ABSTRACT

India Meteorological Department has implemented state level medium range rainfall forecast system applying multi model ensemble technique, making use of model outputs of state-of-the-art global models from the five leading global NWP centers. The pre-assigned grid point weights on the basis of anomaly correlation coefficients (CC) between the observed values and forecast values are determined for each constituent model utilizing two season datasets and the multi model ensemble forecasts are generated at the same resolution on a real-time basis. The ensemble forecast fields are then used to prepare forecasts for each state, taking the average value of all grid points falling in a particular district. In this paper, we describe the development strategy of the technique and performance skill of the system during 15 years of rain fall at different states in india. The study demonstrates the potential of the system for predicting future rainfall forecasts for upcoming years and scale over Indian region. District wise performance of the ensemble rainfall forecast reveals that the technique, in general, is capable of providing reasonably good forecast skill over most states of the country, particularly over the states where the monsoon systems are more dominant.

Rain fall forecasting is one of the most important analysis and prediction method which is required for predicting upcoming rainfall in coming years based on previous dataset. In this project past 10 years dataset for different states and districts are taken as input and weather forecast is predicted. In present studies rainfall prediction was based on NWP numerical weather prediction methods are used which has given better results and played important role. Even though this methods are used in present systems there are many areas accuracy can be increased like dealing with indian monsoon. This is because large variation of data at different times and limitation of NWP models.

1.INTRODUCTION

There has been long demand from the user community for district level quantitative weather forecasts in short to medium range time scale. The quantitative rainfall forecast for smaller spatial distribution such as district level over highly complex inhomogeneous region like India is a very challenging task. For the generation of district level quantitative rainfall forecasts, one has to depend on the forecasts from dynamical Numerical Weather Prediction (NWP) models. During the last two decades, NWP methods have acquired greater skills and are playing an increasingly important role in the operational weather forecasting. But rainfall prediction skill of NWP models is still not adequate to satisfactorily address detailed aspects of Indian summer monsoon.

1.1 Aim

The aim of the project is to analyse the amount of rainfall in different states in India over past 10 years and predict it for further years using Machine leaning models.

Indian meteorological department provides forecasting data required for project. In this project we are planning to work on long term predictions of rainfall.The main motive of the project is to predict the amount of rainfall in a particular division or state well in advance. We predict the amount of rainfall using past data.There has been long demand from the user community for state level quantitative weather forecasts in short to medium range time scale. The quantitative rainfall forecast for smaller spatial distribution such as state level over highly complex inhomogeneous region like India is a very challenging task.

This is because of large spatial and temporal variability of rainfall and some inherent limitations of NWP models. There are various factors like topography, prevailing synoptic situation and its interaction with mesoscale systems, lack of observations, etc., are some of the key factors which pose difficulties for numerical weather prediction of any region, and so Indian region is not an exception. Considering the need of farming sector, India Meteorological Department (IMD) has upgraded the Agro-Meteorological Advisory Service from agro climate zone to district level.

1.2 Existing system

There has been long demand from the user community for state level quantitative weather forecasts in short to medium range time scale. The quantitative rainfall forecast for smaller spatial distribution such as state level over highly complex inhomogeneous region like India is a very challenging task. For the generation of district level quantitative rainfall forecasts, one has to depend on the forecasts from dynamical Numerical Weather Prediction (NWP) models. During the last two decades, NWP methods have acquired greater skills and are playing an increasingly important role in the operational weather forecasting. But rainfall prediction skill of NWP models is still not adequate to satisfactorily address detailed aspects of Indian summer monsoon.

In our previous study (Roy Bhowmik and Durai 2008, 2010), performance skill of MME at 50 km horizontal resolution for district level short range rainfall forecasts during summer monsoon 2007 was demonstrated from the use of four coarser grid models namely (i) IMD limited area model at 75 km horizontal resolution, (ii) IMD MM5 at 45 km horizontal resolution, (iii) National Centre for Medium Range Weather Forecasting (NCMRWF) MM5 at 30 km resolution, and (iv) NCMRWF T-80 (grid space ~156 over the tropics). At 50 km resolution, MME could cover only 250 districts

Disadvantages

1) Existing methods are covering for only few districts which are not using machine learning techniques and there was no prediction for rainfall .

2) In preset system data set was limited for few districts which are restricted for data analysis.

3) The primary problem with Numerical Weather Prediction (NWP) models is,it takes long time to produce its results.

4) Existing methods are covering for only few districts, which in turn are restricting data analysis and data prediction process.

5) There are various factors like topography, its interaction with mesoscale systems, lack of observations, etc., are some of the key factors which pose difficulties.

1.3 Proposed system

Proposed system is designed for long term prediction by applying machine learning models like :

1) **Linear regression** - In statistics, linear regression is a linear approach to modeling the relationship between a scalar response and one or more explanatory variables.

2) **Artificial Neural Network** - ANNs are considered nonlinear statistical data modeling tools where the complex relationships between inputs and outputs are modeled or patterns are found.

3) **Support Vector Machine** - Support Vector Machines are a type of supervised machine learning algorithm that provides analysis of data for classification and regression analysis.

In the present study, we describe the development strategy of the MME technique, used for high resolution rainfall forecasts over Indian region and demonstrate the prediction skill.

Advantages

1. Ability to work with incomplete knowledge

2. Stability :- A small change to the data does not greatly affect the hyperplane and hence the SVM.

3. ANN takes data samples rather than entire data sets to arrive at solutions, which saves both time and money.

4. ANN's are considered fairly simple mathematical models to enhance existing data analysis technologies.

2. LITERATURE SURVEY

Improved weather and seasonal climate forecasts from multimodel super ensemble

India Meteorological Department has implemented district level medium range rainfall forecast system applying multimodel ensemble technique, making use of model outputs of state-of-the-art global models from the five leading global NWP centres. The pre-assigned grid point weights on the basis of anomaly correlation coefficients (CC) between the observed values and forecast values are determined for each constituent model at the resolution of 0.25° ×0.25° utilizing two season datasets (1 June–30 September, 2007 and 2008) and the multimodel ensemble forecasts (day-1 to day-5 forecasts) are generated at the same resolution on a real-time basis. The ensemble forecast fields are then used to prepare forecasts for each district, taking the average value of all grid points falling in a particular district. In this paper, we describe the development strategy of the technique and performance skill of the system during summer monsoon 2009. The study demonstrates the potential of the system for improving rainfall forecasts at five days time scale over Indian region. Districtwise performance of the ensemble rainfall forecast reveals that the technique, in general, is capable of providing reasonably good forecast skill over most states of the country, particularly over the states where the monsoon systems are more dominant.

Improving tropical precipitation forecasts from a multi analysis super ensemble

This paper utilizes forecasts from a multianalysis system to construct a superensemble of precipitation forecasts. This method partitions the computations into two time lines. The first of those is a control (or a training) period and the second is a forecast period. The multianalysis is derived from a physical initialization–based data assimilation of “observed rainfall rates.” The different members of the reanalysis are produced by using different rain-rate algorithms for physical initialization. The basic rain-rate datasets are derived from satellites’ microwave radiometers, including those

from the Tropical Rainfall Measuring Mission (TRMM) satellites and the Special Sensor Microwave Imager (SSM/I) data from three current U.S. Air Force Defense Meteorological Satellite Program (DMSP) satellites. During the training period, 155 experiments were conducted to find the relationship between forecasts from the multianalysis dataset and the best "observed" estimates of daily rainfall totals. This relationship is based on multiple regression and defined by statistical weights (which vary in space.) The forecast phase utilizes the multianalysis forecasts and the statistics from the training period to produce superensemble forecasts of daily rainfall totals. The results for day 1, day 2, and day 3 forecasts are compared to various conventional forecasts with a global model. The superensemble day 3 forecasts of precipitation clearly have the highest skill in such comparisons.

Experimental realtime multi-model ensemble (MME) prediction of rainfall during monsoon 2008: Large scale medium range aspects

Realistic simulation/prediction of the Asian summer monsoon rainfall on various space–time scales is a challenging scientific task. Compared to mid-latitudes, a proportional skill improvement in the prediction of monsoon rainfall in the medium range has not happened in recent years. Global models and data assimilation techniques are being improved for monsoon/tropics. However, multi-model ensemble (MME) forecasting is gaining popularity, as it has the potential to provide more information for practical forecasting in terms of making a consensus forecast and handling model uncertainties. As major centers are exchanging model output in near real-time, MME is a viable inexpensive way of enhancing the forecasting skill and information content. During monsoon 2008, on an experimental basis, an MME forecasting of large-scale monsoon precipitation in the medium range was carried out in real-time at National Centre for Medium Range Weather Forecasting (NCMRWF), India. Simple ensemble mean (EMN) giving equal weight to member models, bias-corrected ensemble mean (BCEMn) and MME forecast, where different weights are given to member models, are the products of the algorithm tested here. In general, the aforementioned products from the multi-model ensemble forecast system have a higher skill than individual model forecasts. The skill score for the Indian domain and other sub-regions indicates that the BCEMn produces the best result, compared to EMN and MME. Giving weights

to different models to obtain an MME product helps to improve individual member models only marginally. It is noted that for higher rainfall values, the skill of the global model rainfall forecast decreases rapidly beyond day-3, and hence for day-4 and day-5, the MME products could not bring much improvement over member models. However, up to day-3, the MME products were always better than individual member models.

High resolution daily gridded rainfall data for Indian Region: Analysis of break and active monsoon spells

In this communication, we discuss the development of a very high resolution (0.5° x 0.5°) daily rainfall data-set for mesoscale meteorological studies over the Indian region. The dataset was developed using quality-controlled rainfall data from more than 3000 rain gauge stations over India. The analysis consists of daily rainfall data for all the seasons for the period 1971-2005. A well-tested interpolation method (Shepard's method) was used to interpolate the station data into regular grids of 0.5° x 0.5° lat. x long. After proper validation, it has been found that the present dataset is better compared to other available datasets. A few case studies have been shown to demonstrate the utility of the dataset for different mesoscale meteorological analyses. However, since the data density is not kept uniform, there is a possibility of temporal inhomogeneity and therefore, the present dataset cannot be used for trend analysis. The dataset is freely available from the India Meteorological Department, Pune.

