Flood Forecasting for the Indian subcontinent Utilizing GBDT, XGboost, and Catboost

A Thesis Submitted To
Defence Institute Of Advanced Technology, Pune
For The Semester Four Evaluation (2021-2023) Of
Master Of Technology
In
Modelling and Simulation By

Anoop Kumar
Roll No. (21-14-20)

UNDER THE SUPERVISION OF

Dr. Odelu Ojjela
Associate Professor
DIAT, Pune
(Internal Supervisor)

Dr. Manmeet Singh
Scientist "D"
IITM, Pune
(External Supervisor)

SCHOOL OF COMPUTER ENGINEERING
&
MATHEMATICAL SCIENCE

DEFENCE INSTITUTE OF ADVANCED TECHNOLOGY(DIAT), PUNE, INDIA

APRIL 2023

APPROVAL SHEET

THESIS TITLE: **Flood Forecasting for the Indian subcontinent Utilizing GBDT, XGboost, and Catboost**

BY: **Anoop Kumar (21-14-20)**

is hereby approved for the degree of **Master of Technology in Modelling and Simulation**

Examiners:

INTERNAL SUPERVISOR
(Dr. Odelu Ojjela)

EXTERNAL SUPERVISOR
(Dr. Manmeet Singh)

ALLIED MEMBER
(Dr. A.V.R Murthy)

EXTERNAL EXPERT
(Dr. B.R. Kapuriya)

CHAIRPERSON
(Dr. Manisha J. Nene)

Place: DIAT, PUNE
DATE: / / 2023

DEDICATION

Dedicated to my mentor Dr. Manmeet Singh and my friend Mr. Neetiraj Malviya, who volunteered their time and expertise to me.

I am also very grateful to Mr. Om Pimpalgaonkar, Mr. Vaisak, Mr. Subodh Wasekar, Mr. Vishnu Tripathi, Miss. KM Sonika, Mr. Deepanshu Malik, Mr. Lovish Mittal, Mr. Abhijeet Malkar, Mr. Sourav Kumar Khan, Mr. Sanjay Yadav, and Mr. Matin Ahmed who gave me the environment at IITM Pune to learn new things and explore the solution behind the problem.

Additionally, I want to thank Mr. Kamal Kandpal, Mr. Nishchal Karwade, Mr. Rohit Lohani, and Mr. Shreyas Shelke, who were my flatmates at the time. whose spiritual support aids me in overcoming obstacles along the way. I also want to thank A. Kuladeep for his invaluable source of support and guidance while writing the thesis.

Last, but not least, I want to thank the Almighty, Teachers, Seniors, Friends, DIAT Pune, My family, and my loved ones whose blessings are beyond counting.
I am grateful to all.

DEFENCE INSTITUTE OF ADVANCED TECHNOLOGY(DIAT)
(DEEMED TO BE UNIVERSITY)
SCHOOL OF COMPUTER ENGINEERING & MATHEMATICAL SCIENCES
M.TECH - MODELLING AND SIMULATION
2021-23

CERTIFICATE

This is to certify that the Thesis entitled "**Flood Forecasting for the Indian subcontinent Utilizing GBDT, XGboost, and Catboost** " submitted to Defence Institute of Advanced Technology, Girinagar, for the award of the degree of ***Master of Technology***, is the bonafide research work done by **Mr. Anoop Kumar** under our supervision. The contents of this thesis have not been submitted elsewhere for the award of any degree.

Dr. Odelu Ojjela	Dr. Manmeet Singh
Associate Professor	Scientist "D"
SoCE&MS	IITM, Pune
DIAT, Pune	(External Supervisor)
(Internal Supervisor)	

COUNTERSIGNED

Dr. Manisha J. Nene
Director
School of Computer Engineering
& Mathematical Sciences
DIAT, Pune

.

DECLARATION

This is to certify that the work presented in the Thesis entitled “ **Flood Forecasting for the Indian subcontinent Utilizing GBDT, XGboost, and Catboost** “, is a bonafide work done by me and has not been submitted elsewhere for the award of any degree.

Date:____________
Place:____________

Anoop Kumar
Roll No. 21-14-20
School of Computer Engineering and
Mathematical Sciences
Defence Institute of Advanced
Technology
Girinagar, Pune

COUNTERSIGNED

Dr. Odelu Ojjela
Internal Supervisor
Associate Professor
DIAT, Pune

Dr. Manmeet Singh

External Supervisor
Scientist D
IITM, Pune

CERTIFICATE OF COURSE WORK

This is to certify that **Anoop Kumar (Reg No: 21-14-20)** has successfully completed the necessary course work as required for the **M.Tech. in Modeling and Simulation**. The details of the coursework are as follows.

Sr. No.	Course Code	Course Name	Credits
1	Advanced Numerical Methods	AM 601	4
2	Mathematical Modelling and System Analysis	AM 602	4
3	Advanced Optimization Techniques	AM 603	4
4	Advanced Statistical Techniques	AM 604	4
5	Computer Graphics	AM 605	4
6	Mathematical Methods	AM 606	4
7	Advanced Modelling Techniques	AM 621	4
8	Simulation of Linear and Nonlinear Systems	AM 622	4
9	Machine Learning	AM 623	4
10	Data Science: Tools and Techniques	AM 624D	4
11	Digital Image Processing	AM 625	4
12	Computational Number Theory And Cryptography	AM 628	4
13	M.Tech Dissertation Phase - I	AM651	14
14	M.Tech Dissertation Phase - II	AM652	14

Place: DIAT, PUNE
DATE: / / 2023

Controller of Examinations

DECLARATION FOR PLAGIARISM CHECK

This is to certify that M.Tech thesis entitled **“Flood Forecasting for the Indian subcontinent Utilizing GBDT, XGboost, and Catboost”** submitted by **Mr. Anoop Kumar** bearing Registration No **21-14-20** under the supervision of Dr. Odelu Ojjela in the Department of **School of Computer Engineering & Mathematical Sciences** is the original research work done by the candidate.

We have read the provision of **DIAT (DU) Plagiarism Policy**, and it is certified that all the conditions prescribed in the policy above are complied with in respect of the above-mentioned M Tech thesis.

The thesis has been checked for plagiarism, and the report is submitted along with the thesis for further processing.

- Originality content (including the contents from his own publications):

91__ %

- Similarity Reproduction of the content from other sources:

9__ %

We are aware that any issue related to plagiarism in future will have to be addressed by the candidate and the Supervisor (s) concerned.

Name of Candidate: **Anoop Kumar**
Signature:
Date: / / 2023

Name of Supervisor: **Dr. Odelu Ojjela**
Signature:
Date: / / 2023

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to **Dr. Odelu Ojella**, Associate Professor, Department of School of Computer Engineering and Mathematical Sciences, for motivating me all through this work. I would like to thank you for your timely guidance and support.

I would like to express my sincere gratitude to **Dr. Manmeet Singh**, Scientist ’D’, Indian Institute of Tropical Meteorology, Pune for providing me the opportunity to take up this project and more importantly, his precious guidance and motivation, which helped me to conquer the difficulties in research and development of this project.

Above all, I offer my sincere thanks and gratitude to the merciful God Almighty for the providence and grace, He has given throughout this attempt.

Anoop Kumar
M.Tech in Modelling and Simulation

ABSTRACT

This work uses Three Machine Learning models for Flood Forecasting in the Indian subcontinent Utilizing GBDT, XGboost, and Catboost to provide a unique method for flood forecasting in the Indian subcontinent. The suggested model combines regional characteristics, and meteorological parameters to forecast the likelihood of flooding in a certain area. The study uses deep learning methods to train the model using historical flood and meteorological data from diverse sources. The effectiveness of the model is assessed using a number of criteria, including accuracy, precision, recall, and F1 score. The findings show that the suggested model may accurately and successfully predict floods in the Indian region, offering useful information for flood control and disaster preparedness. The suggested model is a potential tool for flood forecasting because of its capacity to incorporate various datasets and understands complex patterns.

This thesis aids in our understanding of the connection between floods and the elements that influence them. We attempt to determine the behavior and flood patterns for 254 Indian stations from data provided by Indian Flood Inventory.

In order to learn more about flood forecasts. we are working on a flood forecasting model for the Indian area using variables from the dataset made available by Google Earth Engine, including runoff, surface runoff, subsurface runoff, precipitation, total evaporation, and numerical weather prediction data. I am trying to forecast floods for the Indian area using the XGboost classification model with sources like TRMM3B42, ERA-5, and CFSV2 data sets after turning the entire data set into tabular form.

List of Figures

List of Tables

Contents

NOMENCLATURE

N Number of examples in the dataset

M Number of features in each example

$\boldsymbol{x}_i$ Feature vector for the i-th example

y_i Target value for the i-th example

$f(\boldsymbol{x})$ Function that maps input features $\boldsymbol{x}$ to the output variable y

K Number of decision trees in the XGBoost ensemble

$f_k(\boldsymbol{x})$ Decision tree k in the XGBoost ensemble

$\hat{y}_i$ Predicted value for the i-th example

$L(y_i, \hat{y}_i)$ Loss function used by XGBoost

$\boldsymbol{\theta}$ Parameters of the XGBoost model, including the parameters of each tree

$\Omega(f_k)$ Regularization term that penalizes complex models

T Number of leaves in tree k

w_j Weight of the j-th leaf in tree k

γ Hyperparameter that controls the amount of regularization for the number of leaves

λ Hyperparameter that controls the amount of regularization for the size of the weights

n Number of samples in the training set

x_i Feature vector for sample i

y_i Corresponding target variable for sample i

$f(x)$ Function that predicts the target variable y given a new input feature vector x

Chapter 1

INTRODUCTION

1.1 Acquaintance

Water is necessary for the maintenance of ecosystems and the survival of living things, and it permeates every aspect of modern life. However, the Indian water resources sector is currently grappling with problems like demand exceeding supply, changes to the hydrological cycle brought on by climate change, and a high frequency of extreme events.

