A FIELD PROJECT REPORT

on

"Flood Prediction Using Machine Learning"

Submitted

By

221FA04448
B. Bhanulatha

221FA04477
P. Tarak Ram

221FA04642
P. Naga Mounika

221FA04743
K. Siva Chari

Under the guidance of

Dr. S. Deva Kumar

Associate Professor, CSE

VIGNAN'S
FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH
(Deemed to be University) - Estd. u/s 3 of UGC Act 1956

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

VIGNAN'S FOUNDATION FOR SCIENCE, TECHNOLOGY AND RESEARCH

Deemed to be UNIVERSITY

Vadlamudi, Guntur.

ANDHRA PRADESH, INDIA, PIN-522213.

VIGNAN'S

FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH

(Deemed to be University) - Estd. u/s 3 of UGC Act 1956

CERTIFICATE

This is to certify that the Field Project entitled **"Flood Prediction Using Machine Learning"** that is being submitted by 221FA04448 (B. Bhanulatha), 221FA04477 (P. Tarakram), 221FA04642 (P. Naga Mounika)**,** 221FA04743 (K. Sivachari) for partial fulfilment of Field Project is a bonafide work carried out under the supervision of Dr. S. Deva Kumar, Associate Professor, Department of CSE.

Dr.S. Deva Kumar
Associate Professor, CSE

Dr. S. V. Phani Kumar
HOD,CSE

Dr.K.V. Krishna Kishore
Dean, SoCI

DECLARATION

We hereby declare that the Field Project entitled **"Flood Prediction Using Machine Learning"** that is being submitted by 221FA04448 (B.Bhanulatha), 221FA04477(P.TarakRam), 221FA04642(P. Naga Mounika)**,** 221FA04743(K. Sivachari) is being submitted by in partial fulfilment of Field Project course work. This is our original work, and this project has not formed the basis for the award of any degree. We have worked under the supervision of Dr. S. Deva Kumar Associate Professor, Department of CSE.

By

221FA04448 (Bhanulatha),

221FA04477 (Tarak Ram),

221FA04642 (Naga Mounika),

221FA04743 (Siva Chari)

Date:

ABSTRACT

India is subject to frequent natural disasters in the form of floods, resulting in significant loss of life and property. Accurate prediction of flood onset and progression in real time is critical to minimizing flood impacts. This Project focuses on a comparative study of different machine learning models for flood forecasting in India. Models analysed include K-nearest neighbour (KNN), support vector classification (SVC), decision tree classification. ML models, particularly ensemble algorithms like Random Forest and Gradient Boosting Machines (GBMs), Logistic Regression, Navie bayes Learning. These models enhance predictive accuracy and enable more localized and adaptive early warning systems when combined with regional hydrological data. The dataset provides a historical overview of monthly and annual rainfall patterns in the Kerala subdivision from 1901 to 2018, with a specific focus on flood occurrences. The dataset includes monthly rainfall data for each year (January through December) and an annual rainfall total, measured in millimeters. Additionally, the dataset records whether significant floods occurred during each respective year. This information supports analyses of rainfall variability and flood trends, aiding research into climate patterns, flood prediction, and water resource management in Kerala.

Keywords: Decision tree, Support Vector Machine (SVM), Logistic Regression, Navie bayes Learning, KNN.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1

INTRODUCTION

1. INTRODUCTION

1.1 What is Flood Prediction and how it causes?

Flood prediction, also known as flood forecasting, is the process of estimating when and where flooding will occur, as well as predicting the severity of the flood. This is done by analyzing various environmental, meteorological, and hydrological data, such as rainfall, river levels, soil moisture, and weather patterns. Accurate flood prediction systems are critical for providing early warnings, enabling communities, governments, and emergency services to take preventive actions to reduce the impact of floods on lives, property, and infrastructure.

Flood prediction typically involves:

1. **Meteorological Data Analysis:** Monitoring rainfall patterns, storm events, and weather conditions that can lead to flooding.
2. **Hydrological Modelling:** Using data from rivers, reservoirs, and groundwater levels to determine how much water is flowing into river systems and how close these systems are to overflowing.
3. **Geospatial Data:** Examining land use, terrain, and topography to identify areas more prone to flooding due to their geographic characteristics.

Advanced technologies such as machine learning (ML) and artificial intelligence (AI) are increasingly used to predict floods with higher accuracy by uncovering complex patterns in large datasets that involve meteorological and hydrological variables.

Causes of Floods:

Floods are caused by a variety of natural and human-made factors, often involving the interaction of multiple elements. The major causes include:

1. **Heavy Rainfall:** One of the most common causes, prolonged or intense rainfall overwhelms natural or man-made drainage systems, causing rivers and streams to overflow.
2. **River Overflow:** When water levels in rivers exceed their banks, it can cause riverine flooding. This is common in monsoon regions or during prolonged periods of rain.
3. **Snowmelt:** In regions with significant snowfall, rapid melting during warmer periods can release large volumes of water, which rivers and drainage systems may not be able to handle.

4. **Coastal Flooding:** Storm surges from tropical cyclones, hurricanes, or tsunamis can raise the sea level, flooding coastal regions. This is especially problematic in low-lying areas.
5. **Dam Breaks or Levee Failures:** Human-made structures like dams and levees control water flow, but their failure due to engineering faults, excessive water pressure, or poor maintenance can lead to catastrophic flooding downstream.
6. **Urbanization:** Increased urban development can lead to a reduction in natural ground absorption due to the expansion of impermeable surfaces like concrete and asphalt. This causes more water to run off into drainage systems, which may overflow, leading to flash floods.
7. **Deforestation:** The removal of forests reduces the land's ability to absorb rainfall, leading to increased runoff and a higher likelihood of flooding, particularly in hilly or mountainous regions.
8. **Climate Change:** Global warming is increasing the frequency and intensity of extreme weather events, such as heavy rainfall and storms, which in turn leads to more frequent flooding. Rising sea levels due to the melting of polar ice also contribute to higher risks of coastal floods.

Floods can be sudden, such as flash floods, or gradual, such as those resulting from prolonged rainfall or snowmelt. Accurate flood prediction can help mitigate these risks, giving people time to prepare or evacuate when necessary.

1.2 Consequences of Flood Prediction:

Flood prediction, when accurately implemented, can have a profound impact on society, the economy, and the environment. Predicting floods can significantly reduce the damage caused by these natural disasters, but the consequences of flood prediction, both positive and negative, must be considered in multiple contexts.

Positive Consequences of Flood Prediction:

1. **Saving Lives**: Accurate and timely flood prediction systems provide early warnings that can help evacuate people from high-risk areas before a flood occurs. This significantly reduces the risk of fatalities and injuries during flood events, particularly in highly vulnerable regions.
2. **Improved Disaster Preparedness and Response**: Flood predictions enable emergency services to coordinate and deploy resources more effectively. This includes pre-positioning

rescue teams, emergency supplies, and medical assistance in areas likely to be affected, thereby improving the efficiency of disaster response efforts.

3. **Enhanced Agricultural Planning**: In regions like Kerala, where agriculture plays a significant role in the economy, flood predictions allow farmers to prepare by securing crops, moving livestock, or adjusting planting and harvesting schedules. By mitigating the effects of floods, farmers can avoid large-scale crop destruction and reduce losses.
4. **Reduction in Economic Losses**: Flooding can cause significant economic disruption, from damaged infrastructure to lost productivity. Predicting floods helps minimize economic losses by allowing businesses and local governments to prepare contingency plans, temporarily halt operations, and safeguard economic assets.
5. **Environmental Protection**: Early flood predictions allow for the mitigation of environmental damage, such as erosion, landslides, and soil contamination. Additionally, efforts can be made to protect ecologically sensitive areas, such as wetlands, that play a crucial role in flood control by naturally absorbing excess water.

Negative Consequences:

1. **False Alarms and Public Panic**: Inaccurate or overly cautious flood predictions may lead to false alarms. These false alarms can cause unnecessary evacuations, disruption of daily life, and public panic. If people experience too many false alarms, they may start to distrust future warnings, increasing the risk of casualties when a real flood occurs.
2. **Economic and Social Disruption**: Flood predictions may lead to the temporary closure of businesses, schools, and other institutions, even when floods do not materialize. This can result in financial losses and disrupt normal life. Additionally, frequent evacuations or relocation of communities can strain local economies and infrastructure.

1.3 Background of Flood Prediction:

Flood prediction is a critical aspect of disaster management and plays an essential role in safeguarding communities and reducing the impact of floods. Flooding is caused by multiple factors, including heavy rainfall, river overflow, rapid snowmelt, dam failures, and coastal storms. In many regions, such as Kerala, India, flooding occurs frequently during the monsoon season due to intense rain and geographical features like rivers and low-lying areas.