Rainfall analysis for Indian monsoon region using the merged rain gauge observations and satellite estimates

Objective analysis of daily rainfall at the resolution of 1° grid for the Indian monsoon region has been carried out merging dense land rainfall observations and INSAT derived precipitation estimates. This daily analysis, being based on high dense rain gauge observations was found to be very realistic and able to reproduce detailed features of Indian summer monsoon. The inter-comparison with the observations suggests that the new analysis could distinctly capture characteristic features of the summer monsoon such as north-south oriented belt of heavy rainfall along the Western Ghats with sharp gradient of rainfall between the west coast

heavy rain region and the rain shadow region to the east, pockets of heavy rainfall along the location of monsoon trough/low, over the east central parts of the country, over north-east India, along the foothills of Himalayas and over the north Bay of Bengal. When this product was used to assess the quality of other available standard climate products (CMAP and ECMWF reanalysis) at the gird resolution of 2.5°, it was found that the orographic heavy rainfall along Western Ghats of India was poorly identified by them. However, the GPCC analysis (gauge only) at the resolution of 1° grid closely discerns the new analysis. This suggests that there is a need for a higher resolution analysis with adequate rain gauge observations to retain important aspects of the summer monsoon over India. The case studies illustrated show that the daily analysis is able to capture large-scale as well as mesoscale features of monsoon precipitation systems. This study with data of two seasons (2001 and 2003) has shown sufficiently promising results for operational application, particularly for the validation of NWP models.

3. SYSTEM STUDY

A detailed study to determine whether, to what extent, and how automatic data-processing equipment should be used; it usually includes an analysis of the existing system and the design of the new system, including the development of system specifications which provide a basis for the selection of equipment.

3.1 FEASIABILITY STUDY

Preliminary investigation examines project feasibility; the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All systems are feasible if they are given unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility
- Operation Feasibility
- Economical Feasibility

3.1.1 Technical Feasibility

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipment have the technical capacity to hold the data required to use the new system?
- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- Can the system be upgraded if developed?

3.1.2 Operation Feasibility

Customer will use the forms for their various transactions i.e. for adding new routes, viewing the routes details. Also the Customer wants the reports to view the

various transactions based on the constraints. These forms and reports are generated as user-friendly to the Client.

• The package wills pick-up current transactions on line. Regarding the old transactions, User will enter them in to the system.
• The web server and database server should be protected from hacking, virus etc.,
The application will be developed using standard open source software (Except Oracle) like Java, tomcat web server, Internet Explorer Browser etc these software will work both on Windows and Linux o/s. Hence portability problems will not arise.

• This software will be available always. The system uses the 2-tier architecture. The 1st tier is the GUI, which is said to be front-end and the 2nd tier is the database, which uses My-Sql, which is the back-end.

• The front-end can be run on different systems (clients). The database will be running at the server. Users access these forms by using the user-ids and the passwords.

3.1.3 Economic feasibility

The computerized system takes care of the present existing system's data flow and procedures completely and should generate all the reports of the manual system besides a host of other management reports. It should be built as a web based application with separate web server and database server. This is required as the activities are spread throughout the organization customer wants a centralized database. Further some of the linked transactions take place in different locations. Open source software like TOMCAT, JAVA, Mysql and Linux is used to minimize the cost for the Customer. All systems are feasible if they are given unlimited resources and infinite time. The interpretation of the results from the automated accessibility testing tools requires experience in accessibility techniques with an understanding of technical and usability issues.

3.2 System Requirements

System requirements are all of the requirements at the *system level* that describe the functions which the system as a whole should fulfill to satisfy the stakeholder

needs and requirements, and is expressed in an appropriate combination of textual statements, views, and non-functional requirements; the latter expressing the levels of safety, security, reliability, etc., that will be necessary.

System requirements play major roles in systems engineering, as they

1) Form the basis of system architecture and design activities.
2) Form the basis of system integration and verification activities.
3) Act as reference for validation and stakeholder acceptance.
4) Provide a means of communication between the various technical staff that interact throughout the project

3.3 Non-Functional Requirements

All the other requirements which do not form a part of the above specification are categorized as Non-Functional needs. A system perhaps needed to gift the user with a show of the quantity of records during info. If the quantity must be updated in real time, the system architects should make sure that the system is capable of change the displayed record count at intervals associate tolerably short interval of the quantity of records dynamic. Comfortable network information measure may additionally be a non-functional requirement of a system.

The following are the features

- Accessibility
- Availability
- Maintenance
- Computer Performance
- Portability
- Reliability
- Response Time

Accessibility

Accessibility testing is a subset of usability testing where in the users under consideration are people with all abilities and disabilities. The significance of this testing is to verify both usability and accessibility. Accessibility testing is a subset of usability testing where in the users under consideration are people with all abilities and disabilities. The significance of this testing is to verify both usability and accessibility. The above said automated accessibility testing tools are very good at identifying pages and lines of code that need to be manually checked for accessibility. The interpretation of the results from the automated accessibility testing tools requires experience in accessibility techniques with an understanding of technical and usability issues.

Availability

Availability indicates when a system is operational as well as how reliable it is during operational period, What are the hours that a given system will be available? What days will the system be operational? Not all systems operate on a 24/7 basis. Some internal facing systems may only be needed when there are people in place to operate them. During a system's hours of operation, what reliability (excluding planned outages) is needed? Reliability is usually measured as a percentage. The higher the number, the greater the cost.

Maintenance

The technical meaning of maintenance involves functional checks, servicing, repairing or replacing of necessary devices, equipment, machinery, building infrastructure, and supporting utilities in industrial, business, governmental, and residential installations. Over time, this has come to include multiple wordings that describe various cost-effective practices to keep equipment operational; these activities take place either before or after a failure.

The marine and air transportation, offshore structures, industrial plant and facility management industries depend on maintenance, repair and overhaul (MRO) including scheduled or preventive paint maintenance programs to maintain and restore coatings applied to steel in environments subject to attack from erosion, corrosion and

environmental pollution. Architectural conservation employs MRO to preserve, rehabilitate, restore, or reconstruct historical structures with stone, brick, glass, metal, and wood which match the original constituent materials where possible, or with suitable polymer technologies when not.

Computer Performance

In computing, computer performance is the amount of useful work accomplished by a computer system. Outside of specific contexts, computer performance is estimated in terms of accuracy, efficiency and speed of executing more of the following factors might be involved

- Short response time for a given piece of work.
- High throughput (rate of processing work).
- Low utilization of computing resource(s).
- High availability of the computing system or application.
- Fast (or highly compact) data compression and decompression.
- High bandwidth.
- Short data transmission time.

Portability

Portability in high-level computer programming is the usability of the same software in different environments. The requirement for portability is the generalized abstraction between the application logic and system interfaces. When software with the same functionality is produced for several computing platforms, portability is the key issue for development cost reduction.When operating systems of the same family are installed on two computers with processors with similar instruction sets it is often possible to transfer the files implementing program files between them.

Reliability

Reliability in statistics and psycho metrics is the overall consistency of a meas

- sure. A measure is said to have a high reliability if it produces similar results under consistent conditions. "It is the characteristic of a set of test scores that relates to the amount of random error from the measurement process that might be embedded in the scores. Scores that are highly reliable are accurate, reproducible, and consistent from one testing occasion to another. That is, if the testing process were repeated with a group of test takers, essentially the same results would be obtained. Various kinds of reliability coefficients, with values ranging between 0.00 (much error) and 1.00 (no error), are usually used to indicate the amount of error in the scores." For example, measurements of people's height and weight are often extremely reliable.

Response Time

Response time is the total amount of time it takes to respond to a request for service. That service can be anything from a memory fetch, to a disk IO, to a complex database query, or loading a full web page. Ignoring transmission time for a moment, the response time is the sum of the service time and wait time. The service time is the time it takes to do the work you requested. For a given request the service time varies little as the workload increases – to do X amount of work it always takes X amount of time. The wait time is how long the request had to wait in a queue before being serviced and it varies from zero, when no waiting is required, to a large multiple of the service time, as many requests are already in the queue and have to be serviced first.

3.4 Hardware and Software Requirements

Hardware Requirements

Hard Disk : 120 GB

Input Device : Keyboard, Mouse

Ram : 1GB.

System : Pentium Dual Core

Software Requirements

Operating system : Windows 7 or 10.

Tool : Anaconda (Jupiter)

Language : Python

Version : 3.5

Libraries : numpy, pandas, matplotlib ,sklearn

3.5 Technology Used

Python

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

Python concepts

• Open source general-purpose language.

• Object Oriented, Procedural, Functional

• Easy to interface with C/ObjC/Java/Fortran

• Great interactive environment

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English key words frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

• **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

• **Python is Object - Oriented** − Python supports Object-Oriented style or technique

of programming that encapsulates code within objects.

Python has five standard data types

• Numbers

• String

• List

• Tuple

• Dictionary

Python Numbers

Number data types store numeric values. Number objects are created when you assign a value to them

Python Strings

Strings in Python are identified as a contiguous set of characters represented in the quotation marks. Python allows for either pairs of single or double quotes. Subsets of strings can be taken using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the string and working their way from -1 at the end.

Python Lists

Lists are the most versatile of Python's compound data types. A list contains items separated by commas and enclosed within square brackets ([]). To some extent, lists are similar to arrays in C. One difference between them is that all the items belonging to a list can be of different datatype. The values stored in a list can be accessed using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the list and working their way to end -1. The plus (+) sign is the list concatenation operator, and the asterisk (*) is the repetition operator.

Python Tuples

A tuple is another sequence data type that is similar to the list. A tuple consists of a number of values separated by commas. Unlike lists, however, tuples are enclosed within parentheses. The main differences between lists and tuples are: Lists

are enclosed in brackets ([ ]) and their elements and size can be changed, while tuples are enclosed in brackets (()).

Python Dictionary

Python's dictionaries are kind of hash table type. They work like associative arrays or hashes found in Perl and consist of key-value pairs. A dictionary key can be almost any Python type, but are usually numbers or strings. Values, on the other hand, can be any arbitrary Python object. Dictionaries are enclosed by curly braces ({ }) and values can be assigned and accessed using square braces ([]).

