A project report on

COMPARATIVE ANALYSIS OF FLOOD PREDICTION USING MACHINE LEARNING ALGORITHMS

Submitted in partial fulfillment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

in

Computer Science & Engineering

by

B.SAI SUSHMA	**(184G1A0579)**
P.V.SREEDEVI	**(184G1A05B0)**
M.SRAVANI	**(184G1A0592)**
V.SREEKANTH	**(184G1A0594)**

Under the Guidance of

Mr. LINGAM SUMAN , M.Tech,(Ph.D.)

Assistant Professor

SRIT
Empowering Knowledge

Department of Computer Science & Engineering

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY
(Affiliated to JNTUA & Approved by AICTE)
(Accredited by NAAC with 'A' Grade & Accredited by NBA (EEE, ECE & CSE))
Rotarypuram Village, B K Samudram Mandal, Ananthapuramu-515701.

2021 - 2022

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY

(Affiliated to JNTUA & Approved by AICTE)
(Accredited by NAAC with ‘A’ Grade & Accredited by NBA (EEE, ECE & CSE))
Rotarypuram Village, B K Samudram Mandal, Ananthapuramu-515701.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SRIT
Empowering Knowledge

Certificate

This is to certify that the Project report entitled Comparative Analysis of Flood Prediction using Machine Learning Algorithms is the bonafide work carried out by **B.SAI SUSHMA** bearing Roll Number **184G1A0579, P.V.SREEDEVI** bearing Roll Number **184G1A05B0, M.SRAVANI** bearing Roll Number **184G1A0592** and **V.SREEKANTH** bearing Roll Number **184G1A0594** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** during the academic year 2021-2022.

Signature of the Guide	**Head of the Department**
Mr. Lingam Suman, M.Tech., (Ph.D.)	Mr. P. Veera Prakash, M.Tech., (Ph.D.)
Assistant Professor	Assistant Professor and HOD

Date:	**EXRTERNAL EXAMINER**
Place: Rotarypuram	

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that I have now the opportunity to express my gratitude for all of them.

It is with immense pleasure that I would like to express my indebted gratitude to my Guide **Mr. Lingam Suman, MTech., (Ph.D.), Assistant Professor, Computer Science and Engineering**, who has guided me a lot and encouraged me in every step of the technical seminar. I thank him for the stimulating guidance, constant encouragement and constructive criticism which have made possible to bring out this project work.

I am very much thankful to **Mr. P. Veera Prakash, MTech., (Ph.D.) Assistant Professor & Head of the Department, Computer Science & Engineering,** for his kind support and for providing necessary facilities to carry out the work.

I wish to convey my special thanks to **Dr. G. Bala Krishna, Ph.D., Principal** of **Srinivasa Ramanujan Institute of Technology** for giving the required information in doing my project work. Not to forget, I thank all other faculty and non-teaching staff, and my friends who had directly or indirectly helped and supported me in completing my project work in time.

I also express our sincere thanks to the Management for providing excellent facilities**.**

Finally, I wish to convey my gratitude to my family who fostered all the requirements and facilities that I need.

Project associate

Declaration

We, Ms. B.Sai Sushma with reg no: 184G1A0579, Ms. P.V.Sreedevi with reg no: 184G1A05B0, Ms. M. Sravani with reg no: 184G1A0592 , Mr. V. Sreekanth with reg no: 184G1A0594 students of SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY, RotaryPuram, hereby declare that the dissertation entitled "COMPARATIVE ANALYSIS OF FLOOD PREDICTION USING MACHINE LEARNING ALGORITHMS" embodies the report of our project work carried out by us during IV year Bachelor of Technology under the guidance of Mr. Lingam Suman, M.Tech,(Ph.D.) Department of CSE, SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY, and this work has been submitted for the partial fulfillment of the requirements for the award of the Bachelor of Technology degree.

The results embodied in this project have not been submitted to any other University of Institute for the award of any Degree or Diploma.

B.SAI SUSHMA	Reg no: 184G1A0579
P.V.SREEDEVI	Reg no: 184G1A05B0
M.SRAVANI	Reg no: 184G1A0592
V.SREEKANTH	Reg no: 184G1A0594

CONTENTS Page no.

LIST OF FIGURES

LIST OF SCREENS

Abbreviations

CSV	Comma-separated values
SRS	Software Requirement Specification
UML	Unified modelling language
NumPy	Numerical Python
ML	Machine Learning
KNN	K Nearest Neighbor
SVC	Support Vector Machine

ABSTRACT

Floods are among the most destructive natural disasters, which are highly complex to model. The research on the advancement of flood prediction models has been contributing to risk reduction, policy suggestion, minimizing loss of human life and reducing 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 have highly contributed in the advancement of prediction systems providing better performance and cost-effective solutions. Due to the vast benefits and potential of ML, its popularity has dramatically increased among hydrologists. The main contribution is to demonstrate the state of the art of ML models in flood prediction and give an insight over the most suitable models.

In this a, the Accuracy, Recall, and Receiver Operating Characteristics (ROC) scores of three machine learning algorithms, namely Decision Tree, Logistic Regression, and Support Vector Classification (SVR), were evaluated and compared. Logistic Regression, when compared with the other two algorithms, gives more accurate results and provides high performance accuracy and recall. In addition, the Decision Tree outperformed the Support Vector Classifier. Decision Tree performed reasonably well due to its above-average accuracy and below-average recall scores. We discovered that Support Vector Classification performed poorly with a small size of dataset, with a recall score of 0, below average accuracy score and a distinctly average roc score.

Keywords:

Machine Learning, Flood Prediction, Decision Tree Classification, Logistic Regression, Sigmoid Function, SVM, K Nearest Neighbor ,Random Forest.

CHAPTER 1

INTRODUCTION

Floods are among the most destructive natural disasters and it causes lots of damage to property and human life. The yearly data shows that the amount of rainfall is increasing and it's due to climate change. Flood is predicted in several locations using some advanced technologies which just helps the people to be prepared for upcoming disasters. It is very difficult to create a predictive model using machine learning. Machine learning gives computers the capability to learn without being explicitly programmed. Machine learning has a role in preventing many natural disasters like earthquakes, floods and many more. Machine learning make decisions using past data and these data are fed into the algorithms and the output is predicted. Machine learning(ML) can be classified into three categories Supervised learning, Unsupervised learning and Reinforcement learning. Supervised learning is the types of machine learning in which machines are trained using well “labelled” training data, and on the basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output.

The Supervised learning can be again further divided into two types Regression and Classification. Regression algorithms are used if there is a relationship between the input variable and output variable. It is used for the prediction of continuous variables. Classification algorithms are used when the output variable is categorical which means there are two classes such as Yes/No, Male-Female, True-False. Unsupervised learning is a machine learning technique in which models are not supervised using training dataset. Instead models itself find the hidden patterns and insights from the given data. The Unsupervised learning algorithm can be further categorized into two types: Clustering and Association. Clustering is a method of grouping objects into clusters such that objects with most similarities remain into a group and have less or no similarities with the objects of another group. Association rule is an unsupervised learning method which is used for finding the relationships between variables in the large database. The aim of this project is to develop a flood prediction model which is real time. This could be helpful in the areas where the flash flood occurs.

This system takes the input as the rainfall data all over the India process it using different machine learning models and the best model is determined with the help of accuracy of different algorithms, which would help people prior, save lives and also save lots of meteorological efforts.

1.1 Objective of the Project

The objective of Flood Prediction using Machine Learning is to design a model to predict the flood using the rainfall data. The prediction of different models is taken and compared within each other to find the best model that has high accuracy. The flood can be predicted in different states of India in different months. The confusion matrix of different models in Machine learning is considered to evaluate the accuracy and precision of the system.

1.2 Machine Learning

Machine Learning is the area of study which enables machines to learn without being explicitly programmed. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model of sample data, known as "training data". A Machine Learning system learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it.

Machine learning gives a system the ability to learn automatically and improve its recommendations using data alone, with no additional programming needed. Because retailers generate enormous amounts of data, machine learning technology quickly proves its value. When a machine learning system is fed data—the more, the better— it searches for patterns. Going forward, it can use the patterns it identifies within the data to make better decisions. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately.

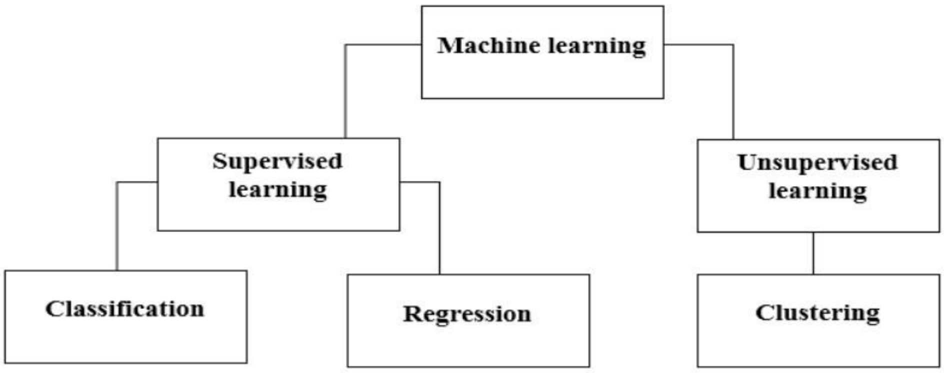


Fig.1.1 Types of Machine Learning

1.2.1. Supervised Learning

Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to find a mapping function to map the input variable(x) with the output variable(y). Supervised learning is the type of machine learning in which machines are trained using well "labeled" trained data. Once the training process is completed, the model is tested on the basis of test data (a subset of the training set), and then it predicts the output.

In supervised learning, models are trained using labelled dataset, where the model learns about each type of data. Once the training process is completed, the model is tested on the basis of test data (a subset of the training set), and then it predicts the output. The working of Supervised learning can be easily understood by the below example and diagram:

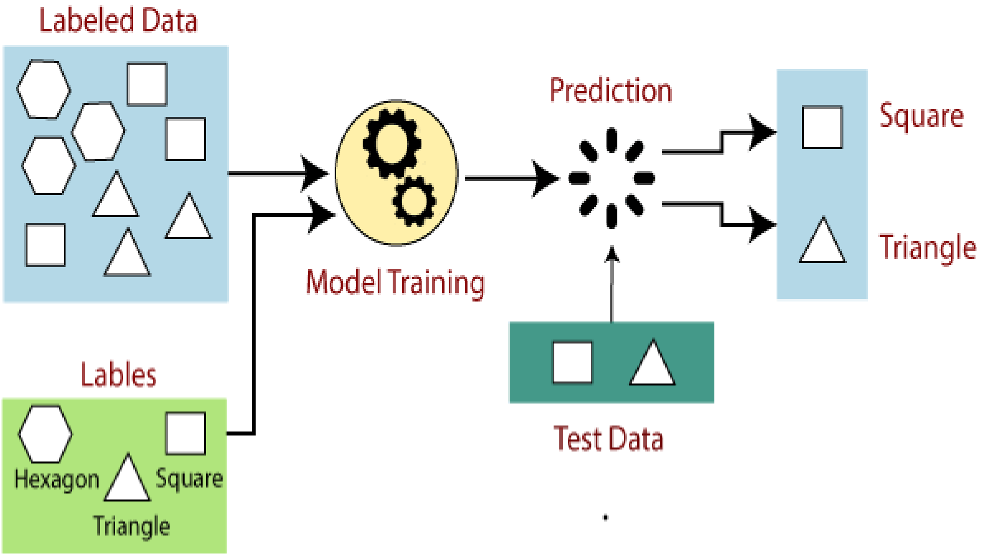


Fig.1.2 : Process of any ML algorithm

Suppose we have a dataset of different types of shapes which includes square, rectangle, triangle, and Polygon. Now the first step is that we need to train the model for each shape.

If the given shape has four sides, and all the sides are equal, then it will be labelled as a Square. If the given shape has three sides, then it will be labelled as a triangle.

If the given shape has six equal sides, then it will be labelled as hexagon. Now, after training, we test our model using the test set, and the task of the model is to identify the shape. The machine is already trained on all types of shapes, and when it finds a new shape, it classifies the shape on the bases of a number of sides, and predicts the output. Supervised learning can be further divided into two types:

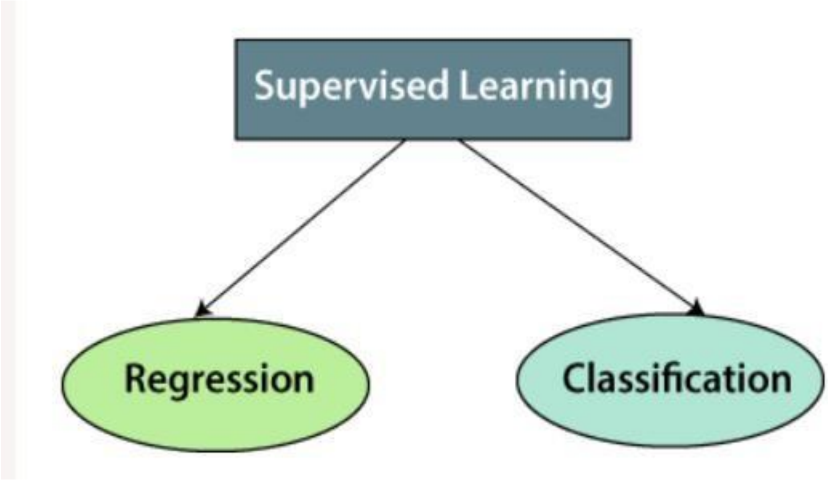


Fig.1.3 : Types of Supervised Learning

1.2.1.1 Regression

Regression is a supervised learning technique which helps in finding the correlation between variables and enables us to predict the continuous output variable based on the one or more predictor variables. It is mainly used for prediction**.** Regression analysis helps us to understand how the value of the dependent variable is changing corresponding to an independent variable when other independent variables are held fixed. It predicts continuous/real values such as price**,** etc. Some real-world examples for regression are predicting the sales based on input parameters etc.

1.2.1.2 Classification

Classification is supervised learning. It can be performed on both structured and unstructured data. Classification is the process of finding a model that helps to separate the data into different categorical classes. In this process, data is categorized under different labels according to some parameters given in input and then the labels are predicted for the data.

1.2.1.3 Classification Algorithms

1.2.1.3.1 Decision Tree

Decision tree learning is a method commonly used in data mining. The goal is to create a model that predicts the value of a target variable based on several input variables. An example is shown in the diagram at right. Each interior node corresponds to one of the input variables; there are edges to children for each of the possible values of that input variable. Each leaf represents a value of the target variable given the values of the input variables represented by the path from the root to the leaf.

A decision tree is a simple representation for classifying examples. For this section, assume that all of the input features have finite discrete domains, and there is a single target feature called the "classification". Each element of the domain of the classification is called a class. A decision tree or a classification tree is a tree in which each internal (non-leaf) node is labeled with an input feature. The arcs coming from a node labeled with an input feature are labeled with each of the possible values of the target or output feature or the arc leads to a subordinate decision node on a different input feature. Each leaf of the tree is labeled with a class or a probability distribution over the classes.

A tree can be "learned” by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning. See the examples illustrated in the figure for spaces that have and have not been partitioned using recursive partitioning, or recursive binary splitting. The recursion is completed when the subset at a node has all the same value of the target variable, or when splitting no longer adds value to the predictions. This process of top-down induction of decision trees is an example of a greedy algorithm, and it is by far the most common strategy for learning decision trees from data. In data mining, decision trees can be described also as the combination of mathematical and computational techniques to aid the description, categorization and generalization of a given set of data. Data comes in records of the form:

$$(X,Y) = (x1, x2, x3\ldots\ldots,xk, Y)$$

The dependent variable, Y, is the target variable that we are trying to understand, classify or generalize. The vector X is composed of the features, x1, x2, x3 etc., that are used for that task.

