

Flood Prediction Using Machine Learning Algorithms Report

Submitted in partial fulfillment of the requirements
for the award of Bachelor of Engineering Degree in
Computer Science and Engineering

By

Yogesh R (38110672)

Vishwa A (38110652)



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING**

SCHOOL OF COMPUTING

**SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600

119

TAMILNADU.

MARCH - 2022

**SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of Yogesh R (38110672) and Vishwa A (38110652) who carried out the project entitled " Flood Prediction Using Multiple Machine Learning Algorithms " under our supervision from ____to____

Internal Guide

Ms. VINODHINI ASST.PROFFESOR

Head of the Department

Dr.S. VIGNESHWARI, M.E, Ph.D.

Submitted for Viva voce Examination held on

Internal Examiner

External Examiner

DECLARATION

I Yogesh R (Reg.no 38110672) hereby declare that the Project Report entitled” Flood Prediction Using Multiple Machine Learning Algorithms” done by me under the guidance of Ms. Vinodhini Asst. Professor is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer science.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of SATHYABAMA for their kind encouragement in doing this project and for completing it successfully.

I am grateful to them. I convey my thanks to Dr.T.SASIKALA, Dean, School of Computing, Dr. L. LAKSHMANAN M.E, Ph.D. and Dr.S.VIGNESHWARI,M.E.,Ph.D., Head Of the Department, and Dept. of Computer Science for providing me necessary Support and details at the time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to My Project Guide Ms. VINODHINI ASST.PROFFESOR , for her valuable guidance, Suggestions and constant encouragement paved way to the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-Teaching staff members of the Department of Computer Science who were helpful in many ways for the completion of my project.

ABSTRACT

Floods are among the most destructive natural disasters, which are highly complex to model. The research on the advancement of flood prediction models contributed to risk reduction, policy suggestion, minimization of the loss of human life, and reduction of the property damage associated with floods. To mimic the complex mathematical expressions of physical processes of floods, during the past two decades, machine learning (ML) methods contributed highly in the advancement of prediction systems providing better performance and cost-effective solutions. Due to the vast benefits and potential of ML, its popularity dramatically increased among hydrologists. Researchers through introducing novel ML methods and hybridizing of the existing ones aim at discovering more accurate and efficient prediction models. The main contribution of this project is to demonstrate the state of the art of ML models in flood prediction and to give insight into the most suitable models. This project presents the Flood prediction and Rainfall analysis using Machine Learning. The main goal of employing this application is to prevent impacts of flood. This application can be easily used by the common people or government to predict the occurrence of flood beforehand. Among many ML techniques, classification is a widely used one. We use multiple algorithms such as Logistic Regression, Naive Bayes, K-Nearest Neighbors and Decision Tree Classifier.

TABLE OF CONTENTS

Chapter number	Title	Page number
1	Introduction	
1.1	Python	
1.2	Outline of the project	
1.3	Problem Statement	
2	Literature Survey	
3	Methodology	
3.1	Work Flow	
3.2	Dataset and Algorithm implementation	
4	Result and Discussion	
5	Conclusion and Future work	
	References	
	Appendix	

CHAPTER-1

INTRODUCTION

The unusual rainfall and global climate change has led to floods in different parts of the world. Floods are one of the worst affecting natural phenomena which causes heavy damage to property, infrastructure and most importantly human life. This paper presents the Flood prediction and Rainfall analysis using Machine Learning. The main goal of employing this application is to prevent immediate impacts of flood. This application can be easily used by the common people or government to predict the occurrence of flood beforehand. Nowadays, machine learning (ML) methods are highly contributed in the advancement of prediction systems. These methods are providing better performance as well as cost effective solutions. The advancement in the information technology, the need for easy accessibility of large cloud storage and processing power is available. Data mining technologies helps us to provide reference for decision makers as summarized information even from the large amount of data. Among many data mining techniques, classification is a widely used one. Past studies proposed many techniques that could be applied to classification, such as decision trees, neural networks, Bayesian classifiers. We use multiple algorithms such as Logistic Regression, Naive Bayes, K-Nearest Neighbors and Decision Tree Classifier using Python environment.

PYTHON:

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

It is used for:

- web development (server-side),
- software development,

- mathematics,
- System scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-orientated way or a functional way.

Python Syntax compared to other programming languages

- Python was designed to for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.

- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

1.1 OUTLINE OF THE PROJECT:

Floods are among the most destructive natural disasters, which are highly complex to model. The main contribution of this project is to demonstrate the state of the art of ML models in flood prediction and to give insight into the most suitable models. This project presents the Flood prediction and Rainfall analysis using Machine Learning. The main goal of employing this application is to prevent impacts of flood. This application can be easily used by the common people or government to predict the occurrence of flood beforehand. Among many ML techniques, classification is a widely used one. We use multiple algorithms such as Logistic Regression, Naive Bayes, K-Nearest Neighbors and Decision Tree Classifier.

1.2 PROBLEM STATEMENT:

- Flood must be predicted beforehand
- Use Machine Learning technique
- Programming language- Python
- Algorithms- 4

CHAPTER-2

LITERATURE SURVEY

- Chowdhury . approached with finding the collinearity of Sea-surface temperature (SST) with a floodaffected area (FAA). They found the statistical relationship using principal component analysis. To construct a model to predict flood,

multiple regression analysis was done. The statistical model uses SST, rainfall, and streamflow in Bangladesh to serve as predictors.