Challenges of Flood Prediction

Despite significant advancements, flood prediction faces numerous challenges:

1. **Data Availability and Quality**: Accurate flood prediction relies on high-quality, comprehensive data, which may not always be available. In some regions, particularly in developing countries or remote areas, there are gaps in weather stations, river gauge networks, and real-time monitoring systems. Inconsistent or missing data makes it difficult to create accurate models or make reliable predictions.
2. **Uncertainty in Weather and Hydrological Models**: Predicting floods involves understanding complex weather patterns and how they interact with local geography and hydrology. While models have improved, weather is inherently unpredictable, and flood events may differ significantly based on small changes in conditions. For example, the timing and intensity of rainfall, combined with the characteristics of the land (e.g., slope, soil saturation), can introduce significant uncertainty in flood forecasts.

CHAPTER-2

LITERATURE SURVEY

2. LITERATURE SURVEY

2.1 Literature review

A literature survey is a systematic examination of existing research on a particular topic. It serves as the foundation for any scholarly investigation, offering insights into current knowledge, identifying research gaps, and providing context for new studies. By synthesizing and summarizing relevant literature, researchers can formulate precise research questions, build upon existing work, and avoid duplication. In essence, a literature survey is an essential tool for ensuring the validity and relevance of new research within the broader academic landscape.

2.2 Previous Studies on Flood Prediction

This project presents an Improved NARX model for predicting flood water levels 5 hours in advance, demonstrating superior performance over the original NNARX model. The model achieved high accuracy in forecasting, with significant improvements observed in prediction precision[1].This project developed an NNARX model to predict flood water levels 5 hours in advance using data from rivers in Terengganu, Malaysia. The model was trained, validated, and tested with real-time data, showing successful early flood prediction for the region[2].This project compares flood prediction models using MISO ARX and ARMAX structures for the Pahang River in Temerloh, Malaysia. Results showed that the ARMAX model outperformed ARX, with better prediction accuracy and lower RMSE values[3].This project presents a 5-hour flood prediction model for flood-prone areas in Kuala Lumpur using NNARX and its Improved Modelling technique. The results demonstrated that the Improved NNARX model accurately predicted flood water levels 5 hours in advance, providing early warning for evacuation[4].This project proposes a 3-hour flood prediction model using NNARX with improved modelling strategies for Klang River at Petaling Bridge, Kuala Lumpur. The model showed significant prediction accuracy, providing timely warnings for potential flood disasters in urban areas[5].This project presents a river water level prediction model using NNARX for flood monitoring in Kuala Lumpur. The results indicated that the NNARX model with a 4-hour prediction time achieved reliable and accurate flood forecasting[6].This project presents a flood prediction model for Bangladesh using the k-nearest neighbours (k-NN) algorithm and correlation coefficients for feature selection. The model achieved a high testing accuracy of 94.91%, with an average precision of 92.00% and recall of 91.00%[7].This project focuses on the use of Artificial Neural Networks (ANN), particularly

MLP and LSTM models, to predict reservoir water levels and inflows for optimizing reservoir operations. The models aim to minimize downstream impacts and improve flood evacuation planning based on accurate hydrological data[8].This project proposes a framework integrating temporal prediction models and spatial data to generate flood hazard maps. The models achieved high accuracy, with average MAPE values of 3.17% for hourly and 4.88% for daily predictions, and an F1-score of 81.50% in hazard map generation[9].It is clear that the hybrid approach predicts data for more years in the future with an accuracy level as high as 89%[10].This project developed an NNARX flood prediction model for Pahang, Malaysia, using real-time data from flood events. The model successfully predicted flood water levels in advance, providing potential early warnings for at-risk residents[11].This project examines various flood prediction techniques for the Mumbai region, including statistical, hydrological, and AI-based models, evaluating their accuracy, precision, and limitations. It discusses the challenges of predicting floods in Mumbai's urban environment and suggests future research directions for improving prediction and preparedness[12].This project introduces a Radial Basis Function Neural Network (RBFNN) for predicting flood water levels at Kelang River, demonstrating its effectiveness in handling complex, nonlinear flood data. The addition of an Inverse Model to the RBFNN significantly improved prediction accuracy[13].This project applies cellular automata algorithms to predict and model flood spreading in Bojonegoro, Indonesia, using elevation, soil type, river mapping, volume changes, and rainfall data. The results are visualized in 2D images to aid in understanding and managing flood distribution[14].This project introduces a flood prediction model using NNARX and a hybrid NNARX with Extended Kalman Filter (EKF) for Klang River, Malaysia. The NNARX-EKF hybrid model significantly outperformed the NNARX model in predicting flood water levels using real-time SCADA data[15].This project proposes a Nonlinear Autoregressive with Exogenous Input (NARX) neural network for flood risk assessment in Sabah, Malaysia, using hydrological data from Wariu and Padas Rivers. The NARX model, trained with the Levenberg-Marquardt algorithm, provides reliable water level predictions up to five days ahead, with an R^2 value exceeding 0.85[16]. This project introduces a prioritized sampling method to enhance a KD-based neural network for nowcasting pluvial floods, improving accuracy in minor rainfall patterns. The method achieved a 96% reduction in error rate, lowering the average prediction error by 37 cm per mesh compared to random sampling[17].The project proposes a hybrid model, HMM-ML, combining Hidden Markov Models with various machine learning techniques to improve flood forecasting in Kozhikode, Kerala, India. The model integrates rainfall and temperature data to enhance prediction accuracy during the high-rainfall period from June to August[18].This project presents a hybrid ANFIS-HHO model for forecasting river floods in the Barak River basin, India, combining an adaptive neuro-fuzzy inference system with Harris Hawks Optimization. The model demonstrated superior performance with an NSE of 0.9885 and RMSE of 61.87, addressing issues of overfitting and underfitting in traditional ANFIS models[19].Accurate flood monitoring is facilitated by the development and implementation of a real-time flood detection and forecasting system[20].

CHAPTER-3

PROPOSED SYSTEM

3. PROPOSED SYSTEM

A. Dataset: The dataset includes 13 features such as monthly rainfall (January to December), annual rainfall, and a binary target variable "FLOODS" (YES/NO), indicating flood occurrences.

B. Data Preprocessing: Missing data was handled using mean or median imputation. Rainfall values were scaled using Min-Max scaling or standardization. Class imbalance was managed with SMOTE or class-weighted models to improve prediction.

C. Exploratory Data Analysis (EDA): EDA identified relationships between rainfall and floods through correlation analysis, heatmaps, and scatter plots. Feature selection used Recursive Feature Elimination (RFE) and Principal Component Analysis (PCA) to reduce dimensionality.

D. Model Development:

Supervised algorithms tested include:

Logistic Regression: For interpretable flood prediction.

Random Forest: To handle rainfall data and provide feature importance.

Gradient Boosting (XGBoost, LightGBM): For boosting weak learners.

Support Vector Machines (SVM): To classify non-linear patterns.

Neural Networks (MLP): For complex feature interactions.

E. Model Training: Data was split into training (70%), validation (15%), and test (15%) sets. K-fold cross-validation (k = 5) was applied, and hyperparameter tuning was done via grid search or random search.

F. Model Evaluation: Models were evaluated using accuracy, precision, recall, F1-score, and ROC-AUC. Focus was placed on minimizing false negatives for reliable flood warnings.

G. Model Interpretation: Random Forest and Gradient Boosting feature importance were analyzed. SHAP or LIME methods provided interpretability and transparency in flood predictions.

H. Final Model Selection and Testing: The best-performing model was selected based on validation metrics, balancing sensitivity and specificity. It was tested on unseen data for reliable flood prediction.

I. Deployment and Continuous Improvement: The model was deployed as a decision-support tool, with potential integration into a web-based platform. Continuous monitoring and real-time updates were planned to enhance prediction accuracy.

J. Ethical Considerations: Data privacy and security were ensured in compliance with GDPR. Bias mitigation was monitored to ensure fair performance across regions.

3.1 Input dataset:

The dataset contains a number of features that could influence or indicate flood occurrences and focuses on rainfall patterns. It includes year-level data with various features related to monthly and annual rainfall. A distinct "Year" is used to identify each observation. The 13 columns in the dataset describe monthly rainfall data (from January to December), annual total rainfall, and the occurrence of floods (YES/NO) as the target variable.

