

HYBRID PREDICTION MODEL FOR RAINFALL FORECASTING

**Sathyabama Institute Of Science And Technology
(Deemed To Be University)**

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

By

**V.SRIKAR (37110853)
Y. GOPI SANDEEP (37110873)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU**

March 2021



SATHYABAMA

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

Accredited with Grade “A” by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI– 600119

www.sathyabamauniversity.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **V. SRIKAR (37110853), Y. GOPISANDEEP (37110873)** who carried out the project entitled **“HYBRID PREDICTION MODEL FOR RAINFALL FORECASTING”** under my supervision from November 2020 to April 2021.

Internal Guide

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

Head of the Department

Dr. Lakshmanan M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **V. SRIKAR & Y. GOPI SANDEEP** hereby declare that the Project Report entitled HYBRID PREDICTION MODEL FOR RAINFALL FORECASTING done by me under the guidance of **Dr. JANCY.S, M.E, Ph.D.**, an_(External) at_(Company name and address) is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

**V.SRIKAR
Y. GOPI SANDEEP**

ACKNOWLEDGEMENT

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

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D., Dean**, School of Computing , **Dr.S.Vigneshwari M.E., Ph.D., and Dr.L.Lakshmanan M.E., Ph.D.**, Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. JANCY.S,M.E,Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

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

ABSTRACT

Rainfall has been one of the most volatile aspects to analyze. Over the past few decades several machine learning algorithms have been applied to predict the rainfalls beforehand. In this study, several hybrid forecasting models are proposed that are combinations two feature selection techniques, Gradient boosting and Random forest with various machine learning techniques, viz Support Vector Machine (SVM), adaboost, Neural Network (NN) and K-Nearest Neighbour (KNN). These model have been applied to the past 11 years (2007 2017) weather data to predict rainfall in town of carry, North caroliana. The performances of these algorithms have been computed on different metrics F-score, precision, recall, accuracy. Adaptive boosting algorithm comes up with F-score of 0.9726 on testing data whereas K-nearest, SVM and neural network shows Fscore of 0.8754, 0.8045 and 0.7946 respectively. The empirical results shown by the Adaboost algorithm is superior to remaining ones. Selecting the relevant attributes is a crucial factor as the performance of algorithm depends largely on relevant features, empirical results shows the supremacy of certain features among others features for better results.

TABLE OF CONTENTS

CHAPTER NO NO	TITLE	PAGE
	ABSTRACT	v
	TABLE OF CONTENT	vi
	LIST OF FIGURES	viii
1	INTRODUCTION	
	1.1 INRODUCTION	1-2
	1.2 INTRODUCTION TO PYTHON	2-3
	1.3 INTRODUCTION TO MACHINE LEARNIN	3-3
2	LITERATURE SURVEY	4-6
3	METHODOLOGY	
	3.1 SYSTEM ANALYSIS	7
	3.2 HARDWARE SOFTWARE SPECIFICATION	7
	3.2.1 HARDWARE REQUIREMENTS	7
	3.2.2 SOFTWARE REQUIREMENTS	8
	3.3 LEARNING ALGORITHMS	
	3.3.1 SUPERVISED LEARNING	8
	3.3.2 UNSUPERVISED LEARNING	9
	3.3.3 SEMISUPERVISED LEARNING	9
	3.4 RANDOM FOREST	
	9-13	
	3.5 CODING AND TESTING	13
	3.5.1 CODING	13
	3.5.2 CODING STANDARDS	13

	3.6 NAMING CONVENTIONS	14
	3.6.1 VALUE CONVENTION	14
	3.7 TEST PROCEDURE	15
4	RESULTS AND DISCUSSIONS, PERFORMANCE ANALYSIS	
	4.1 DESIGN AND IMPLEMENTATION	
	CONSTRAINTS	18
	4.2 NONFUNCTIONAL REQUIREMENTS	
	18-19	
	4.3 SEQUENTIAL DIAGRAM	20
	4.4 USECASE DIAGRAM	
	20-21	
	4.5 ACTIVITY DIAGRAM	
		21-2
		2
	4.6 COLLABORATION DIAGRAM	
		22-2
		3
	4.7 MODULES	
	23-24	
	4.7.1 DATA COLLECTION AND PROCESSING	23
	4.7.2 RANDOM FOREST MODEL GENERATION	24
	4.7.3 PREDICTION, RESULT PRESENTATION	24
5	RESULTS AND DISCUSSIONS	
	5.1 RESLUT	25
	5.2 CONCLUSION	25
	5.3 REFERENCES	
	25-26	

APPENDIX

A.SCREEN SHOTS

27-28

B. SOURCE CODE

29-36

LIST OF FIGURES

FIGURE NO NO	FIGURE NAME	PAGE
1	RANDOM FOREST CASE STUDY	9
2	ARCHITECTURE DIAGRAM	19
3	SEQUENTIAL DIAGRAM	
20		
4	USECASE DIAGRAM	
21		
5	ACTIVITY DIAGRAM	
22		
6	COLLABORATION DIAGRAM	
23		

CHAPTER - 1

INTRODUCTION

1.1 INTRODUCTION

Predicting rainfall is a challenging task and especially when environment factors are in variant state due to global warming. It is multiplex phenomenon challenging problem and require expertise in multiple domain. Aftermath of this invariance cause hazaradious problems in various sectors, agriculture sector gets effected the most. Some related problems is mentioned in. Precipitation expectation is an imperative part of climate forecasting. Exact and opportune precipitation expectation is significant for the arranging and administration of water assets, surge admonitions, development exercises and flight tasks and so on. This paper aims to predict rainfall few days prior and to find the cardinal attributes that play key role in rainfall. With the advancement of technology and science, specifically, the intelligent computing technology in the previous couple of decades, many rising strategies, for example, Support vector machines (SVMs) and Artificial Neural Network (ANNs) have been broadly utilized in the rainfall prediction.

Lets have look on mathematical part. Let m is the total number of data samples having output y for each training samples $(x_1, y_1), \dots, (x_m, y_m)$, where x_i belongs to some domain X , the output labels $y_i \in -1, +1$. Initialize all the weights $Q(i) = 1/m$, for each $i = 1, \dots, m$, where m is the number of training samples. A distribution U_t is calculated over m training data for, it is given by weight α , this alpha provide weights to the learner according to its performance, i.e to minimize the error relative to U_t . A is calculated as: $\alpha(t) = 1/2 * \ln((1 - V_t)/V_t)$ (6) Once all the weights gets calculated and the each learners trains to minimize the error, the combined hypothesis A computes the sign of weighted weak learner combination $R(x) = \text{sign}(\sum_{i=1}^T \alpha_i * h_i(x))$ (7) Pseudo code for this algorithm is given below:

- 1: Input: data x_i, y_i , learning rate ϵ , iterations T .
- 2: Initialize predictions $H = 0$ and selected feature set ω
- 3: for $t = 1$ to T do
- 4: Learn h_t using greedy CART to minimize the impurity

function in Eq. (14).

5: Update $H = H + _ht$.