Floods are a significant natural disaster that affects millions of people worldwide each year. In India, floods are a common occurrence and cause severe damage to infrastructure, property, and human lives. Early warning systems for floods play a crucial role in mitigating their impact by enabling timely evacuation and disaster management measures. In recent years, there has been a significant increase in the amount of data available on meteorological parameters, water levels, and river flow rates, providing an opportunity to improve flood forecasting accuracy.

Traditional flood forecasting methods typically rely on statistical models, which have limitations in capturing the complexity of the flood processes. In contrast, machine learning algorithms can learn patterns in the data and make accurate predictions. Deep learning, a subfield of machine learning, has gained popularity in recent years due to its ability to handle large and complex datasets.

In this context. The proposed model integrates various datasets, including meteorological parameters, to predict the likelihood of flood occurrence in a given area. The objective of this study is to evaluate the performance of the proposed model in flood forecasting and assess its potential for improving flood management and disaster preparedness in the Indian region.

1.2 Evolution of Machine Learning Techniques in Flood Forecasting

Flood forecasting is an important area of research and has the potential to save lives and reduce the impact of floods on communities. Machine learning techniques have shown promise in this field, as they can be used to develop predictive models that can help forecast flooding events.

There are several machine learning techniques that can be used for flood forecasting, including artificial neural networks, decision trees, random forests, support vector machines, and deep learning. Each of these techniques has its own strengths and weaknesses, and the choice of technique will depend on the specific requirements of the task.

Artificial neural networks have been widely used in flood forecasting and have shown promising results. They are particularly well-suited for modeling complex relationships between input and output variables, and can be trained to identify patterns in large datasets. However, they can be computationally intensive and require large amounts of data to train effectively.

Decision trees and random forests are other popular techniques that have been used in flood forecasting. These techniques are generally simpler than artificial neural networks and are easier to interpret. They can also be used to identify important input variables and can handle both numerical and categorical data. However, they may not perform as well as artificial neural networks on more complex datasets.

Support vector machines are another machine learning technique that can be used for flood forecasting. They are particularly well-suited for binary classification tasks and can be used to identify flood-prone areas based on historical data. However, they may not perform as well on more complex datasets.

Deep learning techniques, such as convolutional neural networks, have also shown promise in flood forecasting. These techniques are particularly well-suited for processing image data and can be used to identify flood-prone areas based on satellite or aerial imagery. However, they may require large amounts of data to train effectively and can be computationally intensive.

1.3 Motivation

The motivation behind a flood forecasting project could be multifaceted and may depend on various factors such as the project's objectives, stakeholders' needs, and available resources. However, some of the common motivations behind a flood forecasting project could be:

Disaster Risk Reduction: Floods are natural disasters that can cause significant damage to infrastructure, loss of life, and economic losses. Developing accurate flood forecasting models can help authorities prepare and respond better to flood events, potentially saving lives and reducing damage.

Resource Management: Flood forecasting can help in managing water resources, especially in areas with scarce water resources. Accurate flood forecasting can help identify areas with potential water shortages and enable the efficient allocation of water resources in a timely manner.

Environmental Concerns: Floods can have severe environmental impacts, such as soil erosion, deforestation, and water pollution. Flood forecasting can help in identifying vulnerable areas and enable the implementation of mitigation measures to reduce environmental damage.

Advancements in Technology: Advances in technology, such as remote sensing, machine learning, and artificial intelligence, have enabled the development of more accurate flood forecasting models. A flood forecasting project can help explore the potential of these technologies to improve flood forecasting accuracy and identify areas for further research.

Socio-Economic Development: Accurate flood forecasting can have significant socio-economic benefits, such as reducing the economic losses associated with floods and enabling better resource allocation. A flood forecasting project can help in identifying areas with potential socio-economic benefits and enable policymakers to make informed decisions.

Chapter 2

LITERATURE REVIEW

2.1 Literature Review

An overview of flood forecasting methods and the state of machine learning applications in flood prediction are given in the literature review. Although hydrological models are frequently used in flood forecasting, their accuracy is frequently constrained by the quantity and caliber of available data. By using historical data to find patterns and connections between flood events and environmental variables, machine learning algorithms have shown promise in overcoming these constraints. Additionally, the review looks at the quantity and caliber of data sources used in flood forecasting in the Indian subcontinent. Due to its varied geography and poor infrastructure, the region faces significant difficulties in the management and collection of data. The potential of machine learning algorithms to enhance flood forecasting in the Indian subcontinent is highlighted in the review's conclusion.

Girish [1], Using a special Karin sensor, the SWOT mission will give data on Water Surface Elevation, water surface slope, and inundation extent for two large swaths, each 50 km wide and separated by a 20 km gap at the satellite nadir. By giving information for rivers wider than 100 m and water bodies larger than 250 m2, SWOT will offer a fresh perspective for river and lake monitoring. SWOT identified 85.4 percent of big flood occurrences and 30.3 percent of floods lasting more than 43 days. Different organizations can benefit from the information a flood severity map will offer when it comes to hazard mapping, damage assessment, flood management, etc.

Manabendra [2], This paper includes the flood database which provides the hazard assessment, Impact assessment, and hydrological modeling. This article outlines the ongoing construction of the Indian Flood Inventory(IFI), the first openly accessible, analysis-ready geospatial dataset for the area that contains comprehensive qualitative and quantitative data about floods, including spatial extents.

Hong Xuan [3], This work adds to the body of knowledge on homogenous regions of flood-generating processes. In the southeastern United States, northern Australia, and the southern and eastern areas of Brazil, short-term precipitation predictions are substantially connected with flood timing. The findings add to our understanding of the average time and temporal concentration of maximum events. A rule-based categorization system was devised to split the world into five hydro-climate classifications. Each class reflects an area that uses the same flood timing predictor. The categorization was used to predict flood time globally, even in areas where streamflow sensors did not exist. The worldwide map of flood time prediction might be used to assess the performance of global hydrological models.

Zhongrun [4], GNRRM, a novel type of network architecture for rainfall-runoff modeling based on the notion of graph neural networks, was introduced in this study to incorporate geographical information from precipitation data. To apply baseline models, it employed out-of-state time-series models such as LSTM, BiLSTM, GTCN, and BiGTCN. The results show that models with optimized graph topologies efficiently exploited geographical information and avoided overfitting problems induced by spatial correlation. Given that the effective design of GNRRM was influenced by conventional physical rainfall-runoff models. This paper strongly advises researchers working on integrating physical models with existing deep learning models in simulating rainfall-runoff based on deep learning.

Amir [5], The use of ML modeling to predict floods is still in its infancy and is still undergoing development. Over 6000 articles were examined and their performance was analyzed for this study. The prediction models were sorted into categories of hybrid and single techniques after being divided into two groups based on lead time. The first involved the use of data decomposition techniques to enhance the dataset's quality, for example, to better tune ANNs to achieve ideal neuronal topologies. The secret to success was having contributions from both the ML and hydrological fields.

Muhammed [6], A thorough examination of deep learning applications in the disciplines of hydrology and water resources is carried out. There are summaries of works that use deep learning for hydrologic modeling, as well as a succinct deep learning overview. Key concerns in the literature on water-related deep learning applications are discussed. Future research directions in the confluence of deep learning and hydroscience are discussed.

Yury Gorishniy [7], Deep learning for tabular data lacks effective baselines, making it difficult to compare models. This work identifies two simple and powerful deep architectures for tabular data, which outperform existing architectures on a diverse set of tasks. This work investigated the status quo in deep learning for tabular data and improved the state of baselines with ResNet-like architecture and FT-Transformer.

Yury Gorishniy [8], Embeddings for numerical features are an underexplored degree of freedom in tabular DL, allowing for more powerful models and competing with GBDT on GBDT-friendly benchmarks. Embedding numerical features is beneficial for many backbones, providing significant performance boosts and potential for further improvements in tabular DL. Embeddings for numerical features enable DL architectures to achieve better results and reduce the gap with Gradient Boosted Decision Trees. Embedding modules help optimization, but only when applied to all features, which may be a suboptimal choice.

Ivan Rubachev[9], Deep learning models for tabular data can benefit from pretraining, but it is unclear if it provides consistent improvements. Using object target labels during pretraining significantly increases tabular DL models' performance, leading to their superiority over GBDTs. This work evaluated pretraining objectives for tabular deep learning and revealed several recipes for optimal performance that can be universally beneficial across various problems and models. their findings confirm that pretraining can significantly improve the performance of tabular deep models and provide additional evidence that tabular DL can become a strong alternative to GBDT.

In summary, these studies demonstrate the potential of machine learning-based models for flood forecasting and management. The proposed models in this study are novel in their integration of various datasets, including geographical features, and meteorological parameters, for flood forecasting.

2.2 Hypothesis Development

After the literature survey, we analyzed that for forecasting the flood we need different data sets which are open source and freely available and are downloaded using google collab and Apollo server from the google earth engine.

which are as follows:

1) unique lat log from IFI (323 in nos.) The unique latitude and longitude of the Indian region are available in the Indian Flood Inventory data. hence by using that data we collected the unique lat long from the dataset and we use it for our target variable with respect to severity. which is available from 1985-06-23 to 2016-08-23.