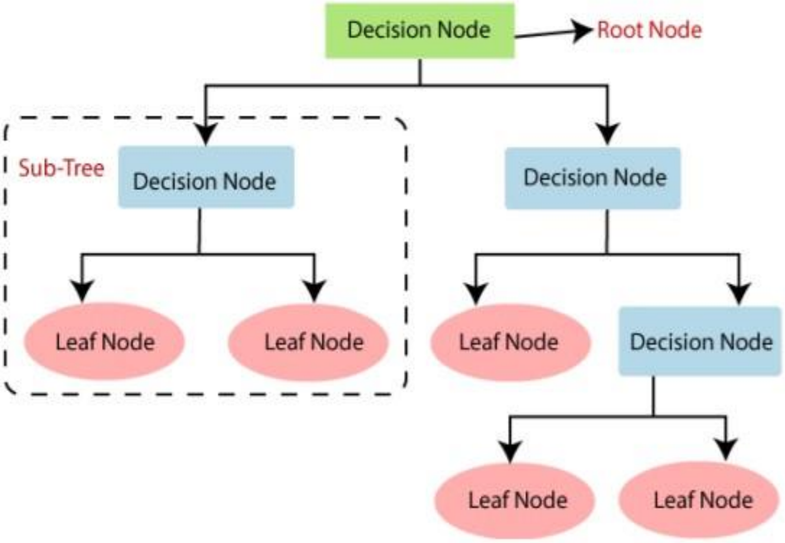


Fig.1.4 Classification algorithm

1.2.1.3.2 Logistic Regression

Logistic Regression may be a machine learning algorithm that predicts the probability of a categorical variable. It is a statistical way of analyzing a group of knowledge that comprises quite one experimental variable that determines the result. The outcome is then measured with a dichotomous variable. The goal of this algorithm is to seek out the simplest model to explain the connection between a dichotomous characteristic of interest and a group of independent variables. In this algorithm, the dependent variable is a binary variable that contains data coded as 1 or 0. In other words, the logistic regression model predicts P(Y=1) as a function of X.

1.2.1.3.3 K Nearest Neighbor

K-Nearest Neighbor is one among the supervised machine learning algorithms that stores all instances like training data points in an n-dimensional space. For real-valued data, the algorithm returns the mean of k nearest neighbors, and in case of receiving unknown discrete data, it analyses the closest k number of instances that is saved and returns the most common class as the result of the prediction. In the distance-weighted nearest neighbor algorithm, the contribution of each of the k neighbors is weighed according to their distance, giving higher weight to the closest neighbors.

The K-Nearest Neighbor algorithm is a classification algorithm and is robust to noisy data as it averages the k-nearest neighbors. The algorithm first takes a bunch of labeled points and analyses them to find out the way to label the opposite points. Hence, to label a new point, it looks at the closest labeled points to that new point and has those neighbors vote, so whichever label most of the neighbors have been the label for the new point. This algorithm makes predictions about the validation set using the whole training set. Only by rummaging through the whole training set to seek out the closest instances, the new instance is predicted. Closeness is a value that is determined using a proximity measurement across all the features involved.

1.2.1.3.4 Support Vector Machines

SVM uses a classifier that categorizes the info set by setting an optimal hyperplane between data. This classifier is chosen as it is incredibly versatile in the number of different kernel functions that can be applied, and this model can yield a high predictability rate. Support Vector Machine is one among the foremost popular and widely used clustering algorithms. It belongs to a gaggle of generalized linear classifiers and is taken into account as an extension of the perceptron. It was developed in the 1990s and continues to be the desired method for a high-performance algorithm with a little tuning.

1.2.1.3.5 Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.

1.2.1.4 Steps Involved in Supervised Learning

1) First Determine the type of training dataset
2) Collect/Gather the labelled training data.
Split the training dataset into training dataset, test dataset, and validation dataset.

Determine the input features of the training dataset, which should have enough knowledge so that the model can accurately predict the output.
Determine the suitable algorithm for the model, such as support vector machine, decision tree, etc.

Execute the algorithm on the training dataset. Sometimes we need validation sets as the control parameters, which are the subset of training datasets.

Evaluate the accuracy of the model by providing the test set. If the model predicts the correct output, which means our model is accurate.

1.2.2 Unsupervised Learning

Unsupervised learning is a machine learning technique in which models are not supervised using training dataset. Models itself find the hidden patterns and insights from the given data. It mainly deals with the unlabeled data. Unsupervised learning cannot be directly applied to a regression or classification problem because unlike supervised learning, we have the input data but no corresponding output data. The goal of unsupervised learning is to find the underlying structure of dataset, group that data according to similarities, and represent that dataset in a compressed format.

1.3. Dimensionality Reduction

Dimensionality reduction technique can be defined as, "It is a way of converting the higher dimensions dataset into lesser dimensions dataset ensuring that it provides similar information." These techniques are widely used in machine learning for obtaining a better fit predictive model while solving the classification and regression problems. It is commonly used in the fields that deal with high dimensional data.

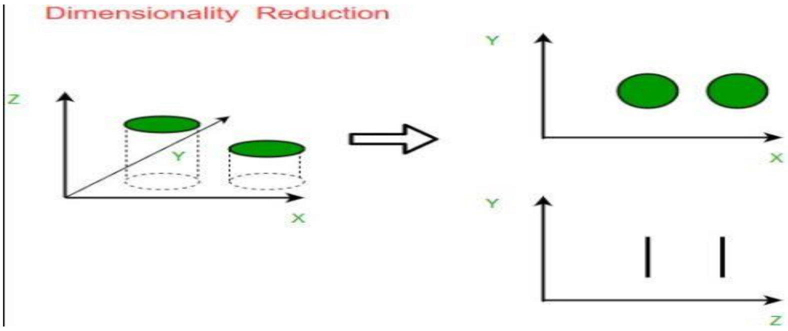


Fig.1.5 : Dimensionality reduction

1.3.1 Feature Selection

Feature selection is the process of selecting the subset of the relevant features and leaving out the irrelevant features present in a dataset to build a model of high accuracy. In other words, it is a way of selecting the optimal features from the input dataset.

Feature Extraction

Feature extraction is the process of transforming the space containing many dimensions into space with fewer dimensions. This approach is useful when we want to keep the whole information but use fewer resources while processing the information.

Benefits of applying Dimensionality Reduction

- By reducing the dimensions of the features, the space required to store the dataset also gets reduced.
- Less Computation training time is required for reduced dimensions of features.
- Reduced dimensions of features of the dataset help in visualizing the data quickly.
- It removes the redundant features (if present) by taking care of multicollinearity.

CHAPTER-2

LITERATURE SURVEY

Classifications algorithms help to predict the sales and also any kind of predictions that has to be made can be easily done using the Machine Learning algorithms. Machine learning is a kind of technology which uses the past data that is available to make different kinds of decisions for the future. So, it can have a wide range of applications that can be built using the Machine learning technology.

Classification will be one of the best techniques for the Prediction of the floods. This Machine Learning technique will help to have best results as it depends on the variables which are Dependent and Independent.

Datasets available may consist of Complete data and some values may be missing. It is very important to handle the missing values to obtain higher accuracy for the models.

2.1. Existing system

The Existing system is used to predict the floods using the Machine learning algorithms. Model trains the previous data of a rainfall provided using the csv file and creates a prediction model using different techniques like fuzzy inference system and IOT and LORA technology.

A. Improving real time flood forecasting using fuzzy inference system.[1] Anil Kumar lohani et al proposed the system in order to improve the real time forecasting of floods, this paper proposes a modified Takagi Sugeno (T–S) fuzzy inference system termed as threshold subtractive clustering-based Takagi Sugeno (TSC-T–S) fuzzy inference system by introducing the concept of rare and frequent hydrological situations in fuzzy modeling system. The proposed modified fuzzy inference systems provide an option of analyzing and computing cluster centers and membership functions for two different hydrological situations, i.e., low to medium flows (frequent events) as well as high to very high flows (rare events) generally encountered in real time flood forecasting. The methodology has been applied for flood forecasting using the hourly rainfall and river flow data of upper Narmada basin, Central India.

B. A Review on Fuzzy Based Flood Warning Expert System using IoT and LoRa Technology.[2] Flood is the most destructive natural hazard. Other than rainfall different parameters like temperature, water flow speed, humidity, moisture level of land. As parameters of flood are uncertain so that algorithm which deals with uncertain inputs that is fuzzy logic is used to develop flood detection system. Fuzzy logic has ability to deal with nonlinearities and uncertainties. Fuzzy logic produces results that have resemblance with human results. As mentioned earlier different parameters are responsible for occurrence of flood so that use of IoT with power of different sensors is appropriate. LoRa is an emerging technique which is a wireless technique accepts data from sensors and transfer this data to main controlling system where this data is processed using fuzzy logic and result is produced.

After training all the algorithms using the data available then they are available to predict the values for the future and then the process of predicting the further values can be made and used in the real world for any kind floods. This will help us to find the floods.

CHAPTER-3

ANALYSIS

3.1 Introduction

The Analysis Phase is where the project life cycle begins. This is the phase where you break down the deliverables in the high-level Project Charter into the more detailed requirements. Gathering requirements is the main attraction of the Analysis Phase. The process of gathering requirements is usually more than simply asking the users what they need and writing their answers down. Depending on the complexity of the application, the process for gathering requirements has a clearly defined process of its own. This process consists of a group of repeatable processes that utilize certain techniques to capture, document, communicate, and manage requirements. This formal process, which will be developed in more detail, consists of four basic steps.

- **Elicitation** – I ask questions, you talk, I listen
- **Validation** – I analyze, I ask follow-up questions
- **Specification** – I document, I ask follow-up questions
- **Verification** – We all agree

Most of the work in the analysis phase is performed by analyst.

3.2 Software Requirement Specifications

SRS is a document created by a system analyst after the requirements are collected. SRS defines how the intended software will interact with hardware, external interfaces, speed of operation, response time of system, portability of software across various platforms, maintainability, speed of recovery after crashing, Security, Quality, Limitations etc.

The requirements received from clients are written in natural language. It is the responsibility of system analysts to document the requirements in technical language so that they can be comprehended and useful by the software development team.

3.3 Hardware Requirements

Processor	- P–IV
RAM	- 2 GB (min)
Hard Disk	- 40 GB
Key Board	- Standard Windows Keyboard
Mouse	- Two or Three Button Mouse

Monitor	-	SVGA 21

3.4 Software Requirements

Operating system	:	Windows 7 Ultimate or above.
Coding Language	:	Python.
Front-End	:	Python.
Back-End	:	Flask
Tools	:	Jupyter notebook
Dataset	:	CSV file

3.5 Jupyter notebook

3.5.1 INTRODUCTION

The Jupyter Notebook is an open-source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter. Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python and R. Jupyter ships with the IPython kernel, which allows you to write your programs in python, but there are currently over 100 other kernels that you can also use. IPython notebook was developed by Fernando Perez as a web-based affront end to IPython kernel.

As an effort to make an integrated interactive computing environment for multiple languages, the Notebook project was shifted under Project Jupyter providing front end for programming environments Juila and R in addition to python.

A notebook document consists of rich text elements with HTML, formatted text, figures, mathematical equations etc. The notebook is also an executable document consisting of code blocks in python or other supporting languages. Jupyter notebook is a client-server application.

The application starts the server on a local machine and opens the notebook interface in the web browser where it can be edited and run from. The notebook is saved as an ipynb file and can be exported as html, pdf and LaTeX files.

Getting Up and Running with Jupyter Notebook

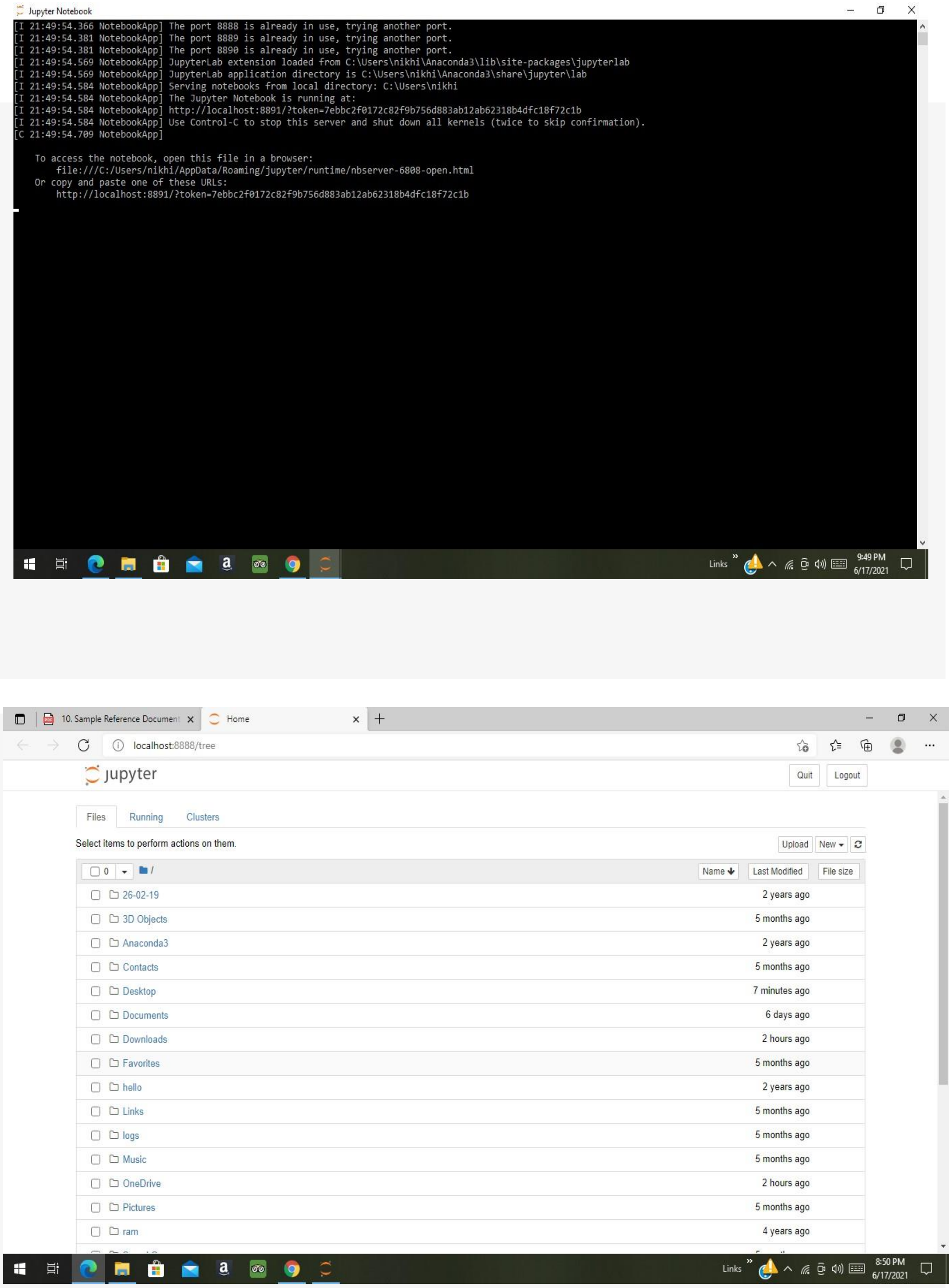
The Jupyter Notebook is not included with Python, so if you want to try it out, you will need to install Jupyter. There are many distributions of the Python language.

This article will focus on just two of them for the purposes of installing Jupyter Notebook. The most popular is CPython, which is the reference version of Python that you can get from their website. It is also assumed that you are using Python 3.

3.5.2. Installation

If so, then you can use a handy tool that comes with Python called pip to install Jupyter. Notebook like this: $pip install jupyter

This will start up Jupyter and your default browser should start (or open a new tab) to



```
[I 21:49:54.366 NotebookApp] The port 8888 is already in use, trying another port.
[I 21:49:54.381 NotebookApp] The port 8889 is already in use, trying another port.
[I 21:49:54.381 NotebookApp] The port 8890 is already in use, trying another port.
[I 21:49:54.569 NotebookApp] JupyterLab extension loaded from C:\Users\nikhi\Anaconda3\lib\site-packages\jupyterlab
[I 21:49:54.569 NotebookApp] JupyterLab application directory is C:\Users\nikhi\Anaconda3\share\jupyter\lab
[I 21:49:54.584 NotebookApp] Serving notebooks from local directory: C:\Users\nikhi
[I 21:49:54.584 NotebookApp] The Jupyter Notebook is running at:
[I 21:49:54.584 NotebookApp] http://localhost:8891/?token=7ebbc2f0172c82f9b756d883ab12ab62318b4dfc18f72c1b
[I 21:49:54.584 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 21:49:54.709 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///C:/Users/nikhi/AppData/Roaming/jupyter/runtime/nbserver-6808-open.html
    Or copy and paste one of these URLs:
        http://localhost:8891/?token=7ebbc2f0172c82f9b756d883ab12ab62318b4dfc18f72c1b
```

Select items to perform actions on them.