Python libraries

1. Requests. The most famous http library written by kenneth reitz. It's a must have for every python developer.

2. Scrapy. If you are involved in web scraping then this is a must have library for you. After using this library you won't use any other.

3. wxPython. A gui toolkit for python. I have primarily used it in place of tkinter. You will really love it.

4. Pillow. A friendly fork of PIL (Python Imaging Library). It is more user friendly than PIL and is a must have for anyone who works with images.

5. SQLAlchemy. A database library. Many love it and many hate it. The choice is yours.

6. BeautifulSoup. I know it's slow but this xml and html parsing library is very useful

for beginners.

7. Twisted. The most important tool for any network application developer. It has a very beautiful api and is used by a lot of famous python developers.

8. NumPy. How can we leave this very important library ? It provides some advance math functionalities to python.

9. SciPy. When we talk about NumPy then we have to talk about scipy. It is a library of algorithms and mathematical tools for python and has caused many scientists to

switch from ruby to python.

10. Matplotlib. A numerical plotting library. It is very useful for any data scientist.

11. Pygame. Which developer does not like to play games and develop them ? This library will help you achieve your goal of 2d game development.

Here's a brief list of Python OOP ideas

• The class statement creates a class object and gives it a name. This creates a new namespace.

• Assignments within the class create class attributes. These attributes are accessed by qualifying the name using dot syntax: ClassName.Attribute.

• Class attributes export the state of an object and its associated behavior. These attributes are shared by all instances of a class.

• Calling a class (just like a function) creates a new instance of the class. This is where the multiple copies part comes in.

Machine Learning

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide.

Artificial Neural Network

Artificial neural networks (ANN) or connectionist systems are computing systems that are inspired by, but not identical to, biological neural networks that constitute animal brains. Such systems "learn" to perform tasks by considering examples, generally without being programmed with task-specific rules.

Support Vector Machine

Support Vector Machine can also be used as a regression method, maintaining all the main features that characterize the algorithm (maximal margin). The Support Vector Regression(SVR) uses the same principles as the SVM for classification, with only a few minor differences.

Linear Regression

In linear regression, the relationships are modeled using linear predictor functions whose unknown model parameters are estimated from the data. Such models are called linear models. Linear regression was the first type of regression analysis to be studied rigorously, and to be used extensively in practical applications. This is because models which depend linearly on their unknown parameters are easier to fit than models which are non-linearly related to their parameters and because the statistical properties of the resulting estimators are easier to determine.

Neural networks are quicker than other methods including regression because they are executing parallel and tolerate more errors and also these networks can make rules without any implicit formula which are understandable in an environment of chaos and implicitly, such as stock exchange which is a very important factor.

4. SYSTEM DESIGN

The purpose of the design phase is to arrange an answer of the matter such as by the necessity document. This part is that the opening moves in moving the matter domain to the answer domain. The design phase satisfies the requirements of the system. The design of a system is probably the foremost crucial issue warm heartedness the standard of the software package. It's a serious impact on the later part, notably testing and maintenance. The output of this part is that the style of the document. This document is analogous to a blueprint of answer and is employed later throughout implementation, testing and maintenance. The design activity is commonly divided into 2 separate phases System Design and Detailed Design.

4.1 System Architecture

System Design conjointly referred to as top-ranking style aims to spot the modules that ought to be within the system, the specifications of those modules, and the way them move with one another to supply the specified results.

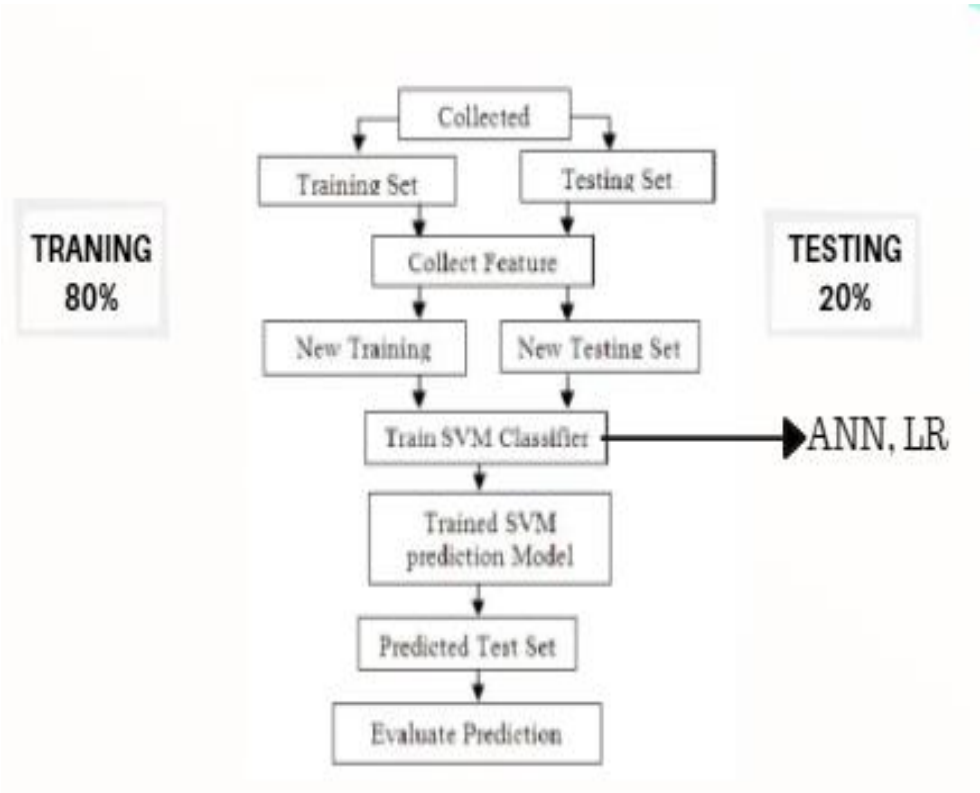


Figure 4.1: Architecture diagram

At the top of the system style all the main knowledge structures, file formats, output formats, and also the major modules within the system and their specifications square measure set. System design is that the method or art of process the design, components, modules, interfaces, and knowledge for a system to satisfy such as needs. Users will read it because the application of systems theory to development.

Detailed Design, the inner logic of every of the modules laid out in system design is determined. Throughout this part, the small print of the info of a module square measure sometimes laid out in a high-level style description language that is freelance of the target language within which the software package can eventually be enforced. In system design the main target is on distinguishing the modules, whereas throughout careful style the main target is on planning the logic for every of the modules.

4.2 Modules

4.2.1 Data Processing

Data processing occurs when data is collected and translated into usable information. Usually performed by a data scientist or team of data scientists, it is important for data processing to be done correctly as not to negatively affect the end product, or data output.Data processing starts with data in its raw form and converts it into a more readable format (graphs, documents, etc.), giving it the form and context necessary to be interpreted by computers and utilized by employees throughout an organization.

Six stages of data collection

1.Data Collection

A data set is a collection of data. In other words, a data set corresponds to the contents of a single database table, or a single statistical data matrix, where every column of the table represents a particular variable, and each row corresponds to a given member of the data set in question.In Machine Learning projects, we need a training data set. Regardless of the field of study or preference for defining data (quantitative, qualitative), accurate data collection is essential to maintaining the integrity of research. Both the selection of appropriate data collection instruments

(existing, modified, or newly developed) and clearly delineated instructions for their correct use reduce the likelihood of errors occurring.

2.Data preparation

Once the data is collected, it then enters the data preparation stage. Data preparation, often referred to as "pre-processing" is the stage at which raw data is cleaned up and organized for the following stage of data processing. During preparation, raw data is diligently checked for any errors.

3. Data input

The clean data is then entered into its destination (perhaps a CRM like Salesforce or a data warehouse like Redshift), and translated into a language that it can understand. Data input is the first stage in which raw data begins to take the form of usable information.

4. Processing

During this stage, the data inputted to the computer in the previous stage is actually processed for interpretation. Processing is done using machine learning algorithms, though the process itself may vary slightly depending on the source of data being processed (data lakes, social networks, connected devices etc.) and its intended use (examining advertising patterns, medical diagnosis from connected devices, determining customer needs, etc.).

5. Data Output/Interpretation

The output/interpretation stage is the stage at which data is finally usable to non-data scientists. It is translated, readable, and often in the form of graphs, videos, images, plain text, etc.). Members of the company or institution can now begin to self-serve the data for their own data analytics projects.

4.2.2 Training and Test Set

Machine learning works on data like temperature values or stock prices or color intensities, and in our case face and eye detection. The data is usually pre processed into features. We might take a database containing 1,000 face images then perform an edge detector on all the faces, and then obtain features such as edge direction, edge

intensity, also offset from the face center for every face. We may obtain up to 500 such values for every face or a feature vector of 500 entries. We may then use machine learning to construct some sort of model from the obtained data. If we want to see how the faces fall into various groups (narrow, wide, etc.), after that a clustering algorithm can be the preferred choice. In case we want to learn how to guess the age of a woman from the pattern of the edges that are detected on her face, then we can use a classifier algorithm.

This method of parameter adjustment for meeting a goal is what is called learning. It is very important to understand how efficiently machine learning methods can work. This might be a delicate task. Usually, we break up the input data set into a very large training set (i.e. 900 faces, in our project) and a relatively small test set (i.e. remainder 100 faces). Then run classifier on the training set in order to learn for a age prediction model, data feature vectors given. Once done, we can then test our age prediction classifier obtained on the remainder of the images left in the set. The test set is not applied for training; also we don't allow the classifier to see the age labels of the test set. Then run the classifier on all of the 100 faces present in the test set and then record how well the ages predicted by the feature vector match the real ages.

When the classifier performs well, we have a possibly lucrative model that can be used on data in the actual world. This system may be used to set the performance of a video game according to age. After the classifier has been setup, it sees faces that it could not see before and it makes decisions with regards to what it had learned during training. Finally, when we develop a classification system, we usually use a validation data set.