- Shafizadeh-Moghadam have implemented eight different machine learning and statistical model and seven ensemble models for assessing flood susceptibility. Their dataset consists of 201 flood incidents in the Haraz watershed. They recognized eleven factors that contribute to flooding events. According to their evaluation parameters, the eight machine learning models: Artificial neural networks (ANN), Classification and regression trees (CART), Flexible discriminant analysis (FDA), Generalized linear model (GLM), Generalized additive model (GAM), Boosted regression trees (BRT), Multivariate adaptive regression splines (MARS) and Maximum entropy (MaxEnt) produce the Area under the ROC curve (AUC) scores of 0.920, 0.643, 0.822, 0.971, 0.962, 0.975, 0.941 and 0.971 respectively.
- Han applied SVM models to forecast floods. They experimented with different kernel tricks and various input combinations. To assess the effectiveness of new models, they compared it with the Naive model, Trend model, and Transfer function (TF) model. For their study, they used the gamma values of 0.001, 0.01, 0.03, 0.05, 0.07, 0.09, 0.1, 0.3, 0.5, 0.7 and 0.9. They applied 5-fold cross-validation for the training and testing phase. It was found that SVM predicted flood with higher accuracy in comparison to the TF model. Their observation showed that the linear function performs better with unknown future rainfall input. Extremely large rainfall data that is not scalable makes the model quite unstable.
- Noymanee attempted to forecast pluvial floods using machine learning techniques. For this purpose, they used open data of the basin of Pattani River, Thailand. They used neural network (NN), bayesian linear (BL), boosted decision tree (BDT), decision forest (DF) and linear regression (LR). They split their dataset into two parts. They used the 80% of dataset for training the model and 20% dataset to test them. For the purpose of examining their models, they used RMSE, MAE and efficiency index (EI). They concluded that BL method provided the most efficient and effective result for a long-time lag. The other models also provided good results that were only acceptable for short time lag prediction.
- Khosravi did a comparative study on decision trees algorithm to create flash flood susceptibility model. The mapping was done at the Haraz Watershed, Iran.

Logistic model trees (LMT), reduced error pruning trees (REPT), Naive Bayes Trees (NBT) and Alternating Decision Trees (ADT) were used for the study. The historical data of 201 flash floods were used. For feature selection, they used information gain ratio and multicollinearity diagnostics method. 70% of the dataset was used for training the model and 30% of it was used to test the model. Statistical evaluation measures, the ROC curve and Freidman and Wilcoxon signed-rank tests were used to evaluate the performances of the four models. After analyzing the results, it was seen that ADT model performed best.

- Duncan proposed a Machine Learning-Based Early Warning System for Urban Flood Management. It works with drainage systems and the use of overall rainfall data concurrently, in order to predict flooding of multiple urban zones using a single multioutput ANN. In their study, there are three input time-series used: cumulative rainfall (mm), rainfall intensity (mm/hour), and the new antecedent precipitation index value (meters). The principal goal of their research is to find the true positive rates for sensitivity analysis. The research findings show that the predictive ability of the system depends on actual rainfall.
- Li established an ANN model to predict the trend of storm flooding in China. The ANN model was used to predict the storm flooding and built to simulate the historical storm surge during typhoon periods. The results showed that the ANN model provides stronger results than other models for flood prediction.
- Xia, and Rao designed a new application that utilized a Basis Prediction artificial (BP- artificial) neural network model to predict the flood level in lakes, rivers, and reservoirs. The errors between predicted and actual value are feedback in the process of learning. Their research primarily compares the data calculated by the model to actual monitoring data from the monitoring station from the city of Chaoan; with the result showing that the algorithm can obtain a better prediction. They compared the values calculated by the algorithm to actual monitoring data from monitoring control centre; the result shows that the BP- artificial neural network model can achieve better prediction in terms of accuracy and credibility, in comparison with other neural network architectures.
- Trombe P. have discussed about the Automatic Classification of Offshore Wind Regimes with Weather Radar Observations. In this paper, the automatic classification of offshore and wind regimes (i.e., wind fluctuations with specific frequency and amplitude) with the help of reflectivity observations from single

weather radar system has been addressed. Spatial continuity, motion of precipitation echoes on the images and global intensity are described from the above attributes. Finally, classification tree was used to find the relationship between wind regimes and precipitation attributes.

- Chawla G. have discussed about the big data analysis which will help in getting a view of the given dataset. In this paper the researchers focuses on Big Data visualization, its challenges, various tools. Researchers have explored a new way to visualize and analyse complex and dynamic datasets using virtual reality. They have also founded that how virtual reality has extremely changed the world of Big Data Visualisation.
- Mariana Belgiu used a technique that compared unsupervised multiresolution segmentation approaches with that of supervised approaches by extracting buildings from images. The classification yielded an overall accuracy that ranged between 82% to 86%. Particle Swarm Optimization was used in dynamic clustering for image segmentation. The approach was called as DCPSO which identified the best possible number of clusters from the tested images.
- Krishna Kant Singh identified flooded area from satellite images using a classifier performance analysis was done by computing the error matrix. Several methods have been proposed by various researchers for classification of only satellite images but not for aerial images.

●

CHAPTER-3

METHODOLOGY

Initially, the data was collected from Kaggle and we filtered and checked the precipitation values for null values and other common errors. Now the data is fed to the machine learning algorithms,

(i) Logistic Regression

(ii) Naive Bayes

(iii) K-Nearest Neighbors

(iv) Decision Tree Classifier and prediction of the occurrence of flood for the particular date and area is calculated based on the risk levels.

The Risk levels decide how bad the flood will affect and we use them for the forecasting of floods. The low level is defined as level 0 flood. Level 1 and level 2 have a moderate and medium-high risk for the flood event and level 3 is the High alert and level 4 is for the very high flood risk.

level 0 =Low

level 1 =Medium

level 2 =Medium-High

level 3 =High

level 4 =Very High

Libraries used:

NumPy - NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices. In Python we have lists that serve the purpose of arrays, but they are slow to process. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.

Pandas - Pandas stands for "Python Data Analysis Library". What's cool about Pandas is that it takes data (like a CSV or TSV file, or a SQL database) and creates a Python object with rows and columns called data frame that looks very similar to table in a statistical software. This is so much easier to work with in comparison to working with lists and/or dictionaries through for loops or list comprehension.

Matplotlib - Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits. Pyplot is a Matplotlib module which provides a MATLAB-like interface. Matplotlib is designed to be as usable as MATLAB, with the ability to use Python, and the advantage of being free and open-source.

Date time - The datetime module supplies classes for manipulating dates and times. While date and time arithmetic is supported, the focus of the implementation is on efficient attribute extraction for output formatting and manipulation. Date and time objects may be categorized as "aware" or "naive" depending on whether or not they include time zone information. With sufficient knowledge of applicable algorithmic and political time adjustments, such as time zone and daylight saving time information, an aware object can locate itself relative to other aware objects. An aware object represents a specific moment in time that is not open to interpretation. A naive object does not contain enough information to unambiguously locate itself relative to other date/time objects. Whether a naive object represents Coordinated Universal Time (UTC), local time, or time in some other time zone is purely up to the program, just like it is up to the program whether a particular number represents meters, miles, or mass.