Name	Last Modified
26-02-19	2 years ago
3D Objects	5 months ago
Anaconda3	2 years ago
Contacts	5 months ago
Desktop	7 minutes ago
Documents	6 days ago
Downloads	2 hours ago
Favorites	5 months ago
hello	2 years ago
Links	5 months ago
logs	5 months ago
Music	5 months ago
OneDrive	2 hours ago
Pictures	5 months ago
ram	4 years ago

Fig.3.1 -Jupyter notebook

the following URL:http://localhost:8888 tree

Creating a Notebook

All you need to do is click on the new button (upper right), and it will open up a list of choices. On my machine, I happen to have Python 2 and Python 3 installed, so I can create a Notebook that uses either of these. For simplicity's sake, let's choose Python 3.

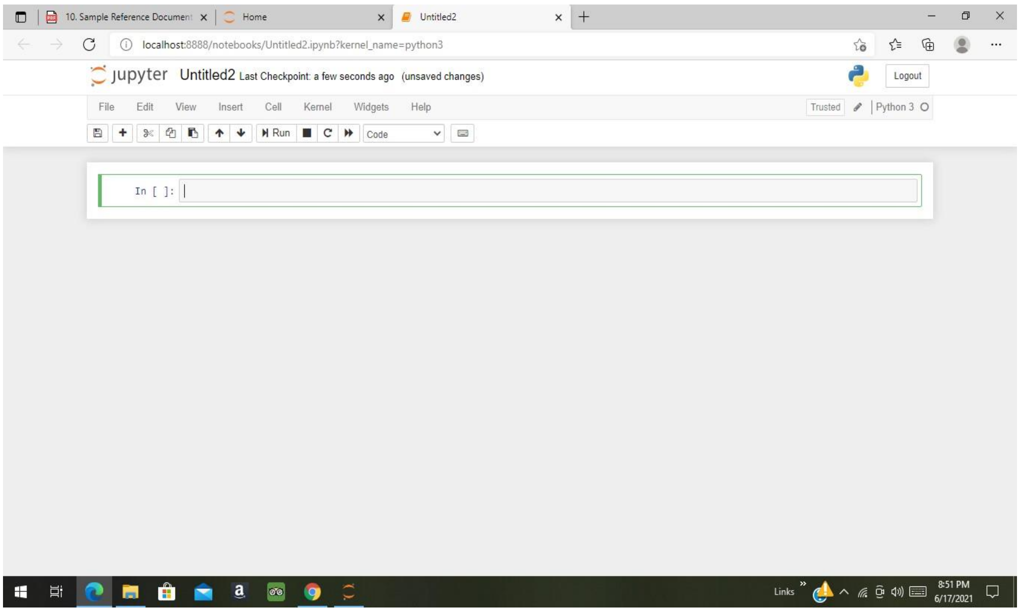
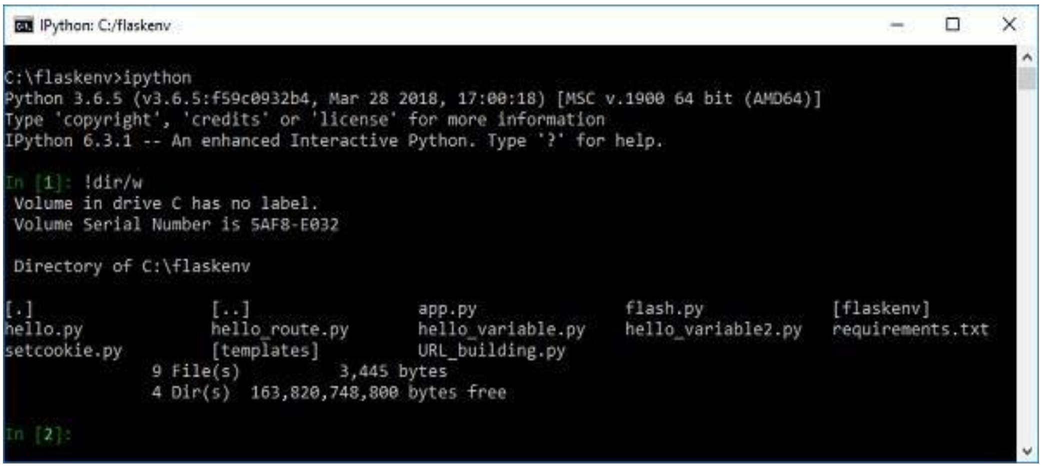

10. Sample Reference Document
Home
Untitled2
localhost:8888/notebooks/Untitled2.ipynb?kernel_name=python3
jupyter
Untitled2 Last Checkpoint: a few seconds ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
In [ ]:
Links
8:51 PM
6/17/2021

Fig.3.2 -Notebook of Jupyter notebook

3.5.3. IPYTHON SYSTEM COMMANDS

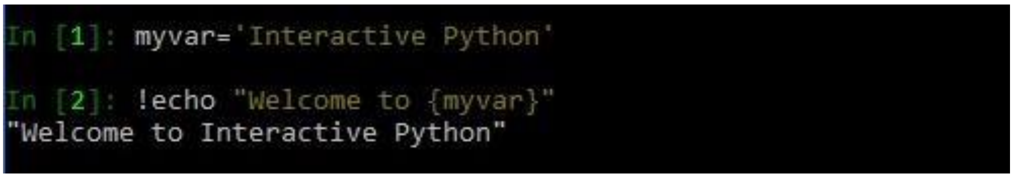
If the statement in the input cell starts with the exclamation symbol (!), it is treated as a system command for the underlying operating system. For example, ls (for Linux) and! dir (for windows) displays the contents of current directory.



```
C:\flaskenv>ipython
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 6.3.1 -- An enhanced Interactive Python. Type '?' for help.

In [1]: !dir/w
 Volume in drive C has no label.
 Volume Serial Number is 5AF8-E032

 Directory of C:\flaskenv

[.]                 [..]                app.py              flash.py            [flaskenv]
hello.py            hello_route.py      hello_variable.py   hello_variable2.py  requirements.txt
setcookie.py        [templates]         URL_building.py
               9 File(s)          3,445 bytes
               4 Dir(s)  163,820,748,800 bytes free

In [2]:
```

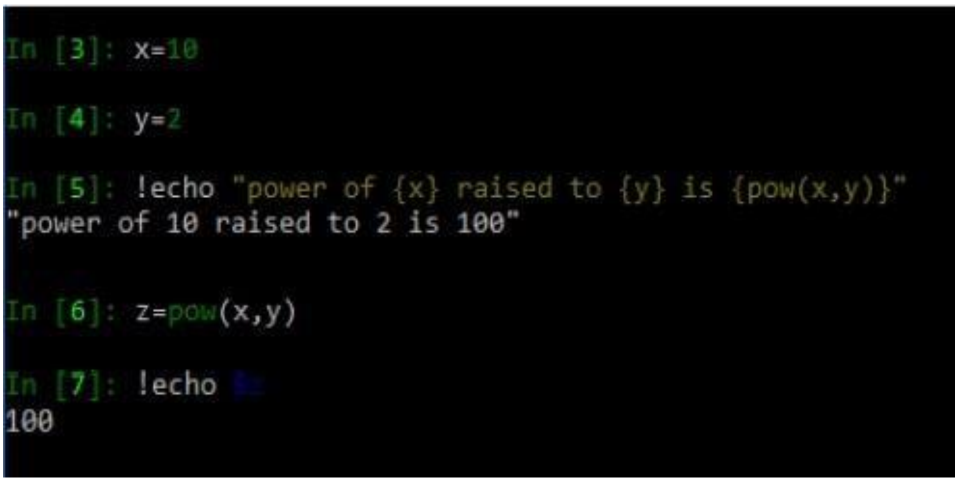
The output of system command can also be assigned to a Python variable as shown below –

The variable stores output without colors and splits at newline characters. It is also possible to combine Python variables or expressions with system command calls. Variables in curly brackets {} can be embedded in command text. Observe the following example –



```
In [1]: myvar='Interactive Python'

In [2]: !echo "Welcome to {myvar}"
"Welcome to Interactive Python"
```

Here is another example to understand that prefixing a Python variable with $ also achieves the same result.



```
In [3]: x=10

In [4]: y=2

In [5]: !echo "power of {x} raised to {y} is {pow(x,y)}"
"power of 10 raised to 2 is 100"

In [6]: z=pow(x,y)

In [7]: !echo $z
100
```

3.5.4. Project Jupyter-Overview

Project Jupyter started as a spin-off from the IPython project in 2014. IPython's language-agnostic features were moved under the name – Jupyter. The name is a reference to core programming languages supported by Jupyter which are Julia, Python and RProducts under the Jupyter project are intended to support interactive data science and scientific computing.

The project Jupyter consists of various products described as under –

IPykernel − This is a package that provides the IPython kernel to Jupyter.

Jupyter client − This package contains the reference implementation of the Jupyter protocol. It is also a client library for starting, managing and communicating with

Jupyter kernels.

Jupyter notebook − This was earlier known as IPython notebook. This is a web-based interface to IPython kernel and kernels of many other programming languages.

Jupyter kernels − Kernel is the execution environment of a programming language for Jupyter products.

Qtconsole − A rich Qt-based console for working with Jupyter kernels

nbconvert − Converts Jupyter notebook files in other formats

JupyterLab − Web based integrated interface for notebooks, editors, consoles etc.

nbviewer − HTML viewer for notebook file.

CHAPTER-4

DESIGN

4.1 UML Introduction

The unified modeling language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic, semantic and pragmatic rules. A UML system is represented using five different views that describe the system from a distinctly different perspective.

UML is specifically constructed through two different domains, they are:

- UML Analysis modeling, this focuses on the user model and structural model views of the systems.
- UML Design modeling, which focuses on the behavioral modeling, implementation modeling and environmental model views.

4.1.2 Usage of UML in Project

As the strategic value of software increases for many companies, the industry looks for techniques to automate the production of software and to improve quality and reduce cost and time to the market. These techniques include component technology, visual programming, patterns and frameworks. Additionally, the development for the World Wide Web, while making some things simpler, has exacerbated these architectural problems. The UML was designed to respond to these needs. Simply, systems design refers to the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements which can be done easily through UML diagrams.

4.2 Use Case diagram

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

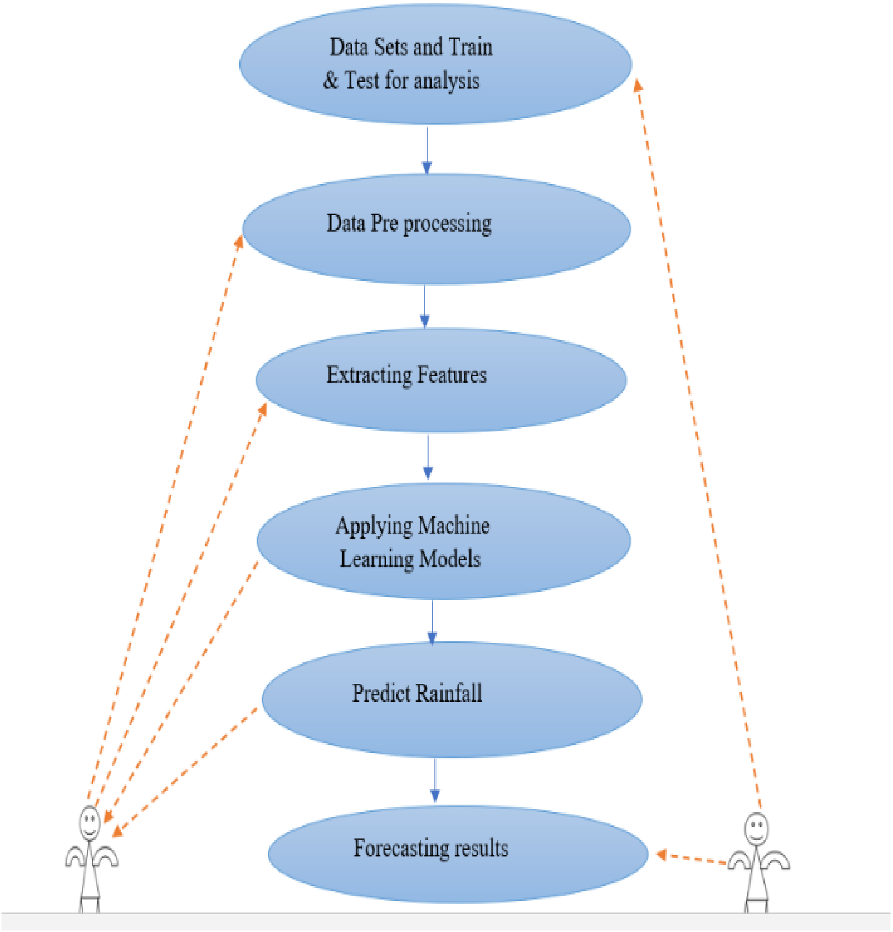


Fig 4.2 : Use case diagram of flood prediction

4.3 Architecture of project

An architecture is a way of representing the flow of data of a process or a system (usually an information system). This also provides information about the outputs and inputs of each entity and the process itself. Machine learning architecture defines the various layers involved in the machine learning cycle and involves the major steps being carried out in the transformation of raw data into training data sets capable for enabling the decision making of a system.

The below architecture describes how flood prediction can be done. Initially obtain the datasets from Kaggle website. After obtaining the datasets, perform data transformation to it in such a way that there shouldn't be any integration problem or any redundancy issue.

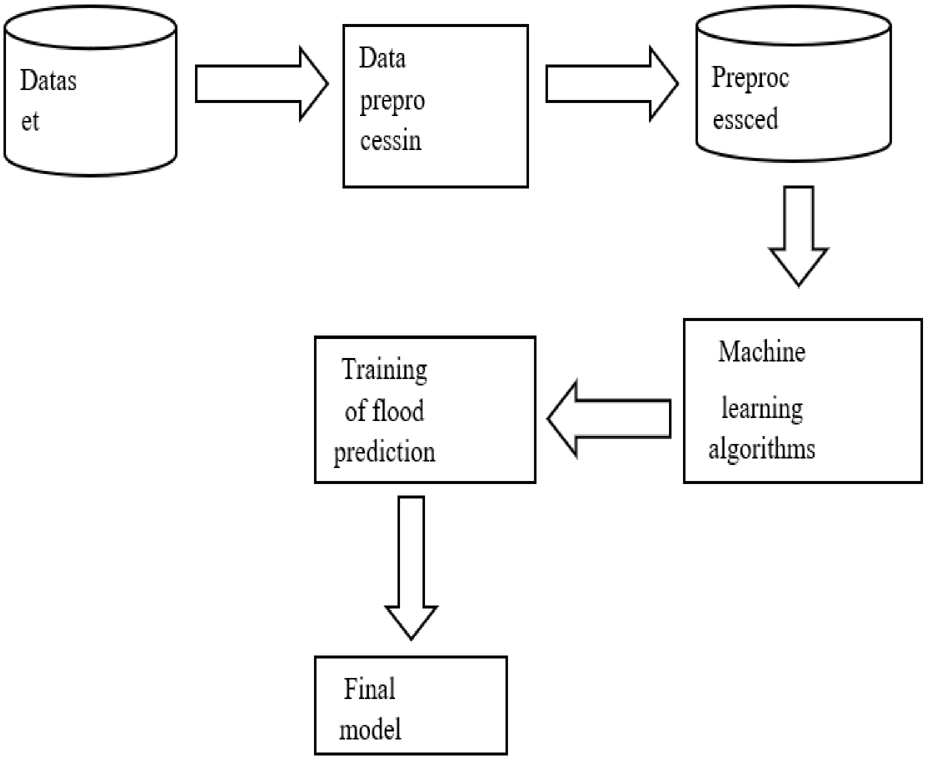


Fig.4.3 : Architecture of the Flood Prediction

Now, apply feature selection techniques to the rainfall dataset which has 5 features in it. Then a subset of features which are most important in the prediction of floods. choose the algorithm that gives the best possible accuracy with the subset of features obtained after feature selection. Applying the algorithms to the dataset actually means that it needs to train the model with the algorithms and test the data so that the model will be fit.