4.2.3 Prediction

Prediction refers to the output of an algorithm after it has been trained on a historical dataset and applied to new data when forecasting the likelihood of a particular outcome, such as whether or not a customer will churn in 30 days. The algorithm will generate probable values for an unknown variable for each record in the new data, allowing the model builder to identify what that value will most likely be.Machine learning model predictions allow businesses to make highly accurate guesses as to the likely outcomes of a question based on historical data, which can be

about all kinds of things – customer churn likelihood, possible fraudulent activity, and more.

4.3 Unified Modeling Language

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. Each view is defined by a set of diagram, which is as follows. The UML acts as a blue print for designing any software application.

4.3.1 Use-case diagram

A usecase diagram is a graphic depiction of the interactions among the elements of a system. A usecase diagram is a graphic depiction of the interactions among the elements of a system. A usecase is a description of set of sequence of actions that a system performs that yields an observable result of value to a particular actor.

Use Case - Draw use cases using ovals. Label with ovals with verbs that represent the system's functions structure the behavioral things in a model.

Actor - Actors are the users of a system. When one system is the actor of another system, label the actor system with the actor stereotype.

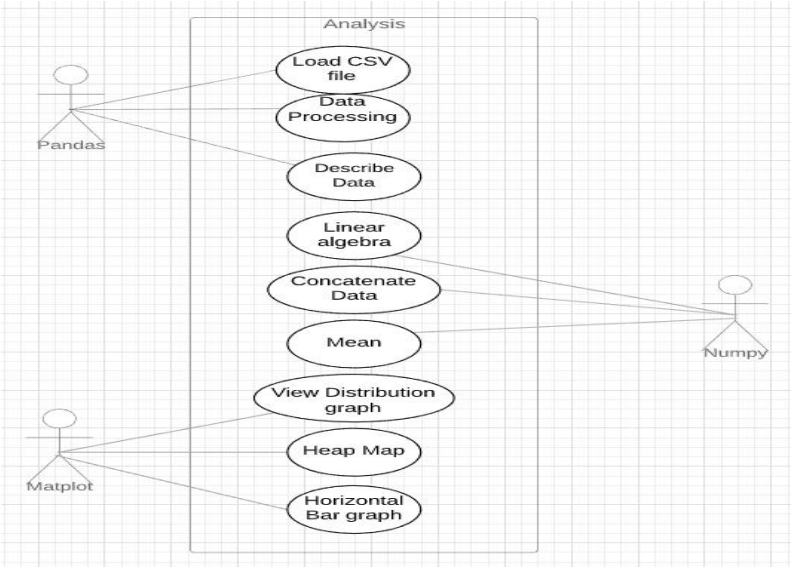


Figure 4.2: Usecase diagram for Analysis

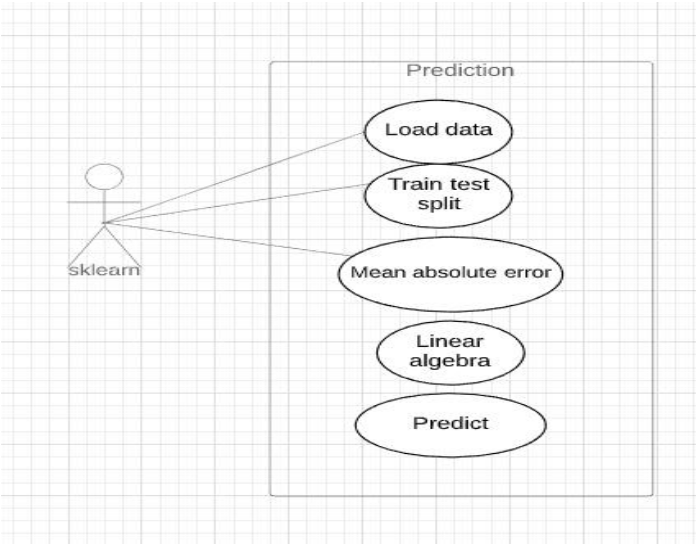


Figure 4.3: Usecase diagram for Prediction

4.3.2 Class diagram

A Class diagram shows a set of classes, interfaces, and collaborations and their relationships. In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

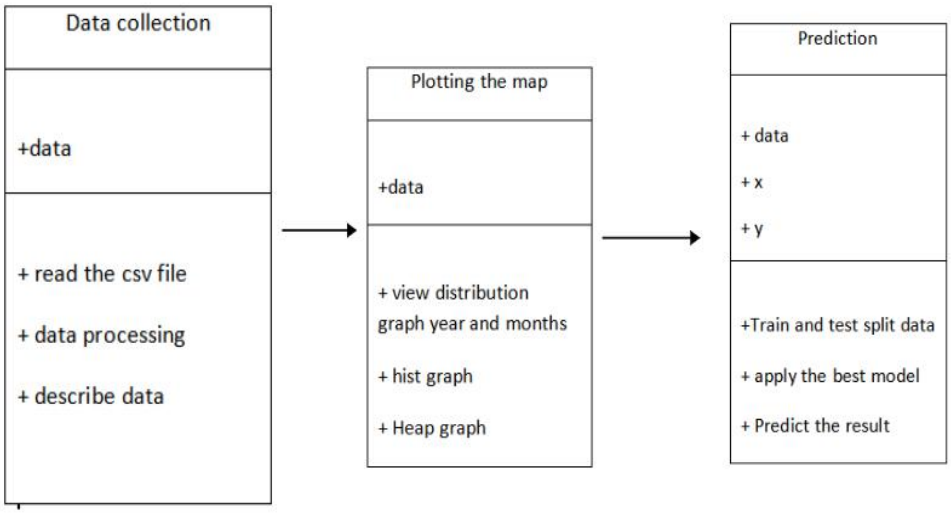


Figure 4.4: Class Diagram

4.3.3 Object Diagram

An object diagram shows a set of objects and their relationships.An object diagram is a UML structural diagram that shows the instances of the classifiers in models. Object diagrams use notation that is similar to that used in class diagrams. A static UML object diagram is an instance of a class diagram; it shows a snapshot of the detailed state of a system at a point in time, thus an object diagram encompasses objects and their relationships at a point in time. It may be considered a special case of a class diagram or a communication diagram.

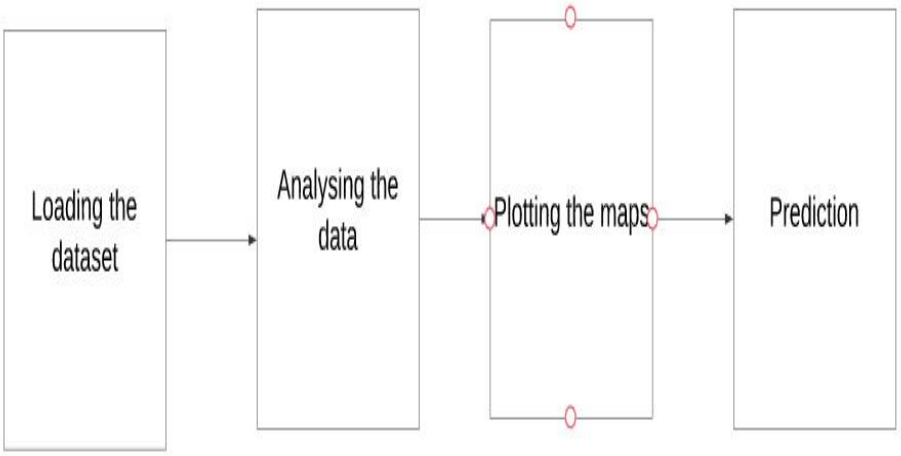


Figure 4.5: Object Diagram

4.3.4 Sequence Diagram

A sequence diagram is an interaction diagram that emphasizes the time ordering of messages.A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process.

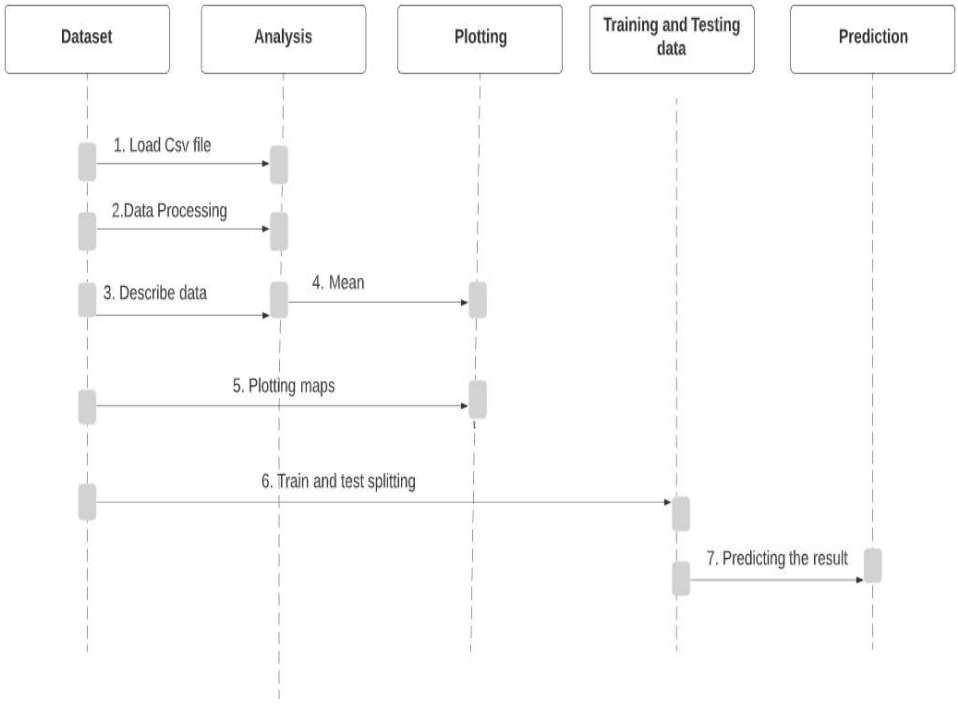


Figure 4.6: Sequence Diagram

4.3.5 Collaboration Diagram

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object.