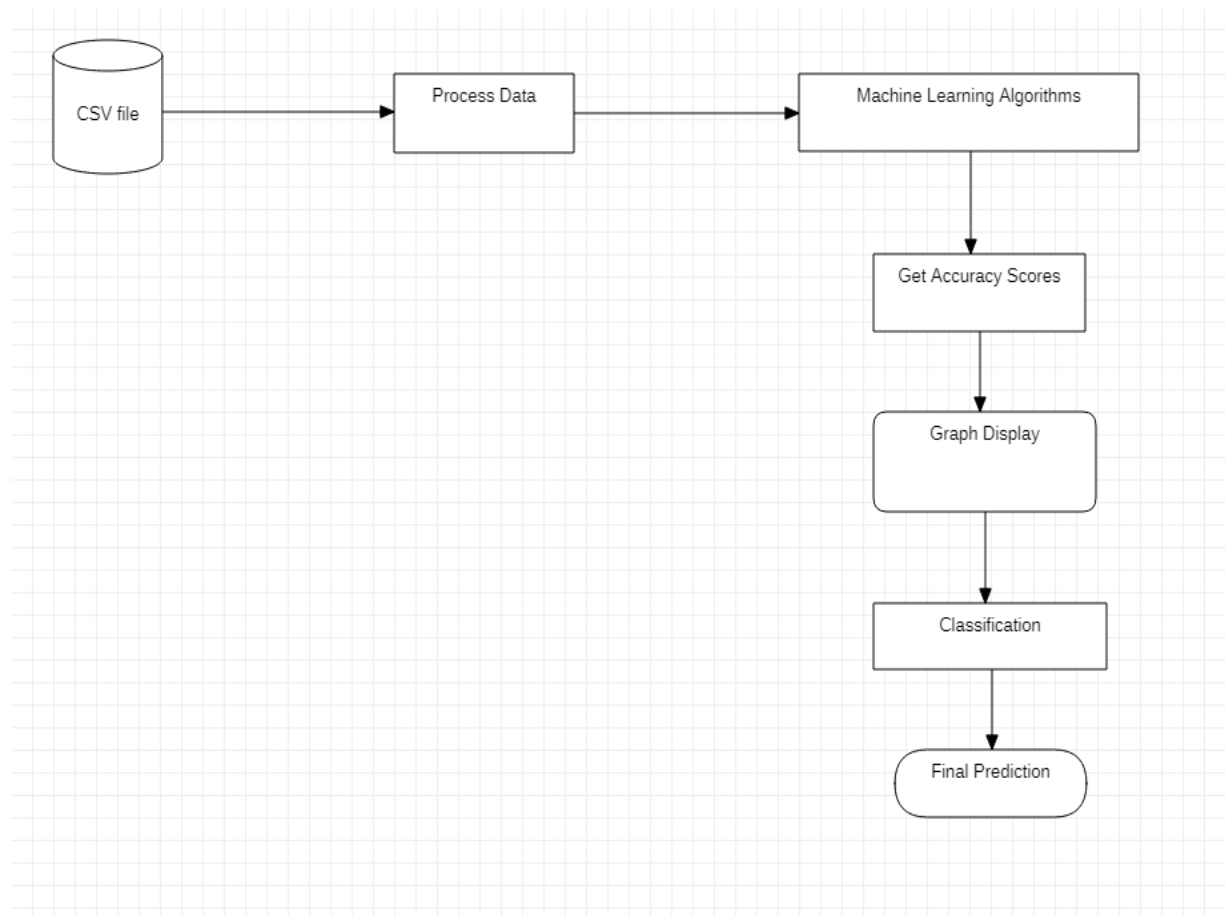
Naive objects are easy to understand and to work with, at the cost of ignoring some aspects of reality.

Seaborn - Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures. Seaborn helps you explore and understand your data. Its plotting functions operate on data frames and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots. Its dataset-oriented, declarative API lets you focus on what the different elements of your plots mean, rather than on the details of how to draw them.

Keras - Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library. Keras contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code. Keras Models provide pre-trained models, which could easily integrated into projects.

Sklearn - sklearn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. sklearn is largely written in Python, and uses NumPy extensively for high-performance linear algebra and array operations. Furthermore, some core algorithms are written in Cython to improve performance.

3.1 WORKFLOW:



DATA COLLECTION AND PREPROCESSING PHASE:

Weather dataset is collected in a comma separated values (CSV) file for the last three or four years of rainfall information. The dataset contains the month wise aggregation. The presence of irrelevant, redundant information, noisy, and unreliable data significantly affects the model outcomes and knowledge discovery, making the training phase more difficult. Data preprocessing is an iterative process for the transformation of the raw data into understandable and useable forms. Raw datasets are usually characterized by incompleteness, inconsistencies, lacking in behavior, and trends while containing errors. The preprocessing is essential to handle the missing values and address inconsistencies. The preprocessing methods involve of removing records which is not complete. Once the clean dataset is available, we have to prepare it to feed to the machine learning algorithm.

IMPLEMENTATION PHASE:

The prediction accuracy of the different models is evaluated using data validation, and the results are compared to get accuracy. We use multiple algorithms to get high accuracy range of prediction. A preprocessed dataset are processed in different machine learning algorithms. Each algorithm gives some accuracy level. Each one undergoes for the comparison.

- (i) Logistic Regression-It is used to model the likelihood of a given class or event, such as pass/fail or true/false.
- (ii) Naive Bayes-It is a family of algorithms that all share a similar concept, namely, that every pair of features being classified is independent of each other.
- (iii) K-Nearest Neighbors-It is a supervised learning algorithm that is widely used in pattern recognition, data mining, and intrusion detection.
- (iv) Random Forest-It combines the output of numerous decision trees to reach a single result; its ease of usage and versatility, as well as its ability to handle both classification and regression problems, have propelled its adoption.