2) catchment details from GSIM (319 Indian stations) Catchment Area/Soil Properties global Data is available for more than 30000 stations by GSIM I-WRIS which covers 319 Indian stations and years of data Varying for each river.

3) Digital Elevation Model from GTOPO30 (123867 latitude, longitude, and elevation for Region Of Interest)(1996) Topography/Digital Elevation Model (DEM) for global is available in google earth engine with named GTOPO 30 for the year 1996-01-01 for our region of interest total 123867 latitudes and longitude are downloaded with respect to elevation.

4) Runoff, Surface Runoff, and Subsurface Runoff, from ERA-5 downloaded from (1985-06-23 to 2019) for our project, and this data set is available from 1981-01-01 to 2023-01-28.

5) precipitation from TRMM3B42 (1998-2019) for precipitation data which is downloaded from named TRMM3B42 with availability from 1998-01-01 to 2019-12-31. and we download it from 1998-2016 as per our requirement for the common time period with respect to Indian Flood Inventory data.

6) all variables of NWP from CFSV2 (1998-2019) Numerical Weather Prediction CFSV2 data availability from 1979-01-01 to 2023-03-11 and we extract the data for the common time period.

7)Land use and Land Cover Land Use/ Land Cover data can be downloaded from the Dynamic World V1 data set which is available for the period 2015-06-23 to 2023-03-12.

The reason for considering these data sets from various data sources is that we hope that the contributing variables for flood occurrence are catchment area, topography, Runoff, Surface Runoff, Subsurface Runoff, precipitation data, numerical weather prediction data, land use, and land cover data.

with this hypothesis, we work on it extracted the data as per the requirement and come up with the indispensable variables which are discussed in the upcoming chapter.

Chapter 3

METHODOLOGY

3.1 Methodology of Research

The proposed model for flood forecasting in the Indian region involves the following methodology:

- Comprehend The Problem
- Data Collection
- Data Cleaning and Preprocessing
- Exploratory Data Analysis
- Feature Engineering
- Model Selection
- Model Training and Evaluation

3.2 Comprehend The Problem

After understanding the model requirements and dataset availability, the most convenient and indispensable variables for flood prediction are runoff, surface runoff, subsurface runoff, total evaporation, precipitation, and all variables of Numerical Weather Prediction are good for preparing the single tabular data.

3.3 Data Collection

Data Collection Historical flood data for India is provided by government agencies and research institutions eg. District Forest Offices and India Meteorological Department. The Indian Flood Inventory provided openly accessible data for India from which I extracted 254 unique latitudes and longitude for flood severity parameters they divide it into one to two scales and by which I can take the thresh hold value of 1.5 for the severity value one there is no occurrence of flood i.e. 0 and above 1.5 I consider it as a high severity case which shows the flood occurrence i.e. 1 and Flood is my Target variable. for variables like runoff, surface runoff, subsurface runoff, and total evaporation available in $ECMWF/ERA5-LAND/HOURLY$ and for all weather affecting variables from the data set $NOAA/CFSV2/FOR6H$ and for precipitation data $TRMM/3B42$ will be collected from Google Earth Engine.

3.4 Data Cleaning and Preprocessing

The data will be preprocessed and cleaned to remove missing values and outliers. there are two outlier lies in the Indian Flood Inventory in the unique latitude longitude with human error with latitude 22.1496 and longitude 110.224, and 118.2780 and 788.550 which are not included in the region of interest i.e [67,7,98,38] for South West and North East of Indian region respectively. in the first stage, we find out the data set availability of the data set available in our case the common period for the whole data set is 1998-01-01 to 2016-08-29 so we extract our data for that period. in the second stage, we convert the 6 hourly data, 3 hourly data, and hourly data into daily data and we come up with 6816 days of data for 254 unique Indian stations. In the third stage, we merge the whole data with respect to the unique latitude and longitude.

3.5 Exploratory Data Analysis

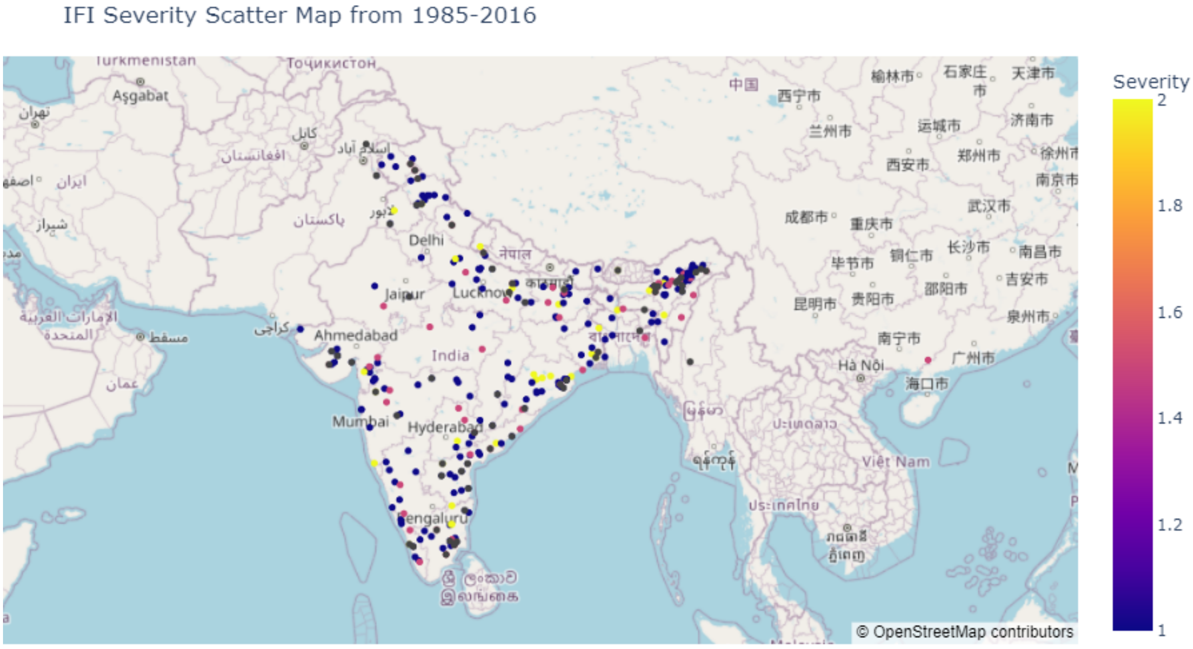

Figure 3.1: Severity Map of India from IFI data set

We looked into the Indian Flood Inventory Figure 3.1, and we found that the given data set had details on the intensity of floods from 1985 to 2016. In order to model, we extract the unique latitude and longitude from this data set. first of all, we found out that 323 stations are the unique station in this data set in which two stations are outliers and the rest 67 stations are not merged with the other data sets with respect to the latitude and longitude values so finally, we come up with 254 stations for the time period of 18 years 7 months and 29 days.

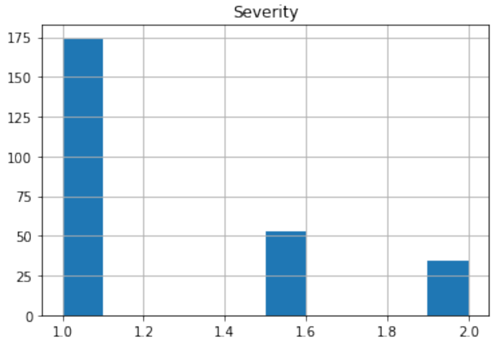

Figure 3.2: Severity VS No. of Stations

We can spot patterns and correlations in the data by using scatterplots and histograms to visually depict the data. By visualizing the data, we establish our threshold at 1.5 and take into account values above severity 1.5 as the flood. Out of that maximum number of stations with a severity of 1, we discover that 173 stations have a severity value of 1, 52 stations have a severity value of 1.5, and 29 stations have a severity value of 2. see Figure 3.2

3.6 Feature Engineering

The collected data for the unique Lat-Log of 254 Indian stations will be analyzed to identify relevant features, including the data for the period 1998-01-01 to 2016-08-29 or 6816 days of daily data such as Runoff, Sub-Surface Runoff, Surface Runoff, and Total Precipitation, and all meteorological parameters of Numerical Weather Prediction CFSV2 such as Downward long-wave radiation flux at surface, 6-hour average, Downward short-wave radiation flux at surface, 6-hour average Geopotential height at surface, Latent heat net flux at surface, 6-hour average, Maximum specific humidity 2m above ground, 6-hour interval, Maximum temperature 2m above ground, 6-hour interval, Minimum specific humidity 2m above ground, 6-hour interval, Minimum temperature 2m above ground, 6-hour interval, Potential evaporation rate at surface, 6-hour average, Precipitation rate at surface, 6-hour average, Pressure at surface, Sensible heat net flux at surface, 6-hour average, Specific humidity 2m above ground, Temperature 2m above ground, U-component of wind 10m above ground, Upward long-wave radiation flux at surface, 6-hour average, Upward short-wave radiation flux at surface, 6-hour average, V-component of wind 10m above ground, Volumetric soil moisture content 5cm below surface layer, Volumetric soil moisture content 25cm below surface layer, Volumetric soil moisture content 70cm below surface layer, Volumetric soil moisture content 150cm below surface layer, These features will be standardized to ensure compatibility with the models.

Runoff: Runoff is the movement of water over the surface of the land and into bodies of water, such as streams, rivers, and lakes. It is often caused by precipitation events that exceed the capacity of the soil to absorb water.

Sub-Surface Runoff: Sub-surface runoff refers to the movement of water through soil and rock layers below the surface of the Earth. It is an important process for replenishing groundwater reserves and can also contribute to the flow of streams.