6: For each feature f used in ht , set $\phi f = 0$ and $\omega = \omega \cup f$.

7: end for

8: Return H and ω

1.2 INTRODUCTION TO PYTHON

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

It is used for:

- web development (server-side),
- software development,
- mathematics,
- System scripting.

What can Python do?

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

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.

- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-orientated way or a functional way.

1.3 INTRODUCTION TO MACHINE LEARNING:

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop a conventional algorithm for effectively performing the task. Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a field of study within machine learning, and focuses on exploratory data analysis through learning. In its application across business problems, machine learning is also referred to as predictive analytics.

CHAPTER - 2

LITERATURE SURVEY

Title 1: A Hybrid Climate Model for Rainfall Forecasting based on Combination of Self-Organizing Map and Analog Method

Merits: The larger cloud providers, such as Amazon Web Services, Microsoft and Rack space, already have good practices and technology in place to lower the risk.

The cloud services are very tempting for DDOS attackers. It's safe to assume that, as more cloud services come into use, DDOS attacks on them become more commonplace.

Demerits: DST and NIEF we need to establish a complex architecture that smaller cloud providers may not have the resources to mount a suitable defense

En resources are not in use. Therefore it is no wonder that academic research and industry are moving towards cloud computing.

Title 2: An Approach for Rainfall Prediction using Soft Computing

Merits: The hybrid SMC technology obtains a lot of advantages in the field of aircraft applications, compared with current composite materials and manufacturing processes, especially for interior components.

main benefits of the SMC technology are cost-efficient processes and materials, the opportunity for parts with complex 3D shapes, a high reproducibility, tailored mechanical properties and short lead times

Demerits: Most of the manufacturing processes for the production of fiber composite aircraft parts are expensive, time-consuming and not automated.

The components mass varies in a comparatively wide range and the surface quality is often not sufficient

Title 3: Application of Artificial Neural Networks to Rainfall Forecasting in the Geum River Basin,

Merits: Flood warning is an efficient way to reduce the flood damage.

Uncertainties of flood forecasts using the Bayesian networks (BN) and to estimate the flood peak in an ensemble flood forecasting.

Demerits:

There is no considerable decrease in accuracy by deleting the Multi-scale Kain-Fritsch scheme, thus it can be concluded that is the least accurate cumulus scheme.

NCEP data are not available for some historical storms. And During the above-mentioned period, a small number of actual flood events occurred in the study area, since the basin is located in a semi-arid region.

Title 4: Development and comparison of Hybrid Rainfall Prediction Model

Merits: In order to improve the effectiveness of the study, method proposed should combine with other feature and extent the number of image database to be tested

Thesis provides with a basic informatory review on the applied fields of data mining which is varied into manufacturing, telecommunication, education, fraud detecting and marketing sector

Demerits:

Image mining is the additional pattern which is quite not clearly visible in the image, association of image data and extraction of hidden data

Also the innovation of image patterns that are noteworthy in a given collection of images.

Title 5: Improved Rainfall Prediction Using CombinedPre-Processing Methods and Feed-Forward Neural Networks

Merits: We identified 180 original and influential articles where the performance and accuracy of at least two machine learning models were compared.

The prediction models were classified into two categories according to lead time, and further divided into categories of hybrid and single methods

Demerits: The feed-forward neural network is a class of ANN, whereby the network's connections are not in cyclical form

The data is scarce or does not cover varieties of the task, their learning falls short, and hence, they cannot perform well when they are put into work

Title 6: Improved Wavelet Neural Network Modeling of Daily Rainfall Forecast

Merits: It is found that optimum combinations of temperature and rainfall are needed to increase the net revenue from agricultural production. The changing climatic scenario and its

impact on various sectors of the economy have emerged as one of the greatest challenges before the scientists and policy makers all over the world in twenty-first century

Demerits: Many developing countries due to market imperfections, land values cannot be obtained in a synchronized way.

The other possible climate change scenarios, there may be negative changes in the net revenue

Title 7: Machine Learning Techniques For Rainfall Prediction: A Review

Merits: A main limiting factor in building accurate models is the quantity and quality of the record of flooding used to train. Demonstrates one way that crowd sourced data, despite its limitations, can provide useful information to flood prediction models.

Demerits: Flood prediction is an essential part of a coastal city's capacity to adapt to and mitigate this growing problem.

Random Forest performed better than Poisson regression at predicting the number of flood reports and had a lower false negative rate.

Title 8: Rainfall Prediction using Data Mining Techniques: A Systematic Literature Review

Merits: Multi-resource demand prediction has emerged as a tough challenge for scientific applications which should be resolved using improved prediction techniques.

To enhance scheduling efficiency for scientific applications of different size, intelligent prediction models can be used. This would help in predicting resources based on time and cost parameters.

Demerits: Existing resource prediction and scheduling techniques are very low accuracy. Unexpected demand for huge resources in future may instigate allocation failure or system hang problem.

CHAPTER 3

METHODOLOGY

3.1 SYSTEM ANALYSIS

EXISTING SYSTEM:

The dataset is stored and processed in structured database like MySQL, Oracle etc. The time consumed to process volumes of data to pre-process and classify were huge. The model is utilized to generate the prediction of future rainfall based on the dataset. MySQL is a flexible computing framework suitable for small applications. Therefore MySQL was used as simple computing and storage platform.

Problem Statement:

Machine Learning process was relatively slow for huge dataset.

- More memory consumption for a given task.

PROPOSED SYSTEM

To overcome the fallback in the existing system we propose a machine learning based system to increase the efficiency and accuracy. To handle voluminous data we are using Hadoop to store and retrieve data from the distributed hadoop file system (hdfs). Hadoop allows the user to load data into a cluster. Random forest algorithm is to be implemented for forecasting rainfall.

3.2 HARDWARE AND SOFTWARE SPECIFICATION

3.2.1 HARDWARE REQUIREMENTS

- Hard disk : 500 GB and above.
- Processor : i3 and above.
- Ram : 4GB and above.

3.2.2 SOFTWARE REQUIREMENTS

- Operating System : Windows 7 and above (64-bit).
- Python : 3.6
- Java : Jdk1.8

3.3 TYPES OF LEARNING ALGORITHMS:

The types of machine learning algorithms differ in their approach, the type of data they input and output, and the type of task or problem that they are intended to solve.

3.3.1 SUPERVISED LEARNING:

Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs. The data is known as training data, and consists of a set of training examples. Each training example has one or more inputs and the desired output, also known as a supervisory signal. In the mathematical model, each training example is represented by an array or vector, sometimes called a feature vector, and the training data is represented by a matrix. Through iterative optimization of an objective function, supervised learning algorithms learn a function that can be used to predict the output associated with new inputs. An optimal function will allow the algorithm to correctly determine the output for inputs that were not a part of the training data. An algorithm that improves the accuracy of its outputs or predictions over time is said to have learned to perform that task.

Supervised learning algorithms include classification and regression. Classification algorithms are used when the outputs are restricted to a limited set of values, and regression algorithms are used when the outputs may have any numerical value within a range. Similarity learning is an area of supervised machine learning closely related to regression and classification, but the goal is to learn from examples using a similarity function that measures how similar or related two objects are. It has

applications in ranking, recommendation systems, visual identity tracking, face verification, and speaker verification.