PREDICTION PHASE:

Preprocessed data are trained and input given by the user goes to the trained dataset. Then the prediction is made for chances of flood in a particular date and city. Prediction refers to the output of an algorithm after it has been trained on a historical dataset and applied to new data when forecasting the likelihood of a particular outcome, such as whether flood will occur or not. The algorithms will generate probable values for an unknown variable for each record in the new data, allowing us to identify what that value will most likely be. Machine learning model predictions allow businesses to make highly accurate guesses as to the likely outcomes of a question based on historical data, which can be about all kinds of things – customer churn likelihood, possible fraudulent activity, and more

in our case it is occurrence of flood. Python matplotlib library can be used to produce the graphical representation of the data in a visual format.

3.2 DATASET AND ALGORITHM IMPLEMENTATION:

Initially, the dataset was collected from Kaggle, it contains major attributes like LOCATION, SUBDIVISION, YEAR, ANNUAL PRECIPITATION, QUARTERLY, PRECIPITATION, SEVERITY, TERRAIN and CLIMATE. Preprocessed test data is fed to the algorithms to get the accuracy scores and the algorithm with best accuracy is chosen to display the prediction output.

(i) Logistic Regression

Logistic regression is a process of modeling the probability of a discrete outcome given an input variable. The most common logistic regression models a binary outcome; something that can take two values such as true/false, yes/no, and so on. Multinomial logistic regression can model scenarios where there are more than two possible discrete outcomes.

Logistic regression is a useful analysis method for classification problems, where you are trying to determine if a new sample fits best into a category. As aspects of cyber security are classification problems, such as attack detection, logistic regression is a useful analytic technique.

(ii) Naive Bayes

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon

the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'. Naïve Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods. Bayes theorem provides a way of calculating posterior probability $P(c|x)$ from $P(c)$, $P(x)$ and $P(x|c)$.

$P(c|x)$ is the posterior probability of class (c, target) given predictor (x, attributes).

$P(c)$ is the prior probability of class.

$P(x|c)$ is the likelihood which is the probability of predictor given class.

$P(x)$ is the prior probability of predictor.

(iii) K-Nearest Neighbors

The K-Nearest Neighbor Algorithm (or KNN) is a popular supervised machine learning algorithm that can solve both classification and regression problems. The algorithm is quite intuitive and uses distance measures to find k closest neighbors to a new, unlabeled data point to make a prediction. Because of this, the name refers to finding the k nearest neighbors to make a prediction for unknown data. The K-Nearest Neighbor algorithm works by calculating a new data points class (in the case of classification) or value (in the case of regression) by looking at its most similar neighbors.

(iv) Decision Tree Classifier

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. The decisions or the test are performed on the basis of features of the given dataset. It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions. It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure. In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm and decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.

CHAPTER-4

RESULT AND DISCUSSION

The system is implemented with python environment with the help of 'sk-learn' library.sk-learn is an open-source machine learning library that supports supervised and unsupervised learning. It also provides various tools for model fitting, data preprocessing, model selection, model evaluation, and many other utilities. After feature selection and scaling, the training data set is fed to the Logistic Regression, K Nearest neighbors, Naive Bayes and Random Forest models and the accuracy is calculated. From the observation it is noted that Random Forest algorithms performs better than the other algorithms. The accuracy scores are:

Logistic Regression: 84.3205574912892
K Nearest neighbors: 87.45644599303137
Naive Bayes: 84.20441347270615
Random Forest: 94.07665505226481

CHAPTER-5

CONCLUSION AND FUTURE WORK

It is very important for India to predict flood as accurately as possible because only then, we will be able to alleviate the after-effects of floods. This Flood risk prediction system has been developed using machine learning that eliminates the extra efforts involved in manual estimation and provides the desired precision for decision-making. And we feel this could be the best prediction system if precision and accuracy are the highest priority. Former studies have revealed that Machine learning algorithms show considerable effectiveness for the preprocessing of environmental time-series datasets, as a precursor to the classification of flood data. For future improvement, we consider using other types of machine learning techniques, including the use of global optimization algorithms, for instance, the genetic algorithm that can expand the scope and scale of this study. In the future, this system will also be deployed on the web with additional features like sending warning SMS to authorities before the time that can help them plan safety measures for the people.

REFERENCES:

- [1] A. M. Kamal, M. Shamsudduha, B. Ahmed, S. K. Hassan, M. S. Islam, I. Kelman, and M. Fordham, "Resilience to flash floods in wetland communities of northeastern bangladesh," *International journal of disaster risk reduction*, vol. 31, pp. 478-488, 2018.
- [2] H. Shafizadeh-Moghadam, R. Valavi, H. Shahabi, K. Chapi, and A. Shirzadi, "Novel forecasting approaches using combination of machine learning and statistical models for flood susceptibility mapping," *Journal of environmental management*, vol. 217, pp. 1-11, 2018.
- [3] L. Xiaodong, G. Zhongyang, X. Ran, D. Xiaoyan, H. Yizhi, and Y. Shufeng, "Application of Artificial Neural Network in the Typhoon Flood Prediction System-A Case Study in Shanghai, China," in *Artificial Intelligence and Computational Intelligence (AICI), 2010 International Conference on*, 2010, vol. 1, pp. 191-195: IEEE.
- [4] H. Xia and Q. Rao, "Flood Level Prediction on the Basis of the Artificial Neural Network," in *2009 First International Conference on Information Science and Engineering*, 2009, pp. 4887-4890: IEEE.
- [5]. Tripathi G., Gupta R., "A New Approach for Rainfall Prediction using Artificial Neural Network", *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, Volume 7 Issue VII, July 2019
- [6]. Bengio Y., Courville A., Vincent P., "Representation learning: A review and new perspectives," *Institute of Electrical and Electronics Engineering*, vol. 35, 2013.
- [7]. Jordan M., Mitchell T., "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, PP. 255-260, 2015.
- [8]. Chakraborty S., Nagwani N., Dey L., "Weather Forecasting using Incremental K-means Clustering", *International Conference in High Performance Architecture and Grid Computing*, Vol.169, Part-2, PP:338- 341,2011.
- [9]. Fuchs R., Waser J., Groller M., "Visual human with machine learning", *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1327–1334, 2009.