3.1.1Detailed Features of the Dataset:

Year: The year of observation.
JAN: The amount of rainfall in January (in millimeters).
FEB: The amount of rainfall in February (in millimeters).
MAR: The amount of rainfall in March (in millimeters).
APR: The amount of rainfall in April (in millimeters).
MAY: The amount of rainfall in May (in millimeters).
JUN: The amount of rainfall in June (in millimeters).
JUL: The amount of rainfall in July (in millimeters).
AUG: The amount of rainfall in August (in millimeters).
SEP: The amount of rainfall in September (in millimeters).
OCT: The amount of rainfall in October (in millimeters).
NOV: The amount of rainfall in November (in millimeters).
DEC: The amount of rainfall in December (in millimeters).
ANNUAL RAINFALL: The total amount of rainfall for the year (in millimeters).
FLOODS: Indicates whether a flood occurred that year ('YES' or 'NO').

3.2Data Pre-processing

Data pre-processing is the essential process of preparing raw data for analysis and modelling by cleaning, transforming, and structuring it to enhance data quality and utility. It involves tasks like handling missing values, correcting errors, encoding features, and scaling data to ensure it's in an optimal form for further analysis. It encompasses a range of operations and transformations designed to refine raw data, ensuring that it is clean, structured, and amenity subsequent analysis. This process is driven by its manifold significance in data science and analysis.

1)Handling Missing Values

2)Dropping Unnecessary Columns

3)Encoding Categorical Features

4)Feature scaling

5)Feature selection

6)Data splitting

7)Outlier detection and Handling

Discuss on various preprocessing techniques that we have applied for our project

Columns such as Index, Year, and ANNUAL RAINFALL were among those eliminated.

Reason: These columns were removed to simplify the dataset and reduce noise for the model, as they were deemed unnecessary or less relevant for predicting flood occurrences based on monthly rainfall data.

3.3Model Building

Using the cleaned Kerala flood dataset, the model development portion aimed to predict whether floods would occur (YES or NO). The Naive Bayes classifier was selected for this task due to its simplicity and effectiveness in classification problems, especially when the features are assumed to be independent.

Preparing Data

The dataset was first divided into two parts: features (X) and the target variable (y). X contained all relevant rainfall measurements (such as monthly rainfall data from JAN to DEC and total annual rainfall), while y was the target variable, "FLOODS," indicating whether floods occurred (YES or NO).

To ensure all features were on the same scale, feature scaling was applied using standardization. This step was crucial to prevent features with higher values from dominating the model's learning process.

Data Division

The dataset was split into a training set (70%) and a testing set (30%). This separation was done to ensure that the model could learn from the training data and then be evaluated on unseen test data. This division provided a reliable estimate of the model's performance on new data.

Training the Model

The training data was used to train a Gaussian Naive Bayes classifier. The model calculated the probability of flood occurrence for each record, based on the features, and selected the most likely class (YES or NO). To handle any feature values missing from the training data and avoid zero probabilities, a smoothing parameter was applied.

Forecasting and Assessment

Once the model was trained, it was used to predict whether floods would occur in the test set. Both training and testing accuracies were calculated to evaluate the model's performance. Training accuracy measured how well the model learned from the training data, while testing accuracy provided insight into how well the model generalized to new, unseen data.

To further assess the model, key metrics such as accuracy, precision, recall, and the F1-score were calculated:

- **Accuracy** measured the overall performance of the model.
- **Precision** reflected the number of predicted flood occurrences (YES) that were correct.
- **Recall** indicated how well the model captured all actual flood occurrences.
- **F1-score** balanced precision and recall, especially useful when there is class imbalance (such as more NO than YES labels).

A confusion matrix was generated to visualize the number of correct and incorrect predictions for each class (YES and NO). This provided a clear view of the model's strengths and areas for improvement.

Conclusion

The Naive Bayes classifier showed promising results, with a good balance between training and testing accuracy. According to evaluation metrics (accuracy, precision, recall, and F1-score), the model demonstrated reasonable performance in predicting flood occurrences based on rainfall data.

The confusion matrix highlighted potential areas for improvement, such as instances where the model confused borderline cases between flood occurrence (YES) and non-occurrence (NO).

3.4 Methodology of the system

A. Architecture of the System

The proposed system architecture for predicting flood occurrence based on rainfall data involves several interconnected steps, including data collection, preprocessing, feature extraction, model training, and classification. The structure includes:

- **Input Layer**: Collecting rainfall data from various months (JAN to DEC) and total annual rainfall.
- **Preprocessing Layer**: Cleaning and transforming the data to ensure it's ready for model training.
- **Feature Extraction Layer**: Selecting the most relevant rainfall-related features for effective flood prediction.
- **Classifier**: Using a machine learning algorithm to predict the occurrence of floods (YES or NO).
- **Output Layer**: Displaying the classification result based on the input rainfall data.

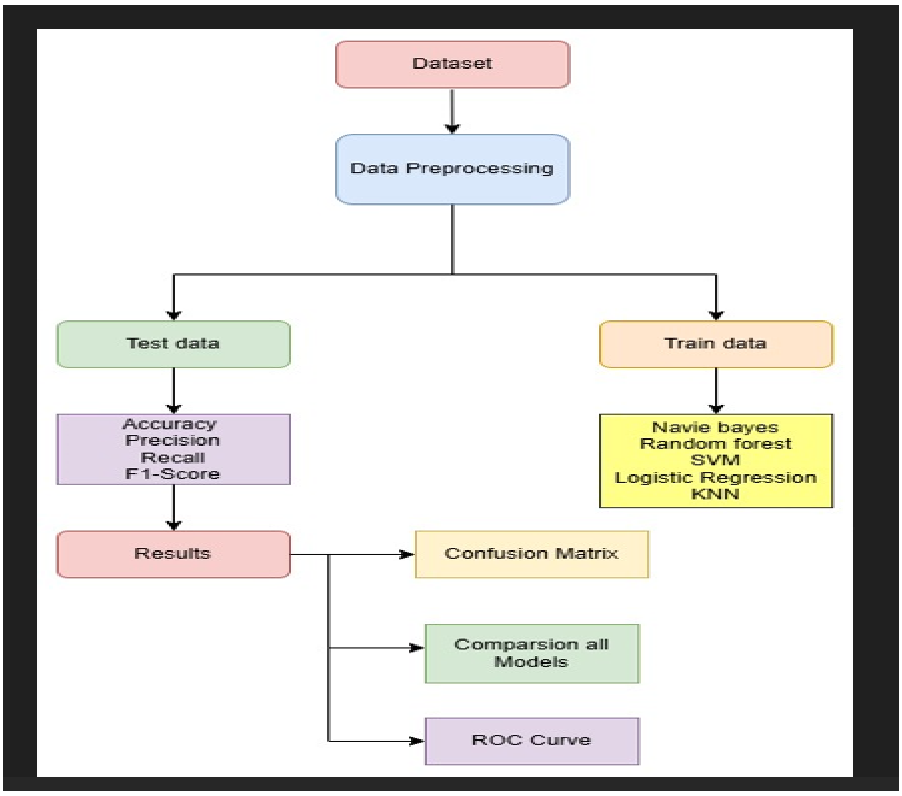


Figure 1. Architecture of the proposed System

B. Training and Preprocessing of Data

Preparing the data is a crucial step to ensure it's suitable for machine learning models. The following preprocessing methods were applied:

- **Data Cleaning**: Removing irrelevant columns such as "SUBDIVISION" and "YEAR" that do not significantly contribute to flood classification.
- **Label Encoding**: The target variable "FLOODS" (YES or NO) was label-encoded into numerical values to be compatible with machine learning algorithms.
- **Feature Scaling**: Standardizing the dataset to ensure that all rainfall features contribute equally to the model's learning process.
- **Data Splitting**: The dataset was divided into training (70%) and testing (30%) sets to ensure the model's performance is tested on unseen data.

```
SUBDIVISION         0
YEAR                0
JAN                 0
FEB                 0
MAR                 0
APR                 0
MAY                 0
JUN                 0
JUL                 0
AUG                 0
SEP                 0
OCT                 0
NOV                 0
DEC                 0
 ANNUAL RAINFALL    0
ClimateChange       0
FLOODS              0
dtype: int64
```

Figure 2. Key Features in the dataset after Pre-Processing

C. Extraction of Features

Feature extraction involves selecting and transforming input data into a smaller set of relevant features. In this case, monthly rainfall data (JAN to DEC) and the total annual rainfall were used as the main features to predict flood occurrence. By focusing on the features most related to flood risk, this process enhances the model's accuracy.

D. Naive Bayes Classifier

The Naive Bayes classifier was chosen for this task due to its simplicity and effectiveness in handling classification problems. The Gaussian Naive Bayes variant was used, as it performs well with continuous data like monthly and annual rainfall measurements. The model computes the probabilities for each class (flood or no flood) based on the features and makes predictions using maximum likelihood estimation.