3.3.2 UNSUPERVISED LEARNING:

Unsupervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms, therefore, learn from test data that has not been labeled, classified or categorized. Instead of responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on the presence or absence of such commonalities in each new piece of data. A central application of unsupervised learning is in the field of density estimation in statistics, though unsupervised learning encompasses other domains involving summarizing and explaining data features.

3.3.3 SEMI SUPERVISED LEARNING:

Semi-supervised learning falls between unsupervised learning (without any labeled training data) and supervised learning (with completely labeled training data). Many machine-learning researchers have found that unlabeled data, when used in conjunction with a small amount of labeled data, can produce a considerable improvement in learning accuracy.

3.4 INTRODUCTION TO RANDOM FOREST :

With increase in computational power, we can now choose algorithms which perform very intensive calculations. One such algorithm is “Random Forest”, which we will discuss in this article. While the algorithm is very popular in various competitions (e.g. like the ones running on Kaggle), the end output of the model is like a black box and hence should be used judiciously.

Before going any further, here is an example on the importance of choosing the best algorithm.

RANDOM FOREST CASE STUDY

Following is a distribution of Annual income Gini Coefficients across different countries:

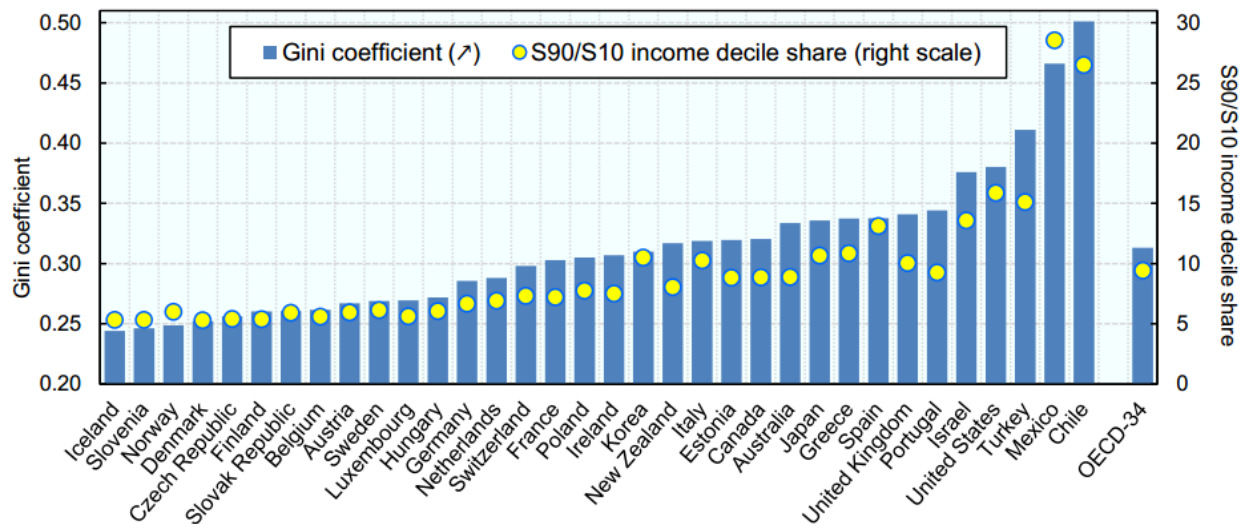


FIG 3.4 : RANDOM FOREST CASE STUDY

Mexico has the second highest Gini coefficient and hence has a very high segregation in annual income of rich and poor. Our task is to come up with an accurate predictive algorithm to estimate annual income bracket of each individual in Mexico. The brackets of income are as follows:

1. below \$40,000
2. \$40,000 – 150,000
3. More than \$150,000

Following are the information available for each individual:

1. Age, 2. Gender, 3. Highest educational qualification, 4. Working in Industry, 5. Residence in Metro/Non-metro

We need to come up with an algorithm to give an accurate prediction for an individual who has following traits:

1. Age: 35 years, 2. Gender: Male, 3. Highest Educational Qualification: Diploma holder, 4. Industry: Manufacturing, 5. Residence: Metro

THE ALGORITHM OF RANDOM FOREST

Random forest is like bootstrapping algorithm with Decision tree (CART) model. Say, we have 1000 observation in the complete population with 10 variables. Random forest tries to build multiple CART model with different sample and different initial variables. For instance, it will take a random sample of 100 observation and 5 randomly chosen initial variables to build a CART model. It will repeat the process (say) 10 times and then make a final prediction on each observation. Final prediction is a function of each prediction. This final prediction can simply be the mean of each prediction.

BACK TO CASE STUDY

Disclaimer: The numbers in this article are illustrative

Mexico has a population of 118 MM. Say, the algorithm Random forest picks up 10k observation with only one variable (for simplicity) to build each CART model. In total, we are looking at 5 CART models being built with different variables. In a real life problem, you will have more number of population sample and different combinations of input variables.

Salary bands:

Band 1: Below \$40,000

Band 2: \$40,000 – 150,000

Band 3: More than \$150,000

Following are the outputs of the 5 different CART model.

CART 1: Variable Age

	Salary Band	1	2	3
Age	Below 18	90%	10%	0%
	19-27	85%	14%	1%
	28-40	70%	23%	7%
	40-55	60%	35%	5%
	More than 55	70%	25%	5%

CART 2: Variable Gender

	Salary Band	1	2	3
Gender	Male	70%	27%	3%
	Female	75%	24%	1%

CART 3: Variable Education

	Salary Band	1	2	3
Education	<=High School	85%	10%	5%
	Diploma	80%	14%	6%
	Bachelors	77%	23%	0%
	Master	62%	35%	3%

CART 4: Variable Residence

	Salary Band	1	2	3
Residence	Metro	70%	20%	10%
	Non-Metro	65%	20%	15%

CART 5: Variable Industry

	Salary Band	1	2	3
Industry	Finance	65%	30%	5%
	Manufacturing	60%	35%	5%
	Others	75%	20%	5%

Using these 5 CART models, we need to come up with single set of probability to belong to each of the salary classes. For simplicity, we will just take a mean of probabilities in this case study. Other than simple mean, we also consider vote method to come up with the final prediction. To come up with the final prediction let's locate the following profile in each CART model:

1. Age: 35 years, 2. Gender: Male, 3. Highest Educational Qualification: Diploma holder, 4. Industry: Manufacturing, 5. Residence: Metro

For each of these CART model, following is the distribution across salary bands:

CART	Band	1	2	3
Age	28-40	70%	23%	7%
Gender	Male	70%	27%	3%
Education	Diploma	80%	14%	6%
Industry	Manufacturing	60%	35%	5%
Residence	Metro	70%	20%	10%
Final probability		70%	24%	6%

The final probability is simply the average of the probability in the same salary bands in different CART models. As you can see from this analysis, that there is 70% chance of

this individual falling in class 1 (less than \$40,000) and around 24% chance of the individual falling in class 2.