4.4 Steps involved in Design

- Data Collection
- Data Preprocessing
- Model Training
- Model Evaluation

4.4.1 Data Collection

- Data is an important asset for developing any kind of Machine learning model. Data collection is the process of gathering and measuring information from different kinds of sources.
- This is an initial step that has to be performed to carry out a Machine learning project. In the present internet world these datasets are available in different websites (Ex: Kaggle, Google public datasets, Data.gov etc.)
- The dataset used in our project is downloaded from the Kaggle website and it contains nearly 116 records and 20 different attributes.
- The dataset consists of 19 independent attributes and one dependent attributes.
- So, the aim of the project is to predict the dependent variables using independent variables.

4.4.2 Data Preprocessing

- Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.
- When creating a machine learning project, it is not always the case that we come across clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put it in a formatted way. So, for this, we use data preprocessing tasks.
- Preprocessing of the data consists of different kinds of steps in which analysis of the data, Data cleaning, Data encoding are part of this.

4.4.2.1 Explanatory Data Analysis

- Exploratory data analysis is an approach of analyzing datasets to summarize their main characteristics, often using statistical graphics and other data visualization methods.
- The main purpose of EDA is to help look at data before making any assumptions.
- It can help identify obvious errors, as well as better understanding patterns within the data, detect outliers or anomalous events, and find interesting relations among the variables.
- Specific statistical functions and techniques you can perform with EDA tools include:

- ➢ Dimension reduction techniques which help create graphical display of high dimensional data containing many variables.
- ➢ Univariate visualization of each field in the raw dataset, with summary statistics.
- ➢ Bivariate visualizations and summary statistics that allows you to assess the relationship between each variable in the dataset and the target variable in the dataset and the target variable you’re looking at.
- ➢ Multivariate visualizations, for mapping and understanding interactions between different fields in the data.
- ➢ This data analysis is of two types:

  a. Univariate analysis

  b. Bivariate analysis

Univariate analysis is the simplest form of data analysis where the data being analyzed contains only one variable. Since it's a single variable it doesn't deal with causes or relationships.

Bivariate data is data that involves two different variables whose values can change.

Bivariate data deals with relationships between these two variables.

4.4.2.2 Filling Missing Data & Data Encoding:

- ➢ The next step of data preprocessing is to handle missing data in the datasets. If our dataset contains some missing data, then it may create a huge problem for our machine learning model. Hence it is necessary to handle missing values present in the dataset.
- ➢ By calculating the mean and Mode: In this way, we will calculate the mean or Mode of that column or row which contains any missing value and will put it on the place of missing value. This strategy is useful for the features which have numeric data such as age, salary, year, etc.
- ➢ Data encoding: Since the machine learning model completely works on mathematics and numbers, but if our dataset would have a categorical variable, then it may create trouble while building the model. So, it is necessary to encode these categorical variables into numbers.
- ➢ Our dataset also consists of different categorical data in which they are encoded in this step.

4.4.3 Training the Model:

In this step the model is trained using the algorithms that are suitable. Flood prediction is a kind of problem in which One variable has to be determined using some independent variables Regression model is suitable for this kind of scenario.

- A training model is a dataset that is used to train an ML algorithm. It consists of the sample output data and the corresponding sets of input data that have an influence on the output. The training model is used to run the input data through the algorithm to correlate the processed output against the sample output.
- Our project implements these algorithms like Logistic Regression, K Nearest Neighbor, Support Vector Machine, Decision Tree, Random Forest.

4.4.3.1 Logistic Regression

Logistic Regression may be a machine learning algorithm that predicts the probability of a categorical variable. It is a statistical way of analyzing a group of knowledge that comprises quite one experimental variable that determines the result. The outcome is then measured with a dichotomous variable. The goal of this algorithm is to seek out the simplest model to explain the connection between a dichotomous characteristic of interest and a group of independent variables. In this algorithm, the dependent variable is a binary variable that contains data coded as 1 or 0. In other words, the logistic regression model predicts P(Y=1) as a function of X.

4.4.3.2 K Nearest Neighbor

K-Nearest Neighbor is one among the supervised machine learning algorithms that stores all instances like training data points in an n-dimensional space. For real-valued data, the algorithm returns the mean of k nearest neighbors, and in case of receiving unknown discrete data, it analyses the closest k number of instances that is saved and returns the most common class as the result of the prediction. In the distance-weighted nearest neighbor algorithm, the contribution of each of the k neighbors is weighed according to their distance, giving higher weight to the closest neighbors. The K-Nearest Neighbor algorithm is a classification algorithm and is robust to noisy data as it averages the k-nearest neighbors. The algorithm first takes a bunch of labeled points and analyses them to find out the way to label the opposite points. Hence, to label a new point, it looks at the closest labeled points to that new point and has those neighbors vote, so whichever label most of the neighbors have been the label for the new point. This algorithm makes predictions about the validation set using the whole training set. Only by rummaging through the whole training set to seek out the closest instances, the new instance is predicted. Closeness is a value that is determined using a proximity measurement across all the features involved.

4.4.3.3 Support Vector Machines

SVM uses a classifier that categorizes the info set by setting an optimal hyperplane between data. This classifier is chosen as it is incredibly versatile in the number of different kernel functions that can be applied, and this model can yield a high predictability rate. Support Vector Machine is one among the foremost popular and widely used clustering algorithms. It belongs to a gaggle of generalized linear classifiers and is taken into account as an extension of the perceptron. It was developed in the 1990s and continues to be the desired method for a high-performance algorithm with a little tuning.

4.4.3.4 Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.

4.4.3.5 Decision Tree

Decision tree learning is a method commonly used in data mining. The goal is to create a model that predicts the value of a target variable based on several input variables. An example is shown in the diagram at right. Each interior node corresponds to one of the input variables; there are edges to children for each of the possible values of that input variable. Each leaf represents a value of the target variable given the values of the input variables represented by the path from the root to the leaf.

4.4.4 Model Evaluation

In this step the trained model is evaluated by determining the accuracy of the model against the test data. various ways to check the performance of our machine learning or deep learning model and why to use one in place of the other. We will discuss terms like:

1.Accuracy

2.Recall

3.ROC Curve

Out of these we used Accuracy for evaluating our model. Accuracy is the most commonly used metric to judge a model and is actually not a clear indicator of the performance. The worst happens when classes are imbalanced.

CHAPTER-5

IMPLEMENTATION

Implementation part is made using CSV file containing 20 different attributes with nearly 116 records. Floods are predicted using the data collected with Machine Learning algorithms like Logistic regression, KNN, SVM, Decision, Random Forest All these algorithms help to predict the Floods. Floods are predicted by implementing all these five algorithms separately and are compared one with another.

5.1. Libraries Used

Python is increasingly being used as a scientific language. Matrix and vector manipulation are extremely important for scientific computations. Both NumPy and Pandas have emerged to be essential libraries for any scientific computation, including machine learning, in python due to their intuitive syntax and high-performance matrix computation capabilities.

Pip

The pip command is a tool for installing and managing Python packages, such as those found in the Python Package Index. It's a replacement for easy installation. The easiest way to install the nfl* python modules and keep them up-to-date is with a Python-based package manager called pip.

pip install (module name)

NumPy

NumPy stands for ‘Numerical Python’ or ‘Numeric Python’. It is an open-source module of Python which provides fast mathematical computation on arrays and matrices. Since arrays and matrices are an essential part of the Machine Learning ecosystem, NumPy along with Machine Learning modules like Scikit-learn, Pandas, Matplotlib, TensorFlow, etc. complete the Python Machine Learning Ecosystem. NumPy provides the essential multi-dimensional array-oriented computing functionalities designed for high-level mathematical functions and scientific computation. NumPy can be imported into the notebook using

import numpy as np.

Pandas

Similar to NumPy, Pandas is one of the most widely used python libraries in data science. It provides high-performance, easy to use structures and data analysis tools. Pandas provides an in-memory 2d table object called Data frame. It is like a spreadsheet with column names and row labels. Hence, with 2d tables, pandas are capable of providing many additional functionalities like creating pivot tables, computing columns based on other columns and plotting graphs. Pandas can be imported into Python using:

import pandas as pd.

Matplotlib

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc. Matplotlib comes with a wide variety of plots. Plots help to understand trends, patterns, and to make correlations. They're typically instruments for reasoning about quantitative information. Matplotlib can be imported into Python using:

import matplotlib. pyplot as plt

Seaborn

Seaborn is a data visualization library built on top of matplotlib and closely integrated with pandas' data structures in Python. Visualization is the central part of Seaborn which helps in exploration and understanding of data. Seaborn offers the functionalities like Dataset oriented API to determine the relationship between variables, Automatic estimation and plotting of linear regression plots, it supports high-level abstractions for multi-plot grids and Visualizing univariate and bivariate distribution.

Sklearn

Scikit-learn is a free software machine library for Python programming language. It features various classification, regression and clustering algorithms. In our project we have used different features of sklearn library like:

from sklearn.preprocessing import LabelEncoder

In machine learning, we usually deal with datasets which contain multiple labels in one or more than one column. These labels can be in the form of words or numbers. To make the data understandable or in human readable form, the training data is often labeled in words.

Label Encoding refers to converting the labels into numeric form so as to convert it into the machine-readable form. Machine learning algorithms can then decide in a better way on how those labels must be operated. It is an important preprocessing step for the structured dataset in supervised learning.

Label encoding converts the data in machine readable form, but it assigns a unique number (starting from 0) to each class of data. This may lead to the generation of priority issues in training of data sets. A label with high value may be considered to have high priority than a label having lower value.

from sklearn.preprocessing import PolynomialFeatures

Polynomial features of sklearn are mainly useful for the implementation of Polynomial regression algorithm of different kind of degrees like 1,2,3,4…..... This feature of sklearn help to fit the dataset with Polynomial regression algorithm. So that it helps to Predict the sales.

from sklearn.model_selection import train_test_split

The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model. It is a fast and easy procedure to perform, the results of which allow you to compare the performance of machine learning algorithms for your predictive modeling problem. Although simple to use and interpret, there are times when the procedure should not be used, such as when you have a small dataset and situations where additional configuration is required, such as when it is used for classification and the dataset is not balanced.

- **Train Dataset**: Used to fit the machine learning model.
- **Test Dataset**: Used to evaluate the fit machine learning model.

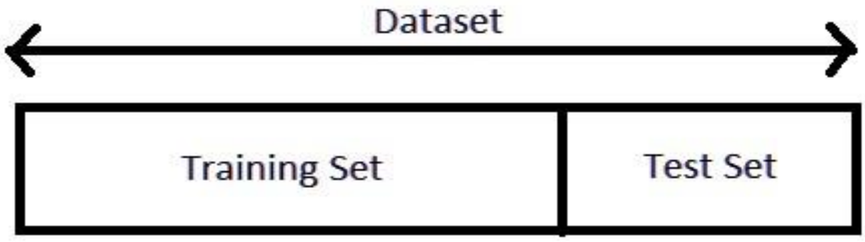


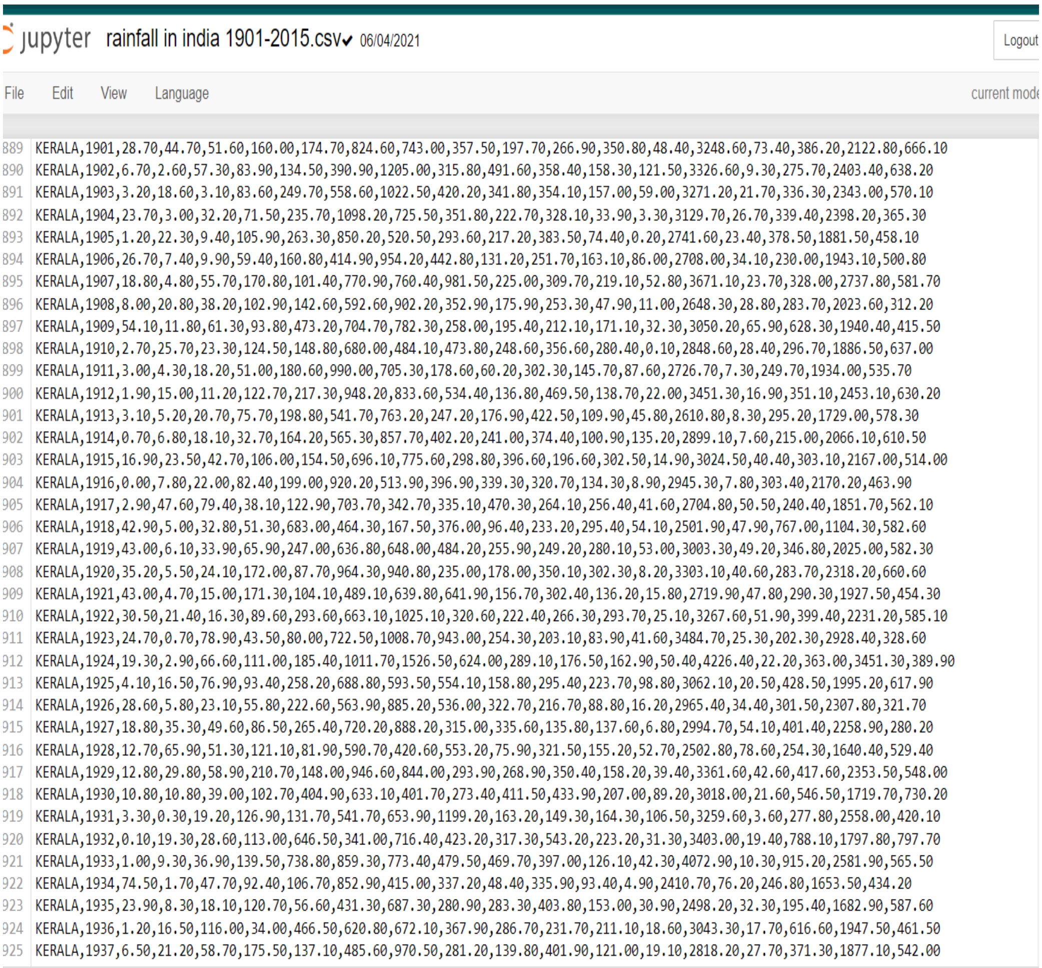
Fig.5.1 Training and test data set

CSV file

The dataset used in this project is a .CSV file. In computing, a comma-separated values (CSV) file is a delimited text file that uses a comma to separate values. A CSV file stores tabular data (numbers and text) in plain text. Each line of the file is a data record. Each record consists of one or more fields, separated by commas. The use of the comma as a field separator is the source of the name for this file format. CSV is a simple file format used to store tabular data, such as a spreadsheet or database. Files in the CSV format can be imported to and exported from programs that store data in tables, such as Microsoft Excel or OpenOffice Calc. Its data fields are most often separated, or delimited, by a comma.

A CSV is a comma-separated values file, which allows data to be saved in a tabular format. CSVs look like a garden-variety spreadsheet but with a .csv extension. CSV files can be used with most any spreadsheet program, such as Microsoft Excel or Google Spreadsheets.