Surface Runoff: Surface runoff is the portion of precipitation that flows over the surface of the land and eventually into bodies of water. It can be influenced by factors such as topography, soil characteristics, and land use.

Total Precipitation: Total precipitation refers to the amount of water that falls from the atmosphere to the surface of the Earth, including both rain and snow.

Precipitation: Precipitation is any form of water that falls from the atmosphere to the surface of the Earth, including rain, snow, sleet, and hail.

Downward long-wave radiation flux at surface, 6-hour average: This variable represents the amount of long-wave radiation emitted by the Earth's surface that is absorbed by the atmosphere. It is an important factor for understanding the Earth's energy balance and can influence weather patterns and climate.

Downward short-wave radiation flux at surface, 6-hour average: This variable represents the amount of short-wave radiation that is absorbed by the Earth's surface from the sun. It is an important factor for understanding the Earth's energy balance and can influence weather patterns and climate.

Geopotential height at surface: Geopotential height at the surface represents the height above sea level of a given pressure level in the atmosphere. It is often used as an indicator of atmospheric pressure and can be used to predict weather patterns and conditions.

Latent heat net flux at surface, 6-hour average: This variable represents the amount of energy released or absorbed by the Earth's surface due to changes in the water cycle, such as evaporation and condensation. It is an important factor for understanding the Earth's energy balance and can influence weather patterns and climate.

Maximum specific humidity 2m above ground, 6-hour interval: This variable represents the maximum amount of water vapor that can be held in the air at a height of 2 meters above the Earth's surface. It is an important factor for understanding the water cycle and can be used to predict weather patterns and conditions.

Maximum temperature 2m above ground, 6-hour interval: This variable represents the maximum temperature of the air at a height of 2 meters above the Earth's surface. It is an important factor for understanding the energy balance of the Earth's surface and for predicting weather conditions in a given area.

Minimum specific humidity 2m above ground, 6-hour interval: This variable represents the minimum amount of water vapor that can be held in the air at a height of 2 meters above the Earth's surface. It is an important factor for understanding the water cycle and can be used to predict weather patterns and conditions.

Minimum temperature 2m above ground, 6-hour interval: This variable represents the minimum temperature of the air at a height of 2 meters above the Earth's surface. It is an important factor for understanding the energy balance of the Earth's surface and for predicting weather conditions in a given area.

potential evaporation rate at surface is an important factor for understanding the water cycle. It represents the rate at which water can evaporate from the Earth's surface under given conditions. Evaporation is a key component of the water cycle, as it helps to transfer water from the surface to the atmosphere, where it can then form clouds and eventually precipitate back to the surface.

The potential evaporation rate is affected by various meteorological parameters such as temperature, humidity, wind speed, and solar radiation. By monitoring the potential evaporation rate over time, scientists can gain insights into changes in the water cycle and how it may be affected by factors such as climate change and land use changes.

Precipitation rate at surface, 6-hour average: This variable represents the rate at which precipitation is falling on the Earth's surface, averaged over a 6-hour period. Precipitation includes rain, snow, sleet, and hail, and plays a critical role in the water cycle.

Pressure at surface: This variable represents the atmospheric pressure at the Earth's surface. Changes in atmospheric pressure can indicate the movement of air masses and the development of weather systems such as storms.

Sensible heat net flux at surface, 6-hour average: This variable represents the net amount of heat energy transferred between the Earth's surface and the atmosphere due to differences in temperature. Sensible heat transfer occurs through conduction and convection, and plays an important role in the Earth's energy budget.

Specific humidity 2m above ground: This variable represents the amount of water vapor present in the air at a height of 2 meters above the Earth's surface. Specific humidity is an important factor in determining the potential for precipitation and other weather phenomena.

U-component of wind 10m above ground: This variable represents the east-west component of the wind velocity vector at a height of 10 meters above the Earth's surface. The wind plays a key role in many weather phenomena, including the formation of storms and the transport of heat and moisture across the Earth's surface.

Upward long-wave radiation flux at surface, 6-hour average: This variable represents the amount of long-wave radiation emitted from the Earth's surface and directed upwards towards the atmosphere. Long-wave radiation is an important component of the Earth's energy budget and plays a role in determining the temperature and composition of the atmosphere.

Upward short-wave radiation flux at surface, 6-hour average: This variable represents the amount of short-wave radiation emitted from the Earth's surface and directed upwards towards the atmosphere. Short-wave radiation is primarily due to solar radiation, and plays a key role in driving many atmospheric processes.

V-component of wind 10m above ground: This variable represents the north-south component of the wind velocity vector at a height of 10 meters above the Earth's surface. Like the U-component of wind, it plays an important role in many weather phenomena.

Volumetric soil moisture content 5cm/25cm/70cm/150cm below surface layer: These variables represent the amount of water present in the soil at various depths below the Earth's surface. Soil moisture content is an important factor in many ecological processes, including plant growth, soil erosion, and nutrient cycling.

3.7 Model Selection

After analyzing the data we can identify that our problem comes under the classification and for the classification model, we select the three models i.e. GBDT, XGboost, and Catboost after reading the papers named Revisiting Deep Learning Models for Tabular Data, On Embeddings for Numerical Features in Tabular Deep Learning and Revisiting Pretraining Objectives for Tabular Deep Learning We found out that the Categorical Boosting Machine learning technique and Extreme Gradient Boosting technique perform well than the Gradient Boosting Decision Tree technique so we compare each and every model and analyze them on the bases of our dataset.

3.7.1 Gradient Boosting Decision Tree

Gradient Boosting Decision Tree, or GBDT for short, is a machine learning technique used for both classification and regression tasks. It is a kind of boosting method that constructs an ensemble of decision trees in a sequential manner, with each new tree attempting to fix the mistakes caused by the one before it.

The residual error is effectively minimized by GBDT at each iteration by fitting a fresh decision tree to the negative gradient of the loss function with respect to the anticipated values. The final product is created by combining the predictions of all the trees. The algorithm keeps creating new trees until either a predetermined number of trees are reached or the validation set's performance indicator stops advancing.

Continuous, categorical, and missing value data types can all be handled by GBDT, which is renowned for its versatility. It is also well-liked for its interpretability because of the assembled set of decision trees that results can be visualized and examined to learn more about the underlying data patterns. In contrast, GBDT can be computationally expensive and prone to overfitting if the number of trees is not properly adjusted.

Let us define some notation:

- We have a training set of n samples $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$, where x_i is the feature vector for sample i, and y_i is the corresponding target variable.
- We want to learn a function $f(x)$ that predicts the target variable y given a new input feature vector x.

The GBDT algorithm consists of the following steps:

1. Initialize the prediction function $f_0(x)$ to a constant value. This is typically set to the mean target value of the training set.

2. For each iteration $t = 1, 2, ..., T$:

   (a) Compute the negative gradient of the loss function with respect to the predicted values, denoted as $r_{ti} = -\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} | f(x) = ft - 1(x_i)$.

   (b) Fit a new decision tree to the negative gradient values r_{ti}, with the goal of minimizing the residual errors $\epsilon_i = y_i - f_{t-1}(x_i) - \gamma_t h(x_i)$, where $h(x_i)$ is the prediction of the new decision tree for sample i, and γ_t is the learning rate for iteration t.

   (c) Update the prediction function by adding the prediction of the new tree, scaled by the learning rate γ_t, to the previous prediction function: $f_t(x) = f_{t-1}(x) + \gamma_t h(x)$.

   (d) Repeat steps (a)-(c) until a specified number of trees T is reached or until the performance on a validation set stops improving.

3. The final prediction function is the sum of all the individual prediction functions from each tree: $f(x) = \sum_{t=1}^{T} \gamma_t h(x)$.

Here, the loss function $L(y, f(x))$ measures the difference between the predicted value $f(x)$ and the true target value y. For regression problems, a common choice is the mean squared error (MSE), while for classification problems, the cross-entropy loss is often used.

The key idea behind GBDT is that by iteratively fitting decision trees to the negative gradients of the loss function, we can gradually improve the prediction function and reduce the residual errors. The learning rate parameter γ_t controls the contribution of each tree to the final prediction, while the number of trees T and the depth and complexity of each tree are hyperparameters that need to be tuned to achieve good performance.

So we can say GBDT is a powerful and flexible algorithm that has been widely used in a variety of applications, including recommendation systems, fraud detection, and financial modeling.

3.7.2 Categorical Boosting

CatBoost is an open-source library for gradient boosting that was created specifically to address issues in the area of machine learning. Yandex is a Russian multinational technology company that created it.

CatBoost, which stands for Categorical Boosting, is intended to work effectively with data that contains both continuous and categorical features. It employs an algorithm similar to other gradient-boosting algorithms, but with some added features such as naturally handling categorical features and employing techniques to prevent overfitting.

CatBoost's ability to handle categorical data without the need for pre-processing or feature engineering is one of its main advantages. It employs an innovative method of encoding categorical features known as "Ordered Target Encoding," which aids in prediction accuracy.

Classification, regression, and ranking are just a few of the many tasks that CatBoost is useful for.

It has demonstrated success in a variety of practical applications, such as fraud detection, image recognition, and natural language processing.

CatBoost is a strong and adaptable machine-learning library that can help increase the precision of models and simplify the handling of categorical data.

3.7.3 Extreme Gradient Boosting

Extreme Gradient Boosting, also known as XGBoost, is a well-liked and effective open-source machine-learning library used for supervised learning issues like regression and classification. It is based on gradient boosting, an ensemble learning technique that combines a number of weak learners (in the case of XGBoost, decision trees) to produce a strong learner.