3.5 CODING AND TESTING

3.5.1 CODING :

Once the design aspect of the system is finalized the system enters into the coding and testing phase. The coding phase brings the actual system into action by converting the design of the system into the code in a given programming language. Therefore, a good coding style has to be taken whenever changes are required it easily screwed into the system.

3.5.2 CODING STANDARDS:

Coding standards are guidelines to programming that focuses on the physical structure and appearance of the program. They make the code easier to read, understand and maintain. This phase of the system actually implements the blueprint developed during the design phase. The coding specification should be in such a way that any programmer must be able to understand the code and can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows:

- Program should be simple, clear and easy to understand.

- Naming conventions

- Value conventions

- Script and comment procedure

- Message box format

- Exception and error handling

3.6 NAMING CONVENTIONS

Naming conventions of classes, data member, member functions, procedures etc., should be self-descriptive. One should even get the meaning and scope of the variable by its name. conventions are adopted for easy understanding of the intended message by the user.

Class names are problem domain equivalence and begin with capital letter and have mixed cases.

Member function and data member name begins with a lowercase letter with each

subsequent letters of the new words in uppercase and the rest of letters in lowercase.

3.6.1 VALUE CONVENTIONS:

Value conventions ensure values for variable at any point of time. This involves the following:

Proper default values for the variables.

Proper validation of values in the field.

Proper documentation of flag values.

SCRIPT WRITING AND COMMENTING STANDARD:

Script writing is an art in which indentation is utmost important. Conditional and looping statements are to be properly aligned to facilitate easy understanding. Comments are included to minimize the number of surprises that could occur when going through the code.

MESSAGE BOX FORMAT:

When something has to be prompted to the user, he must be able to understand it properly. To achieve this, a specific format has been adopted in displaying messages to the user. They are as follows:

X – User has performed illegal operation.

! – Information to the user.

3.7 TEST PROCEDURE

3.7.1 SYSTEM TESTING

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example the design must not have any logic faults in the design

is detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walkthrough.

Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

Static analysis is used to investigate the structural properties of the Source code.

Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

3.7.2 TEST DATA AND OUTPUT:

UNIT TESTING

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were heavily employed for unit testing.

FUNCTIONAL TESTS

Functional test cases involved exercising the code with nominal input values for which the expected results are known, as well as boundary values and special values, such as logically related inputs, files of identical elements, and empty files.

Three types of tests in Functional test:

Performance Test

Stress Test

Structure Test

PERFORMANCE TEST

It determines the amount of execution time spent in various parts of the unit, program throughput, and response time and device utilization by the program unit.

STRESS TEST

Stress Test is those test designed to intentionally break the unit. A Great deal can be learned about the strength and limitations of a program by examining the

manner in which a programmer in which a program unit breaks.

STRUCTURED TEST

Structure Tests are concerned with exercising the internal logic of a program and traversing particular execution paths. The way in which White-Box test strategy was employed to ensure that the test cases could Guarantee that all independent paths within a module have been have been exercised at least once.

Exercise all logical decisions on their true or false sides.

Execute all loops at their boundaries and within their operational bounds.

Exercise internal data structures to assure their validity.

Checking attributes for their correctness.

Handling end of file condition, I/O errors, buffer problems and textual errors in output information

INTEGRATION TESTING

Integration testing is a systematic technique for construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules which makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules. Critical modules should be tested as early as possible. One approach is to wait until all the units have passed testing, and then combine them and then tested. This approach is evolved from unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A small set of modules are integrated together and tested, to which another module is added and tested in combination. And so on. The advantages of this approach are that, interface dispenses can be easily found and corrected.

TESTING TECHNIQUES / TESTING STRATEGIES

TESTING:

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet –undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system

works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system.

WHITE BOX TESTING

This testing is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

Basis path testing:

- Flow graph notation

- Cyclometric complexity

- Deriving test cases

- Graph matrices Control

BLACK BOX TESTING

In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised.

The steps involved in black box test case design are:

- Graph based testing methods

- Equivalence partitioning

- Boundary value analysis

- Comparison testing

CHAPTER 4

RESULTS AND DISCUSSIONS, PERFORMANCE ANALYSIS

4.1 DESIGN AND IMPLEMENTATION CONSTRAINTS:

Constraints in Analysis:

- ◆ Constraints as Informal Text
- ◆ Constraints as Operational Restrictions
- ◆ Constraints Integrated in Existing Model Concepts
- ◆ Constraints as a Separate Concept
- ◆ Constraints Implied by the Model Structure

Constraints in Design:

- ◆ Determination of the Involved Classes
- ◆ Determination of the Involved Objects
- ◆ Determination of the Involved Actions
- ◆ Determination of the Require Clauses
- ◆ Global actions and Constraint Realization

Constraints in Implementation:

A hierarchical structuring of relations may result in more classes and a more complicated structure to implement. Therefore it is advisable to transform the hierarchical relation structure to a simpler structure such as a classical flat one. It is rather straightforward to transform the developed hierarchical model into a bipartite, flat model, consisting of classes on the one hand and flat relations on the other. Flat relations are preferred at the design level for reasons of simplicity and implementation ease.

4.2 OTHER NONFUNCTIONAL REQUIREMENTS:

Performance Requirements:

The application at this side controls and communicates with the following three main general components.

embedded browser in charge of the navigation and accessing to the web service;

Server Tier: The server side contains the main parts of the functionality of the proposed architecture. The components at this tier are the following.

Safety Requirements:

1. The software may be safety-critical. If so, there are issues associated with its integrity level
2. The software may not be safety-critical although it forms part of a safety-critical system. For example, software may simply log transactions.
3. If a system must be of a high integrity level and if the software is shown to be of that integrity level, then the hardware must be at least of the same integrity level.
4. If a computer system is to run software of a high integrity level then that system should not at the same time accommodate software of a lower integrity level.
5. Systems with different requirements for safety levels must be separated.