The difference between CSV and XLS file formats is that CSV format is a plain text format in which values are separated by commas (Comma Separated Values), while XLS file format is an Excel Sheets binary file format which holds information about all the worksheets in a file, including both content and formatting.



```
KERALA,1901,28.70,44.70,51.60,160.00,174.70,824.60,743.00,357.50,197.70,266.90,350.80,48.40,3248.60,73.40,386.20,2122.80,666.10
KERALA,1902,6.70,2.60,57.30,83.90,134.50,390.90,1205.00,315.80,491.60,358.40,158.30,121.50,3326.60,9.30,275.70,2403.40,638.20
KERALA,1903,3.20,18.60,3.10,83.60,249.70,558.60,1022.50,420.20,341.80,354.10,157.00,59.00,3271.20,21.70,336.30,2343.00,570.10
KERALA,1904,23.70,3.00,32.20,71.50,235.70,1098.20,725.50,351.80,222.70,328.10,33.90,3.30,3129.70,26.70,339.40,2398.20,365.30
KERALA,1905,1.20,22.30,9.40,105.90,263.30,850.20,520.50,293.60,217.20,383.50,74.40,0.20,2741.60,23.40,378.50,1881.50,458.10
KERALA,1906,26.70,7.40,9.90,59.40,160.80,414.90,954.20,442.80,131.20,251.70,163.10,86.00,2708.00,34.10,230.00,1943.10,500.80
KERALA,1907,18.80,4.80,55.70,170.80,101.40,770.90,760.40,981.50,225.00,309.70,219.10,52.80,3671.10,23.70,328.00,2737.80,581.70
KERALA,1908,8.00,20.80,38.20,102.90,142.60,592.60,902.20,352.90,175.90,253.30,47.90,11.00,2648.30,28.80,283.70,2023.60,312.20
KERALA,1909,54.10,11.80,61.30,93.80,473.20,704.70,782.30,258.00,195.40,212.10,171.10,32.30,3050.20,65.90,628.30,1940.40,415.50
KERALA,1910,2.70,25.70,23.30,124.50,148.80,680.00,484.10,473.80,248.60,356.60,280.40,0.10,2848.60,28.40,296.70,1886.50,637.00
KERALA,1911,3.00,4.30,18.20,51.00,180.60,990.00,705.30,178.60,60.20,302.30,145.70,87.60,2726.70,7.30,249.70,1934.00,535.70
KERALA,1912,1.90,15.00,11.20,122.70,217.30,948.20,833.60,534.40,136.80,469.50,138.70,22.00,3451.30,16.90,351.10,2453.10,630.20
KERALA,1913,3.10,5.20,20.70,75.70,198.80,541.70,763.20,247.20,176.90,422.50,109.90,45.80,2610.80,8.30,295.20,1729.00,578.30
KERALA,1914,0.70,6.80,18.10,32.70,164.20,565.30,857.70,402.20,241.00,374.40,100.90,135.20,2899.10,7.60,215.00,2066.10,610.50
KERALA,1915,16.90,23.50,42.70,106.00,154.50,696.10,775.60,298.80,396.60,196.60,302.50,14.90,3024.50,40.40,303.10,2167.00,514.00
KERALA,1916,0.00,7.80,22.00,82.40,199.00,920.20,513.90,396.90,339.30,320.70,134.30,8.90,2945.30,7.80,303.40,2170.20,463.90
KERALA,1917,2.90,47.60,79.40,38.10,122.90,703.70,342.70,335.10,470.30,264.10,256.40,41.60,2704.80,50.50,240.40,1851.70,562.10
KERALA,1918,42.90,5.00,32.80,51.30,683.00,464.30,167.50,376.00,96.40,233.20,295.40,54.10,2501.90,47.90,767.00,1104.30,582.60
KERALA,1919,43.00,6.10,33.90,65.90,247.00,636.80,648.00,484.20,255.90,249.20,280.10,53.00,3003.30,49.20,346.80,2025.00,582.30
KERALA,1920,35.20,5.50,24.10,172.00,87.70,964.30,940.80,235.00,178.00,350.10,302.30,8.20,3303.10,40.60,283.70,2318.20,660.60
KERALA,1921,43.00,4.70,15.00,171.30,104.10,489.10,639.80,641.90,156.70,302.40,136.20,15.80,2719.90,47.80,290.30,1927.50,454.30
KERALA,1922,30.50,21.40,16.30,89.60,293.60,663.10,1025.10,320.60,222.40,266.30,293.70,25.10,3267.60,51.90,399.40,2231.20,585.10
KERALA,1923,24.70,0.70,78.90,43.50,80.00,722.50,1008.70,943.00,254.30,203.10,83.90,41.60,3484.70,25.30,202.30,2928.40,328.60
KERALA,1924,19.30,2.90,66.60,111.00,185.40,1011.70,1526.50,624.00,289.10,176.50,162.90,50.40,4226.40,22.20,363.00,3451.30,389.90
KERALA,1925,4.10,16.50,76.90,93.40,258.20,688.80,593.50,554.10,158.80,295.40,223.70,98.80,3062.10,20.50,428.50,1995.20,617.90
KERALA,1926,28.60,5.80,23.10,55.80,222.60,563.90,885.20,536.00,322.70,216.70,88.80,16.20,2965.40,34.40,301.50,2307.80,321.70
KERALA,1927,18.80,35.30,49.60,86.50,265.40,720.20,888.20,315.00,335.60,135.80,137.60,6.80,2994.70,54.10,401.40,2258.90,280.20
KERALA,1928,12.70,65.90,51.30,121.10,81.90,590.70,420.60,553.20,75.90,321.50,155.20,52.70,2502.80,78.60,254.30,1640.40,529.40
KERALA,1929,12.80,29.80,58.90,210.70,148.00,946.60,844.00,293.90,268.90,350.40,158.20,39.40,3361.60,42.60,417.60,2353.50,548.00
KERALA,1930,10.80,10.80,39.00,102.70,404.90,633.10,401.70,273.40,411.50,433.90,207.00,89.20,3018.00,21.60,546.50,1719.70,730.20
KERALA,1931,3.30,0.30,19.20,126.90,131.70,541.70,653.90,1199.20,163.20,149.30,164.30,106.50,3259.60,3.60,277.80,2558.00,420.10
KERALA,1932,0.10,19.30,28.60,113.00,646.50,341.00,716.40,423.20,317.30,543.20,223.20,31.30,3403.00,19.40,788.10,1797.80,797.70
KERALA,1933,1.00,9.30,36.90,139.50,738.80,859.30,773.40,479.50,469.70,397.00,126.10,42.30,4072.90,10.30,915.20,2581.90,565.50
KERALA,1934,74.50,1.70,47.70,92.40,106.70,852.90,415.00,337.20,48.40,335.90,93.40,4.90,2410.70,76.20,246.80,1653.50,434.20
KERALA,1935,23.90,8.30,18.10,120.70,56.60,431.30,687.30,280.90,283.30,403.80,153.00,30.90,2498.20,32.30,195.40,1682.90,587.60
KERALA,1936,1.20,16.50,116.00,34.00,466.50,620.80,672.10,367.90,286.70,231.70,211.10,18.60,3043.30,17.70,616.60,1947.50,461.50
KERALA,1937,6.50,21.20,58.70,175.50,137.10,485.60,970.50,281.20,139.80,401.90,121.00,19.10,2818.20,27.70,371.30,1877.10,542.00
```

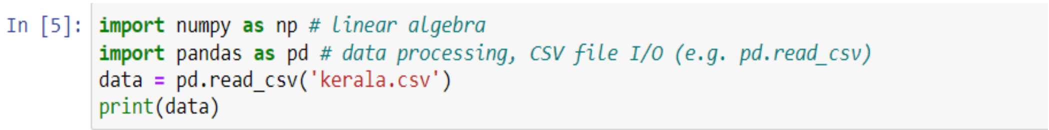
Fig.5.2 Dataset with 19 different attributes

5.2 Implementation

5.2.1 Importing all the required Modules and Libraries

All the required libraries like Sklearn and Modules like NumPy, Pandas, Matplotlib, Seaborn are imported into the Jupyter notebook initially into the file created in the notebook.

After importing all the modules and libraries into the notebook, A csv file has to be loaded using Pandas into the notebook. The implementation of these will be as follows:



```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
data = pd.read_csv('kerala.csv')
print(data)
```

Fig.5.3 Importing of modules and libraries

5.2.2. Data Visualization

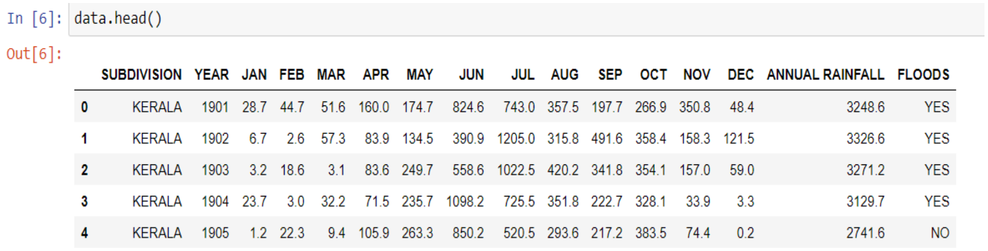
As it contains large amounts of data it is not possible to analyze with the human eye normally so the feature of Data Visualization helps to analyze the entire data. The relation between any two features can be only analyzed with the Data Visualization technique. This Visualization can be made in different forms by representing the data in the pictorial forms like graph, bar chart and many other forms.

Some Visualizations are made for the dataset that is collected for the Prediction of floods. They are as follows:

5.2.2.1. Descriptive Analyzation:

It is important to know about the information of each and every attribute, such information can be easily predicted and is analyzed as follows:

Panda’s head () method is used to return top n (5 by default) rows of a data frame or series.



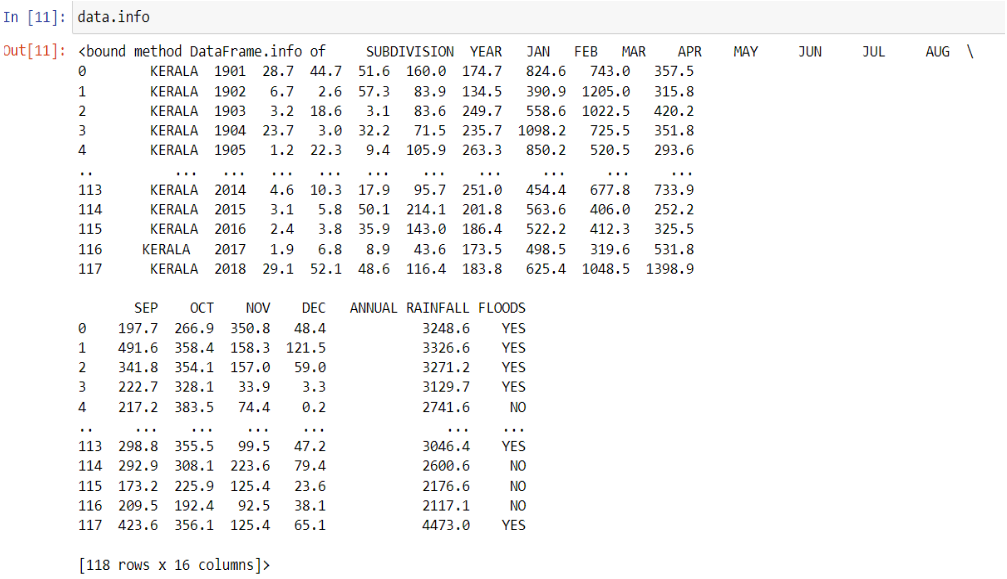
```
In [6]: data.head()
Out[6]:
```

	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL RAINFALL	FLOODS
0	KERALA	1901	28.7	44.7	51.6	160.0	174.7	824.6	743.0	357.5	197.7	266.9	350.8	48.4	3248.6	YES
1	KERALA	1902	6.7	2.6	57.3	83.9	134.5	390.9	1205.0	315.8	491.6	358.4	158.3	121.5	3326.6	YES
2	KERALA	1903	3.2	18.6	3.1	83.6	249.7	558.6	1022.5	420.2	341.8	354.1	157.0	59.0	3271.2	YES
3	KERALA	1904	23.7	3.0	32.2	71.5	235.7	1098.2	725.5	351.8	222.7	328.1	33.9	3.3	3129.7	YES
4	KERALA	1905	1.2	22.3	9.4	105.9	263.3	850.2	520.5	293.6	217.2	383.5	74.4	0.2	2741.6	NO

Fig.5.4 Pandas head method

The info () function is used to print a concise summary of a Data Frame. This method prints information about a Data Frame including the index dtype and column dtypes, non-null values and memory usage.

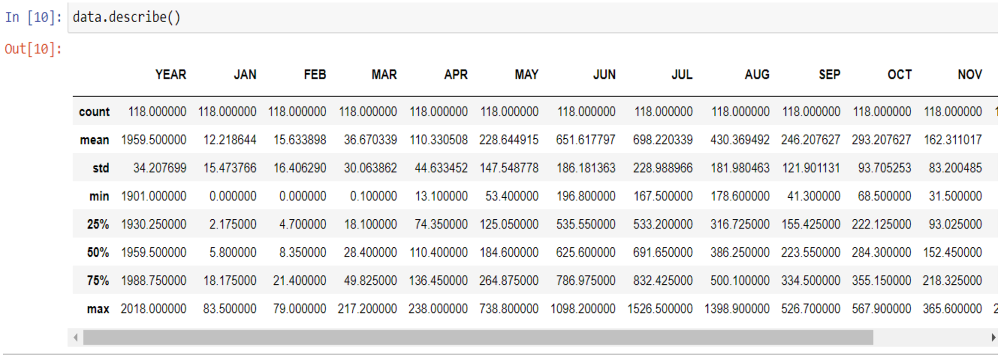
This method helps to provide a very small and important summary of the entire dataset so that it is easy to have an idea on the entire dataset that is present. It provides information about Count of the attribute, Type of the attribute along with the attribute name etc.



```
In [11]: data.info

Out[11]: <bound method DataFrame.info of      SUBDIVISION  YEAR   JAN   FEB   MAR    APR    MAY     JUN     JUL     AUG  \
         0        KERALA  1901  28.7  44.7  51.6  160.0  174.7   824.6   743.0   357.5
         1        KERALA  1902   6.7   2.6  57.3   83.9  134.5   390.9  1205.0   315.8
         2        KERALA  1903   3.2  18.6   3.1   83.6  249.7   558.6  1022.5   420.2
         3        KERALA  1904  23.7   3.0  32.2   71.5  235.7  1098.2   725.5   351.8
         4        KERALA  1905   1.2  22.3   9.4  105.9  263.3   850.2   520.5   293.6
         ..          ...   ...   ...   ...   ...    ...    ...     ...     ...     ...
         113      KERALA  2014   4.6  10.3  17.9   95.7  251.0   454.4   677.8   733.9
         114      KERALA  2015   3.1   5.8  50.1  214.1  201.8   563.6   406.0   252.2
         115      KERALA  2016   2.4   3.8  35.9  143.0  186.4   522.2   412.3   325.5
         116     KERALA   2017   1.9   6.8   8.9   43.6  173.5   498.5   319.6   531.8
         117      KERALA  2018  29.1  52.1  48.6  116.4  183.8   625.4  1048.5  1398.9

                SEP    OCT    NOV    DEC   ANNUAL RAINFALL FLOODS
         0    197.7  266.9  350.8   48.4            3248.6    YES
         1    491.6  358.4  158.3  121.5            3326.6    YES
         2    341.8  354.1  157.0   59.0            3271.2    YES
         3    222.7  328.1   33.9    3.3            3129.7    YES
         4    217.2  383.5   74.4    0.2            2741.6     NO
         ..     ...    ...    ...    ...               ...    ...
         113  298.8  355.5   99.5   47.2            3046.4    YES
         114  292.9  308.1  223.6   79.4            2600.6     NO
         115  173.2  225.9  125.4   23.6            2176.6     NO
         116  209.5  192.4   92.5   38.1            2117.1     NO
         117  423.6  356.1  125.4   65.1            4473.0    YES

         [118 rows x 16 columns]>
```

Fig.5.5 : Pandas info method()

Pandas **describe ()** is used to view some basic statistical details like percentile, mean, std etc. of a data frame or a series of numeric values. When this method is applied to any kind of dataset the output will be as follows:



```
In [10]: data.describe()

Out[10]:
```

	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV
count	118.000000	118.000000	118.000000	118.000000	118.000000	118.000000	118.000000	118.000000	118.000000	118.000000	118.000000	118.000000
mean	1959.500000	12.218644	15.633898	36.670339	110.330508	228.644915	651.617797	698.220339	430.369492	246.207627	293.207627	162.311017
std	34.207699	15.473766	16.406290	30.063862	44.633452	147.548778	186.181363	228.988966	181.980463	121.901131	93.705253	83.200485
min	1901.000000	0.000000	0.000000	0.100000	13.100000	53.400000	196.800000	167.500000	178.600000	41.300000	68.500000	31.500000
25%	1930.250000	2.175000	4.700000	18.100000	74.350000	125.050000	535.550000	533.200000	316.725000	155.425000	222.125000	93.025000
50%	1959.500000	5.800000	8.350000	28.400000	110.400000	184.600000	625.600000	691.650000	386.250000	223.550000	284.300000	152.450000
75%	1988.750000	18.175000	21.400000	49.825000	136.450000	264.875000	786.975000	832.425000	500.100000	334.500000	355.150000	218.325000
max	2018.000000	83.500000	79.000000	217.200000	238.000000	738.800000	1098.200000	1526.500000	1398.900000	526.700000	567.900000	365.600000

Fig.5.6 pandas describe() method

Mean: Mean value of the entire column and in the dataset.