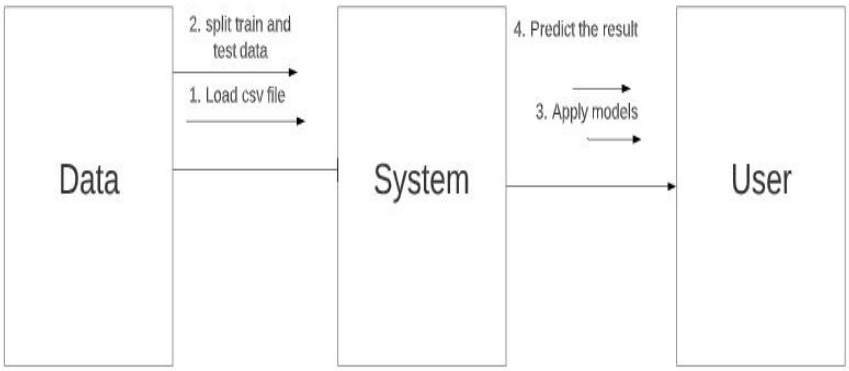
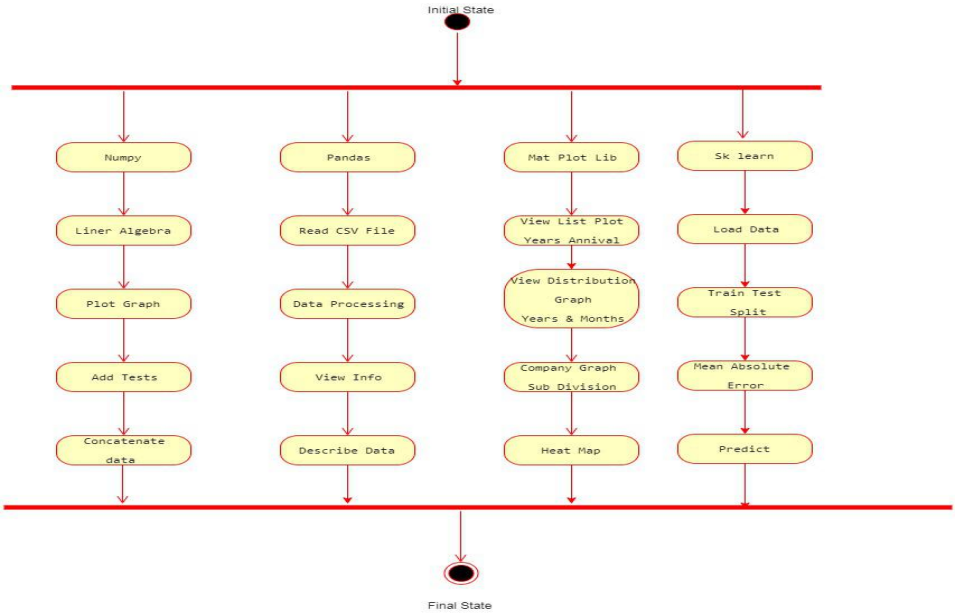


Figure 4.7 : Collaboration Diagram

4.3.6 Activity diagram

Activity diagrams address the dynamic view of a system. Activity diagram shows the flow of one activity to other activity.Activity diagram is defined as a UML diagram that focuses on the execution and flow of the behavior of a system instead of implementation. It is also called object-oriented flowchart. Activity diagrams consist of activities that are made up of actions which apply to behavioral modeling technology.



Figuree 4.8: Activity Diagram

4.3.7 State Chart Diagram

Statechart diagram is one of the five UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events.

Statechart diagrams are useful to model the reactive systems.Statechart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of Statechart diagram is to model lifetime of an object from creation to termination.

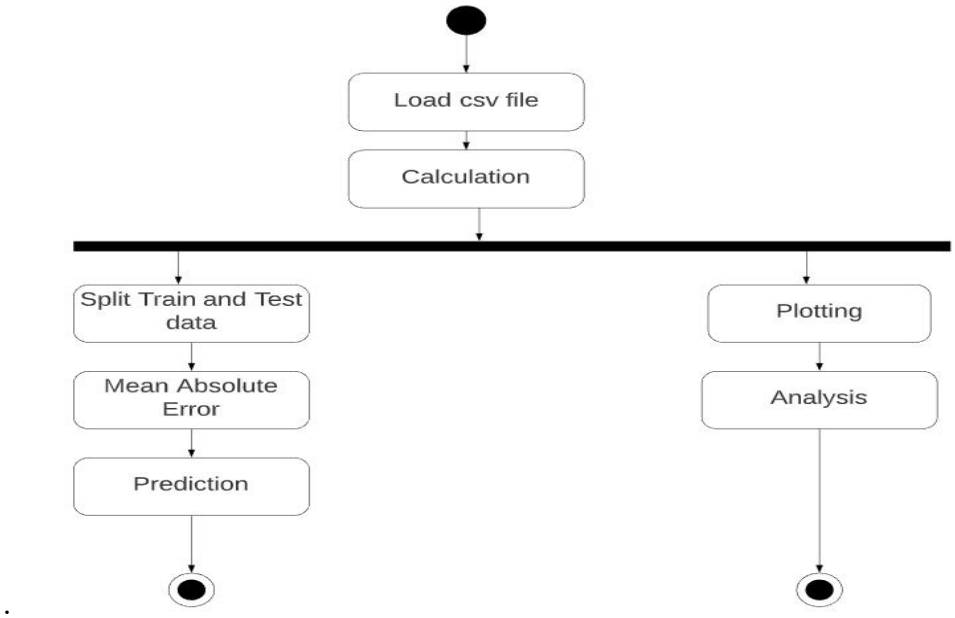


Figure 4.9: State chart Diagram

4.3.8 Component diagram

A component diagram depicts how components are wired together to form larger components and or software systems. A component is a physical building block of the system. It is represented as a rectangle with tab.A component is a physical building block of the system. It is represented as a rectangle with tab.

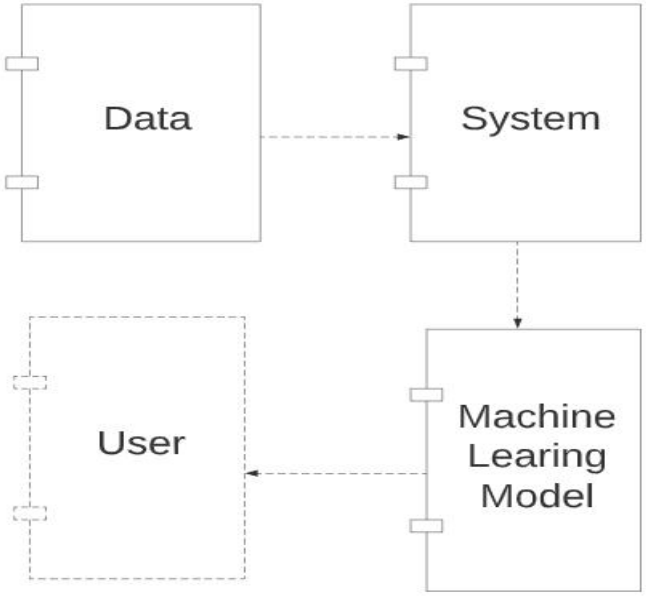


Figure 4.10: Component Diagram

4.3.9 Deployment diagram

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed. The term Deployment itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components, where software components are deployed. Component diagrams and deployment diagrams are closely related. Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware.UML is mainly designed to focus on the software artifacts of a system. However, these two diagrams are special diagrams used to focus on software and hardware components.

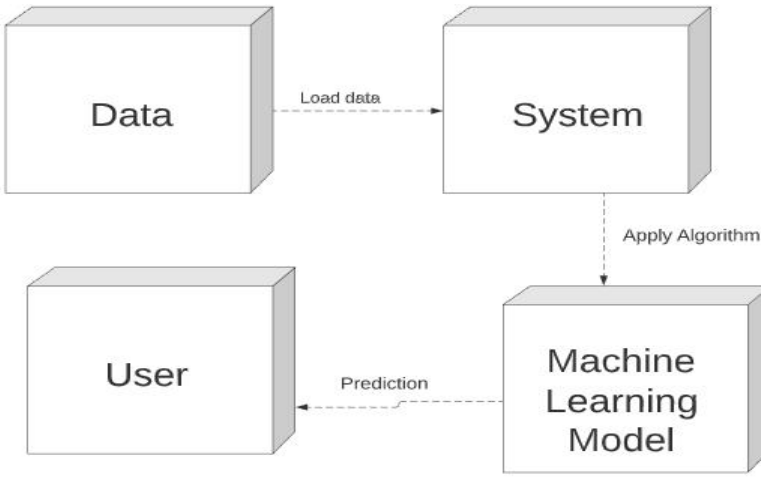


Figure 4.11: Deployment Diagram

5.IMPLEMENTATION

Data Collection

- Fist step is to collect rain fall dataset of different districts and states for few yesrs .
- Dataset(https://data.gov.in/resources/district-rainfall-normal-mm-monthly-seasonal-and-annual-data-period-1951-2000) .This dataset has average rainfall from 1951-2000 for each district, for every month.
- For some of the subdivisions data is from 1950 to 2015. All the attributes has the sumof amount of raianfall in mm.

Prediction

For prediction format data in the way, given the rainfall in the last three months to predict the rainfall in the next consecutive month.

• For all the experiments we used 80:20 training and test ratio.

– Linear regression

– SVR

– Artificial neural nets

• Testing metrics: We used Mean absolute error to train the models.

• We also shown the amount of rainfall actually and predicted with the histogram plots.

5.1 Algorithm

Step 1 : Load the .csv file in the program.