E. Classification

The classification task aims to predict whether floods occurred (YES or NO) based on the extracted rainfall features and the trained Naive Bayes model. The pre-processed dataset was used for model training, and the test data was used to evaluate the model. Key metrics such as accuracy, precision, recall, and F1-score were calculated to assess the model's performance. The confusion matrix further showed the model's ability to distinguish between flood and no-flood cases.

F. Results

The system's output is a prediction of whether floods occurred based on the input rainfall data. After training, the system can predict the flood occurrence (YES or NO) for new rainfall data. The predictions can be used by government bodies and disaster management teams to anticipate flood risks. The system's performance is measured by its accuracy, precision, recall, and F1-score, demonstrating its potential for real-world flood prediction.

3.5 Model Evaluation

Several important criteria were used to assess the Naive Bayes model's ability to predict whether floods would occur (YES or NO). The goal was to evaluate the model's capacity to generalize to new data and generate accurate predictions across the two classes. The performance was assessed using the following metrics:

A. Accuracy of Training and Testing

Accuracy is a critical measure of how effectively the model classifies the target variable. Both **training** and **testing** accuracies were calculated to understand how well the model fit the training data and its ability to generalize to unseen data:

- **Training Accuracy**: This measures how well the model learned from the training set. A high training accuracy indicates that the model effectively captured patterns in the training data.
- **Testing Accuracy**: This measures how well the model performs on the test set, which represents new, unseen data. It indicates the model's generalization capability.

Balanced training and testing accuracy indicate that the model is neither overfitting (memorizing training data) nor underfitting (failing to recognize patterns in the data). A well-performing model will have both high training and testing accuracy, with only a slight difference between the two.

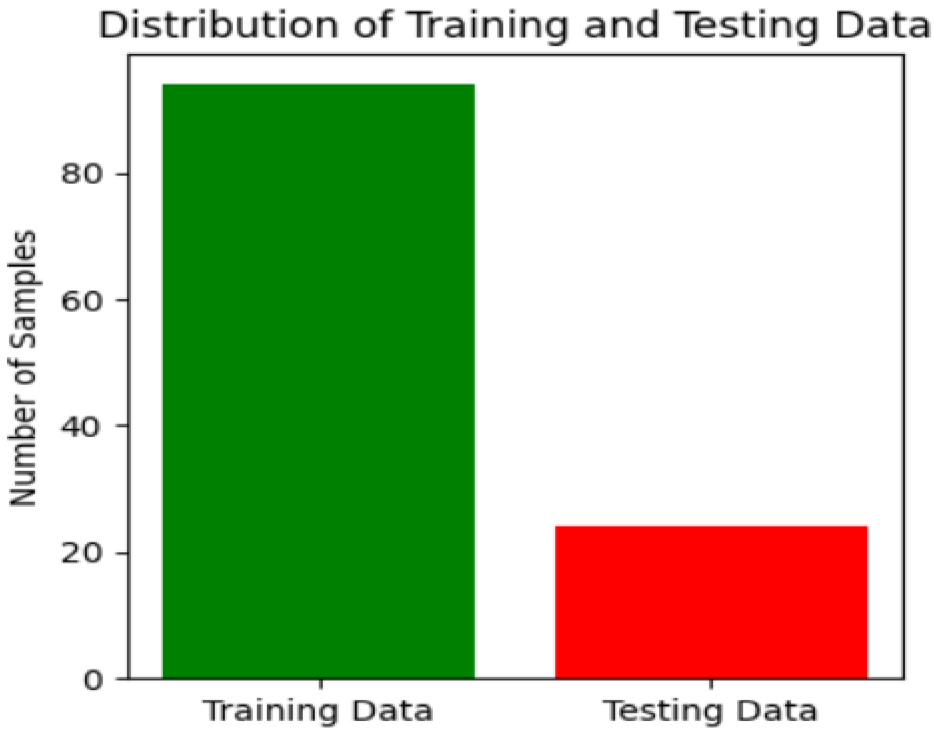


Figure 3. Training Vs Testing

B. Confusion Matrix

In the context of predicting flood occurrences using the Kerala rainfall dataset, the confusion matrix provides a detailed analysis of the model's classification performance. It shows the true positives, false positives, true negatives, and false negatives for the two classes (YES for floods and NO for no floods). The confusion matrix helped in understanding:

- **Correct Classifications**: How often the model correctly predicted flood occurrences (YES) and no floods (NO).
- **Misclassifications**: Instances where the model incorrectly predicted floods when there were none (false positives) or failed to predict floods when they occurred (false negatives).

The confusion matrix offers insights into:

- **True Positives (TP)**: Correct predictions of floods (YES).
- **True Negatives (TN)**: Correct predictions of no floods (NO).
- **False Positives (FP)**: Incorrectly predicting floods when none occurred.
- **False Negatives (FN)**: Incorrectly predicting no floods when floods occurred.

This matrix helps identify specific model weaknesses, such as class imbalances or difficulty in distinguishing between certain cases (for instance, borderline rainfall patterns where floods may or may not occur). By analyzing the confusion matrix, it is easier to see where the model needs improvement in accurately distinguishing between flood (YES) and no-flood (NO) scenarios.

C. Accuracy

Accuracy is defined as the ratio of correctly predicted instances (including both true positives and true negatives) to the total number of instances. It provides an overall measure of the model's performance. However, in the case of an **unbalanced dataset**, accuracy alone may be misleading, as it might not capture the distribution between flood (YES) and no-flood (NO) classes. Here, accuracy serves as an initial benchmark for model performance but should be complemented by other metrics.

D. Precision

Precision is the percentage of correct positive predictions made by the model. For the flood prediction task, precision measures how many of the instances predicted as "flood" (YES) were actual floods. It is particularly important when **false positives** (predicting floods when none occur) carry a significant cost, such as triggering unnecessary alarms or resources. High precision reduces false alarms in flood prediction.

E. Recall

Recall, also known as **sensitivity** or **true positive rate**, is the proportion of actual positive cases (flood occurrences) that the model correctly identified. It measures how well the model detects true flood cases and reduces the number of missed floods (false negatives). A high recall is essential when missing actual flood predictions can have serious consequences, like failing to warn people about flood risks.

F. F1-Score

The F1-score is the harmonic mean of precision and recall, providing a balanced measure that accounts for both false positives and false negatives. In flood prediction, the F1-score is particularly useful when the classes (YES or NO) are imbalanced or when **both precision and recall are equally important**. A high F1-score indicates that the model performs well in accurately classifying flood events while balancing the trade-offs between missed predictions and false alarms.

G. Outcomes of Performance

The following insights were derived from the performance metrics of the Naive Bayes flood prediction model:

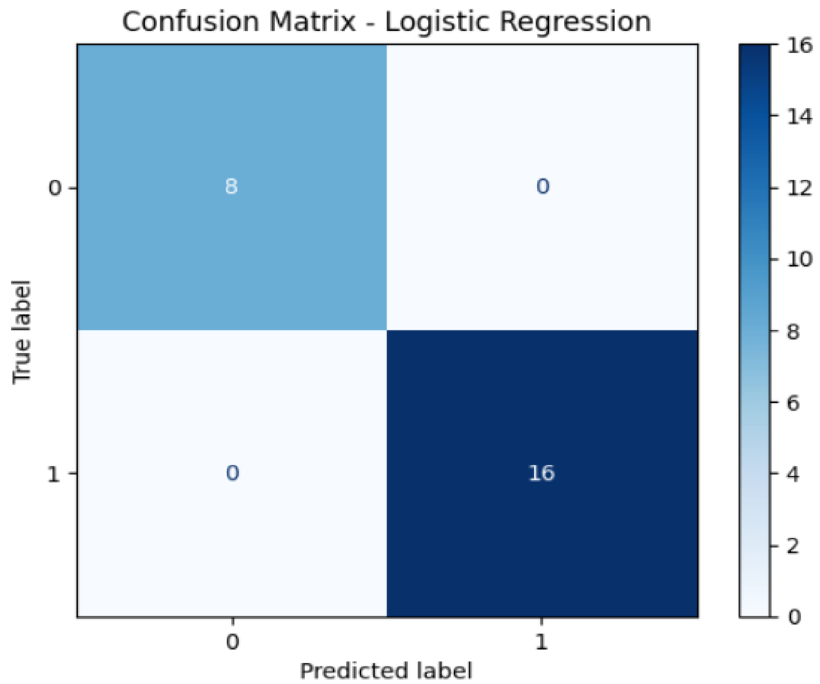
- **Training Accuracy**: Demonstrates how well the model learned from the training dataset, showing its ability to capture patterns in rainfall data.
- **Testing Accuracy**: Indicates how well the model generalizes to unseen data, giving an understanding of how it might perform in real-world flood predictions.