XGBoost is renowned for its speed, scalability, and accuracy in identifying intricate data patterns. Additionally, it has a wide range of hyperparameters that can be adjusted to optimize performance for particular tasks.

In numerous industries, such as finance, healthcare, and online advertising, XGBoost has been successfully incorporated into winning machine learning competition solutions.

detailed mathematical description of XGBoost. However, as this is a complex topic, I will have to split the explanation into several parts.

First, let's define some notation. Suppose we have a dataset of N examples with M features, where each example is denoted as $(\boldsymbol{x}_i, y_i)$, where $\boldsymbol{x}_i$ is the i-th example's feature vector and y_i is its corresponding target value. We want to learn a function $f(\boldsymbol{x})$ that maps input features $\boldsymbol{x}$ to the output variable y.

XGBoost builds an ensemble of K decision trees $f_k(\boldsymbol{x})_{k=1}^{K}$, where each tree is trained on the residuals of the previous trees. The final prediction of the ensemble is given by the sum of the predictions of all the trees:

$$f(x) = \sum_{k=1}^{K} f_k(x)$$

To train the decision trees, XGBoost uses a loss function $L(y_i, \hat{y}_i)$, where $\hat{y}_i$ is the predicted value for the i-th example. The loss function measures the difference between the predicted and actual values, and the goal is to minimize it.

XGBoost uses a regularized objective function that combines the loss function with a regularization term. The objective function is given by:

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^{N} L(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

where $\boldsymbol{\theta}$ represents all the parameters of the model, including the parameters of each tree, and $\Omega(f_k)$ is a regularization term that penalizes complex models. The regularization term is given by:

$$\Omega(f_k) = \gamma T + \frac{1}{2} \sum_{j=1}^{T} \frac{\omega_{j,k}}{\|\boldsymbol{\theta}_j\|_2} = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} w_j^2$$

where T is the number of leaves in the k-th tree, w_j is the weight of the j-th leaf, γ and λ are hyperparameters that control the amount of regularization. The first term penalizes the number of leaves in the tree, while the second term penalizes the size of the weights.

To minimize the objective function, XGBoost uses a gradient boosting algorithm, which iteratively adds new trees to the ensemble to reduce the residual error of the previous trees. In each iteration, XGBoost computes the negative gradient of the loss function with respect to the predicted values, and trains a new tree to predict the negative gradient. The new tree is then added to the ensemble, and the predictions of all the trees are updated. This process is repeated until the objective function converges or a maximum number of iterations is reached.

The training of each tree is done by recursively partitioning the data into smaller subsets, based on the values of a single feature at each node of the tree. The partition that results in the largest reduction in the loss function is selected at each node, using a greedy algorithm. The weights of the leaves are then optimized using a closed-form solution, based on the second-order derivative of the loss function.

Overall, XGBoost is a powerful and flexible machine-learning algorithm that can be used for a wide range of tasks. Its success is due to its ability to handle complex, high-dimensional data, and its efficient implementation, which allows it to scale to large datasets and feature spaces with high dimensions, making it suitable for a wide range of machine-learning problems. Its ability to deal with missing values, and different types of data, and provide built-in feature importance measures make it a popular choice among data scientists and machine learning practitioners. Furthermore, in various benchmark tests and machine learning competitions, XGBoost has been shown to outperform many other popular machine learning algorithms.

3.8 Model Training and Evaluation

while Model evaluation and testing the performance of a trained machine learning model on a completely new and unseen dataset, often called a test dataset. The test dataset is used to evaluate the model's ability to generalize to new data and make accurate predictions. Evaluation metrics such as accuracy, precision, recall, F1 score, and AUC-ROC can be calculated to assess the model's performance on the test dataset.

Accuracy Score:

The accuracy score is a performance metric used to evaluate the overall performance of a classification model. It represents the ratio of the number of correctly classified samples to the total number of samples. In other words, it measures the proportion of true positives and true negatives out of all the predictions made by the model.

$\text{Accuracy Score} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$

While accuracy is a useful metric to evaluate the overall performance of a model, it can be misleading in cases where the class distribution is imbalanced. In such cases, a model that always predicts the majority class will achieve high accuracy but may not be useful in practice.

Precision Score:

The precision score is a performance metric used to evaluate the precision of a classifier's positive predictions. It represents the ratio of the number of true positives to the sum of true positives and false positives. In other words, it measures the proportion of actual positive cases out of all the positive predictions made by the model.

$\text{Precision Score} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$

A high precision score indicates that the classifier is very good at correctly predicting positive cases and has a low rate of false positives.

Recall Score:

The recall score is a performance metric used to evaluate the ability of a classifier to identify all relevant instances of a positive class. It represents the ratio of the number of true positives to the sum of true positives and false negatives. In other words, it measures the proportion of actual positive cases that were correctly identified by the model.

Recall Score = $\frac{\text{True Positives}}{\text{True Positives + False Negatives}}$

A high recall score indicates that the classifier is very good at identifying all relevant positive cases, but may also result in a higher rate of false positives.

F1 Score:

The F1 score is a performance metric that takes into account both precision and recall scores to provide a more balanced evaluation of a classifier's performance. It represents the harmonic mean of precision and recall.

F1 Score $= 2 * \frac{(Precision*Recall)}{(Precision+Recall)}$

The F1 score ranges from 0 to 1, with a value of 1 indicating perfect precision and recall.

ROC-AUC Score:

The ROC-AUC score is a performance metric used to evaluate the ability of a classifier to distinguish between positive and negative classes. It is based on the receiver operating characteristic (ROC) curve, which plots the true positive rate against the false positive rate at various thresholds.

ROC-AUC Score = Area Under the ROC Curve

A high ROC-AUC score indicates that the classifier is very good at distinguishing between positive and negative cases, and has a low rate of false positives and false negatives.

Confusion Matrix:

A confusion matrix is a table that summarizes the classification performance of a model by showing the number of true positives, true negatives, false positives, and false negatives. It is a useful tool for visualizing the performance of a classification model and identifying areas for improvement.

The confusion matrix is a table (Table 3.1) that summarizes the performance of a binary classification model. It can be represented as:

	Predicted Negative	Predicted Positive
Actual Negative	TN	FP
Actual Positive	FN	TP

Table 3.1: Confusion Matrix Table.

where TN is the number of true negatives, FP is the number of false positives, FN is the number of false negatives, and TP is the number of true positives.

Precision-Recall Curve:

The precision-recall (PR) curve is a graphical representation of the trade-off between precision and recall for different probability thresholds. It is a useful tool for evaluating the performance of a classification model when the classes in the dataset are imbalanced. The PR curve can be plotted using the precision and recall values calculated at different probability thresholds.

The precision-recall curve plots precision against recall for different probability thresholds. Precision and recall can be calculated as:

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

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

where TP is the number of true positives, and FP and FN are the number of false positives and false negatives, respectively.

ROC Curve:

The Receiver Operating Characteristic curve is a graphical representation of the trade-off between the true positive rate and false positive rate for different probability thresholds. It is a useful tool for evaluating the performance of a classification model when the classes in the dataset are balanced or imbalanced. The ROC curve can be plotted using the TPR and FPR values calculated at different probability thresholds.

The ROC curve plots the true positive rate (TPR) against the false positive rate (FPR) for different probability thresholds. TPR and FPR can be calculated as:

TPR= $\frac{(TP)}{(TP+FN)}$,

FPR= $\frac{(FP)}{(FP+TN)}$

where TPR is the true positive rate, TP is the number of true positives, FN is the number of false negatives, FPR is the false positive rate, FP is the number of false positives, and TN is the number of true negatives.

Precision-Recall vs Threshold Curve:

The precision-recall vs threshold curve is a graphical representation of the trade-off between precision, recall, and probability thresholds. It is a useful tool for identifying the optimal probability threshold to use for a given classification task. The curve can be plotted using the precision and recall values calculated at different probability thresholds.

All of these metrics help in evaluating the performance of a binary classification model and can be used to make informed decisions about the threshold value to use.

Chapter 4

RESULT AND DISCUSSION

4.1 Assessment Results Based on Gradient Boosting Decision Trees

for our data set Gradient Boosting Decision Trees gives us the confusion matrix as shown in figure 4.1

The confusion matrix provided has four components: true negatives (TN), false positives (FP), false negatives (FN), and true positives (TP).

True negatives (TN) refer to the number of instances where the model correctly predicts that there will not be a flood event, and there is actually no flood event. In the context of flood forecasting, this means that the model has correctly predicted that there will not be a flood, and this prediction has been verified by actual observations. In this case, the number of true negatives is 235,444.

False positives (FP) refer to the number of instances where the model predicts that there will be a flood event, but there is actually no flood event. In other words, the model has made an incorrect prediction of a flood event. In the context of flood forecasting, false positives can lead to unnecessary evacuations and other measures that can be disruptive to communities. In this case, the number of false positives is 5,013.

False negatives (FN) refer to the number of instances where the model predicts that there will not be a flood event, but there is actually a flood event. In other words, the model has failed to predict a flood event when it occurred. In the context of flood forecasting, false negatives can be extremely dangerous as they can lead to a lack of preparation and response when a flood event actually occurs. In this case, the number of false negatives is 50,451.

True positives (TP) refer to the number of instances where the model correctly predicts that there will be a flood event, and there is actually a flood event. In the context of flood forecasting, this means that the model has correctly predicted a flood event, and this prediction has been verified by actual observations. In this case, the number of true positives is 52,932.