Step 2 : Describe the data

Step 3 : Plot different graphs to analyse the rainfall dataset

Step 4 : Split the train and test data

Step 5 : Apply ANN,LR and SVR on the train data

Step 6 : Find the Mean Absolute Error

Step 7 : Use the best model among the three for prediction

5.2 Sample Code

```
import numpy as np # linear algebras

import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

import matplotlib.pyplot as plt

import seaborn as sns

# In[2]:

data = pd.read_csv("../data/rainfall_in_india_1901-2015.csv",sep=",")

data = data.fillna(data.mean())

data.info()

# In[3]:

data.head()

# In[4]:

data.describe()

# In[5]:

data.hist(figsize=(24,24));

# In[6]:

data.groupby("YEAR").sum()['ANNUAL'].plot(figsize=(12,8));

# In[7]:

data[['YEAR', 'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL','AUG', 'SEP', 'OCT',
'NOV', 'DEC']].groupby("YEAR").sum().plot(figsize=(13,8));

# In[8]:

data[['YEAR','Jan-Feb', 'Mar-May', 'Jun-Sep',
'Oct-Dec']].groupby("YEAR").sum().plot(figsize=(13,8));

# In[9]:

data[['SUBDIVISION', 'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL','AUG', 'SEP',
```

```
'OCT', 'NOV','DEC']].groupby("SUBDIVISION").mean() .plot.barh(stacked=True,

figsize=(13,8));

# In[10]:

data[['SUBDIVISION', 'Jan-Feb', 'Mar-May' 'Jun-Sep',

'Oct-Dec']].groupby("SUBDIVISION").sum().plot.barh(stacked=True,figsize=(16,8));


# In[11]:

plt.figure(figsize=(11,4))

sns.heatmap(data[['Jan-Feb','Mar-May','Jun-Sep','Oct-Dec','ANNUAL']].corr(),annot=
True)

plt.show()

# In[12]:

plt.figure(figsize=(11,4))


sns.heatmap(data[['JAN','FEB','MAR','APR','MAY','JUN','JUL','AUG','SEP','OCT','N
OV','DEC','ANNUAL']].corr(),annot=True)

plt.show()

# In[14]:

# seperation of training and testing data

from sklearn.model_selection import train_test_split

from sklearn.metrics import mean_absolute_error

division_data = np.asarray(data[['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',

        'AUG', 'SEP', 'OCT', 'NOV', 'DEC']])

X = None; y = None

for i in range(division_data.shape[1]-3):

    if X is None:

        X = division_data[:, i:i+3]

        y = division_data[:, i+3]
```

```
    else:
        X = np.concatenate((X, division_data[:, i:i+3]), axis=0)
        y = np.concatenate((y, division_data[:, i+3]), axis=0)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1,
random_state=42)
# In[15]:
```

#test 2010

```
temp = data[['SUBDIVISION','JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
'AUG', 'SEP', 'OCT', 'NOV', 'DEC']].loc[data['YEAR'] == 2010]
data_2010 = np.asarray(temp[['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
'AUG', 'SEP', 'OCT', 'NOV', 'DEC']].loc[temp['SUBDIVISION'] == 'TELANGANA'])
X_year_2010 = None; y_year_2010 = None
for i in range(data_2010.shape[1]-3):
    if X_year_2010 is None:
        X_year_2010 = data_2010[:, i:i+3]
        y_year_2010 = data_2010[:, i+3]
    else:
        X_year_2010 = np.concatenate((X_year_2010, data_2010[:, i:i+3]), axis=0)
        y_year_2010 = np.concatenate((y_year_2010, data_2010[:, i+3]), axis=0)
# In[16]:
```

#test 2005

```
temp = data[['SUBDIVISION','JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
'AUG', 'SEP', 'OCT', 'NOV', 'DEC']].loc[data['YEAR'] == 2005]data_2005 =
np.asarray(temp[['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL','AUG', 'SEP',
'OCT', 'NOV', 'DEC']].loc[temp['SUBDIVISION'] == 'TELANGANA'])
X_year_2005 = None; y_year_2005 = None
```

```
for i in range(data_2005.shape[1]-3):
    if X_year_2005 is None:
        X_year_2005 = data_2005[:, i:i+3]
        y_year_2005 = data_2005[:, i+3]
    else:
        X_year_2005 = np.concatenate((X_year_2005, data_2005[:, i:i+3]), axis=0)
        y_year_2005 = np.concatenate((y_year_2005, data_2005[:, i+3]), axis=0)
# In[17]:
#test 2015
temp = data[['SUBDIVISION','JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
'AUG', 'SEP', 'OCT', 'NOV', 'DEC']].loc[data['YEAR'] == 2015]

data_2015 = np.asarray(temp[['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
'AUG', 'SEP', 'OCT', 'NOV', 'DEC']].loc[temp['SUBDIVISION'] == 'TELANGANA'])

X_year_2015 = None; y_year_2015 = None
for i in range(data_2015.shape[1]-3):
    if X_year_2015 is None:
        X_year_2015 = data_2015[:, i:i+3]
        y_year_2015 = data_2015[:, i+3]
    else:
        X_year_2015 = np.concatenate((X_year_2015, data_2015[:, i:i+3]), axis=0)
        y_year_2015 = np.concatenate((y_year_2015, data_2015[:, i+3]), axis=0)
# In[18]:
from sklearn import linear_model
# linear model
reg = linear_model.ElasticNet(alpha=0.5)
```

```
reg.fit(X_train, y_train)

y_pred = reg.predict(X_test)

print ("LR=", mean_absolute_error(y_test, y_pred))

# In[19]:

#2005

y_year_pred_2005 = reg.predict(X_year_2005)

#2010

y_year_pred_2010 = reg.predict(X_year_2010)

y_year_pred_2015 = reg.predict(X_year_2015)

print ("MEAN 2005")

print (np.mean(y_year_2005),np.mean(y_year_pred_2005))

print ("Standard deviation 2005")

print (np.sqrt(np.var(y_year_2005)),np.sqrt(np.var(y_year_pred_2005)))

print ("MEAN 2010")

print (np.mean(y_year_2010),np.mean(y_year_pred_2010))

print ("Standard deviation 2010")

print (np.sqrt(np.var(y_year_2010)),np.sqrt(np.var(y_year_pred_2010)))

print ("MEAN 2015")

print (np.mean(y_year_2015),np.mean(y_year_pred_2015))

print ("Standard deviation 2015")

print (np.sqrt(np.var(y_year_2015)),np.sqrt(np.var(y_year_pred_2015)))

plot_graphs(y_year_2005,y_year_pred_2005,"Year-2005")

plot_graphs(y_year_2010,y_year_pred_2010,"Year-2010")

plot_graphs(y_year_2015,y_year_pred_2015,"Year-2015")

from sklearn.svm import SVR
```

 # SVM model

```
clf = SVR(gamma='auto', C=0.1, epsilon=0.2)

clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)

print ("SVR =", mean_absolute_error(y_test, y_pred))

#2005

y_year_pred_2005 = clf.predict(X_year_2005)

#2010

y_year_pred_2010 = clf.predict(X_year_2010)

#2015

y_year_pred_2015 = clf.predict(X_year_2015)

print ("MEAN 2005")

print (np.mean(y_year_2005),np.mean(y_year_pred_2005))

print ("Standard deviation 2005")

print (np.sqrt(np.var(y_year_2005)),np.sqrt(np.var(y_year_pred_2005)))

print ("MEAN 2010")

print (np.mean(y_year_2010),np.mean(y_year_pred_2010))

print ("Standard deviation 2010")

print (np.sqrt(np.var(y_year_2010)),np.sqrt(np.var(y_year_pred_2010)))

print ("MEAN 2015")

print (np.mean(y_year_2015),np.mean(y_year_pred_2015))

print ("Standard deviation 2015")

print (np.sqrt(np.var(y_year_2015)),np.sqrt(np.var(y_year_pred_2015)))

plot_graphs(y_year_2005,y_year_pred_2005,"Year-2005")

plot_graphs(y_year_2010,y_year_pred_2010,"Year-2010")
```

```
plot_graphs(y_year_2015,y_year_pred_2015,"Year-2015")

from keras.models import Model

from keras.layers import Dense, Input, Conv1D, Flatten

# NN model

inputs = Input(shape=(3,1))

x = Conv1D(64, 2, padding='same', activation='elu')(inputs)

x = Conv1D(128, 2, padding='same', activation='elu')(x)

x = Flatten()(x)

x = Dense(128, activation='elu')(x)

x = Dense(64, activation='elu')(x)

x = Dense(32, activation='elu')(x)

x = Dense(1, activation='linear')(x)

model = Model(inputs=[inputs], outputs=[x])

model.compile(loss='mean_squared_error', optimizer='adamax', metrics=['mae'])

model.summary()

model.fit(x=np.expand_dims(X_train, axis=2), y=y_train, batch_size=64, epochs=10,
verbose=1, validation_split=0.1, shuffle=True)

y_pred = model.predict(np.expand_dims(X_test, axis=2))

print ("ANN=",mean_absolute_error(y_test, y_pred))

#2005

y_year_pred_2005 = model.predict(X_year_2005)

#2010

y_year_pred_2010 = model.predict(X_year_2010)

#2015

y_year_pred_2015 = model.predict(X_year_2015)

print ("MEAN 2005")
```

```
print (np.mean(y_year_2005),np.mean(y_year_pred_2005))
print ("Standard deviation 2005")
print (np.sqrt(np.var(y_year_2005)),np.sqrt(np.var(y_year_pred_2005)))
print ("MEAN 2010")
print (np.mean(y_year_2010),np.mean(y_year_pred_2010))
print ("Standard deviation 2010")
print (np.sqrt(np.var(y_year_2010)),np.sqrt(np.var(y_year_pred_2010)))
print ("MEAN 2015")
print (np.mean(y_year_2015),np.mean(y_year_pred_2015))
print ("Standard deviation 2015")
print (np.sqrt(np.var(y_year_2015)),np.sqrt(np.var(y_year_pred_2015))
plot_graphs(y_year_2005,y_year_pred_2005,"Year-2005")
plot_graphs(y_year_2010,y_year_pred_2010,"Year-2010")
plot_graphs(y_year_2015,y_year_pred_2015,"Year-2015")
```

6.TESTING

Testing is a process, which reveals errors in the program. Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. The increasing visibility of software as a system element and attendant costs associated with a software failure are motivating factors for we planned, through testing. Testing is the process of executing a program with the intent of finding an error. The design of tests for software and other engineered products can be as challenging as the initial design of the product itself It is the major quality measure employed during software development. During testing, the program is executed -with a set of test cases and the output of the program for the test cases is evaluated to determine if the program is performing as it is expected to perform.A test case contains all the information necessary to verify some particular functionality of the software.