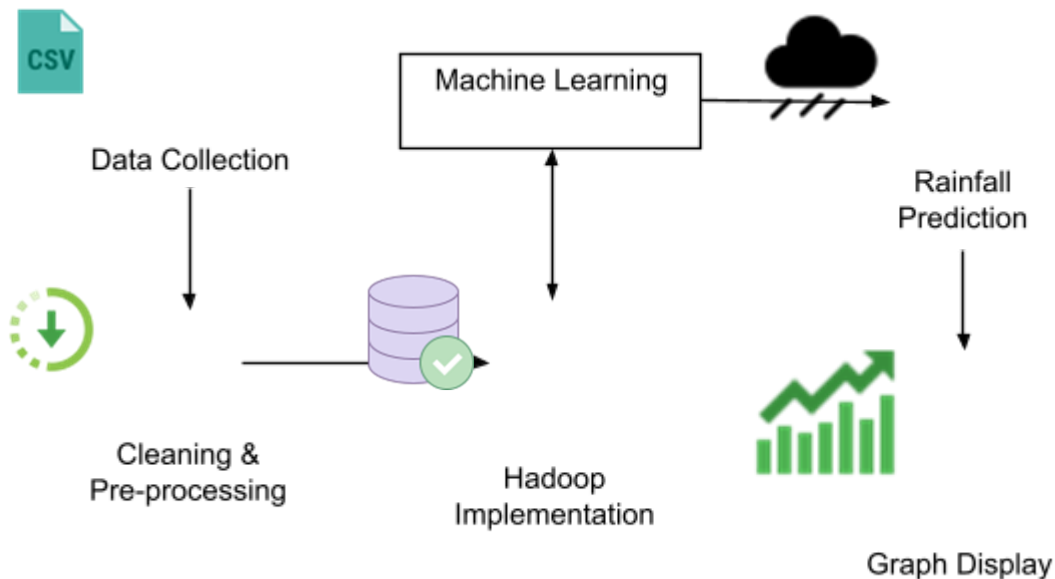


FIG: 4.2.2 ARCHITECTURE DIAGRAM

4.3 Sequence Diagram:

A Sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of Message Sequence diagrams are sometimes called event diagrams, event sceneries and timing diagram.

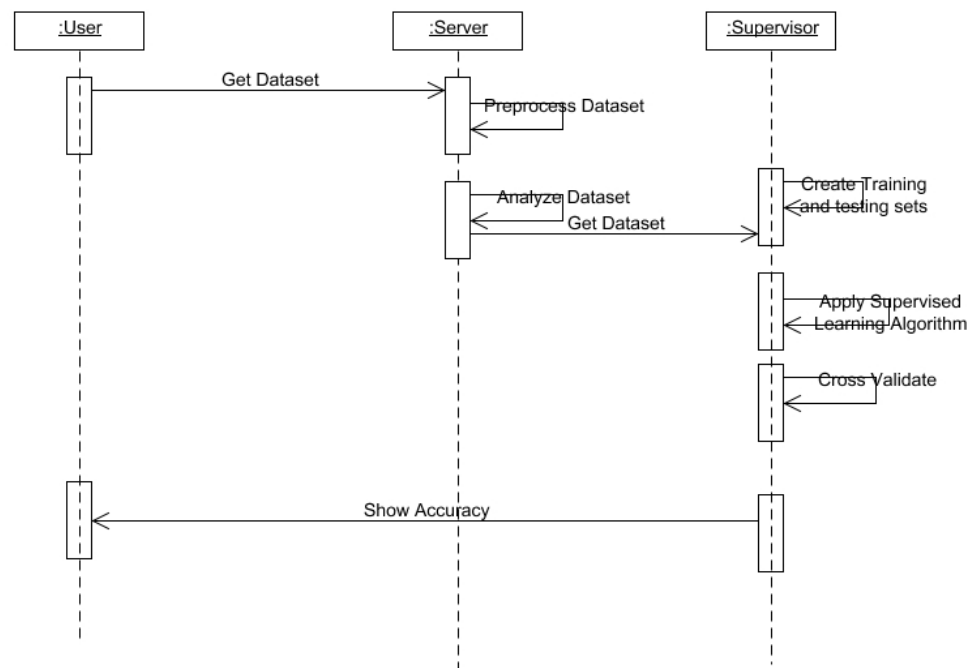


FIG: 4.3 SEQUENTIAL DIAGRAM

4.4 Use Case Diagram:

Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of software engineering. The standard is managed and was created by the Object Management Group. UML includes a set of graphic notation techniques to create visual models of software intensive systems.

Use case diagram consists of two parts:

Use case: A use case describes a sequence of actions that provided something of measurable value to an actor and is drawn as a horizontal ellipse.

Actor: An actor is a person, organization or external system that plays a role in one or more interaction with the system.

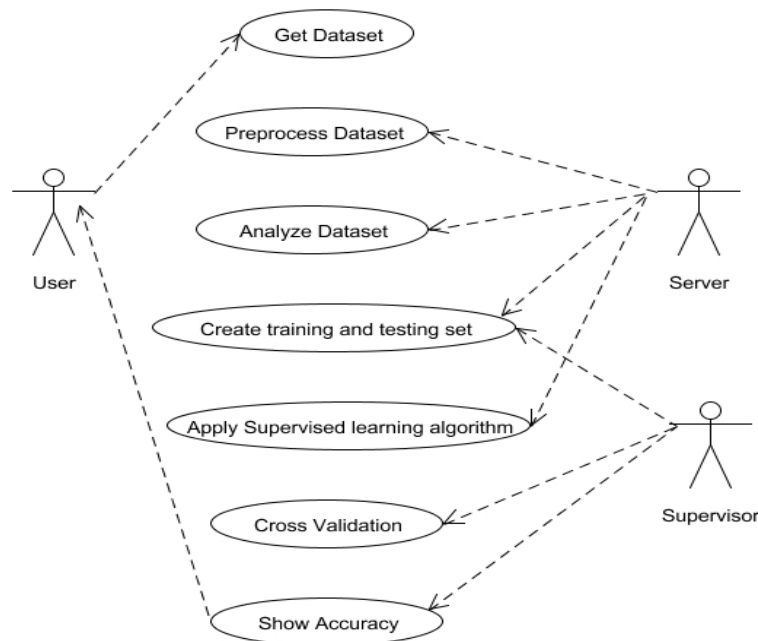


FIG: 4.4 USECASE DIAGRAM

4.5 Activity Diagram:

Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

The most important shape types:

- Rounded rectangles represent activities.
- Diamonds represent decisions.
- Bars represent the start or end of concurrent activities.
- A black circle represents the start of the workflow.
- An encircled circle represents the end of the workflow.

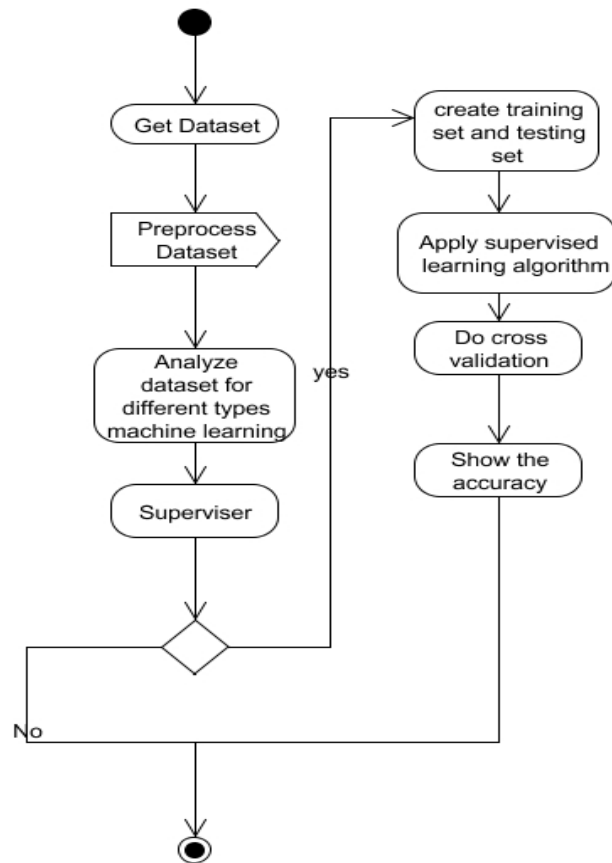


FIG: 4.5 ACTIVITY DIAGRAM

4.6 Collaboration Diagram:

UML Collaboration Diagrams illustrate the relationship and interaction between software objects. They require use cases, system operation contracts and domain model to already exist. The collaboration diagram illustrates messages being sent between classes and objects.