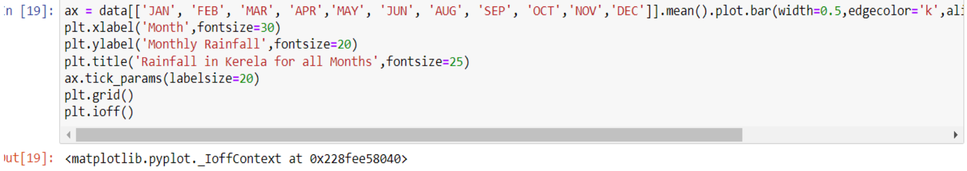
Min & Max: Minimum and Maximum values of the entire column.

5.2.2.2. Univariate Analysis

This kind of analysis which is Univariate helps to find and analyze all the information about a single attribute and Variable in the dataset. So that we can determine the uniformity and any kind of uneven nature of the variable can also be predetermined.

5.2.2.3. Bivariate analysis

Bivariate analysis means the analysis of the bivariate data. This is a single statistical analysis that is used to find out the relationship that exists between two value sets. The variables that are involved are X and Y.



```
In [19]: ax = data[['JAN', 'FEB', 'MAR', 'APR','MAY', 'JUN', 'AUG', 'SEP', 'OCT','NOV','DEC']].mean().plot.bar(width=0.5,edgecolor='k',ali
         plt.xlabel('Month',fontsize=30)
         plt.ylabel('Monthly Rainfall',fontsize=20)
         plt.title('Rainfall in Kerela for all Months',fontsize=25)
         ax.tick_params(labelsize=20)
         plt.grid()
         plt.ioff()

Out[19]: <matplotlib.pyplot._IoffContext at 0x228fee58040>
```

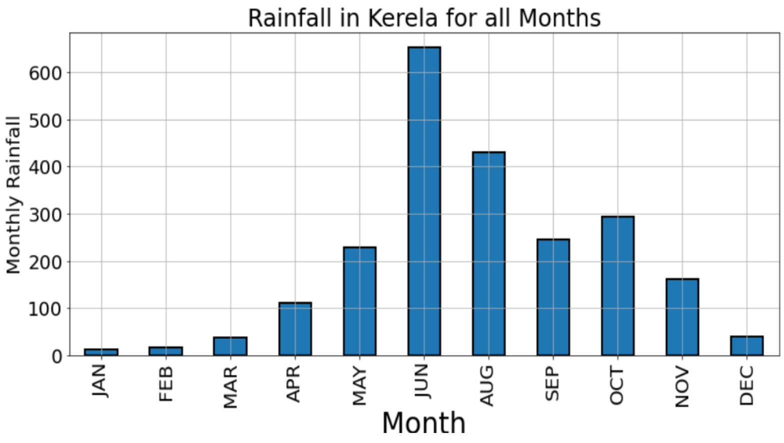
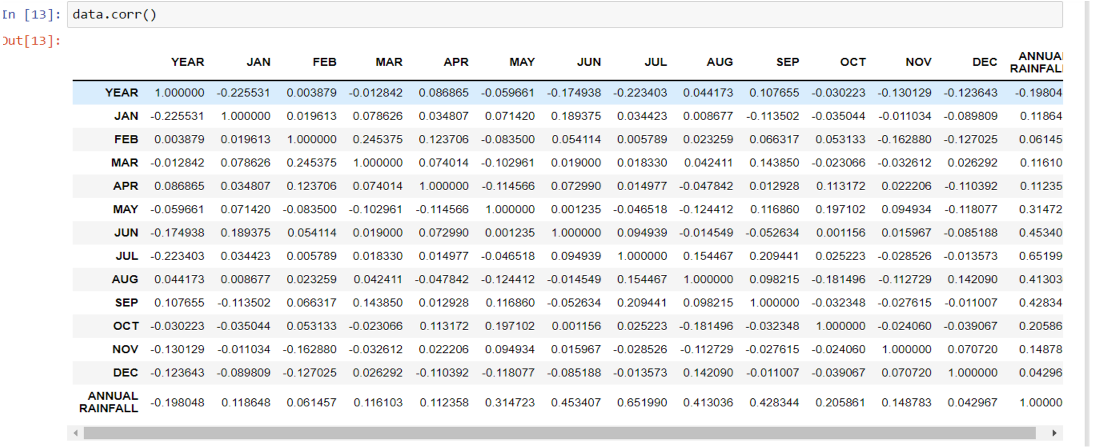


Fig.5.7 : month Vs monthly rainfall analysis

5.2.2.3. Correlation function

Correlation corr() is used to find the pairwise correlation of all columns in the data frame. Any na values are automatically excluded. For any non-numeric data type columns in the data frame, it is ignored.



```
In [13]: data.corr()
Out[13]:
```

	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL RAINFALL
YEAR	1.000000	-0.225531	0.003879	-0.012842	0.086865	-0.059661	-0.174938	-0.223403	0.044173	0.107655	-0.030223	-0.130129	-0.123643	-0.19804
JAN	-0.225531	1.000000	0.019613	0.078626	0.034807	0.071420	0.189375	0.034423	0.008677	-0.113502	-0.035044	-0.011034	-0.089809	0.11864
FEB	0.003879	0.019613	1.000000	0.245375	0.123706	-0.083500	0.054114	0.005789	0.023259	0.066317	0.053133	-0.162880	-0.127025	0.06145
MAR	-0.012842	0.078626	0.245375	1.000000	0.074014	-0.102961	0.019000	0.018330	0.042411	0.143850	-0.023066	-0.032612	0.026292	0.11610
APR	0.086865	0.034807	0.123706	0.074014	1.000000	-0.114566	0.072990	0.014977	-0.047842	0.012928	0.113172	0.022206	-0.110392	0.11235
MAY	-0.059661	0.071420	-0.083500	-0.102961	-0.114566	1.000000	0.001235	-0.046518	-0.124412	0.116860	0.197102	0.094934	-0.118077	0.31472
JUN	-0.174938	0.189375	0.054114	0.019000	0.072990	0.001235	1.000000	0.094939	-0.014549	-0.052634	0.001156	0.015967	-0.085188	0.45340
JUL	-0.223403	0.034423	0.005789	0.018330	0.014977	-0.046518	0.094939	1.000000	0.154467	0.209441	0.025223	-0.028526	-0.013573	0.65199
AUG	0.044173	0.008677	0.023259	0.042411	-0.047842	-0.124412	-0.014549	0.154467	1.000000	0.098215	-0.181496	-0.112729	0.142090	0.41303
SEP	0.107655	-0.113502	0.066317	0.143850	0.012928	0.116860	-0.052634	0.209441	0.098215	1.000000	-0.032348	-0.027615	-0.011007	0.42834
OCT	-0.030223	-0.035044	0.053133	-0.023066	0.113172	0.197102	0.001156	0.025223	-0.181496	-0.032348	1.000000	-0.024060	-0.039067	0.20586
NOV	-0.130129	-0.011034	-0.162880	-0.032612	0.022206	0.094934	0.015967	-0.028526	-0.112729	-0.027615	-0.024060	1.000000	0.070720	0.14878
DEC	-0.123643	-0.089809	-0.127025	0.026292	-0.110392	-0.118077	-0.085188	-0.013573	0.142090	-0.011007	-0.039067	0.070720	1.000000	0.04296
ANNUAL RAINFALL	-0.198048	0.118648	0.061457	0.116103	0.112358	0.314723	0.453407	0.651990	0.413036	0.428344	0.205861	0.148783	0.042967	1.00000

Fig.5.8 : Correlation values

The above values show the correlation between two variables. They help to determine the rate of change between two variables.

5.3. Feature Engineering

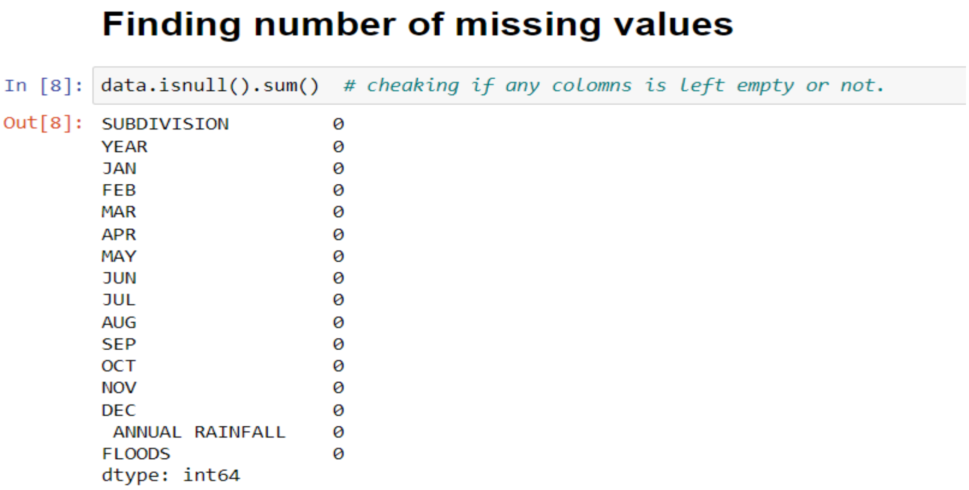
What is a feature and why do we need the engineering of it? Basically, all machine learning algorithms use some input data to create outputs. This input data comprises features, which are usually in the form of structured columns. Algorithms require features with some specific characteristics to work properly. Here, the need for feature engineering arises. I think feature engineering efforts mainly have two goals:

- Preparing the proper input dataset, compatible with the machine learning algorithm requirements.
- Improving the performance of machine learning models.

Item Visibility is one of the features of the dataset in which maximum number of values in that row are Zeros so they have to be normalized and are set to some value. So, mean of the entire column of data_visibilty is calculated and the value is set to that mean value. This make all the Zeros of the column to particular value and accuracy of the model can be made more efficient .

5.4. Checking Missing values

In order to check missing values in Pandas Data Frame, we use a function isnull() and notnull() . Both function help in checking whether a value is NaN or not. This function can also be used in Pandas Series in order to find null values in a series.

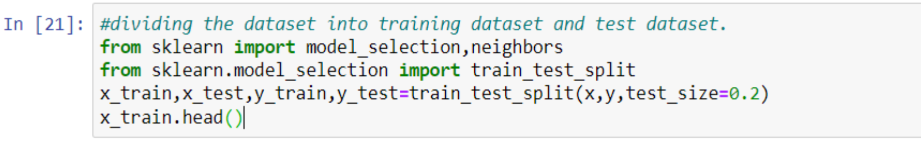


Finding number of missing values

```
In [8]:  data.isnull().sum()  # cheaking if any colomns is left empty or not.

Out[8]:  SUBDIVISION         0
         YEAR                0
         JAN                 0
         FEB                 0
         MAR                 0
         APR                 0
         MAY                 0
         JUN                 0
         JUL                 0
         AUG                 0
         SEP                 0
         OCT                 0
         NOV                 0
         DEC                 0
          ANNUAL RAINFALL    0
         FLOODS              0
         dtype: int64
```

Fig.5.9 : Finding missing value

5.4.1. Splitting the dataset



```
In [21]:  #dividing the dataset into training dataset and test dataset.
          from sklearn import model_selection,neighbors
          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)
          x_train.head()
```

Fig.5.10 : Train and Test split of data

For all the Machine learning models to train with any algorithm of their choice the dataset has to be divided into two parts called Training dataset and Testing dataset. Generally, the training dataset will be 80% of the entire dataset and 20% of the data as the Testing dataset.

Training dataset will be used to train the model and testing dataset will be used to find the accuracy of our predicted model. Performance evaluation can be made with the accuracy of the trained model with required algorithm.

Those splitting of the dataset to train and test splitting can be made using the command from the sklearn library as shown above.

X_train-Represents train dataset.

X_test-Represents test dataset.

5.5. Model prediction

In this prediction the entire data is trained with five different models in which each model provides different output values of different accuracies. All the models are compared and the conclusions are made.

5.5.1. Modelling with KNN Algorithm

Step1: Load the dataset

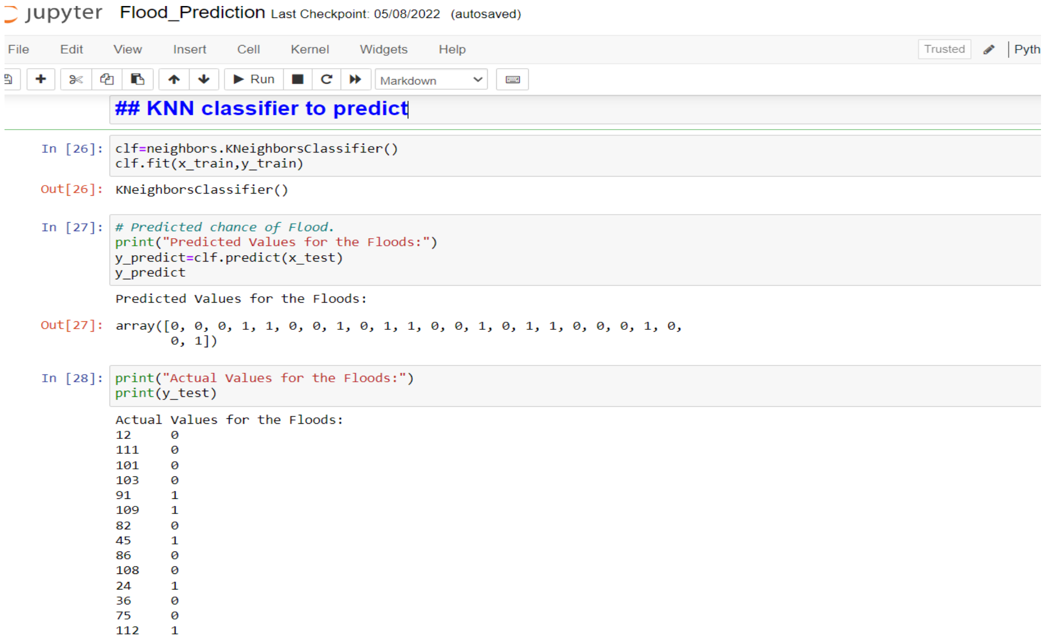
Step2: Divide the dataset

Step3: Assign the KNN algorithm to a variable.

Step4: Fit the training dataset using KNN algorithm

Step5: Predict the values for the testing dataset using a trained model.

Step6: Check the accuracy of the model.



KNN classifier to predict

```
In [26]: clf=neighbors.KNeighborsClassifier()
         clf.fit(x_train,y_train)

Out[26]: KNeighborsClassifier()

In [27]: # Predicted chance of Flood.
         print("Predicted Values for the Floods:")
         y_predict=clf.predict(x_test)
         y_predict

         Predicted Values for the Floods:

Out[27]: array([0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0,
                0, 1])

In [28]: print("Actual Values for the Floods:")
         print(y_test)

         Actual Values for the Floods:
         12     0
         111    0
         101    0
         103    0
         91     1
         109    1
         82     0
         45     1
         86     0
         108    0
         24     1
         36     0
         75     0
         112    1
```

Fig.5.11 : Fitting model with KNN

5.5.2. Modelling with Logistic regression Algorithm

Step1: Load the dataset

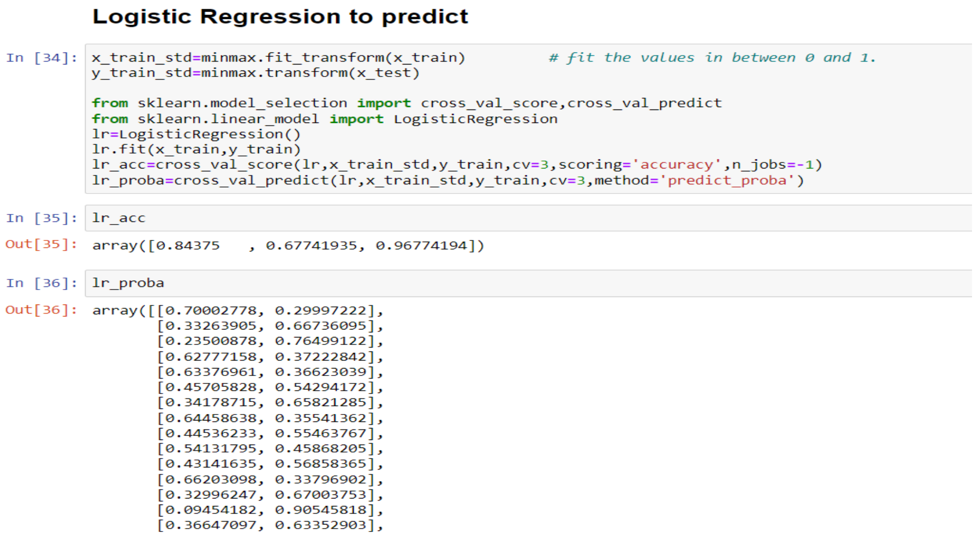
Step2: Divide the dataset

Step3: Assign the Logistic regression algorithm to a variable.