Purpose - Describe the features of the software to be tested, and the particular behavior being verified by this test.

Requirement Traceability - A cross reference to the numbers of the requirements (in the system specification) which are being verified in this test.

Setup - Describe all the steps necessary to setup the software environment necessary to carry out the test.

Test data - Write the actual input data to be provided and the expected output for your actual working product. You must provide the actual input data values, not just a description.

Test Plan - A test plan outlines the strategy that will be used to test an application, the resources that will be used, the test environment in which testing will be performed, and the limitations of the testing and the schedule of testing activities.

6.1 Types Testing

The development of software systems involves of a series of production activities where opportunities for injection of human falibilities are enormous.

Errors may begin to occur at the very inception of the process where the objectives may be erroneously or imperfectly specified, as well as in later design and development stages. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects).In order to make sure that the system does not have errors, the different levels of testing strategies that are applied at differing phases of software development are:

6.1.1 Unit Testing

Unit Testing is done on individual modules as they are completed and become executable. It is confined only to the designer's requirements.

6.1.2 Functional Testing

This is a type of black-box testing that is based on the specifications of the software that is to be tested. The application is tested by providing input and then the results are examined that need to conform to the functionality it was intended for. Functional testing of a software is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.

6.1.3 Black Box Testing

In this strategy some test cases are generated as input conditions that fully execute all functional requirements for the program. This testing has been uses to find errors in the following categories Incorrect or missing functions

- Interface errors
- Errors in data structure or external database access
- Performance errors

6.1.4 White Box Testing

In this the test cases are generated on the logic of each module by drawing flow graphs of that module and logical decisions are tested on all the cases.
It has been uses to generate the test cases in the following cases:

- Execute all logical decisions on their true and false Sides.
- Guarantee that all independent paths have been Executed.

6.1.5 Regression Testing

Whenever a change in a software application is made, it is quite possible that other areas within the application have been affected by this change. Regression testing is performed to verify that a fixed bug hasn't resulted in another functionality or business rule violation.

6.1.6 Acceptance Testing

This is arguably the most important type of testing, as it is conducted by the Quality Assurance Team who will gauge whether the application meets the intended specifications and satisfies the client's requirement.

6.1.7 Performance Testing

It is mostly used to identify any bottlenecks or performance issues rather than finding bugs in a software. There are different causes that contribute in lowering the performance of a software:

- Network delay
- Client-side processing
- Database transaction processing
- Load balancing between servers
- Data rendering
- Load Testing

It is a process of testing the behaviour of a software by applying maximum load in terms of software accessing and manipulating large input data.

6.1.8 Usability Testing

Usability testing is a black-box technique and is used to identify any error(s) and improvements in the software. Usability can be defined in terms of five factors, i.e. efficiency of use, learn-ability, memory-ability, errors/safety, and satisfaction.

6.1.9 Security Testing

Security testing involves testing a software in order to identify any flaws and gaps from security and vulnerability point of view. Listed below are the main aspects that security testing should ensure:

- Confidentiality
- Integrity
- Authentication
- Availability
- Integrating Testing

Integration testing ensures that software and subsystems work together a whole. It tests the interface of all the modules to make sure that the modules behave properly when integrated together.

6.1.10 System Testing

System testing involves in-house testing of the entire system before delivery to the user. Its aim is to satisfy the user the system meets all requirements of the client's specifications.

6.1.11 Acceptance Testing

It is a pre-delivery testing in which entire system is tested at client's site on real world data to find errors.

6.2 Test Cases

Table 6.1 : Test Cases

No.	Test Case	Input	Actual Output	Expected Output	Is actual out -put same as expected Output?
1.	Process without loading the data	nothing	It will show error	It will show error	Yes

2.	Load a Csv file	.csv file	Dataset is loaded		Yes
3.	Without import packages	nothing	Error	It will show error	Yes

6.3 Test Approach

Testing can be done in two ways :

6.3.1 Bottom up Approach

Testing can be performed starting from smallest and lowest level modules and proceeding one at a time. When bottom level modules are tested attention turns to those on the next level that use the lower level ones they are tested individually and then linked with the previously examined lower level modules.

6.3.2 Top down Approach

This type of testing starts from upper level modules. Since the detailed activities usually performed in the lower level routines are not provided stubs are written. A stub is a module shell called by upper level module and that when reached properly will return a message to the calling module indicating that proper interaction occurred.

7. RESULTS

These are the following results which we get during the execution of the process which are mentioned in level by level detail.

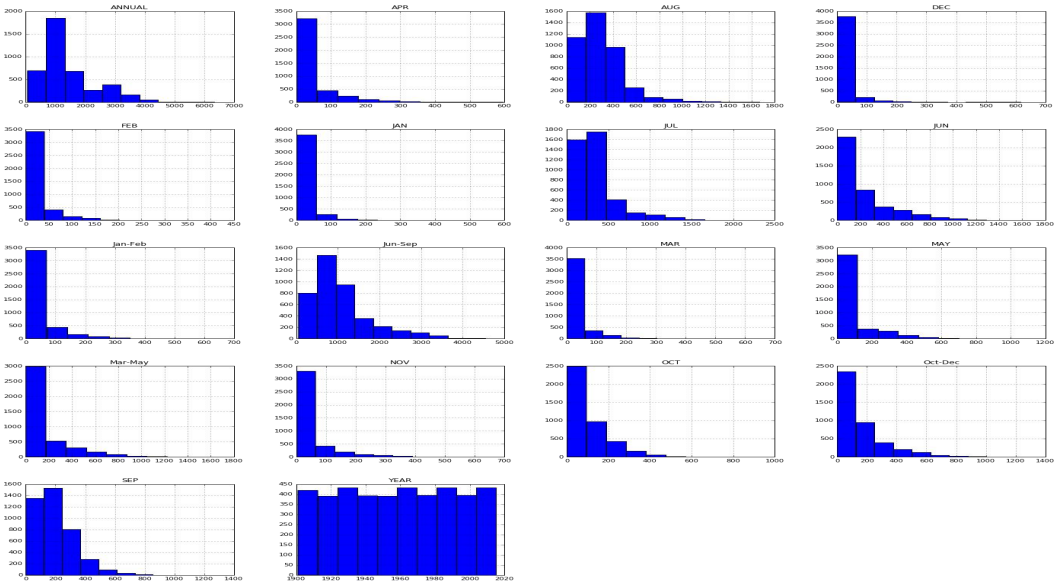


Figure 7.1 : Histogram representation of rainfall

Above histograms show the distribution of rainfall over months.Observed increase in amount of rainfall over months July, August, September.

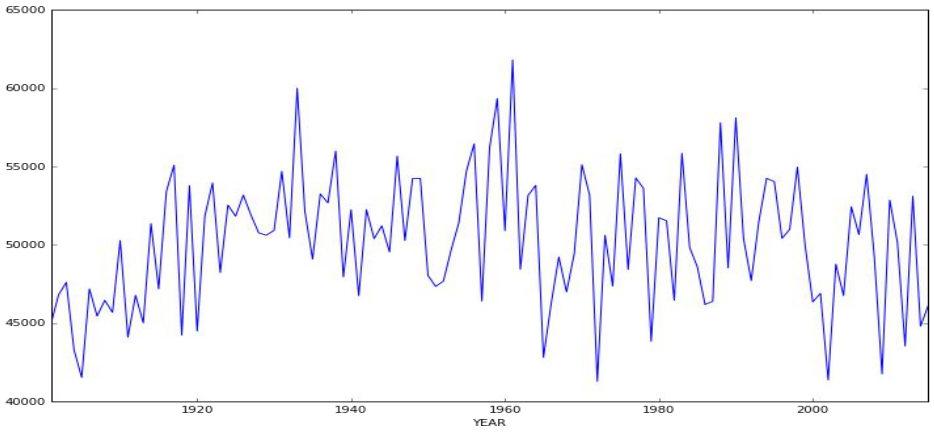


Figure 7.2 : Plot graph

Above Plot graph shows distribution of rainfall over years. Observed high amount of rainfall in 1950s.