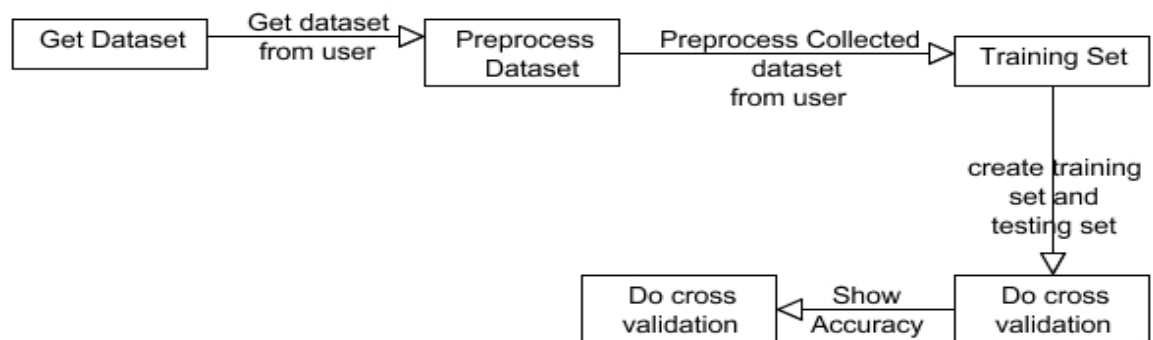


FIG 4.6 :COLLABARATION DIAGRAM

4.7 MODULES:

- ✓ **Data collection and pre-processing**
- ✓ **Random Forest model generation**
- ✓ **Prediction, result presentation**

4.7.1 DATA COLLECTING AND PRE-PROCESSING :

Weather dataset is collected in a comma separated values (CSV) file for the last three or four years of rainfall information. The dataset contains the month wise aggregation. The dataset might contain empty values, negative values or error. Dataset is cleaned in the pre-processing. The preprocessing methods involve of removing records which is not complete. Once the clean dataset is available we have to prepare it to feed to the machine learning algorithm

4.7.2 RANDOM FOREST MODEL GENERATION:

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of

decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction. One big advantage of random forest is, that it can be used for both classification and regression problems, which form the majority of current machine learning systems. We train our system with dataset and create the model for future prediction

4.7.3 PREDICTION, RESULT PRESENTATION:

Advantage of random forest is that it can be used for both regression and classification tasks and that it's easy to view the relative importance it assigns to the input features. Random Forest is also considered as a very handy and easy to use algorithm, because its default hyper parameters often produce a good prediction result. The number of hyper parameters is also not that high and they are straightforward to understand. The Random forest trained model is used to predict the rainfall for a particular month. The prediction duration is for a couple of month. Python matplotlib library can be used to produce the graphical representation of the data in a visual format. The predicted data is also plotted in the graph along with the existing history data.

CHAPTER 5

SUMMARY AND CONCLUSIONS

5.1 RESULTS:

1. The performance of the four predictive algorithms and hybrid models are calculated on two metrics accuracy and F-score as mention earlier.
2. Results showed that the highest F-score is achieved by GB-Adaboost classifier and has perform better than others predictive classifiers.
3. Both feature selection techniques (Gradient-boosting and Random forest) are applied to each predictive models but classifiers (Adaboost, SVM, NN) combined with gradient boosting feature selection achieve higher F-score than the classifier without feature selection.
4. As gradient boosting feature selection played major role in achieving higher F-score it is clear that the feature selection techniques improves the results than the individual ones.
5. Python matplotlib library can be used to produce the graphical representation of the data in a visual format.

5.2 CONCLUSIONS:

Random forest gives much more accurate predictions when compared to simple CART/CHAID or regression models in many scenarios. These cases generally have high number of predictive variables and huge sample size. This is because it captures the variance of several input variables at the same time and enables high number of observations to participate in the prediction

5.3 REFERENCES:

- [1] Cao, Li-Juan, and Francis Eng Hock Tay. "Support vector machine with adaptive parameters in financial time series forecasting." IEEE Transactions on neural networks 14.6 (2003): 1506-1518.
- [2] Charaniya, Nizar Ali, and Sanjay V. Dudul. "Committee of artificial neural networks for monthly rainfall prediction using wavelet transform." Business, Engineering and Industrial

Applications (ICBEIA), 2011 International Conference on. IEEE, 2011.

[3] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995), pp. 273-297

[4] French, Mark N., Witold F. Krajewski, and Robert R. Cuykendall. "Rainfall forecasting in space and time using a neural network." Journal of hydrology 137.1-4 (1992): 1-31.

[5] French, Mark N., Witold F. Krajewski, and Robert R. Cuykendall. "Rainfall forecasting in space and time using a neural network." Journal of hydrology 137.1-4 (1992): 1-31

[6] Guhathakurta, P. "Long-range monsoon rainfall prediction of 2005 for the districts and sub-division Kerala with artificial neural network." Current Science 90.6 (2006): 773-779.

[7] Hornik, Kurt, Maxwell Stinchcombe, and Halbert White. "Multilayer feedforward networks are universal approximators." Neural networks 2.5 (1989): 359-366.

[8] Haykin, Simon. Neural networks: a comprehensive foundation. Prentice Hall PTR, 1994