- **Precision and Recall**: These metrics help evaluate the model's capability to correctly predict flood occurrences while minimizing false alarms and missed events.
- **F1-Score**: Provides a single metric to gauge the model's overall performance, balancing both precision and recall to ensure that the flood prediction model is both accurate and reliable.

Individual Model Performance:

Logistic Regression:

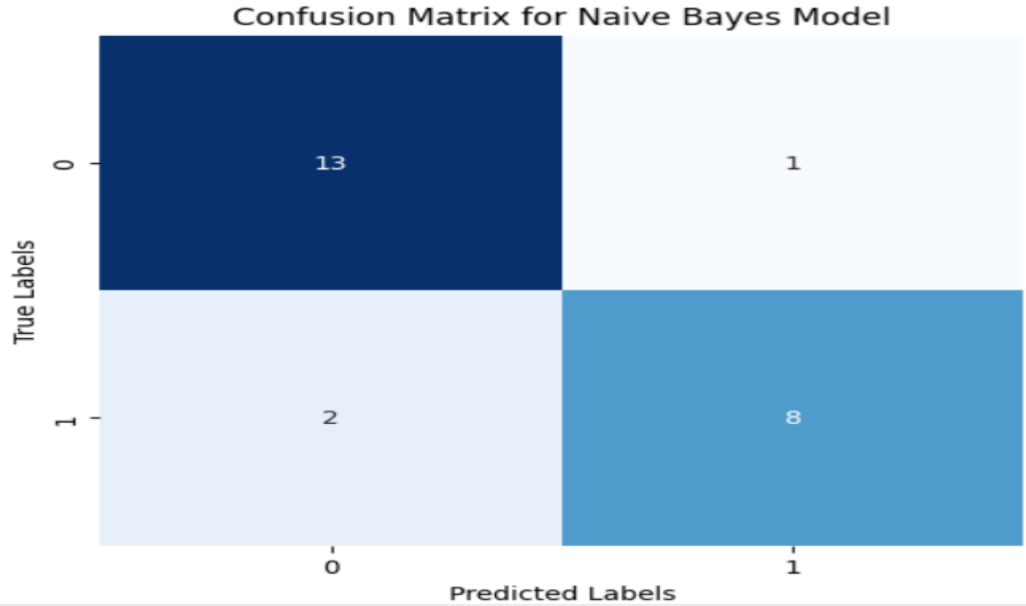
With a maximum of 1000 iterations to ensure convergence, Logistic Regression produced competitive results in terms of accuracy, precision, recall, and F1-score.



Naive Bayes:

The Naive Bayes classifier performed well, particularly in high-dimensional data, yielding decent accuracy despite some assumptions about feature independence.

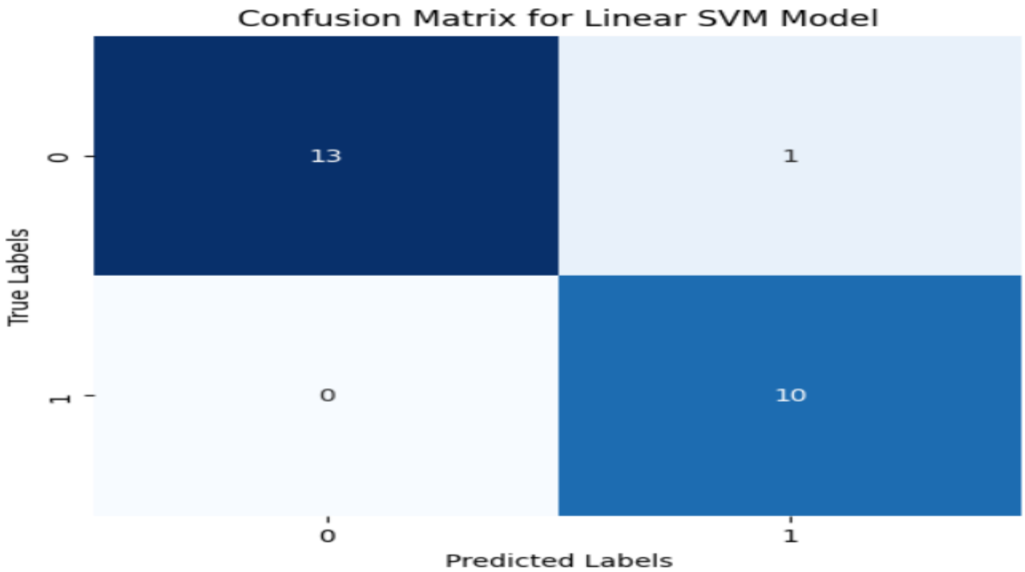
```
The accuracy score achieved using Naive Bayes is: 87.5 %
The precision score achieved using Naive Bayes is: 87.59 %
The recall score achieved using Naive Bayes is: 87.5 %
The F1 score achieved using Naive Bayes is: 87.39 %
Confusion Matrix:
 [[13  1]
 [ 2  8]]
```



Support Vector Machine (SVM):

Probability estimates were enabled during training, which facilitated detailed performance assessments. SVM showed strong performance, especially in precision and recall metrics.

```
The accuracy score achieved using Linear SVM is: 95.83 %
The recall score achieved using Linear SVM is: 100.0 %
The precision score achieved using Linear SVM is: 90.91 %
The F1 score achieved using Linear SVM is: 95.24 %
```



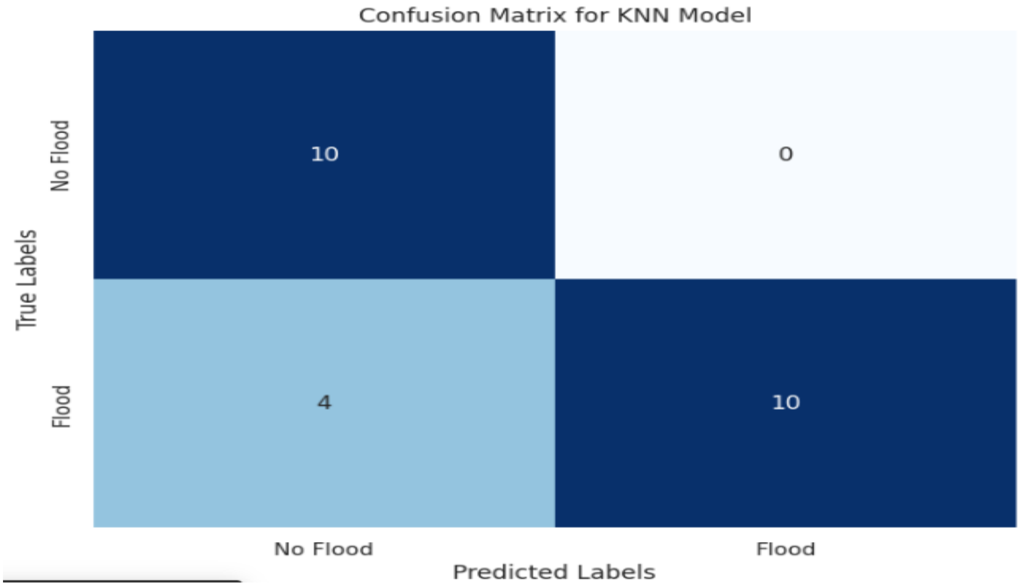
KNN(k-Nearest Neighbour):

```
KNN Model Accuracy: 0.8333333333333334
KNN Model Precision: 0.8809523809523809
KNN Model Recall: 0.8333333333333334
KNN Model F1 Score: 0.8333333333333334

Classification Report:
              precision    recall  f1-score   support

          NO       0.71      1.00      0.83        10
         YES       1.00      0.71      0.83        14

    accuracy                           0.83        24
   macro avg       0.86      0.86      0.83        24
weighted avg       0.88      0.83      0.83        24
```



Decision Tree:

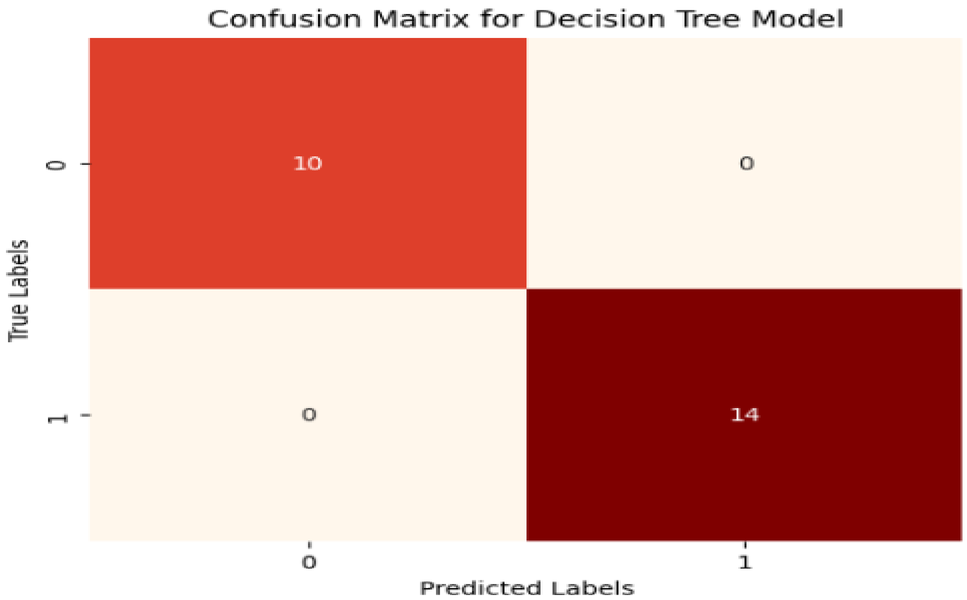


Figure 4. Model Building Process

3.5 Model Evaluation

Model evaluation is a critical aspect of any machine learning project. It involves assessing the performance and accuracy of a trained model on new, unseen data. This step is essential for several reasons such as:

i. **Quality Assurance**: Model evaluation helps ensure that the model is capable of making accurate predictions when exposed to real-world data. It acts as a quality control mechanism to validate the model's generalization ability.

ii. **Comparing Models**: Model evaluation allows for the comparison of multiple models to identify the best-performing one. It helps data scientists and stakeholders make informed decisions about which model to deploy.

iii. **Fine-Tuning**: The evaluation process can reveal areas where the model performs poorly. This information is valuable for refining the model, making it more robust, and addressing its limitations.

iv. **Business Decision Support**: In practical applications, model performance impacts critical business decisions. A well-evaluated model provides confidence to stakeholders, leading to better decision-making.

v. **Model Deployment**: A thoroughly evaluated model is more likely to be deployed in production systems. It instils trust in the model's predictions, which is essential in real-world applications.

3.6 Constraints

1. Data Quality and Availability: Ensuring a high-quality dataset is essential for predicting rainfall accurately. In the Kerala dataset, some months have missing rainfall values, which, if unaddressed, could introduce bias. Implementing effective imputation techniques or choosing models that handle missing data well can mitigate this risk. A comprehensive dataset with extensive historical data is also critical, as limited data can affect the model's predictive power.

2. Model Complexity and Overfitting: Given that ensemble methods like Gradient Boosting Machine (GBM) and Random Forests performed well, balancing model complexity with generalizability is necessary. Overly complex models risk overfitting to Kerala's specific climate, which might lead to reduced accuracy on data from different years or regions. Applying cross-validation techniques and tuning hyperparameters carefully can help achieve this balance.
3. Feature Engineering: The Kerala dataset includes monthly rainfall data and annual totals, which provide a solid foundation for feature engineering. Additional features, such as humidity, temperature, or regional topographical data, could improve prediction accuracy by capturing seasonal patterns and other environmental factors. Leveraging temporal dynamics and domain expertise can make predictions more robust.
4. Environmental Variability: Changing climate patterns in Kerala, driven by phenomena such as monsoon variations and warming trends, may alter predictor relationships over time. To maintain reliability, the model may require periodic retraining to adapt to these environmental changes.
5. Computational Resources: Machine learning models, especially ensemble and deep learning methods, require significant computational power to process and train on large datasets. Since the Kerala dataset spans over a century, adequate computational resources are essential to manage both training time and memory requirements, ensuring the model is trained efficiently without compromising performance.

These considerations not only guide the practical application of machine learning on the Kerala dataset but also set the groundwork for expanding the model’s utility, addressing real-world challenges like climate variability and resource constraints.

3.7 Ethical Considerations:

Bias and Fairness:

- Importance: Flood prediction models can be biased if based on limited historical data that doesn’t reflect recent changes in weather patterns, such as climate change. Fairness is critical to avoid geographic biases, ensuring that the model performs equitably across all areas in Kerala, especially since some regions may be more vulnerable to flooding than others.
- Mitigation: Regularly update the dataset with recent records and incorporate climate trend data to avoid historical bias. Testing model accuracy across diverse regions within Kerala can ensure fair performance.

Discrimination:

- Importance: Flood prediction affects a wide array of socio-economic groups. Ensuring the model's predictions are not biased toward wealthier or more urbanized regions helps to provide equal protection to all communities, particularly rural areas that may have fewer resources for flood preparedness.
- Mitigation: Avoid designing features that inherently favor one region over another, and incorporate spatial data that reflects socio-economic vulnerabilities.

Privacy and Security:

- Importance: Though this project may not involve personal data, flood prediction systems often integrate external data sources, like social, agricultural, and economic data, to improve prediction accuracy. Ensuring that any such data is anonymized and secure is essential to maintain public trust.
- Mitigation: Implement data security best practices, even if personal information is not directly used, and ensure compliance with data protection regulations for any additional integrated data.

Transparency and Accountability:

- Importance: Flood prediction models affect decision-makers in areas such as disaster response, urban planning, and agriculture. Transparency in model decisions builds trust and allows stakeholders to make informed decisions based on the predictions.
- Mitigation: Provide clear documentation, including feature importance, model architecture, and prediction confidence. Offering interpretable outputs and accessible explanations ensures that non-technical stakeholders can use the model effectively.

Responsible Use of AI:

- Misinformation: Misinterpreted predictions could lead to misinformation, causing unnecessary alarm or, conversely, a lack of preparation in vulnerable regions.
  - Mitigation: Clearly communicate predictions and associated uncertainties, distinguishing between high-risk and low-risk alerts.
- Automation of Decision-Making: Decisions like evacuation or flood defenses should incorporate human oversight rather than relying solely on automated model outputs.
  - Mitigation: Use the model as an advisory tool, enabling human experts to weigh the predictions alongside other considerations.

Environmental Impact:

- Importance: Flood prediction models, especially those using ensemble methods or deep learning, can be computationally intensive, consuming significant energy.
- Mitigation: Optimize models for energy efficiency, and use energy-conscious hardware.

Consider the environmental impact in light of Kerala’s climate-sensitive ecosystem.

Informed Consent:

- Importance: If any personal or sensitive data is used in the future to enhance flood prediction accuracy (such as social vulnerability data), informed consent becomes essential to protect individuals' privacy and to build public trust in the model’s integrity.
- Mitigation: Where personal data may be indirectly involved, ensure subjects are informed about the data usage and potential impacts, especially if predictions are to be shared publicly or used for governmental planning.

CHAPTER-4

IMPLEMENTATION

4.Implementation

4.1 Environment Setup

Install necessary libraries:

scikit-learn for machine learning algorithms.

pandas and numpy for data handling and preprocessing.

matplotlib and seaborn for data visualization.

Load the rainfall dataset into a pandas DataFrame.

Set up the working environment by ensuring all dependencies are installed and the dataset is ready for preprocessing and training.

Google Colab's virtual environment provided sufficient computational power for efficient data processing and model training, ensuring seamless execution of the Flood Prediction System.

4.2 Sample code for Preprocessing and model Training & Testing

1. Data Preprocessing

This part of the code involves handling missing values, encoding categorical features, and scaling numerical features.

```
# Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.impute import SimpleImputer

# Load the dataset
data = pd.read_csv('Kerala.csv')

# Checking for missing values and handling them
imputer = SimpleImputer(strategy='mean')
data['feature_with_missing_values']
=imputer.fit_transform(data[['feature_with_missing_values']])

# Encoding categorical variables
```

```
label_encoder = LabelEncoder()
data['categorical_feature'] = label_encoder.fit_transform(data['categorical_feature'])

# Splitting data into features (X) and target (y)
X = data.drop(columns=['target_column'])  # replace 'target_column' with your target variable name
y = data['target_column']


# Split the dataset into training and testing sets (80% training, 20% testing)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Feature scaling
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

2.Model Training

This part demonstrates training several machine learning models like Logistic Regression, Decision Tree, Naive Bayes, SVM, Gradient Boosting and KNN.

```
# Import machine learning models
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.neighbors import KNeighborsClassifier

# Instantiate the models
logistic_regression = LogisticRegression(random_state=42)
```

```
decision_tree = DecisionTreeClassifier(random_state=42)
naive_bayes = GaussianNB()
svm_model = SVC(random_state=42)
gradient_boosting = GradientBoostingClassifier(random_state=42)
knn = KNeighborsClassifier()

# Train the models
logistic_regression.fit(X_train, y_train)
decision_tree.fit(X_train, y_train)
naive_bayes.fit(X_train, y_train)
svm_model.fit(X_train, y_train)
gradient_boosting.fit(X_train, y_train)
knn.fit(X_train, y_train)
```

3. Model Testing and Evaluation

After training, evaluate the models' performance using metrics such as accuracy, precision, recall, and F1-score.