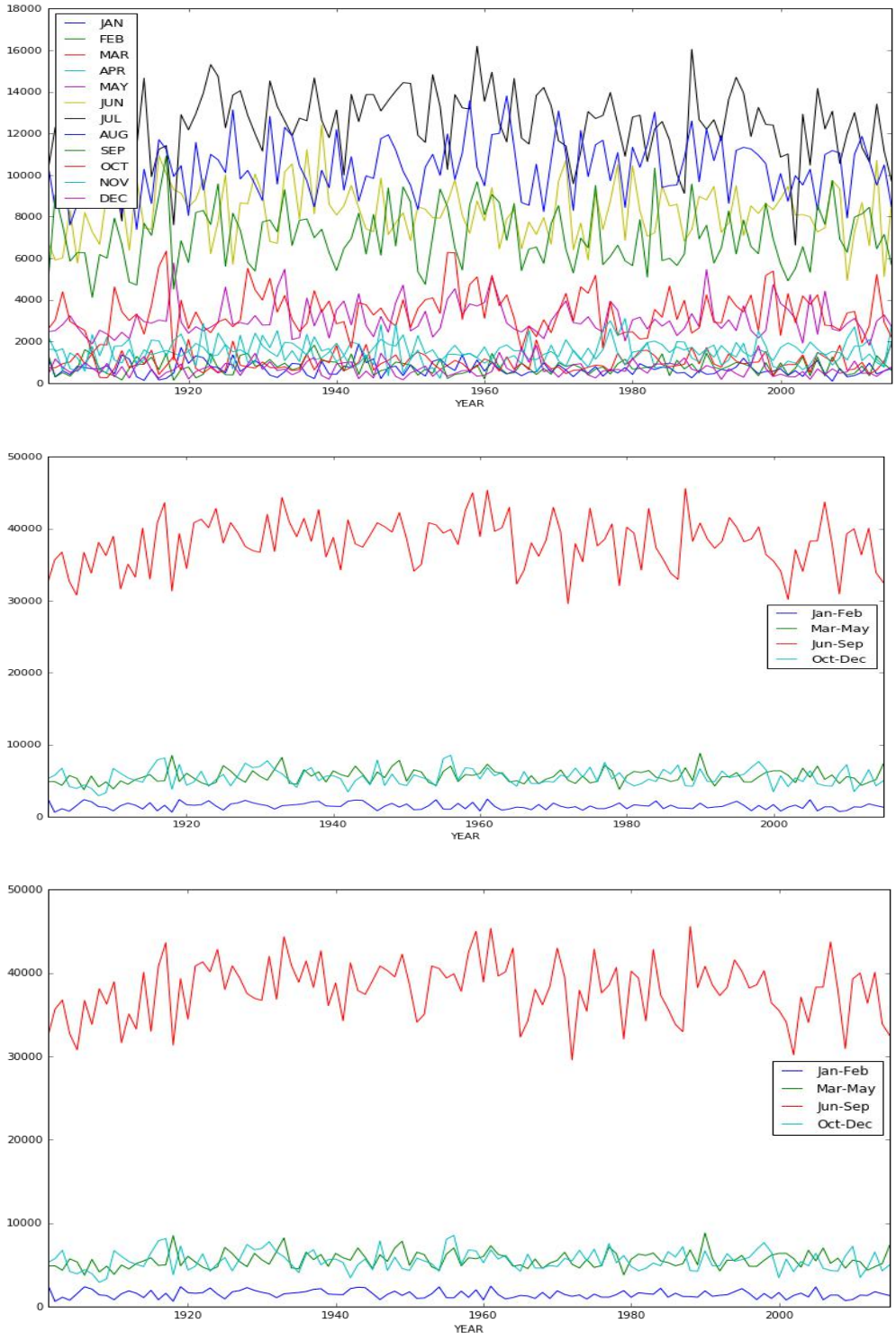
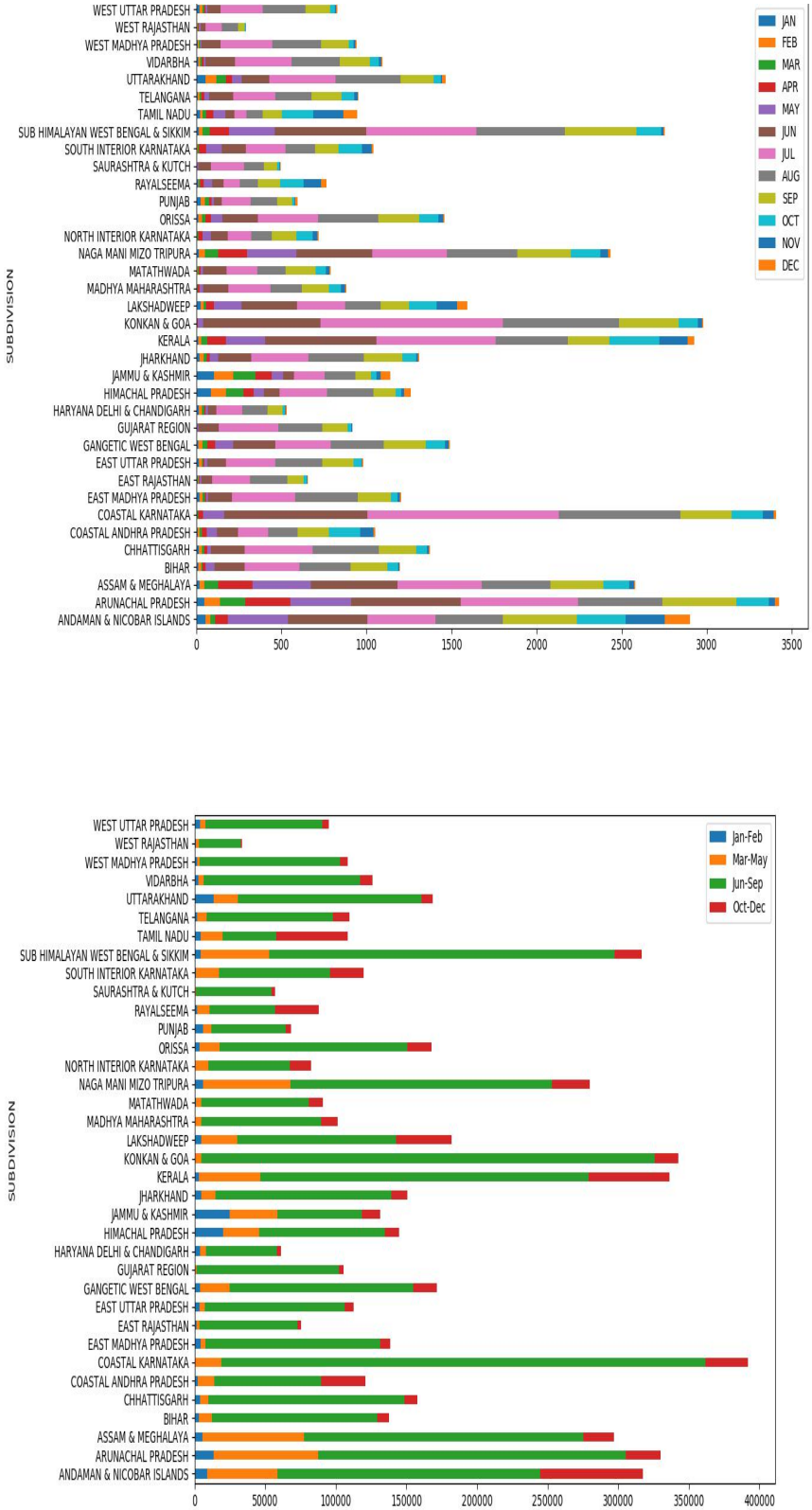


Figure 7.3 : Distribution of rainfall over months

The graphs clearly shows that amount of rainfall in high in the months july, aug, sep which is monsoon season in India.



Figu 7.4 : Horizontal bar graph

Above two graphs shows that the amount of rainfall is reasonably good in the months of march, april, may in eastern India.

Figure 7.5 : Heap graph

Heat Map shows the co-relation (dependency) between the amounts of rainfall over months. From above it is clear that if amount of rainfall is high in the months of July, august, September then the amount of rainfall will be high annually. It is also observed that if amount of rainfall in good in the months of October, November, December then the rainfall is going to b good in the overall year.

Algorithm	MAE
Linear Regression	94.94821727619338
SVR	127.74073860203839
Artificial neural nets	85.2648713528865

Figure 7.6 : Mean Absolute Error

In statistics, mean absolute error is a measure of difference between two continuous variables.The mean absolute error is given by:

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n}$$

```
MEAN 2005
121.2111111111111 134.68699821349804
Standard deviation 2005
123.77066107608005 90.86310230416441
MEAN 2010
139.93333333333334 144.80501326515915
Standard deviation 2010
135.71320250194282 95.9493136360173
MEAN 2015
88.52222222222223 119.64752006738826
Standard deviation 2015
86.62446123324875 62.36355370163376
```

Figure 7.7 : Prediction

8. CONCLUSION

The Mean absolute error over the long-term rainfall data gives a lowest value for ANN, hence it is concluded that ANN is the most reliable model for rainfall forecast prediction.The amount of rainfall is reasonably good in the months of march, april, may in eastern India. Observed high amount of rainfall in 1950s. if amount of rainfall is high in the months of July, august, September then the amount of rainfall will be high annually. In the current scenario forecasting the rainfall is measured to be an important and thought provoking task, as it's closely associated with the agriculture, economy and human life. Accuracy of a rainfall forecasting is importance for countries like India whose economy majorly depends on agriculture. The rainfall prediction is to predict the state of current weather condition. The weather is dynamic in nature, statistical techniques are unsuccessful to provide the decent accuracy of rainfall. So Machine Learning Techniques and their Algorithms are being used in getting the better accuracy of rainfall.

8.1 Future Work

Though some significant improvement in accuracy and reliability of NWP product has been driven by adopting MME approach, however, limitations remain, particularly in the prediction of intensity and mesoscale rainfall features causing inland flooding. During recent years, Ensemble Prediction System (EPS) has emerged as a powerful tool for improving medium range weather forecasts. In the EPS, single model is used with multiple sets of initial conditions (Brooks and Doswell 2012) to obtain the final forecast. While Singular Vector and Bred Vector (BV) methods are still widely used in generating initial perturbations, Ensemble Transform of BV, Ensemble Transform Kalman Filter and Ensemble Data Assimilation are also implemented in various centres. Currently, 10 global centres operate EPS for medium range forecasts and they exchange model outputs at the native resolution among themselves.

9. REFERENCES

• Brooks H E and Doswell C A 1999 New technology and numerical weather prediction; Weather 48 173–177. Krishnamurti T N, Kishtawal C M, Larow T, Bachiochi D, Zhang Z, Willford E C, Gadgil S and Surendran S 1999 Improved weather and seasonal climate forecasts from multimodel super ensemble; Science 285 1548–1550.

• Krishnamurti T N, Kishtawal C M, Shin D W and Willford E C 2000 Improving tropical precipitation forecasts from a multi analysis super ensemble; J. Climate 13 4217–4227.

• Mishra A K and Krishnamurti T N 2007 Current status of multi-model super-ensemble operational NWP forecast of the Indian summer monsoon; J. Earth Syst. Sci. 116(5) 1–16.

• Roy Bhowmik S K and Durai V R 2015 Multimodel ensemble forecasting of rainfall over Indian monsoon region; Atmosfera 21(3) 225–239. Roy Bhowmik S K and Durai V R 2010 Application of multimodel ensemble techniques for real-time district level forecasts in short range time scale over Indian region; Meteor. Atmos. Phys.

• Rajeevan M and Bhate J 2010 A high resolution daily gridded rainfall dataset (1971–2005) for meso-scale meteorological studies; Curr. Sci. 96(4) 558–562. Roy Bhowmik S K and Das Ananda K 2007 Rainfall analysis for Indian monsoon region using the merged rain gauge observations and satellite estimates: Evaluation of monsoon rainfall features; J. Earth Syst. Sci.