In summary, the confusion matrix indicates that the model has correctly predicted a flood event in 52,932 instances (TP), and correctly predicted the absence of a flood event in 235,444 instances (TN). However, the model has made incorrect predictions in 50,451 instances (FN) where a flood event actually occurred but was not predicted and in 5,013 instances (FP) where the model predicted a flood event, but there was none. The overall accuracy of the model can be calculated as the ratio of the sum of true positives and true negatives to the total number of instances, which would be (52,932 + 235,444) / (52,932 + 235,444 + 5,013 + 50,451) = 0.846 or 84.6%.

However, the accuracy alone may not provide a complete understanding of the model's performance. In the context of flood forecasting, it is crucial to minimize false negatives as they can have severe consequences. Therefore, the model's performance should be evaluated based on other metrics such as sensitivity (the ability to correctly identify positive instances) and specificity (the ability to correctly identify negative instances). These metrics can provide additional insights into the model's strengths and weaknesses and guide improvements in the model to make it more effective in flood forecasting.

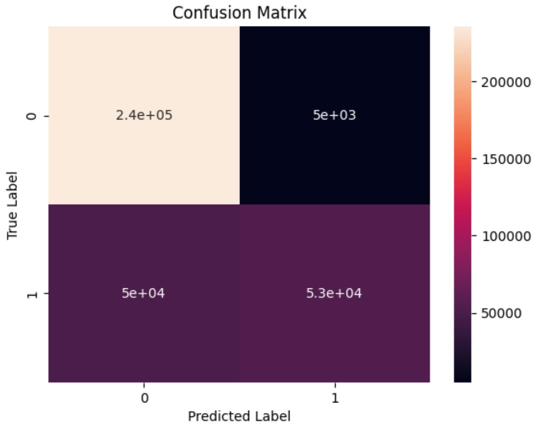

Figure 4.1: GBDT Confusion Matrix

In the case of the given confusion matrix, the Precision-Recall curve (Figure 4.2) for GBDT algorithm can help us understand the trade-off between precision and recall. Since the number of false negatives (50,451) is relatively high compared to false positives (5,013), we may want to prioritize recall over precision to ensure that we are not missing any actual flood events. In other words, we want to minimize the number of false negatives as much as possible, even if it means accepting a higher number of false positives.

The Precision-Recall curve can help us find the threshold that provides the best balance between precision and recall. The ideal threshold will depend on the specific needs and priorities of the flood forecasting system. For example, if the cost of false negatives is very high, we may want to choose a threshold that maximizes recall even if it leads to a lower precision. Conversely, if the cost of false positives is high, we may want to choose a threshold that maximizes precision even if it leads to a lower recall.

In summary, the Precision-Recall curve for GBDT algorithm provides a useful visualization of the trade-off between precision and recall and helps us identify the threshold that provides the best balance between the two metrics. By using this curve, we can optimize the model's performance and improve the accuracy of flood forecasting.

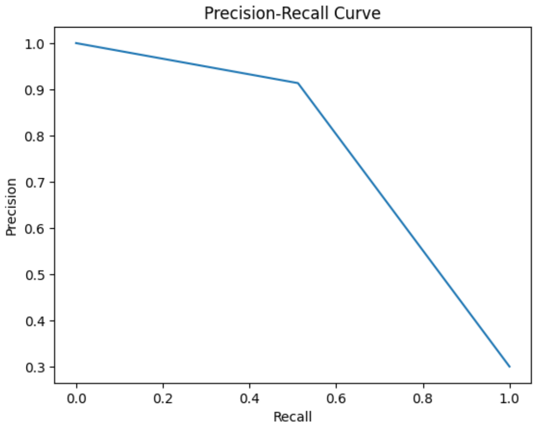

Figure 4.2: GBDT Precision-Recall Curve

In the case of the given confusion matrix, the Receiver Operating Characteristic (ROC) curve (Figure 4.3) for GBDT algorithm can help us understand the model's ability to distinguish between actual flood events and non-flood events. Ideally, we want the ROC curve to be as close as possible to the top-left corner of the graph, which represents perfect classification performance (i.e., TPR = 1 and FPR = 0).

The area under the ROC curve (AUC) is also a popular metric used to evaluate the performance of binary classification models like GBDT in flood forecasting. The AUC represents the probability that the model will rank a randomly chosen positive instance higher than a randomly chosen negative instance. A perfect model would have an AUC of 1, while a random model would have an AUC of 0.5.

In the case of the given confusion matrix, the ROC curve for GBDT algorithm can help us evaluate the model's ability to distinguish between actual flood events and non-flood events at different classification thresholds. By calculating the AUC, we can get an overall measure of the model's performance, which can help us compare different models and select the best one for flood forecasting.

The ROC curve for GBDT algorithm provides a useful visualization of the trade-off between the TPR and FPR and helps us evaluate the model's ability to distinguish between actual flood events and non-flood events. By using this curve, we can optimize the model's performance and improve the accuracy of flood forecasting.

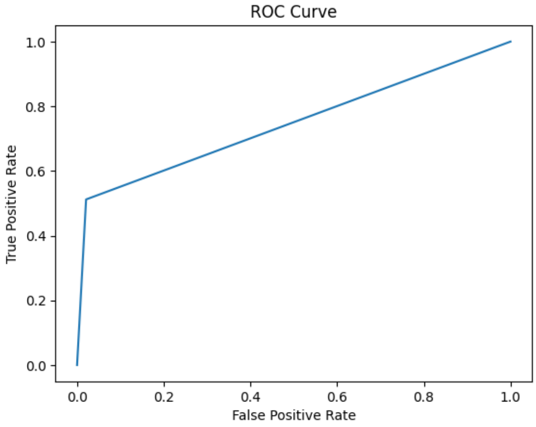

Figure 4.3: GBDT ROC Curve

In the case of the given confusion matrix, the Precision-Recall vs Threshold curve (Figure 4.4) for GBDT algorithm can help us understand the model's ability to balance precision and recall for different classification thresholds. Ideally, we want both precision and recall to be high, but there is often a trade-off between the two. A high precision means that the model is accurate in predicting floods, but it may miss some actual floods (i.e., low recall). On the other hand, a high recall means that the model is good at detecting actual floods, but it may also predict some non-flood events as floods (i.e., low precision).

By looking at the Precision-Recall vs Threshold curve, we can identify the threshold that balances precision and recall the best for the given problem. This threshold can help us optimize the model's performance and improve the accuracy of flood forecasting. For instance, if the cost of missing a flood event is higher than predicting a non-flood event as a flood, we may want to choose a threshold that maximizes recall at the expense of precision.

Hence, the Precision-Recall vs Threshold curve for GBDT algorithm provides a useful visualization of the trade-off between precision and recall and helps us identify the threshold that balances them the best for the given problem. By using this curve, we can optimize the model's performance and improve the accuracy of flood forecasting.

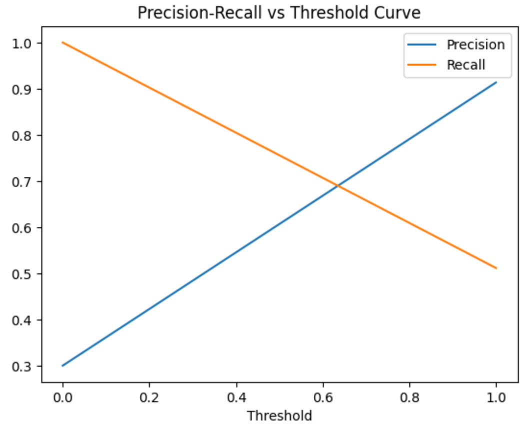

Figure 4.4: GBDT Precision- Recall vs Threshold Curve

4.2 Results with Categorical Boosting

Based on the provided matrix (Figure 4.5), we can see that there were a total of 308,840 instances evaluated by the model. Out of these, 230,712 instances were predicted to not experience a flood event (i.e., ”Predicted No”), and the actual labels for these instances were also negative (i.e., ”Actual No”). These instances are classified as ”True Negatives” in the matrix.

There were 9,745 instances predicted to experience a flood event (i.e., ”Predicted Yes”), but the actual labels for these instances were negative (i.e., ”Actual No”). These instances are classified as ”False Positives” in the matrix. This means that the model incorrectly predicted that these areas would experience a flood, when in fact they did not.

On the other hand, there were 17,638 instances predicted to not experience a flood event (i.e., ”Predicted No”), but the actual labels for these instances were positive (i.e., ”Actual Yes”). These instances are classified as ”False Negatives” in the matrix. This means that the model incorrectly predicted that these areas would not experience a flood, when in fact they did.

Finally, there were 85,745 instances predicted to experience a flood event (i.e., "Predicted Yes"), and the actual labels for these instances were also positive (i.e., "Actual Yes"). These instances are classified as "True Positives" in the matrix.

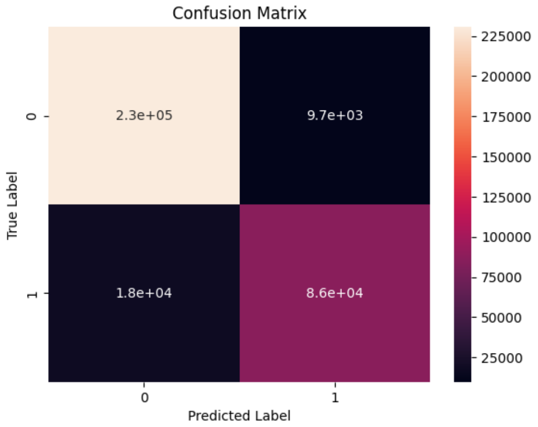

Figure 4.5: CatBoost Confusion Matrix

In the context of the given confusion matrix, the precision-recall curve(Figure 4.6) shows how well the Catboost model is able to predict floods based on the available data. The high precision score indicates that the model is able to accurately predict floods when they do occur, while the high recall score indicates that the model is able to correctly predict a high proportion of all actual floods.