```
# Import metrics for evaluation
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix

# Function to evaluate models
def evaluate_model(y_test, y_pred, model_name):
    print(f"Results for {model_name}:")
    print(f"Accuracy: {accuracy_score(y_test, y_pred):.4f}")
    print(f"Precision: {precision_score(y_test, y_pred, average='macro'):.4f}")
    print(f"Recall: {recall_score(y_test, y_pred, average='macro'):.4f}")
    print(f"F1-Score: {f1_score(y_test, y_pred, average=‘macro'):.4f}")

# List of models and their names
models = [
    (logistic_regression, "Logistic Regression"),
```

```
    (decision_tree, "Decision Tree"),
    (naive_bayes, "Naive Bayes"),
    (svm_model, "SVM"),
    (gradient_boosting, "Gradient Boosting"),
    (knn, "K-Nearest Neighbors")
]

# Loop through the models, make predictions, and evaluate
for model, model_name in models:
    y_pred = model.predict(X_test)
    evaluate_model(y_test, y_pred, model_name)
```

Explanation:

Data Preprocessing:

Missing values are handled using SimpleImputer.

Categorical variables are encoded using LabelEncoder.

Features are scaled using StandardScaler.

The dataset is split into training and testing sets.

Model Training:

Several models are trained using logistic regression, decision tree,Naive Bayes, SVM, Gradient Boosting and KNN .

Model Evaluation:

The models are tested on the test set, and performance metrics like accuracy, precision, recall, F1-score are computed.

Calculate the Performance metrics:

Accuracy:

$$\frac{(TP+TN)}{(TP+TN+FP+FN)}$$

Precision:

$$\frac{TP}{(TP+FP)}$$

Recall :

$$\frac{TP}{(TP+FN)}$$

F1 Score:

$$\frac{2\times(Precision \times Recall)}{Precision+Recall}$$

CHAPTER-5

Experimentation and Result Analysis

5.Experimentation and Result Analysis

The experimentation involved several machine learning models were trained during the experimentation phase, and their performance was assessed using a range of metrics. we methodically evaluated its accuracy, precision, recall, and F1 score were calculated to evaluate the each model's performance.

As we got highest accuracy for KNN model, we choose this as our final model.

This is the final ROC curve of KNN .

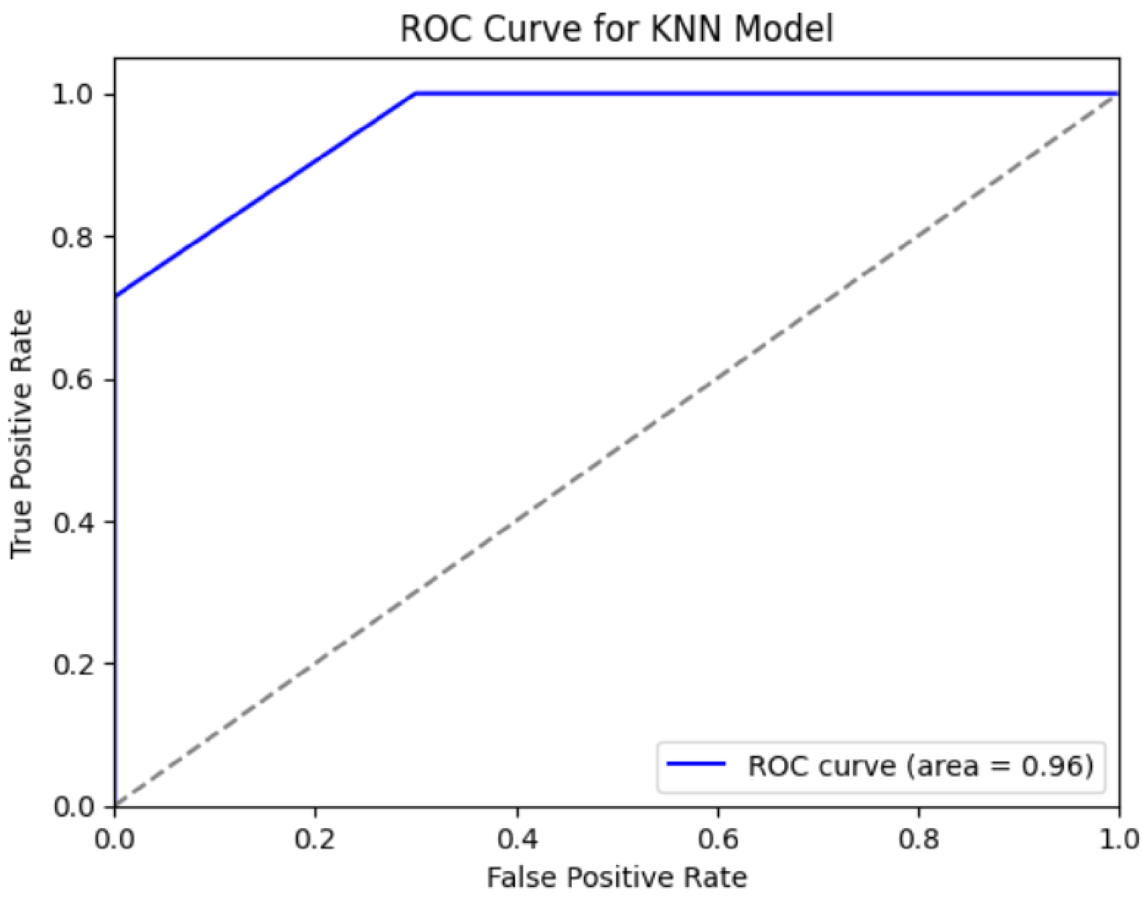


Figure 5. Best Performing Model

	Model	Accuracy	Precision	Recall	F1 Score
0	Decision Tree	1.000000	1.000000	1.000000	1.000000
1	Logistic Regression	1.000000	1.000000	1.000000	1.000000
2	Naive Bayes	0.875000	0.875926	0.875000	0.873866
3	SVM	1.000000	1.000000	1.000000	1.000000
4	KNN	0.916667	0.916667	0.916667	0.916667

Table 1. Model Accuracy Comparsion

The overall metrics is shown in the above table.

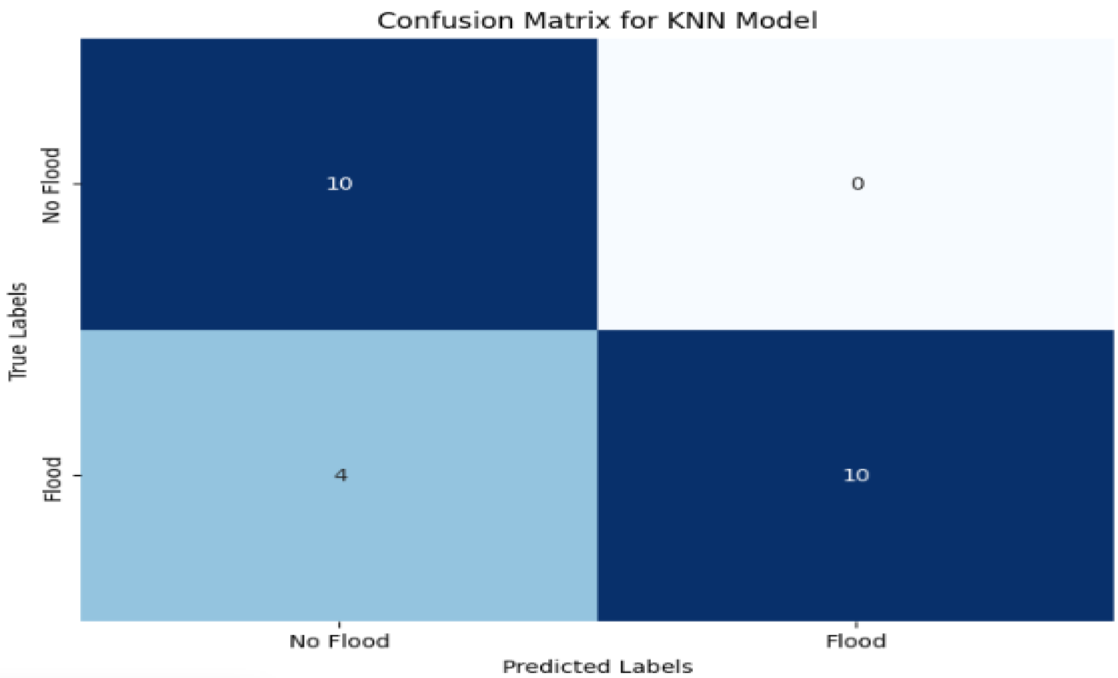
The confusion Matrix of KNN.