- [10]. Trombe P., and Pinson P., "Automatic classification of offshore wind regimes with weather radar observations", IEEE journal of selected topics in applied earth and remotesensing, volume: 07, No: 1, PP: 116-125, January 2014.
- [11]. Tino P., Nabney I., "Hierarchical GTM: Constructing localized nonlinear projection manifolds in a principled way", IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(5):639–656, 2002.
- [12]. Majumdar N., Bhatnagar A., and Shukla S., "Using Linear Discriminant Analysis for Dimensionality Reduction for Predicting Anomalies of BGP data", International Journal of Innovative Technology and Exploring Engineering (IJITEE), 2019 Volume 8- 11, pp. 1989-1995.
- [13]. Bhatnagar A. Majumdar N., and Shukla S., "BGP Anomaly Detection using Decision Tree Based Machine Learning Classifiers", International Journal of Innovative Technology and Exploring Engineering (IJITEE), 2019 Volume 8-12, pp. 4015- 4020.
- [14] Febus Reidj G. Cruz , Matthew G.Binag , Marlou Ryan G. Ga , Francis Aldrine A. Uy. Flood Prediction Using Multi-Layer Artificial Neural Network in Monitoring System with Rain Gauge, Water Level, Soil Moisture Sensors, IEEE,28-31 Oct. 2018, DOI: 10.1109/TENCON.2018.8650387
- [15] Abhijit Paul, Prodipto Das, Floodpredictionmodel using artificial neural network, International Journal of Computer Applications Technology and Research Volume 3– Issue 7, 473- 478, 2014, ISSN: 2319–8656.
- [16] D.Lekkas,C.Onof,M.Lee,and E.Baltas,"Application of artificial Neural Networks For flood forecasting,"Global Nest Journal,vol.6,no.3,pp.205-211,2004
- [17] M. Mousa, X. Zhang, and C. Claudel, "Flash flood detection in urban cities using ultrasonic and infrared sensors," IEEE Sensors Journal, vol. 16, no. 19, pp. 7204–7216, 2016
- [18] Indrastanti R. Wideasari, Lukito Edi Nugoho2,Widyawan, Rissal Efendi ,Contextbased Hydrology Time Series Data for A flood Prediction Model Using LSTM , IEEE,27-28Sept.2018,DOI:10.1109/ICITACEE.2018.857 600
- [19] Gurleen Kaur, Anju Bal, An Efficient Automated Hybrid Algorithm to Predict Floods in Cloud Environment, IEEE, 27-28 Sept. 2018, DOI: 10.1109/ICITACEE.2018.8576900

APPENDICES:

SAMPLE CODE:

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g.pd.read_csv)
import matplotlib.pyplot as plt
from matplotlib import rcParams
import seaborn as sns

data = pd.read_csv(r"data.csv",sep=",")
data = data.fillna(data.mean())
data.info()
data.tail()
data.describe()
data.shape
data[0:6]
data.isnull().sum()
data.isnull()
data.sort_values(['YEAR','TERRAIN'])
data.hist(figsize=(24,24));
data.groupby("YEAR").sum()['ANNUAL
PRECIPITATION'].plot(figsize=(12,8));
data[['LOCATION','ANNUAL
PRECIPITATION']].groupby("LOCATION").mean().plot.barh(stacked=True
,figsize=(13,8),color={"ANNUAL PRECIPITATION": "blue"});

rcParams['figure.figsize'] = 17,6
plt.bar(data['TERRAIN'].unique(), data['TERRAIN'].value_counts(), color =
['#64B8B1',
'#38B261','#E9D59C','#89CFF0','#8C92AC','#E3813B','#F6C324','#DC63
86','#F34100','#FF7F50'])
plt.xticks([0, 1,2,3,4,5,6,7,8,9,10,11])
plt.xlabel('Target Classes')
plt.ylabel('Count')
plt.title('Count of each SEVERITY')
rcParams['figure.figsize'] = 8,6
plt.bar(data['SEVERITY'].unique(), data['SEVERITY'].value_counts(),
color = ['#F34100','#2D8F4E',
'#FF5D22','#FFA500','#880ED4','#DC143C'])
plt.xticks([0, 1,2,3,4,5])
plt.xlabel('TERRAIN')
plt.ylabel('Count')
plt.title('Count of each TERRAIN')
```

```
from datetime import datetime
import datetime
import numpy as np
import keras.models
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import rcParams
import seaborn as sns
import sys
import os
```

```
dataset = pd.read_csv(r'dataset/data.csv')
dataset = dataset[ dataset['YEAR']>=1980 ]
dataset = dataset.dropna()
#without climate
X = dataset.iloc[:,[1,2,4,5]].values
y = dataset.iloc[:,9].values
dataset
print(X)
print(y)
```

```
#importing preprocessing libraries
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
#from sklearn.cross_validation import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.2,random_state = 0)
```

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors = 24, metric = 'minkowski', p =
2)
knn.fit(X_train, y_train)
print(y_train)
print(X_train)
```

```
# Logistic Regression Algorithm
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression(random_state = 42)
logreg.fit(X_train, y_train)
```

```

# Naive Bayes Algorithm
from sklearn.naive_bayes import GaussianNB
nb = GaussianNB()
nb.fit(X_train, y_train)
# Random forest Algorithm
from sklearn.ensemble import RandomForestClassifier
ranfor = RandomForestClassifier(n_estimators = 11, criterion = 'entropy',
random_state = 42)
ranfor.fit(X_train, y_train)

Y_pred_logreg = logreg.predict(X_test)
Y_pred_knn = knn.predict(X_test)
Y_pred_nb = nb.predict(X_test)
Y_pred_ranfor = ranfor.predict(X_test)

# Evaluating using accuracy_score metric
from sklearn.metrics import accuracy_score
accuracy_logreg = accuracy_score(y_test, Y_pred_logreg)
accuracy_knn = accuracy_score(y_test, Y_pred_knn)
accuracy_nb = accuracy_score(y_test, Y_pred_nb)
accuracy_ranfor = accuracy_score(y_test, Y_pred_ranfor)

Logistic_Regression = accuracy_logreg * 100
K_Nearest_neighbors = accuracy_knn * 100
Naive_Bayes = accuracy_nb * 100
Random_Forest = accuracy_ranfor * 100

# Accuracy on test set
print("Logistic Regression: " + str(accuracy_logreg * 100))
print("K Nearest neighbors: " + str(accuracy_knn * 100))
print("Naive Bayes: " + str(accuracy_nb * 100))
print("Random Forest: " + str(accuracy_ranfor * 100))

height = [Logistic_Regression, K_Nearest_neighbors, Naive_Bayes, Random_Forest]
bars = ('LR', 'KNN', 'NB', 'RF')
x_pos = np.arange(len(bars))

# Create bars with different colors
plt.bar(x_pos, height, color=['#FF007F', 'red', 'green', 'blue'])

# Create names on the x-axis
plt.xticks(x_pos, bars)