Step4: Fit the training dataset using Logistic regression algorithm of the required degree. For this project the Logistic Regression degree used is 1.

Step5: Predict the values for the testing dataset using a trained model.

Step6: Check the accuracy of the model.



Logistic Regression to predict

```
In [34]: x_train_std=minmax.fit_transform(x_train)        # fit the values in between 0 and 1.
         y_train_std=minmax.transform(x_test)

         from sklearn.model_selection import cross_val_score,cross_val_predict
         from sklearn.linear_model import LogisticRegression
         lr=LogisticRegression()
         lr.fit(x_train,y_train)
         lr_acc=cross_val_score(lr,x_train_std,y_train,cv=3,scoring='accuracy',n_jobs=-1)
         lr_proba=cross_val_predict(lr,x_train_std,y_train,cv=3,method='predict_proba')

In [35]: lr_acc

Out[35]: array([0.84375   , 0.67741935, 0.96774194])

In [36]: lr_proba

Out[36]: array([[0.70002778, 0.29997222],
                [0.33263905, 0.66736095],
                [0.23500878, 0.76499122],
                [0.62777158, 0.37222842],
                [0.63376961, 0.36623039],
                [0.45705828, 0.54294172],
                [0.34178715, 0.65821285],
                [0.64458638, 0.35541362],
                [0.44536233, 0.55463767],
                [0.54131795, 0.45868205],
                [0.43141635, 0.56858365],
                [0.66203098, 0.33796902],
                [0.32996247, 0.67003753],
                [0.09454182, 0.90545818],
                [0.36647097, 0.63352903],
```

Fig.5.12: Fitting model with Logistic regression

lr-Variable of Polynomial regression algorithm degree=-degree of the algorithm

.fit()-fit method of sklearn

. predict()-predict method for predicting the values

X_train,Y_train-Training data

X_test,Y_test-Testing data

5.5.3. Modelling with Decision Tree Algorithm:

Step1: Load the dataset

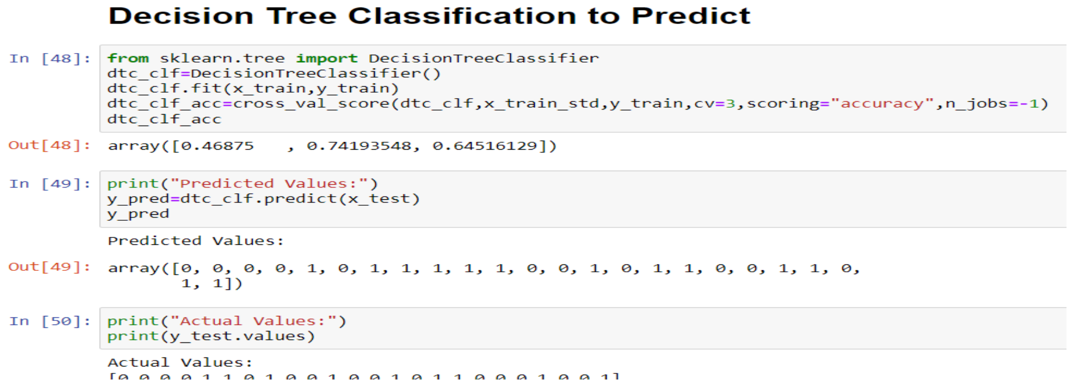
Step2: Divide the dataset

Step3: Assign the Decision Tree to a variable.

Step4: Fit the training dataset using Decision Tree algorithm.

Step5: Predict the values for the testing dataset using a trained model.

Step6: Check the accuracy of the model.



Decision Tree Classification to Predict

```
In [48]: from sklearn.tree import DecisionTreeClassifier
         dtc_clf=DecisionTreeClassifier()
         dtc_clf.fit(x_train,y_train)
         dtc_clf_acc=cross_val_score(dtc_clf,x_train_std,y_train,cv=3,scoring="accuracy",n_jobs=-1)
         dtc_clf_acc

Out[48]: array([0.46875   , 0.74193548, 0.64516129])

In [49]: print("Predicted Values:")
         y_pred=dtc_clf.predict(x_test)
         y_pred

         Predicted Values:

Out[49]: array([0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0,
                1, 1])

In [50]: print("Actual Values:")
         print(y_test.values)

         Actual Values:
         [0 0 0 0 1 1 0 1 0 0 1 0 0 1 0 1 1 0 0 0 1 0 0 1]
```

Fig.5.13 : Fitting model with Decision Tree

dtc-Variable of Decision Tree algorithm

.fit()-fit method of sklearn

.predict()-predict method for predicting the values

X_train,Y_train-Training data

X_test,Y_test-Testing data

5.5.2. Modelling with SVC

Algorithm

Step1: Load the dataset

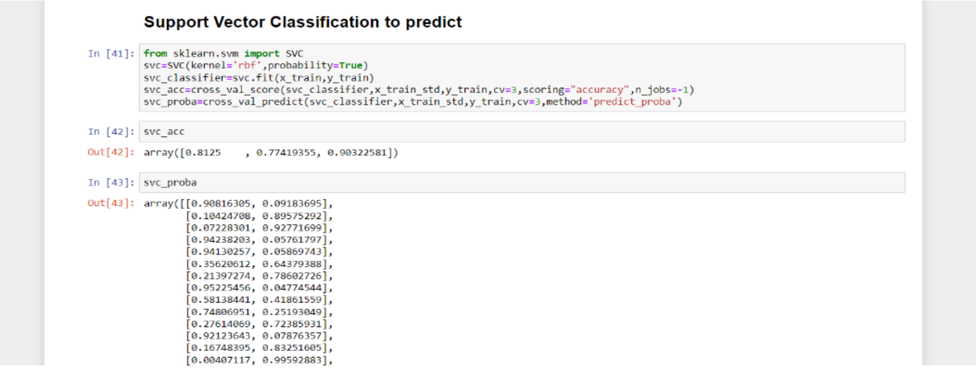
Step2: Divide the dataset

Step3: Assign the svc to a variable.

Step4: Fit the training dataset using svc algorithm.

Step5: Predict the values for the testing dataset using a trained model.

Step6: Check the accuracy of the model.



Support Vector Classification to predict

```
In [41]: from sklearn.svm import SVC
         svc=SVC(kernel='rbf',probability=True)
         svc_classifier=svc.fit(x_train,y_train)
         svc_acc=cross_val_score(svc_classifier,x_train_std,y_train,cv=3,scoring="accuracy",n_jobs=-1)
         svc_proba=cross_val_predict(svc_classifier,x_train_std,y_train,cv=3,method='predict_proba')

In [42]: svc_acc

Out[42]: array([0.8125    , 0.77419355, 0.90322581])

In [43]: svc_proba

Out[43]: array([[0.90816305, 0.09183695],
                [0.10424708, 0.89575292],
                [0.07228301, 0.92771699],
                [0.94238203, 0.05761797],
                [0.94130257, 0.05869743],
                [0.35620612, 0.64379388],
                [0.21397274, 0.78602726],
                [0.95225456, 0.04774544],
                [0.58138441, 0.41861559],
                [0.74806951, 0.25193049],
                [0.27614069, 0.72385931],
                [0.92123643, 0.07876357],
                [0.16748395, 0.83251605],
                [0.00407117, 0.99592883],
```

Fig.5.14: Fitting model with SVC

svc-Variable of SVC algorithm

.fit()-fit method of sklearn

.predict()-predict method for predicting the values

X_train,Y_train-Training data

X_test,Y_test-Testing data

5.5.2. Modelling with Random Forest Algorithm

Step1: Load the dataset

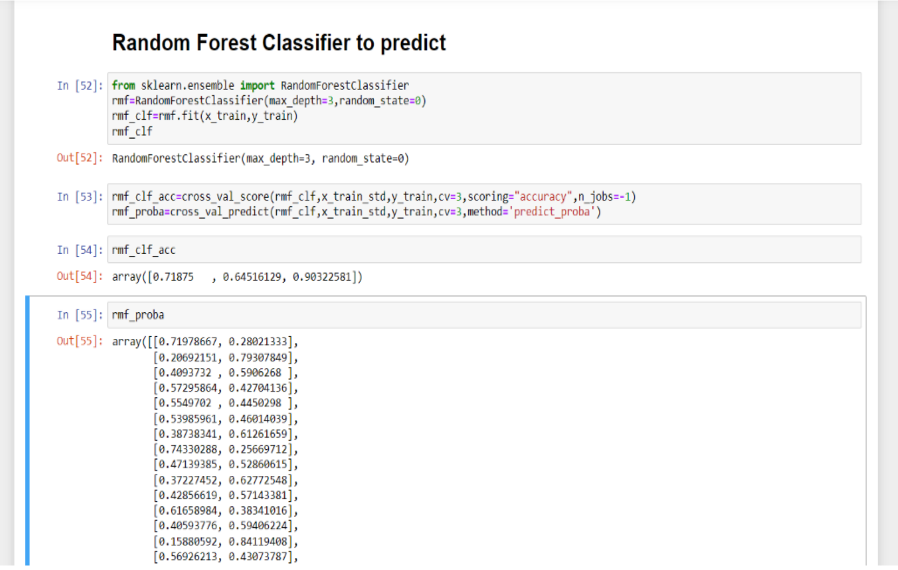
Step2: Divide the dataset

Step3: Assign the Random Forest Classifier to a variable.

Step4: Fit the training dataset using Random Forest Classifier algorithm.

Step5: Predict the values for the testing dataset using a trained model.

Step6: Check the accuracy of the model.



Random Forest Classifier to predict

```
In [52]: from sklearn.ensemble import RandomForestClassifier
         rmf=RandomForestClassifier(max_depth=3,random_state=0)
         rmf_clf=rmf.fit(x_train,y_train)
         rmf_clf

Out[52]: RandomForestClassifier(max_depth=3, random_state=0)

In [53]: rmf_clf_acc=cross_val_score(rmf_clf,x_train_std,y_train,cv=3,scoring="accuracy",n_jobs=-1)
         rmf_proba=cross_val_predict(rmf_clf,x_train_std,y_train,cv=3,method='predict_proba')

In [54]: rmf_clf_acc

Out[54]: array([0.71875   , 0.64516129, 0.90322581])

In [55]: rmf_proba

Out[55]: array([[0.71978667, 0.28021333],
                [0.20692151, 0.79307849],
                [0.4093732 , 0.5906268 ],
                [0.57295864, 0.42704136],
                [0.5549702 , 0.4450298 ],
                [0.53985961, 0.46014039],
                [0.38738341, 0.61261659],
                [0.74330288, 0.25669712],
                [0.47139385, 0.52860615],
                [0.37227452, 0.62772548],
                [0.42856619, 0.57143381],
                [0.61658984, 0.38341016],
                [0.40593776, 0.59406224],
                [0.15880592, 0.84119408],
                [0.56926213, 0.43073787],
```

Fig.5.15: Fitting model with Random Forest Classifier

Rmf_clf-Variable of algorithm

.fit()-fit method of sklearn

.predict()-predict method for predicting the values

X_train,Y_train-Training data

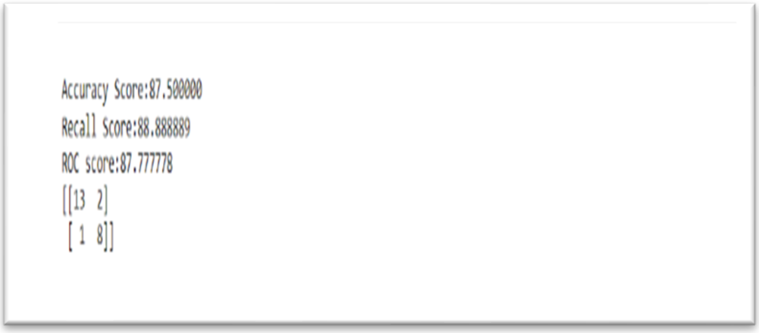
X_test,Y_test-Testing data

5.6. Accuracy of models

5.6.1. Prediction of outputs

After training all the models the output values are viewed according to our requirements. The outputs of each algorithm will be as follows:

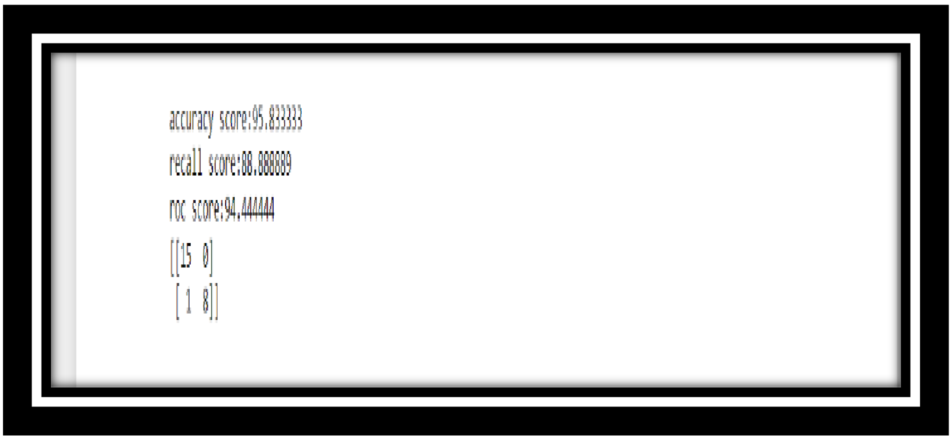
Accuracy of KNN



```
Accuracy Score:87.500000
Recall Score:88.888889
ROC score:87.777778
[[13  2]
 [ 1  8]]
```

Fig.5.16: Output values of KNN Classifier

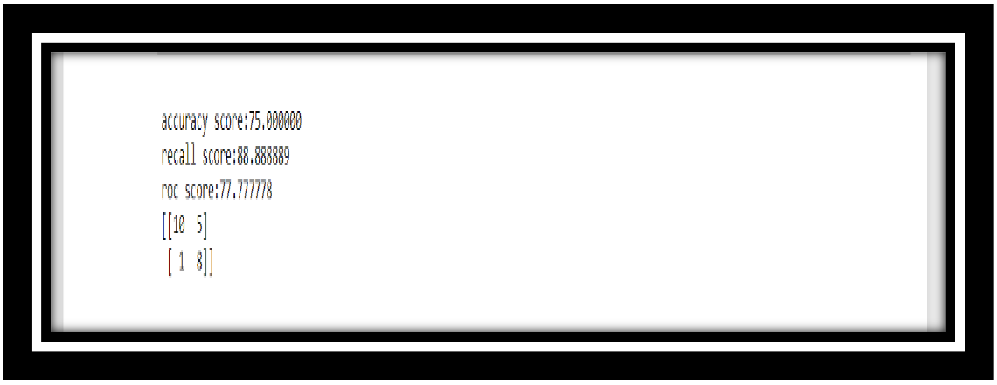
Accuracy of Logistic Regression



```
accuracy score:95.833333
recall score:88.888889
roc score:94.444444
[[15  0]
 [ 1  8]]
```