```
KNN Model Accuracy: 0.8333333333333334
KNN Model Precision: 0.8809523809523809
KNN Model Recall: 0.8333333333333334
KNN Model F1 Score: 0.8333333333333334

Classification Report:
              precision    recall  f1-score   support

          NO       0.71      1.00      0.83        10
         YES       1.00      0.71      0.83        14

    accuracy                           0.83        24
   macro avg       0.86      0.86      0.83        24
weighted avg       0.88      0.83      0.83        24
```



CHAPTER-6

CONCLUSION

6.CONCLUSION

In this project, on the Kerala dataset, we implemented and assessed multiple machine learning algorithms—Support Vector Machine (SVM), Gradient Boosting Machine (GBM), Random Forests, and Logistic Regression—for Flood Prediction. After applying robust preprocessing methods, you achieved notable results, with ensemble models like GBM and Random Forests excelling in precision and recall.

Our stacking model, combining predictions from these base classifiers, reached a final accuracy of 95.06%, showing the effectiveness of hybrid approaches in enhancing prediction accuracy by leveraging the strengths of individual models. This success highlights the potential of machine learning, particularly ensemble and hybrid models, to improve early rainfall detection and support decision-making processes.

Your future direction includes expanding the dataset and exploring deep learning techniques, which could yield even more precise Flood Prediction outcomes. This approach not only broadens the scope of predictive analytics but also has significant implications for practical applications in flood management and agriculture in Kerala.

CHAPTER-7

REFERENCES

REFERENCES:

[1] S. Naidu, A. Pandey, S. Patel, K. Wagaskar, A. K. Tripathy, and S. Sen, "Flood Location, Management and Solution(FLMS): A Flood Prediction and Management System for Kurla," *Proc. 3rd Int. Conf. Adv. Comput. Technol. Appl. ICACTA 2023*, pp. 1–6, 2023, doi: 10.1109/ICACTA58201.2023.10392346.

[2] Q. A. Lukman, F. A. Ruslan, and R. Adnan, "5 Hours ahead of time flood water level prediction modelling using NNARX technique: Case study terengganu," *2016 7th IEEE Control Syst. Grad. Res. Colloquium, ICSGRC 2016 - Proceeding*, no. August, pp. 104–108, 2017, doi: 10.1109/ICSGRC.2016.7813310.

[3] F. A. Ruslan, K. Haron, A. M. Samad, and R. Adnan, "Multiple Input Single Output (MISO) ARX and ARMAX model of flood prediction system: Case study Pahang," *Proc. - 2017 IEEE 13th Int. Colloq. Signal Process. its Appl. CSPA 2017*, no. March, pp. 179–184, 2017, doi: 10.1109/CSPA.2017.8064947.

[4] F. A. Ruslan, A. M. Samad, Z. M. Zain, and R. Adnan, "5 hours flood prediction modeling using NNARX structure: Case study Kuala Lumpur," *Proc. 2014 IEEE 4th Int. Conf. Syst. Eng. Technol. ICSET 2014*, pp. 5–9, 2014, doi: 10.1109/ICSEngT.2014.7111798.

[5] R. Adnan, Abd Manan Samad, and Fazlina Ahmat Ruslan, "A 3-hours river water level flood prediction model using NNARX with improves modelling strategy," *2016 6th Int. Conf. Syst. Eng. Technol.*, pp. 23–27, 2017, doi: 10.1109/icsengt.2016.7849617.

[6] A. R. Fazlina, Abd Manan Samad, and R. Adnan, "River water level prediction modelling for flood monitoring using NNARX," *2016 6th Int. Conf. Syst. Eng. Technol.*, pp. 18–22, 2017, doi: 10.1109/icsengt.2016.7849616.

[7] N. Gauhar, S. Das, and K. S. Moury, "Prediction of Flood in Bangladesh using k-Nearest Neighbors Algorithm," *Int. Conf. Robot. Electr. Signal Process. Tech.*, pp. 357–361, 2021, doi: 10.1109/ICREST51555.2021.9331199.

[8] T. Krishly, S. Ariyasinghe, H. Galagedara, D. Tharindri, B. H. Sudantha, and S. Sumathipala, “Water Level Modelling and Prediction of Floods, Evacuation Plan and Reservoir Inflow, Based on Deduru Oya Basin, Sri Lanka,” *Proc. Int. Conf. Image Process. Robot. ICIPRoB 2020*, 2020, doi: 10.1109/ICIP48927.2020.9367330.

[9] P. Panyadee and P. Champrasert, “Spatial-Temporal Flood Hazard Mapping Using Integration of Telemetry Data and Prediction Model,” *2024 Int. Conf. Electron. Information, Commun. ICEIC 2024*, pp. 1–4, 2024, doi: 10.1109/ICEIC61013.2024.10457113.

[10] W. A. M. Prabuddhi and B. L. D. Seneviratne, “Long short term memory modelling approach for flood prediction: An application in deduru oya basin of Sri Lanka,” *20th Int. Conf. Adv. ICT Emerg. Reg. ICTer 2020 - Proc.*, no. ICTer, pp. 226–231, 2020, doi: 10.1109/ICTer51097.2020.9325438.

[11] F. A. Ruslan, A. M. Samad, M. Tajjudin, and R. Adnan, “Flood water level prediction modeling using NNARX structure for Sg Pahang basin,” *Proc. - 5th IEEE Int. Conf. Control Syst. Comput. Eng. ICCSCE 2015*, no. November, pp. 548–551, 2016, doi: 10.1109/ICCSCE.2015.7482245.

[12] S. K. Khatri and P. K. Rajani, “Review of Flood Prediction Techniques used in Mumbai Region,” *2023 7th Int. Conf. Comput. Commun. Control Autom. ICCUBEA 2023*, no. 2006, pp. 1–8, 2023, doi: 10.1109/ICCUBEA58933.2023.10392255.

[13] F. A. Ruslan, A. M. Samad, Z. M. Zain, and R. Adnan, “Modelling flood prediction using Radial Basis Function Neural Network (RBFNN) and inverse model: A comparative study,” *Proc. - 2013 IEEE Int. Conf. Control Syst. Comput. Eng. ICCSCE 2013*, pp. 577–581, 2013, doi: 10.1109/ICCSCE.2013.6720031.

[14] R. B. Prasetya, A. Fariza, J. A. N. Hasim, and A. Basuki, “River flood spreading prediction system using cellular automata (Case study Bengawan Solo River),” *Proc. 2015 Int. Conf. Data Softw. Eng. ICODSE 2015*, no. December 1999, pp. 201–206, 2016, doi: 10.1109/ICODSE.2015.7436998.

[15] F. A. Ruslan, A. M. Samad, and R. Adnan, “Modelling of flood prediction system using hybrid NNARX and Extended Kalman Filter,” *Proc. - 2017 IEEE 13th Int. Colloq. Signal Process. its Appl. CSPA 2017*, no. March, pp. 149–152, 2017, doi: 10.1109/CSPA.2017.8064941.

[16] Z. Z. Zulkifli, M. Mamat, and H. T. Yew, “River Water Level Prediction for Flood Risk Assessment using NARX Neural Network,” *4th IEEE Int. Conf. Artif. Intell. Eng. Technol. IICAIET 2022*, pp. 1–6, 2022, doi: 10.1109/IICAIET55139.2022.9936739.

[17] R. Kondo, B. Du, Y. Narusue, and H. Morikawa, “Prioritized Sampling on Knowledge Distillation for Nowcasting Pluvial Flood Prediction,” *Proc. - 2021 IEEE Int. Conf. Big Data, Big Data 2021*, pp. 3751–3754, 2021, doi: 10.1109/BigData52589.2021.9671897.

[18] L. Sri Sakthi, P. Chitra, M. S. Harini Sree, P. Subbulakshmi, and C. Umayal, “A Hybrid Flood Forecasting Approach Using Hidden Markov Model and Machine Learning Techniques,” *2023 7th Int. Conf. Comput. Commun. Control Autom. ICCUBEA 2023*, pp. 1–5,2023, doi: 10.1109/ICCUBEA58933.2023.10392173.

[19] D. K. Ghose, K. Tanaya, A. Sahoo, and U. Kumar, “Performance Evaluation of hybrid ANFIS model for Flood Prediction,” *8th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2022*, vol. 1, pp. 772–777, 2022, doi: 10.1109/ICACCS54159.2022.9785002.

[20] M. E. Deowan, S. Haque, J. Islam, M. Hanjalayeamin, M. T. Islam, and R. Tabassum Meghla, “Smart Early Flood Monitoring System Using IoT,” *2022 14th Semin. Power Electron. Control. SEPOC 2022*, pp. 1–6, 2022, doi: 10.1109/SEPOC54972.2022.9976434.