```

```

# Show graph
plt.show()

dataset = pd.read_csv(r'dataset/data.csv')
dataset = dataset[ dataset['YEAR']>=1980 ]
dataset = dataset.dropna()
#without climate
X = dataset.iloc[:,[0,1,3,4]].values
y = dataset.iloc[:,8].values

dataset

print(X)
print(y)

#importing preprocessing libraries
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
#from sklearn.cross_validation import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.2,random_state = 0)

print(y_train)
print(X_train)

# Random forest Algorithm
from sklearn.ensemble import RandomForestClassifier
ranfor = RandomForestClassifier(n_estimators = 11, criterion = 'entropy',
random_state = 42)
ranfor.fit(X_train, y_train)

Y_pred_ranfor = ranfor.predict(X_test)
from sklearn.metrics import accuracy_score
accuracy_ranfor = accuracy_score(y_test, Y_pred_ranfor)
print("Random Forest: " + str(accuracy_ranfor * 100))
date = str(input("Date----->"))
month = datetime.datetime.strptime(date, '%d/%m/%Y').strftime('%m')
month
Year = datetime.datetime.strptime(date, '%d/%m/%Y').strftime('%Y')
Year

import random

```

```
if(month == '01' or month == '02'):
```

```
    precipitation=random.randint(1,300)
    a=(precipitation)
    print('1')
    z="1"
```

```
elif(month == '03' or month == '04' or month == '05'):
```

```
    precipitation=random.randint(300,400)
    a=(precipitation)
    print('2')
    z="2"
```

```
elif(month == '06' or month == '07' or month == '08' or month == '09'):
```

```
    precipitation=random.randint(1000,3500)
    a=(precipitation)
    print('3')
    z="3"
```

```
elif(month == '10' or month == '11' or month == '12'):
```

```
    precipitation=random.randint(800,900)
    a=(precipitation)
    print('4')
    z="4"
```

```
state = str(input("STATE----->"))
```

```
count=28
```

```
if(state == 'chennai' or state == 'madurai',state == 'trichy' or state ==
'kovai',state == 'salem' or state == 'dindugal',state == 'virudhunagar' or
state == 'kanchipuram',state == 'trivallur' or state == 'kanyakumari',state
== 'namakkal' or state == 'nagarkovil',state == 'thanjavur' or state ==
'vilupuram'):
```

```
    print("result")
```

```
elif(State == 'bengalore' or state == 'belegavi',State == 'kolar' or state ==
'raichur',State == 'ballari bidar' or state == 'shivamogga',State ==
'ramanagara' or state == 'chitradurga',State == 'davanagere' or state ==
'vijayapura',State == 'uttara' or state == 'bagalkot',State == 'yadgir' or state
== 'dharwad'):
```

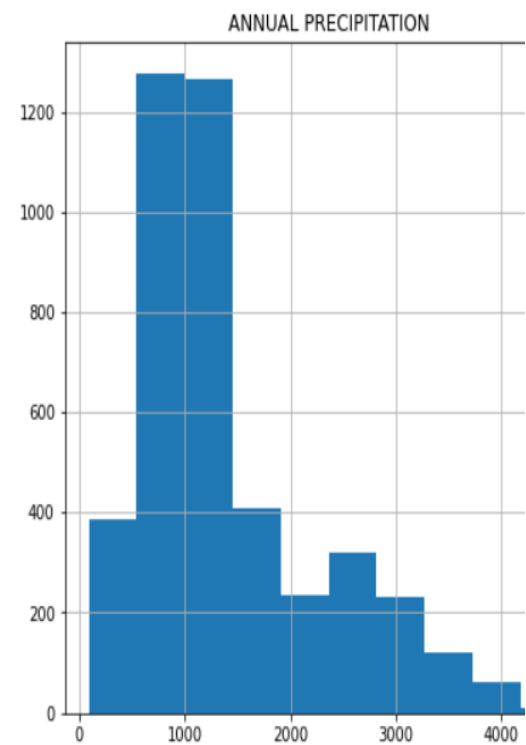
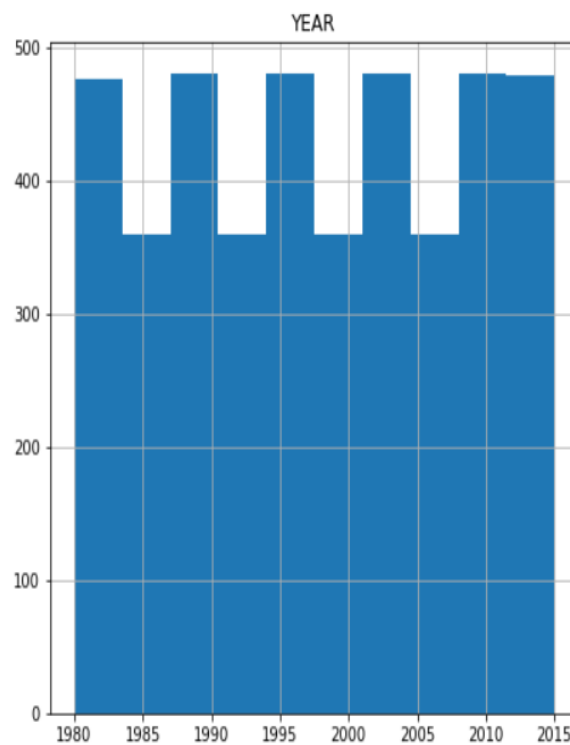
```
    print("result")
```

```
elif(State == 'malapuram' or state == 'plakkad',State == 'idukki' or state ==
'kottiyam',State == 'alappuzha' or state == 'kollam',State == .....
```