Fig.5.17: Output values of Logistic Regression

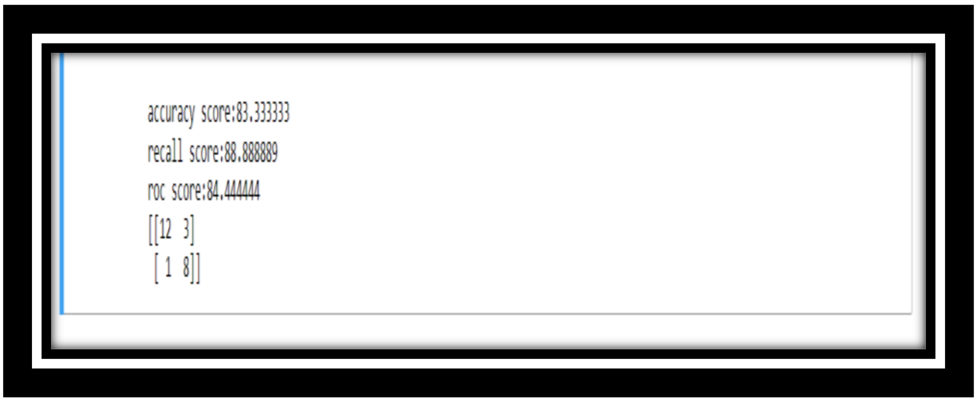
Accuracy of Decision Tree



```
accuracy score:75.000000
recall score:88.888889
roc score:77.777778
[[10  5]
 [ 1  8]]
```

Fig.5.18: Output values of Decision Tree

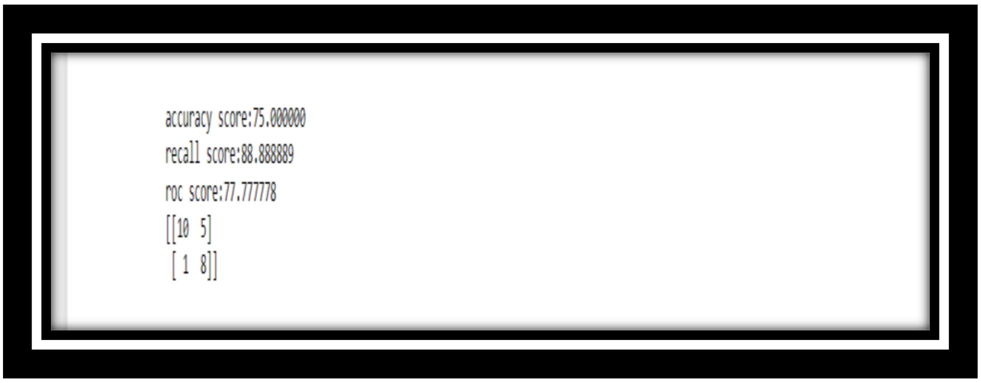
Accuracy of SVC



```
accuracy score:83.333333
recall score:88.888889
roc score:84.444444
[[12  3]
 [ 1  8]]
```

Fig.5.19: Output values of SVC

Accuracy of Random Forest



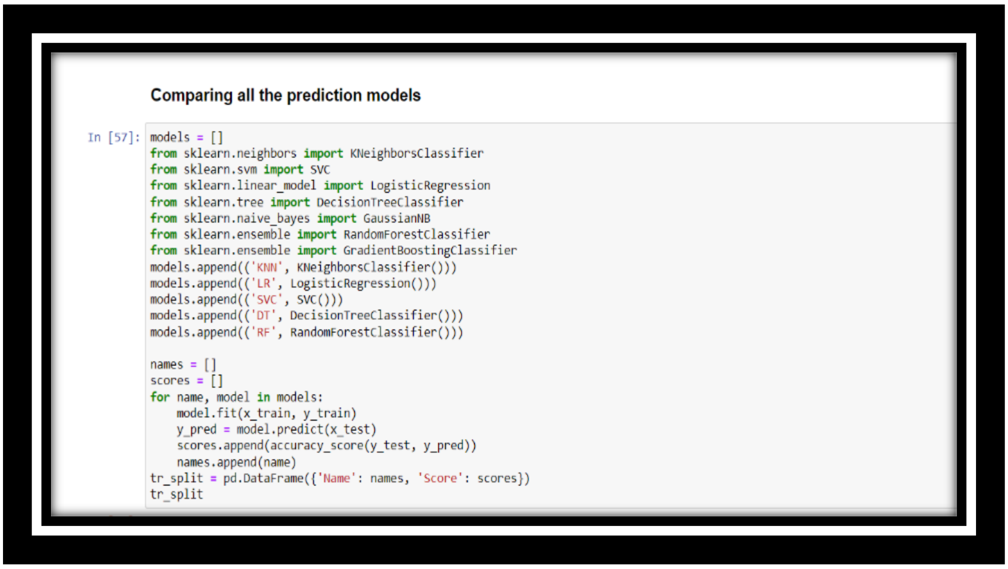
```
accuracy score:75.000000
recall score:88.888889
roc score:77.777778
[[10  5]
 [ 1  8]]
```

Fig.5.20: Output values of Random Forest

5.6.2. Comparing all the Prediction models

Here the accuracy of all models is predicted and the models are compared for finding best accuracy.

The model with best accuracy will be used for prediction.



Comparing all the prediction models

```
In [57]: models = []
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import GradientBoostingClassifier
models.append(('KNN', KNeighborsClassifier()))
models.append(('LR', LogisticRegression()))
models.append(('SVC', SVC()))
models.append(('DT', DecisionTreeClassifier()))
models.append(('RF', RandomForestClassifier()))

names = []
scores = []
for name, model in models:
    model.fit(x_train, y_train)
    y_pred = model.predict(x_test)
    scores.append(accuracy_score(y_test, y_pred))
    names.append(name)
tr_split = pd.DataFrame({'Name': names, 'Score': scores})
tr_split
```

Fig.5.21: Comparing all the prediction model

In this we have compared all the models and outputs are predicted. And these outputs are represented in the form of graph, and is as follows:

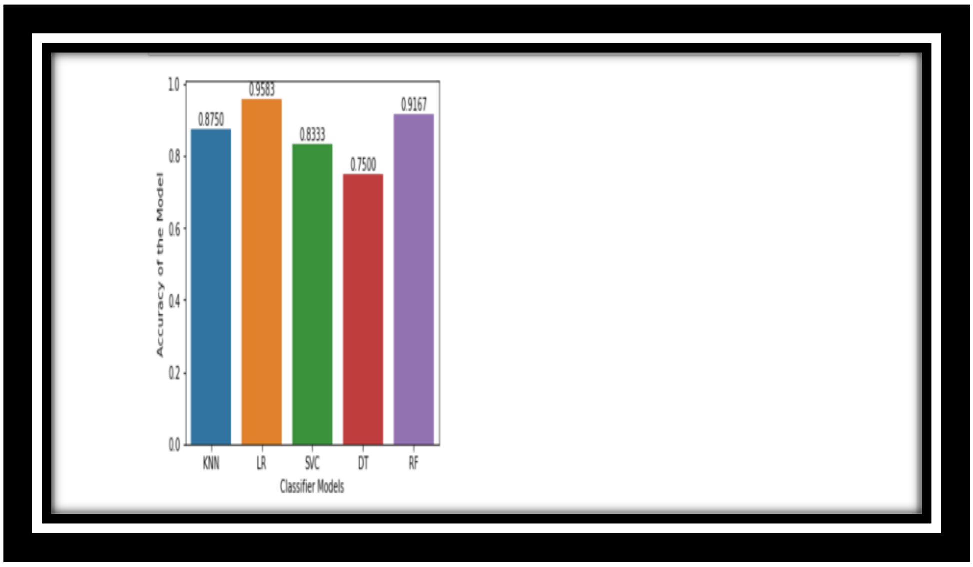
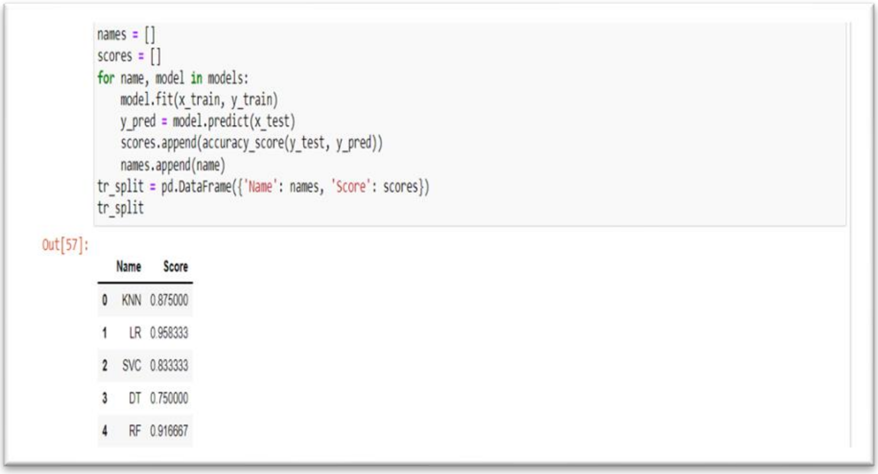


Fig.5.22: 5 Graphical representation of models



```
names = []
scores = []
for name, model in models:
    model.fit(x_train, y_train)
    y_pred = model.predict(x_test)
    scores.append(accuracy_score(y_test, y_pred))
    names.append(name)
tr_split = pd.DataFrame({'Name': names, 'Score': scores})
tr_split
```

```
Out[57]:
   Name     Score
0   KNN  0.875000
1    LR  0.958333
2   SVC  0.833333
3    DT  0.750000
4    RF  0.916667
```

5.7. Dimensionality Reduction

Dataset consists of 20 different attributes where 19 are independent attributes in which Dimensionality reduction is the process of removing few attributes from those which improve the performance of the model.

CHAPTER-6

TESTING

6.1 INTRODUCTION

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say. Testing is a process of executing a program with the intent of finding an error.

- A successful test is one that unconverse an as yet undiscovered error
- A good test case is one that has a high probability of finding error, if it exists.

The first approach is what is known as Black box testing and the second approach is White box testing. We apply white box testing techniques to ascertain the functionalities top-down and then we use black box testing techniques to demonstrate that everything runs as expected.

6.2 Black-Box Testing

This technique of testing is done without any knowledge of the interior workings of the application. The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black-box test, a tester will interact with the system’s user interface by providing inputs and examining the outputs without knowing how and where the inputs are worked upon. Code access is not required. Clearly separates user’s perspectives from the developer’s perspective through visibly defined roles.

6.3 White-Box Testing

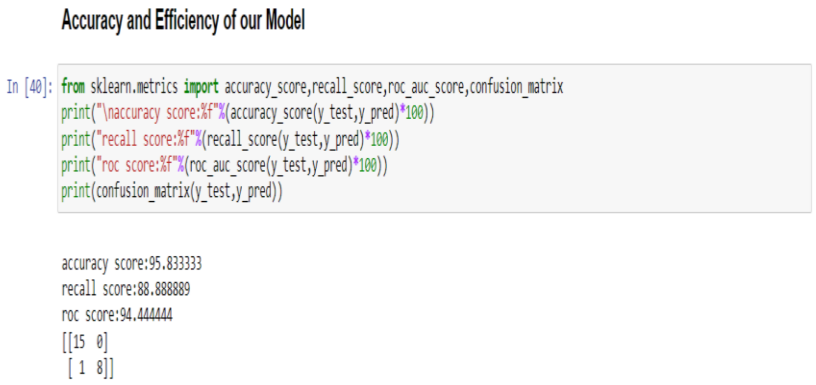
White-box testing is the detailed investigation of internal logic and structure of the code. It is also called “glass testing” or “open-box testing”. In order to perform white box testing on an application, a tester needs to know the internal workings of the code.

The tester needs to look inside the source code and find out which part of the code is working inappropriately. In this, the test cases are generated on the logic of each module. It has been uses to generate the test cases in the following cases:

- ➢ Guarantee that all independent modules have been executed.
- ➢ Execute all logical decisions and loops.
- ➢ Execute through proper plots and curves.

6.4. Performance Evaluation

Score method: It is a kind of method used to evaluate the performance of the model. Performance evaluation is made for this project using Score method of the sklearn library of Python. The score method is applied for all three algorithms as follows:

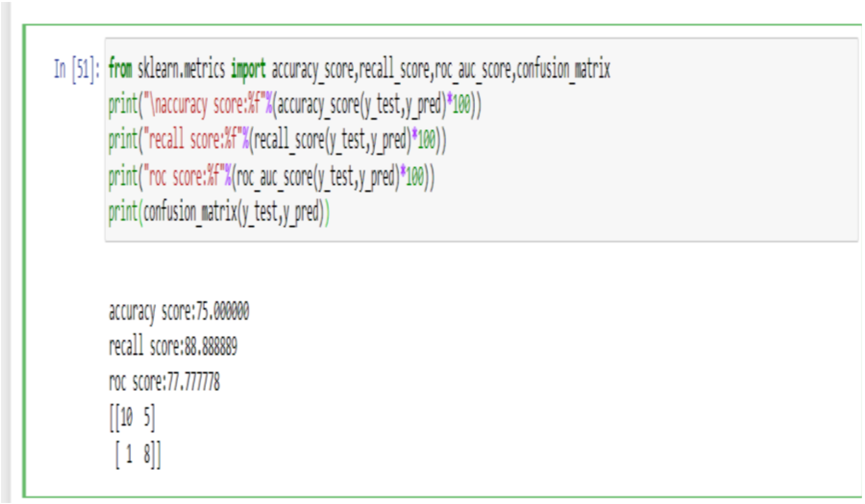


Accuracy and Efficiency of our Model

```
In [40]: from sklearn.metrics import accuracy_score,recall_score,roc_auc_score,confusion_matrix
         print("\naccuracy score:%f"%(accuracy_score(y_test,y_pred)*100))
         print("recall score:%f"%(recall_score(y_test,y_pred)*100))
         print("roc score:%f"%(roc_auc_score(y_test,y_pred)*100))
         print(confusion_matrix(y_test,y_pred))
```

```
accuracy score:95.833333
recall score:88.888889
roc score:94.444444
[[15  0]
 [ 1  8]]
```

Fig 6.1: Accuracy of Logistic Regression



```
In [51]: from sklearn.metrics import accuracy_score,recall_score,roc_auc_score,confusion_matrix
         print("\naccuracy score:%f"%(accuracy_score(y_test,y_pred)*100))
         print("recall score:%f"%(recall_score(y_test,y_pred)*100))
         print("roc score:%f"%(roc_auc_score(y_test,y_pred)*100))
         print(confusion_matrix(y_test,y_pred))
```

```
accuracy score:75.000000
recall score:88.888889
roc score:77.777778
[[10  5]
 [ 1  8]]
```

Fig 6.2: Accuracy of decision tree

Accuracy of the models of the algorithms are as follows

KNN Classifier:87.50%

Logistic Regression:95.8%

Decision tree:75.0%

Support vector classifier:83.3%

Random Forest:91.6%

CONCLUSION

The machine learning system ignited by data cleaning and processing, replacing or removing the null values, model building and evaluation. At the end the flood prediction model has given different accuracy results from four different models. From the above results and analysis, the best algorithm for flood prediction is Logistic Regression with (95%). In this , the simple machine learning algorithm to predict the accuracy of the flood occurrence is implemented. The desired algorithm shows the results of occurrence of flood in the upcoming year. When compared with the other algorithms, the Logistic Regression algorithm gives more accurate results and provide high performance accuracy and easy to understand. As the compared results shows that the Logistic Regression gives more accuracy compared to other simple machine learning algorithm. It can provide historical dataset with more mutable and adaptable form.

REFERENCES

[1]. Anil Kumar Lohani et al. “Improving real time flood forecasting using fuzzy inference system”.

[2]. Deepak Jayant Dattawadkar, Sunita Babanrao Vani “A Review on Fuzzy Based Flood Warning Expert System using IoT and LoRa Technology”.

[3]. A.Kolvankar “International Journal of Trend in Research and Development”, Volume 6(6), ISSN: 2394-9333.

[4]. Abhishek, K., Kumar, A., Ranjan, R., & Kumar, S. (2012). A Rainfall Prediction Model using Artificial Neural Network. IEEE Control and System Graduate Research Colloquium, 1- 5.

[5]. Agnihotri, G., & Panda, J. (2014). Comparison of Rainfall from Ordinary and Automatic Rain Gauges in Karnataka. Mausam, 65(4), 1-8. Ahn, J. (2017). Analysis of a neural network model for building energy hybrid controls for inbetween season. Architecture of Complexity, (pp. 1-5).