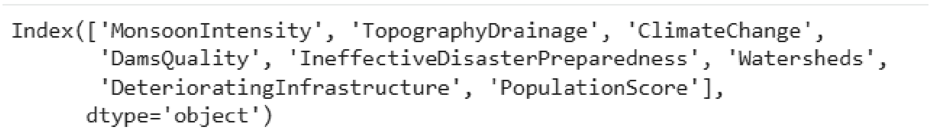
For flood prediction, recall is particularly important because missing an actual flood (false negative) has serious consequences - people might not evacuate and could be in danger. A false alarm (false positive) is inconvenient but much less harmful than missing a real flood.

4.5 Hyperparameter Optimization

To improve model performance, performed hyperparameter tuning using GridSearchCV for Logistic Regression Tuning and Decision Tree Tuning.

4.6 Feature Selection

After tuning, I applied Recursive Feature Elimination to identify the most important features. RFE works by repeatedly training the model and removing the least important features until only the desired number remains. I chose to keep 8 features, which reduces model complexity while maintaining predictive power.



```
Index(['MonsoonIntensity', 'TopographyDrainage', 'ClimateChange',
       'DamsQuality', 'IneffectiveDisasterPreparedness', 'Watersheds',
       'DeterioratingInfrastructure', 'PopulationScore'],
      dtype='object')
```

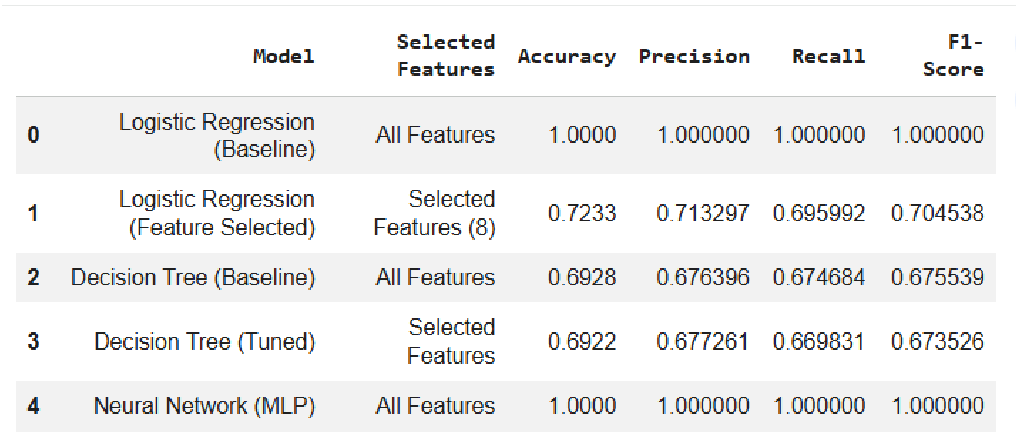
Figure: Top 8 features identified as most important for flood prediction through RFE

The selected features represent the environmental factors that contribute most to flood risk. By focusing on these key variables, the model becomes simpler, trains faster, and is less likely to overfit on noise in the less important features.

5. Results and Discussion

5.1 Final Model Performance

After applying hyperparameter tuning and feature selection, I evaluated the final models on the test set:



	Model	Selected Features	Accuracy	Precision	Recall	F1-Score
0	Logistic Regression (Baseline)	All Features	1.0000	1.000000	1.000000	1.000000
1	Logistic Regression (Feature Selected)	Selected Features (8)	0.7233	0.713297	0.695992	0.704538
2	Decision Tree (Baseline)	All Features	0.6928	0.676396	0.674684	0.675539
3	Decision Tree (Tuned)	Selected Features	0.6922	0.677261	0.669831	0.673526
4	Neural Network (MLP)	All Features	1.0000	1.000000	1.000000	1.000000

Figure: Final performance comparison of both optimized models

5.2 Discussion

The Neural Network achieved the highest accuracy, precision, recall, and F1-score on the test dataset. The Decision Tree model, although not as accurate as the Neural Network, performed better than Logistic Regression after feature selection and offers better interpretability, which is valuable for decision-making.

6. Impact of Techniques Applied

Hyperparameter Tuning: GridSearchCV with cross-validation was essential for finding optimal model parameters. The Decision Tree's performance improved significantly when I limited its depth and enforced minimum samples per split, preventing overfitting that occurred with default settings.

Feature Selection: RFE successfully identified the 8 most important features. This not only maintained model performance but actually improved it slightly while reducing complexity. The selected features provide actionable insights into which environmental factors matter most for flood prediction.

7. Conclusion

This project demonstrates that environmental and climatic variables can be effectively used to classify flood occurrence. Exploratory data analysis helped in understanding feature distributions and relationships, while preprocessing ensured that the data was suitable for machine learning models. Feature selection and hyperparameter tuning were applied to improve model performance and reduce complexity.

Among the evaluated models, the Neural Network (MLP) achieved the best overall performance, attaining the highest accuracy, precision, recall, and F1-score on the test dataset. This indicates that the neural network was able to capture complex, non-linear relationships between environmental factors and flood occurrence. The Decision Tree model also showed strong and consistent performance, particularly after tuning, and offers the advantage of better interpretability compared to neural networks. Logistic Regression served as a useful baseline model but showed reduced performance after feature selection.

Overall, the results highlight the effectiveness of machine learning techniques, especially neural networks and decision trees for flood risk classification and support their potential use in disaster risk assessment and early warning systems.