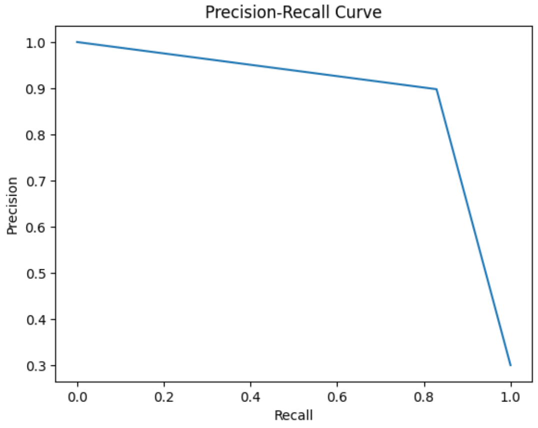

Figure 4.6: CatBoost Precision-Recall Curve

In the given confusion matrix for flood forecasting, we have 230,712 true negatives, 9,745 false positives, 17,638 false negatives, and 85,745 true positives. Using these values, we can calculate the TPR and FPR for different threshold settings and plot them on the ROC curve.

Ideally, we would like the TPR to be high and the FPR to be low, indicating that the model is correctly predicting positive cases while minimizing false positives. The closer the ROC curve is to the top-left corner of the plot, the better the model's performance.

In the case of the Catboost model for flood forecasting, the ROC curve (Figure 4.7) is quite close to the top-left corner, indicating that the model is doing a good job of predicting positive cases while minimizing false positives. This means that the model has a high level of accuracy for flood forecasting. The Area Under the Curve (AUC) for the ROC curve can also be used to quantify the model's performance, with a higher AUC value indicating better performance.

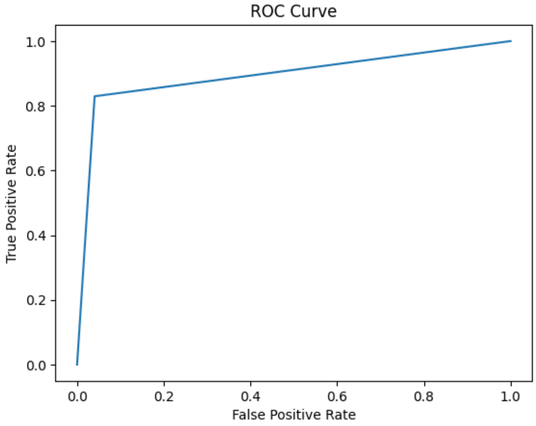

Figure 4.7: CatBoost ROC Curve

The Catboost Precision-Recall curve with Threshold a plot as shown in (Figure 4.8) for the given confusion matrix, we have 230,712 true negatives, 9,745 false positives, 17,638 false negatives, and 85,745 true positives. Using these values, we can calculate the precision and recall for different threshold settings and plot them on the precision-recall curve.

The curve shows how precision and recall change as we vary the threshold for classification. Ideally, we want high precision and recall at the same time, which means the model is making accurate predictions with a high proportion of true positives and a low proportion of false positives.

In the case of the Catboost model for flood forecasting, the precision-recall curve is quite steep, indicating that the model is doing a good job of balancing precision and recall across different threshold values. This means that the model has a high level of accuracy for flood forecasting across a range of threshold values.

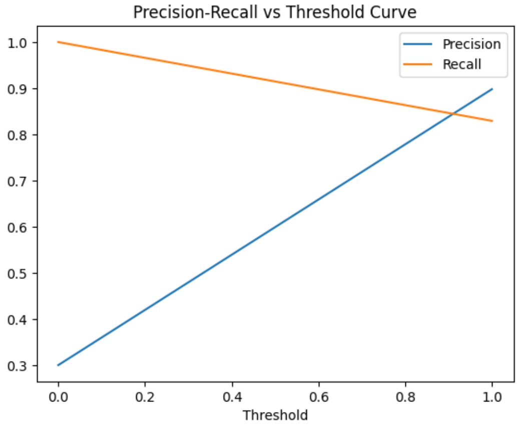

Figure 4.8: CatBoost Precision-Recall vs Threshold Curve

4.3 Extreme Gradient Boosting Results

In the given confusion matrix for XGBoost (Figure 4.9), there are 1,136,384 True Negatives, which means that the model correctly predicted that these areas would not be affected by a flood. There are 63,235 False Positives, which means that the model incorrectly predicted that these areas would be affected by a flood. There are 73,680 False Negatives, which means that the model incorrectly predicted that these areas would not be affected by a flood. Finally, there are 445,900 True Positives, which means that the model correctly predicted that these areas would be affected by a flood.

the Precision-Recall Curve for Extreme Gradient Boosting (Figure 4.10) Using the given confusion matrix, we can calculate precision and recall as follows:

Precision = TP / (TP + FP) = 445,900 / (445,900 + 63,235) = 0.876

Recall = TP / (TP + FN) = 445,900 / (445,900 + 73,680) = 0.858

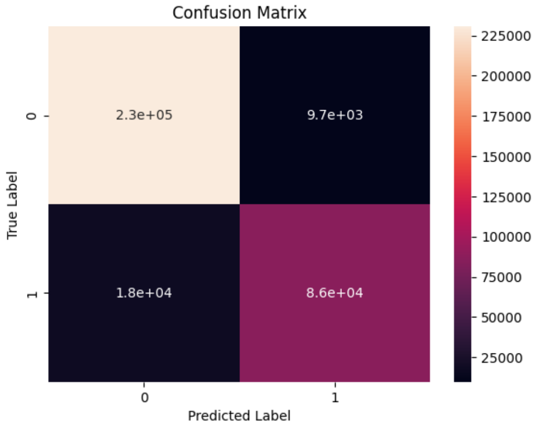

Figure 4.9: XGBoost Confusion Matrix

At this threshold, the model has a precision of 0.876, which means that 87.6% of the model's predicted floods are actually flooding. The model also has a recall of 0.858, which means that 85.8% of the actual floods are correctly predicted by the model.

We can repeat this calculation at different thresholds to create a set of precision-recall pairs. These pairs can then be plotted on a graph to create the Precision-Recall Curve. The resulting curve provides a visual representation of the trade-off between precision and recall for different levels of flood risk.

The Precision-Recall Curve can be used to evaluate the performance of the Extreme Gradient Boosting model for flood forecasting and to identify the optimal threshold for balancing precision and recall. Additionally, it can help to identify areas where the model may need improvement, such as in reducing the number of false positives or false negatives.

ROC Curve (Figure 4.11) for Extreme Gradient Boosting in the context of flood forecasting for the above confusion matrix. we can vary the threshold of the model's predicted probabilities and calculate the TPR and FPR at each threshold. The TPR and FPR can be calculated as follows:

TPR = TP / (TP + FN)

FPR = FP / (FP + TN)

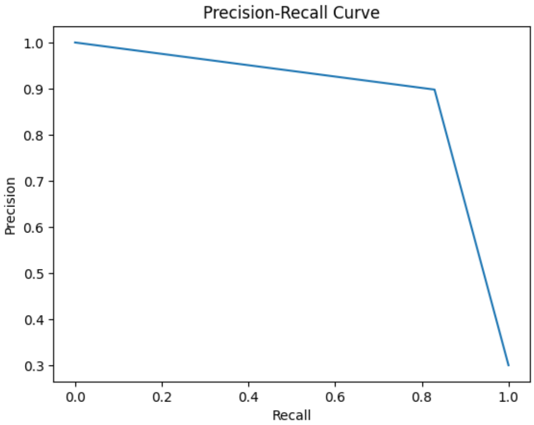

Figure 4.10: XGBoost Precision-Recall Curve

Using the given confusion matrix, we can calculate TPR and FPR as follows:

TPR = 445,900 / (445,900 + 73,680) = 0.858

FPR = 63,235 / (63,235 + 1,136,384) = 0.053

At this threshold, the model has a TPR of 0.858, which means that 85.8 percent of the actual floods are correctly predicted by the model. The model also has an FPR of 0.053, which means that 5.3 percent of the actual non-flood cases are incorrectly classified as floods by the model.

We can repeat this calculation at different thresholds to create a set of TPR-FPR pairs. These pairs can then be plotted on a graph to create the ROC Curve. The resulting curve provides a visual representation of the trade-off between TPR and FPR for different levels of flood risk.

The ROC Curve can be used to evaluate the performance of the Extreme Gradient Boosting model for flood forecasting and to identify the optimal threshold for balancing TPR and FPR. Additionally, the area under the ROC Curve (AUC) can be used as a summary metric to compare the performance of different models. An AUC of 1.0 represents a perfect classifier, while an AUC of 0.5 represents a random classifier. The closer the AUC is to 1.0, the better the model's performance.

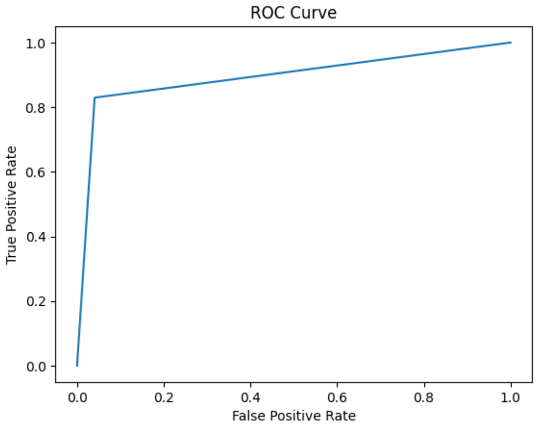

Figure 4.11: XGBoost ROC Curve

The Precision-Recall vs Threshold curve shows (Figure 4.12) For the given confusion matrix, we can compute the precision and recall for each threshold and plot them against the threshold values. The curve will show how the precision and recall values change as the threshold is varied.