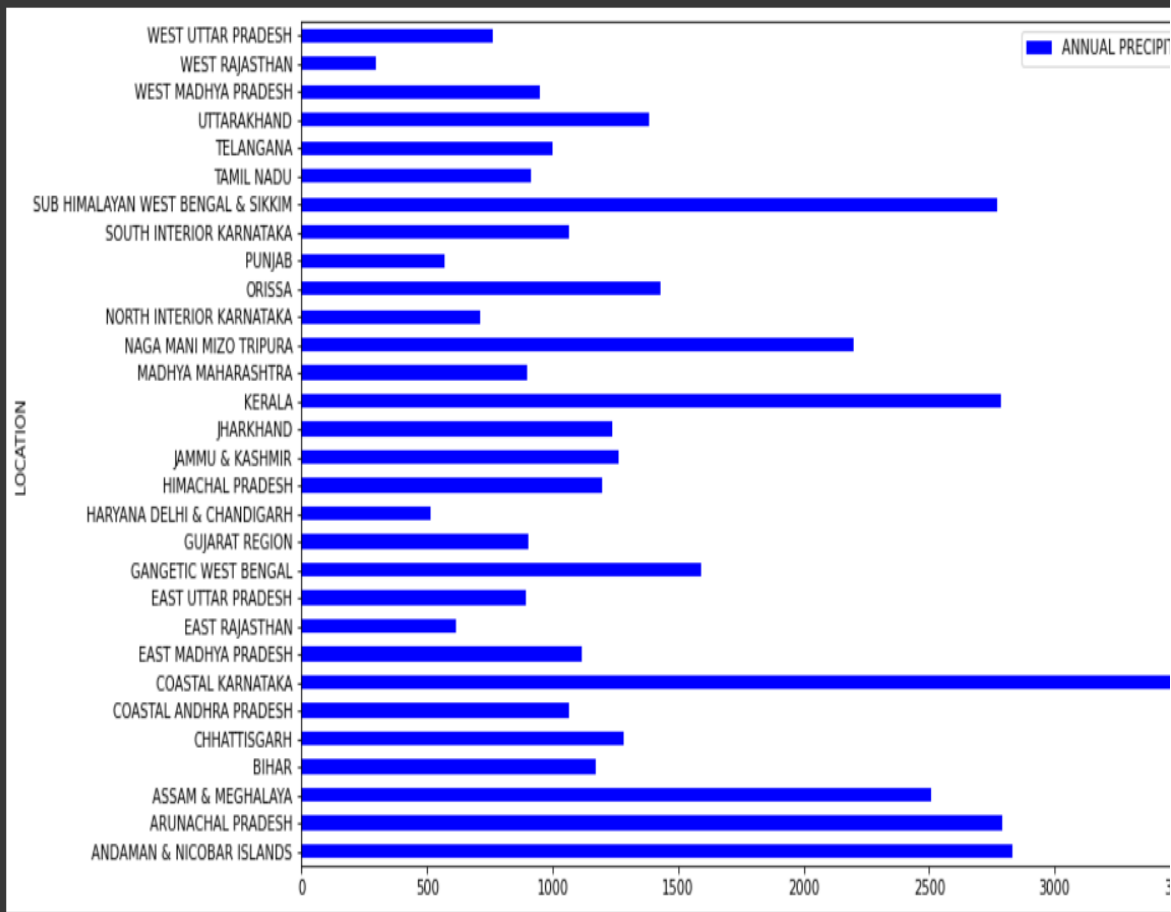
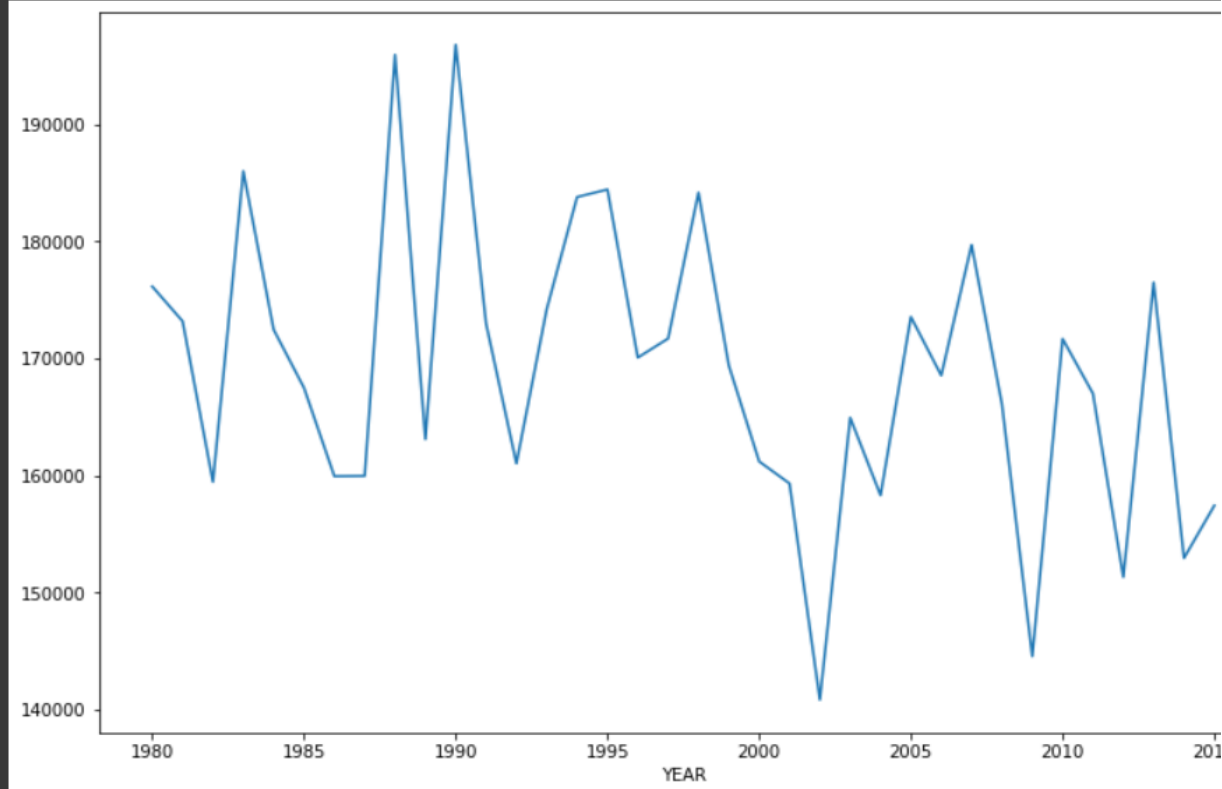
SCREENSHOTS:

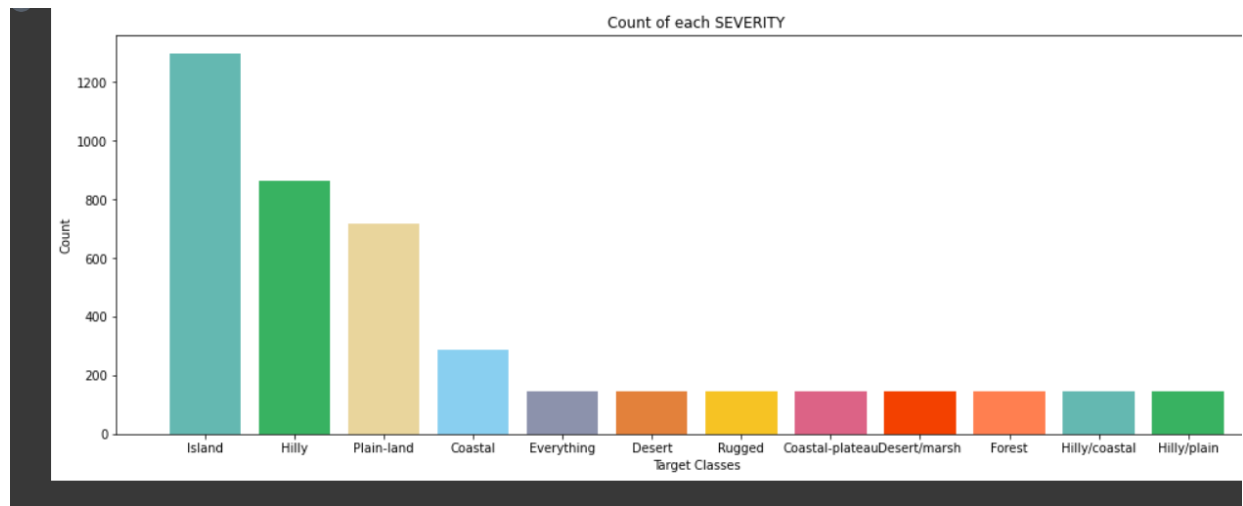
```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 4316 entries, 0 to 4315  
Data columns (total 10 columns):  
#   Column                                Non-Null Count  Dtype    
---  -  
0   LOCATION                             4316 non-null   object   
1   SUBDIVISION                         4316 non-null   int64    
2   YEAR                                4316 non-null   int64    
3   ANNUAL PRECIPITATION                4316 non-null   float64  
4   QUARTER                             4316 non-null   int64    
5   PRECIPITATION                       4316 non-null   float64  
6   SEVERITY1                           4316 non-null   int64    
7   TERRAIN                             4316 non-null   object   
8   CLIMATE                             4316 non-null   object   
9   SEVERITY                            4316 non-null   int64    
dtypes: float64(2), int64(5), object(3)  
memory usage: 337.3+ KB
```

Dataset attributes



Precipitation visualization



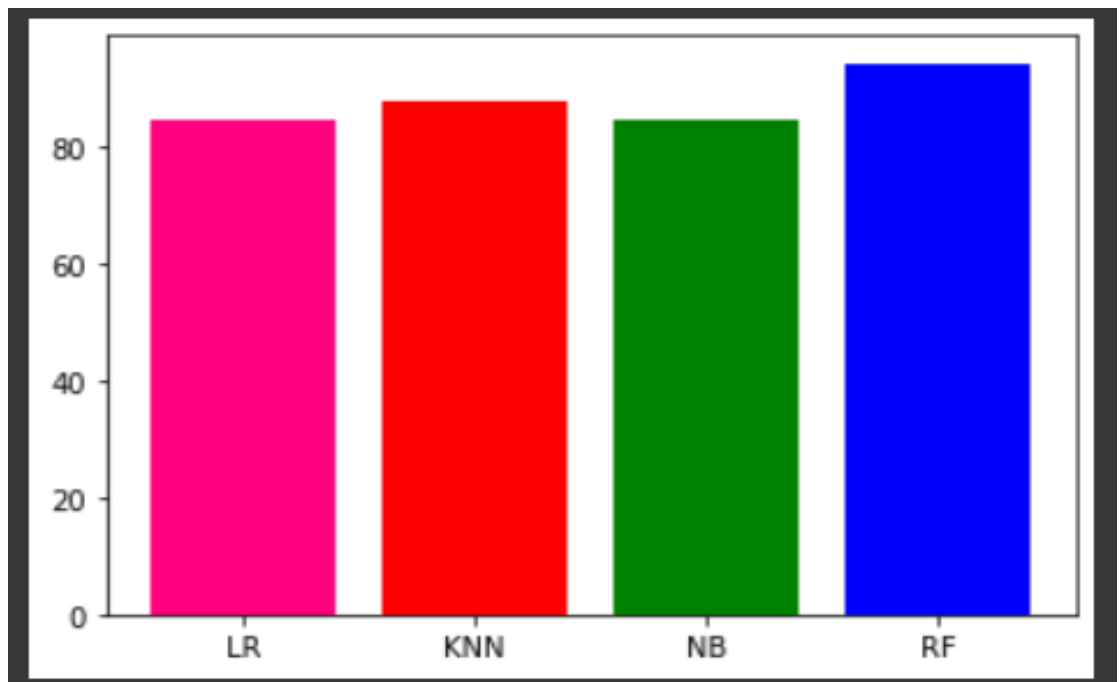


Severity recorded in different terrain

```
# Accuracy on test set
print("Logistic Regression: " + str(accuracy_logreg * 100))
print("K Nearest neighbors: " + str(accuracy_knn * 100))
print("Naive Bayes: " + str(accuracy_nb * 100))
print("Random Forest: " + str(accuracy_ranfor * 100))
```

```
Logistic Regression: 84.3205574912892
K Nearest neighbors: 87.45644599303137
Naive Bayes: 84.20441347270615
Random Forest: 94.07665505226481
```

Accuracy scores



Algorithm performance