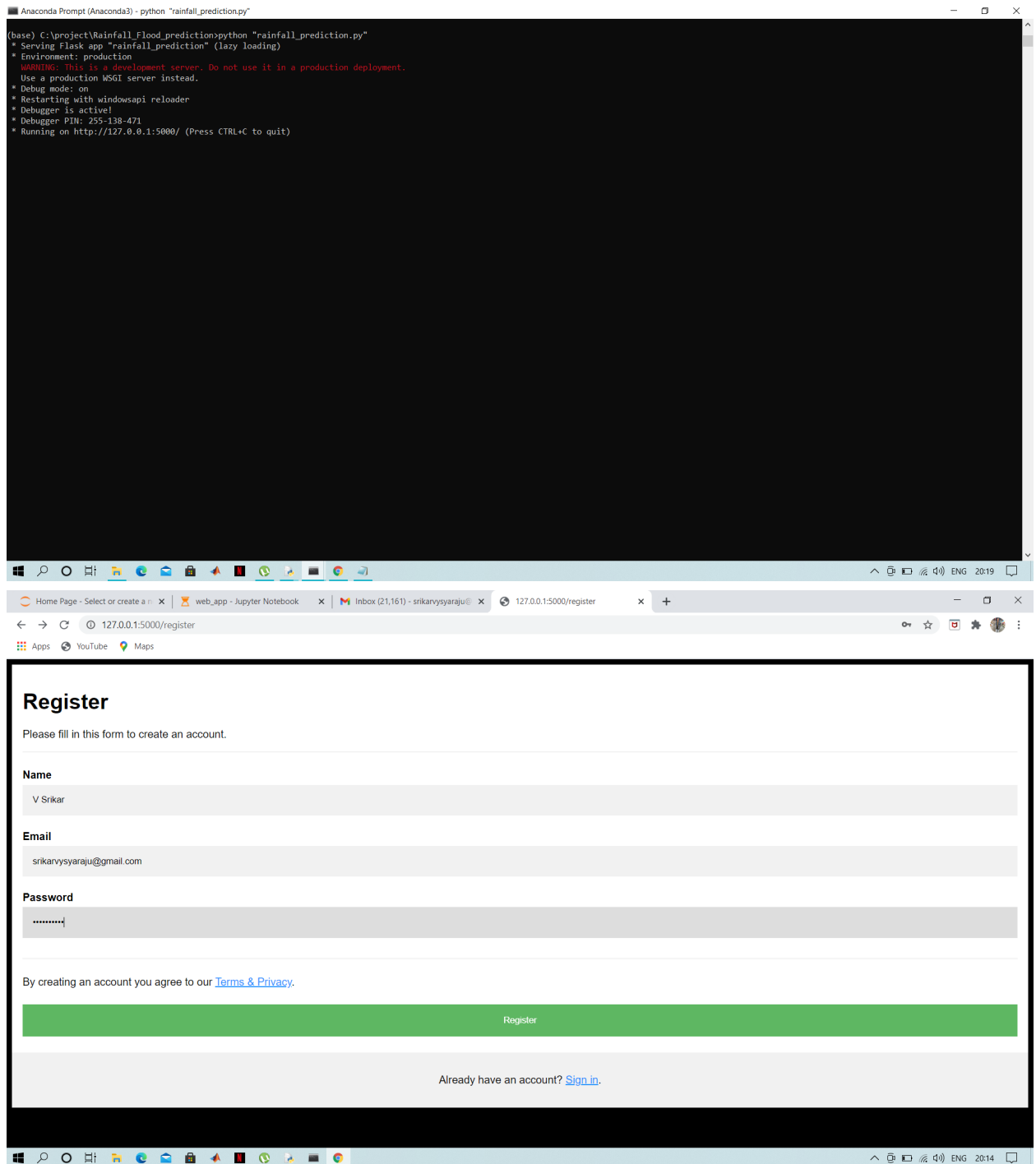
[9] Kim, Kyoung-jae. "Financial time series forecasting using support vector machines." Neurocomputing 55.1-2 (2003): 307-319.

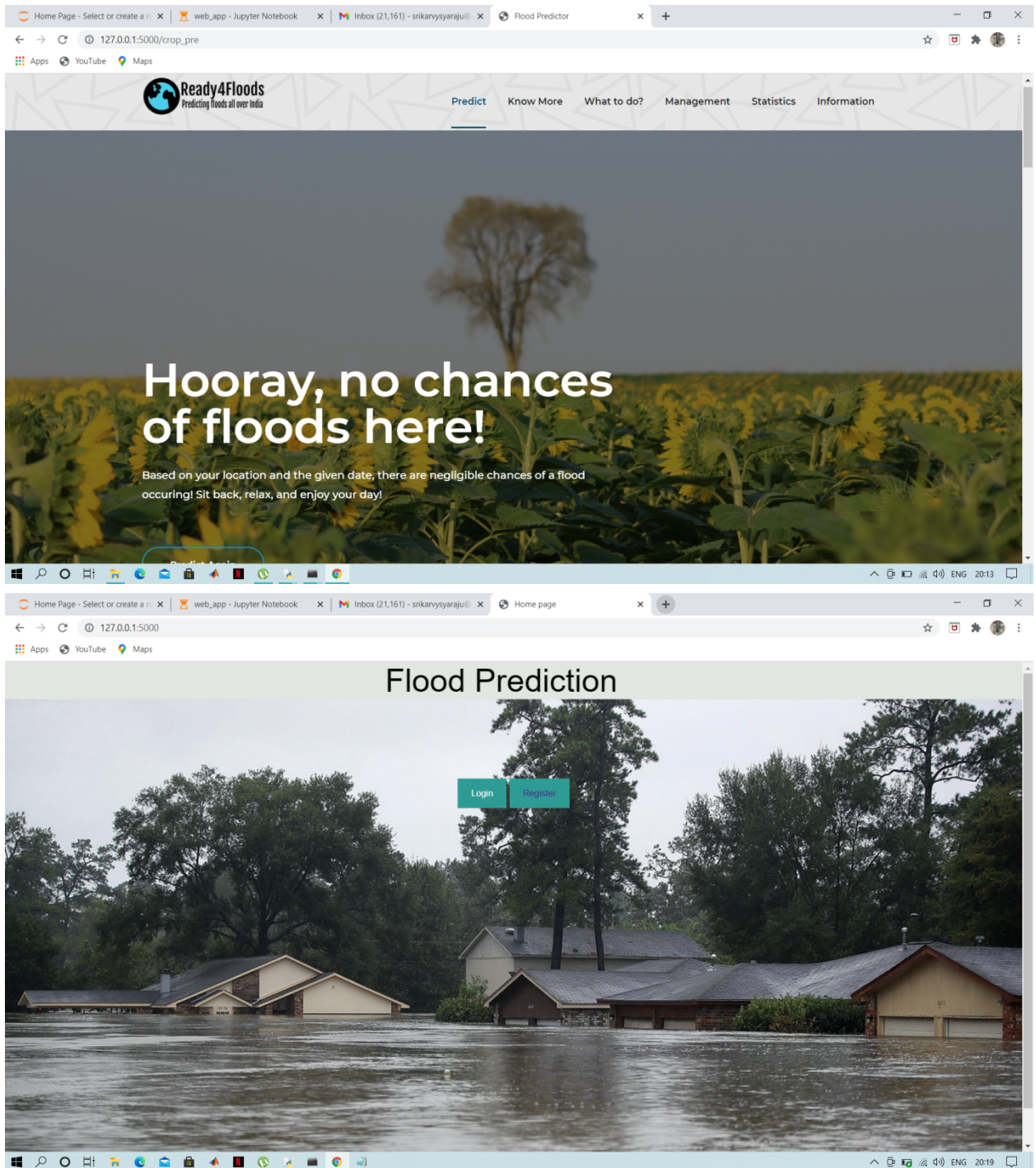
[10] Luk, K. C., James E. Ball, and Ashish Sharma. "A study of optimal model lag and spatial inputs to artificial neural network for rainfall forecasting." Journal of Hydrology 227.1-4 (2000): 56-65.

[10] Noone, David, and Harvey Stern. "Verification of rainfall forecasts from the Australian Bureau of Meteorology's Global Assimilation and Prognosis(GASP) system." Australian Meteorological Magazine 44.4 (1995): 275-286.