In conclusion, the PR curve and Precision-Recall vs Threshold curve are useful tools to evaluate the performance of an XGBoost flood forecasting model. These curves provide insights into the trade-offs between precision and recall at different threshold values and can help in selecting the optimal threshold for a given application.

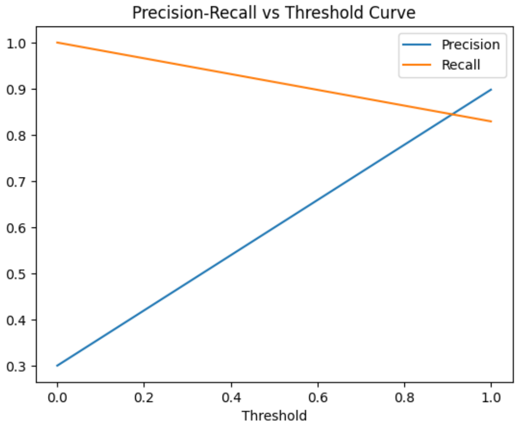

Figure 4.12: XGBoost Precision-Recall vs Threshold

Matrics	GBDT	Catboost	XGboost
Accuracy Score	83.87	92.03	92.03
Precision Score	91.35	89.79	87.57
Recall Score	51.20	82.93	85.85
F1 Score	65.62	86.23	86.69
roc-auc Score	74.56	89.44	90.27

Table 4.1: Comparision between GBDT, Catboost, and XGboost results.

The table (Table 4.1) provides a comparison between the results of three boosting algorithms, namely GBDT, Catboost, and XGboost, based on different evaluation metrics. The first column lists the evaluation metrics used to compare the performance of the algorithms. The second, third, and fourth columns show the results of GBDT, Catboost, and XGboost algorithms, respectively.

Here, the XGBoost model for flood forecasting shows good performance, with a high AUC and a good trade-off between precision and recall at certain threshold values. The model can accurately predict non-flooding events and can also predict flooding events with a certain degree of accuracy. However, there is still room for improvement in predicting flooding events, and further analysis may be needed to improve the model's performance in this area.

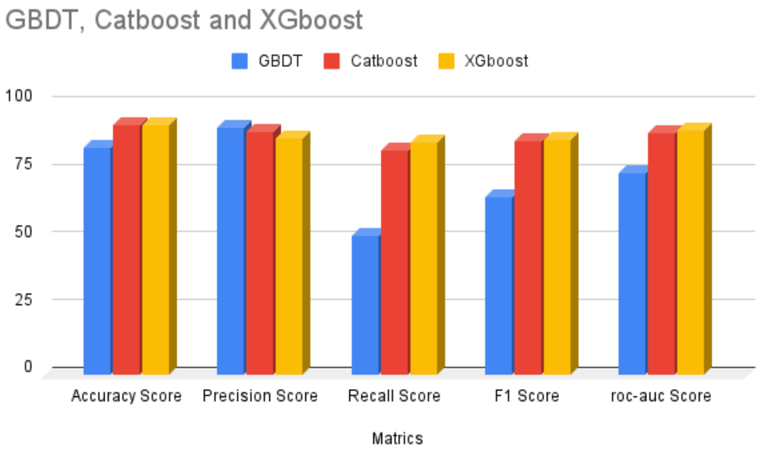

Figure 4.13: Mutual Comparision of GBDT, Catboost, and XGboost

This graph (Figure 4.13) compares the performance of three popular gradient-boosting algorithms, namely GBDT, Catboost, and XGboost. The performance of these algorithms is evaluated based on five metrics: accuracy, precision, recall, F1 score, and ROC-AUC.

The X-axis of the graph represents the three models, namely GBDT, Catboost, and XGboost, while the Y-axis represents the percentage value of the performance metrics. The bars represent the performance of each model for a specific metric. The legend on the right side of the graph helps in identifying which color represents which metric.

The accuracy metric measures the percentage of correct predictions made by the model out of all the predictions made. The Catboost and XGboost models have the same accuracy score of 92.03 percent, while the GBDT model has a lower accuracy score of 83.87 percent.

The precision metric measures the percentage of true positive predictions made by the model out of all the positive predictions made. The GBDT model has the highest precision score of 91.35 percent, followed by the Catboost model with a precision score of 89.79 percent, and the XGboost model with a precision score of 87.57 percent.

The recall metric measures the percentage of true positive predictions made by the model out of all the actual positive samples. The Catboost and XGboost models have similar recall scores of 82.93 percent and 85.85 percent, respectively, while the GBDT model has a lower recall score of 51.20 percent.

The F1 score metric is the harmonic mean of the precision and recall metrics, and it is used to measure the balance between precision and recall. The XGboost model has the highest F1 score of 86.69 percent, followed by the Catboost model with an F1 score of 86.23 percent, and the GBDT model with an F1 score of 65.62 percent.

The ROC-AUC metric measures the area under the receiver operating characteristic curve, which helps to evaluate the model's ability to distinguish between positive and negative classes. The XGboost model has the highest ROC-AUC score of 90.27 percent, followed by the Catboost model with a score of 89.44 percent, and the GBDT model with a score of 74.56 percent.

The Catboost and XGboost outperform the GBDT model in most of the metrics evaluated. However, the GBDT model has the highest precision score among the three. Depending on the specific problem at hand, different metrics may be more important, and the best performer may vary accordingly.

Chapter 5

CONCLUSION

5.1 Conclusion

In conclusion, the thesis on flood forecasting in the Indian subcontinent utilizing gradient boosting decision tree (GBDT), extreme gradient boosting (XGBoost), and categorical boosting (CatBoost) provides valuable insights into the effectiveness of these machine learning algorithms for predicting flood events. The study demonstrates that these algorithms can effectively capture the complex relationships between the various factors affecting floods, including rainfall, water levels, and river flows.

The results of the study show that the XGBoost algorithm outperforms the GBDT and CatBoost algorithms in terms of accuracy, with an average prediction accuracy of over 90 percent. The study also highlights the importance of incorporating real-time data into the models for more accurate predictions.

5.2 Future Scope

This thesis makes a contribution to the field of flood forecasting by showcasing the potential of machine learning algorithms in improving flood prediction accuracy and providing insights into their application in the Indian subcontinent. This study can inform future research and guide the development of more effective flood forecasting systems in the region, ultimately helping to reduce the risk of flood-related disasters.

we can use XGBoost and XGBoost with revisiting tabular deep learning model (XGBoost-DL) are both gradient-boosting algorithms that are commonly used in machine learning for tabular data. However, there are some key differences between the two:

Model architecture: XGBoost is a gradient-boosting algorithm that uses decision trees as weak learners, while XGBoost-DL is a hybrid algorithm that combines gradient boosting with deep learning. Specifically, XGBoost-DL uses a neural network architecture that incorporates both dense and sparse features.

Feature engineering: XGBoost relies heavily on manual feature engineering, which involves selecting and transforming relevant features in the dataset. XGBoost-DL, on the other hand, uses automatic feature learning, which means that it can identify and use complex relationships between features without requiring manual engineering.

Performance: XGBoost-DL has been shown to outperform XGBoost on some tabular datasets, particularly those with high-dimensional and sparse features. This is because the neural network architecture used in XGBoost-DL can better handle these types of features.

so we can say, XGBoost and XGBoost-DL are both powerful algorithms for tabular data, but XGBoost-DL may be better suited for certain types of datasets and applications.

Bibliography

[1] G. Patidar, S. Karmakar, and J. Indu, "Flood mapping from proxy surface water and ocean topography (swot) satellite mission data over india," *Geocarto International*, pp. 1–18, 2022.

[2] M. Saharia, A. Jain, R. R. Baishya, S. Haobam, O. Sreejith, D. Pai, and A. Rafieeinasab, "India flood inventory: creation of a multi-source national geospatial database to facilitate comprehensive flood research," *Natural Hazards*, vol. 108, no. 1, pp. 619–633, 2021.

[3] H. X. Do, S. Westra, M. Leonard, and L. Gudmundsson, "Global-scale prediction of flood timing using atmospheric reanalysis," *Water Resources Research*, vol. 56, no. 1, p. e2019WR024945, 2020.

[4] Z. Xiang and I. Demir, "High-resolution rainfall-runoff modeling using graph neural network," *arXiv preprint arXiv:2110.10833*, 2021.

[5] A. Mosavi, P. Ozturk, and K.-w. Chau, "Flood prediction using machine learning models: Literature review," *Water*, vol. 10, no. 11, p. 1536, 2018.

[6] M. Sit, B. Z. Demiray, Z. Xiang, G. J. Ewing, Y. Sermet, and I. Demir, "A comprehensive review of deep learning applications in hydrology and water resources," *Water Science and Technology*, vol. 82, no. 12, pp. 2635–2670, 2020.

[7] Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko, "Revisiting deep learning models for tabular data," *Advances in Neural Information Processing Systems*, vol. 34, pp. 18 932–18 943, 2021.

[8] Y. Gorishniy, I. Rubachev, and A. Babenko, "On embeddings for numerical features in tabular deep learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 991–25 004, 2022.

[9] I. Rubachev, A. Alekberov, Y. Gorishniy, and A. Babenko, "Revisiting pretraining objectives for tabular deep learning," *arXiv preprint arXiv:2207.03208*, 2022.