APPENDIX

A. SCREENSHOTS





B.SOURCE CODE

```
#Flask Libraries
from flask import Flask, redirect, url_for, render_template, request
import requests
import urllib
from datetime import datetime
import datetime

#Numpy
import numpy as np
#for importing our keras model
import keras.models
import pandas as pd
#system level operations (like loading files)
import sys
#for reading operating system data
import os

dataset = pd.read_csv(r'C:\Users\GPIndia1\Downloads\m-3\rain1.csv')
dataset = dataset[ dataset["YEAR"]>=1980 ]
dataset = dataset.dropna()
#without climate
X = dataset.iloc[:,[0,1,3,4]].values
y = dataset.iloc[:,8].values
print(X)
print(y)

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

0)

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

Logistic Regression Algorithm

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

Decision tree Algorithm

```
from sklearn.tree import DecisionTreeClassifier
dectree = DecisionTreeClassifier(criterion = 'entropy', random_state = 42)
dectree.fit(X_train, y_train)
```

Naive Bayes Algorithm

```
from sklearn.naive_bayes import GaussianNB
nb = GaussianNB()
nb.fit(X_train, y_train)
```

Random forest Algorithm

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

Making predictions on test dataset

```
Y_pred_logreg = logreg.predict(X_test)
Y_pred_knn = knn.predict(X_test)
#Y_pred_svc = svc.predict(X_test)
```

```

Y_pred_nb = nb.predict(X_test)
Y_pred_dectree = dectree.predict(X_test)
Y_pred_ranfor = ranfor.predict(X_test)

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

# Accuracy on test set
print("Logistic Regression: " + str(accuracy_logreg * 100))
print("K Nearest neighbors: " + str(accuracy_knn * 100))
print("Support Vector Classifier: " + str(accuracy_svc * 100))
print("Naive Bayes: " + str(accuracy_nb * 100))
print("Decision tree: " + str(accuracy_dectree * 100))
print("Random Forest: " + str(accuracy_ranfor * 100))

date = str(input("Date"))
month = datetime.datetime.strptime(date, '%d/%m/%Y').strftime('%m')
#year = str(input("Year"))
month = datetime.datetime.strptime(date, '%d/%m/%Y').strftime('%m')
month
Year = datetime.datetime.strptime(date, '%d/%m/%Y').strftime('%Y')
Year

import random
if(month == '01' or month == '02'):

```

```

precipitation=random.randint(1,300)
a=(precipitation)
print('1')
z="1"
    elif(month == '03' or month == '04' or month == '05'):
precipitation=random.randint(300,400)
a=(precipitation)
print('2')
z="2"
    elif(month == '06' or month == '07' or month == '08' or month == '09'):
precipitation=random.randint(1000,3500)
a=(precipitation)
print('3')
z="3"
elif(month == '10' or month == '11' or month == '12'):
    precipitation=random.randint(800,900)
    a=(precipitation)
    print('4')
    z="4"
state = str(input("STATE"))
if(state == 'chennai' or state == 'madurai',state == 'trichy' or state == 'kovai',state ==
'selem' or state == 'dindugal',state == 'virudhunagar' or state == 'kanchipuram',state
== 'trivallur' or state == 'kanyakumari',state == 'namakkal' or state ==
'nagarkovil',state == 'thanjavur' or state == 'vilupuram'):
    print('25')
    b="25"
elif(State == 'bengalore' or state == 'belegavi',State == 'kolar' or state ==
'raichur',State == 'ballari bidar' or state == 'shivamogga',State == 'ramanagara' or
state == 'chitradurga',State == 'davanagere' or state == 'vijayapura',State == 'uttara'
or state == 'bagalkot',State == 'yadgir' or state == 'dharwad'):
    print('7')
    b="7"

```

```
elif(State == 'malapuram' or state == 'plakkad',State == 'idukki' or state ==  
'kottiyam',State == 'alappuzha' or state == 'kollam',State == 'thiruvananthapuram' or  
state == 'wayanad',State == 'kannur' or state == 'kasaragod',State == 'kozhikode' or  
state == 'thrissur',State == 'pathanamthitta' or state == 'madurai):
```

```
    print('17')
```

```
    b="17"
```

```
elif(State == 'hyderabad' or state == 'kamareddy',State == 'khammam' or state ==  
'nagarkurnool',State == 'nizamabad' or state == 'karimnagar):
```

```
    print('26')
```

```
    b="26"
```

```
elif(state == 'GANGETIC WEST BENGAL):
```

```
    print('1')
```

```
    b="1"
```

```
elif(state == 'ANDAMAN & NICOBAR ISLANDS):
```

```
    print('2')
```

```
    b="2"
```

```
elif(state == 'ARUNACHAL PRADESH):
```

```
    print('3')
```

```
    b="3"
```

```
elif(state == 'ASSAM & MEGHALAYA):
```

```
    print('4')
```

```
    b="4"
```

```
elif(state == 'BIHAR):
```

```
    print('5')
```

```
    b="5"
```

```
elif(state == 'CHATTISGARH):
```

```
    print('6')
```

```
    b="6"
```

```
elif(state == 'COASTAL ANDHRA PRADESH):
```

```
    print('8')
```

```
    b="8"
```

```
elif(state == 'COASTAL KARNATAKA):
```

```
print('9')
b="9"
elif(state == 'EAST MADHYA PRADESH'):
    print('10')
    b="10"
elif(state == 'EAST RAJASTHAN'):
    print('11')
    b="11"
elif(state == 'EAST UTTAR PRADESH'):
    print('12')
    b="12"
elif(state == 'GUJARAT REGION'):
    print('13')
    b="13"
elif(state == 'HARYANA DELHI & CHANDIGARH'):
    print('14')
    b="14"
elif(state == 'HIMACHAL PRADESH'):
    print('15')
    b="15"
elif(state == 'JAMMU & KASHMIR'):
    print('16')
    b="16"
elif(state == 'JHARKHAND'):
    print('20')
    b="20"
elif(state == 'KONKAN & GOA'):
    print('18')
    b="18"
elif(state == 'LAKSHWADEEP'):
    print('19')
    b="19"
```

```

elif(state == 'MADHYA MAHARASHTRA'):
    print('21')
    b="21"
elif(state == 'MATATHWADA'):
    print('22')
    b="22"
elif(state == 'NAGA MANI MIZO TRIPURA'):
    print('23')
    b="23"
elif(state == 'NORTH INTERIOR KARNATAKA'):
    print('24')
    b="24"
elif(state == 'ORISSA'):
    print('29')
    b="29"
elif(state == 'PUNJAB'):
    print('27')
    b="27"
output=[b,Year,z,a]
response=ranfor.predict([output])
if(int(response) == 0 ):
    print("Hooray, no chances of floods here!")
elif(int(response) == 1):
    print("Get your weather boots out!")
elif(int(response) == 2):
    print("Looks like you are going to be working from home!")
elif(int(response) == 3):
    print("Beware, heavy floods incoming!")
elif(int(response) == 4):
    print("Brace yourself! Severe floods incoming!")
else:
    print("Act fast! You are in grave danger!")

```

