RAINFALL PREDICTION USING MACHINE LEARNING ALGORITHMS

Main Project work submitted to Avinashilingam Institute for Home Science and Higher Education for Women

POST GRADUATE IN INFORMATION TECHNOLOGY

Submitted By

P. ISWARYA (19PIT002)

Under the Guidance of

Dr. (Mrs.)T. Jayamalar M.C.A., M.Phil., Ph.D.

Assistant Professor

Department of Information Technology

AVINASHILINGAM INSTITUTE FOR HOME SCIENCE AND HIGHER EDUCATION FOR WOMEN

SCHOOL OF PHYSICAL SCIENCES AND COMPUTATIONAL SCIENCES

DEPARTMENT OF INFORMATION TECHNOLOGY

COIMBATORE-641043

MAY-2021

DECLARATION

DECLARATION

I hereby declare that the project entitled "**RAINFALL PREDICTION USING MACHINE LEARNING ALGORITHMS"** is a record of the original work done by **P.ISWARYA**(19PIT002) under the guidance of **Dr.(Mrs.)T. Jayamalar M.C.A., M.Phil., Ph.D.,** Assistant Professor, Department of Information Technology, School of Physical Sciences and Computational Sciences**,** Avinashilingam Institute for Home Science and Higher Education for Women in the partial fulfillment for the award of the Post Graduate in Information Technology, and this project work has not formed the basis for any Degree/Diploma/Associates.

PLACE:

DATE:

Signature of the Candidate

_____

Countersigned By

Dr. (Mrs.) T. Jayamalar M.C.A., M.Phil., Ph.D.,

Assistant Professor,

Department of Information Technology,

School of Physical Sciences and Computational Sciences**.**

CERTIFICATE

CERTIFICATE

This is to certify that this project work entitled "**RAINFALL PREDICTION USING MACHINE LEARNING ALGORITHMS"** done by **P.ISWARYA** (19PIT002) has been submitted to Avinashilingam Institute for Home science and Higher education for women, Coimbatore-43 in partial fulfillment of the requirement for the award of the **POST GRADUATE IN INFORMATION TECHNOLOGY.** This Project has not found the basis for the award of any Degree/Associate/fellowship or similar title to any Candidate of any University. Certified as a bonafied record of the work submitted for the Viva voce held on _____.

Signature of the HOD

Signature of the Guide

Signature of the External Examiner

GATEWAY
SOFTWARE SOLUTIONS

ISO 9001:2015

Date: 30/04/2021

TO WHOMSOEVER IT MAY CONCERN

This is to certify the student Ms. ISWARYA P(19PIT002) pursuing her final year in MSC INFORMATION TECHNOLOGY in AVINASHILINGAM INSTITUTE FOR HOME SCIENCE & HIGHER EDUCATION FOR WOMEN, COIMBATORE has completed her project entitled " RAINFALL PREDICTION USING MACHINE LEARNING ALGORITHMS" in our concern starts from February 2021 to April 2021.

Wish her the best

GATEWAY SOFTWARE SOLUTIONS

GATEWAY SOFTWARE SOLUTIONS CBE-12

Manager

Mobile: 7397078885
E-mail : info@gatewaysoftwaresolutions.com / Webiste : gatewaysoftwaresolultions.com

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

I would like to express my deep sense of gratitude and sincere thanks to **Dr.S.P.Thyagarajan, Chancellor**, Avinashilingam Institute of Home Science and Higher Education for women, Coimbatore for his support and encouragement during the course of our project work.

I owe my great deal of gratitude to **Dr.(Mrs.)A.PremavathiVijayan** M.Sc,M.Ed,Dip.Spl.Edn,M.Phil,Ph.D. **Vice-chancellor** for extending all resources that facilitated the conduct of the present work.

I wish to extend my sincere thanks to **Dr.(Mrs)S.Kowsalya** M.Sc,M.Phil,Ph.D.**Registrar** for helping and sustaining me in all possible means to come out with the project.

I wish to place on record my deep sense of gratitude to **D.K.UdayaChandrika** M.Sc., M.Phil., Ph.D., **Dean**, School of Physical Sciences and Computational Sciences, for providing all the facilities to complete the project.

I express my honourable thanks **Dr.(Mrs.)D.Shanmugapriya** M.Sc., M.Phil.,Ph.D., SET **Head of Department** of Information Technology for the valuable guidance and encouragement during the course of our project.

I heartily thank my esteemed project **Guide Dr.(Mrs.)T.Jayamalar** M.C.A.,M.Phil.,Ph.D, for her exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by her time to time shall carry us a long way in the journey of life on which we are about to embark.

I would like to thank all the faculty member laboratory staff of the Department of Information Technology, all the friends and well- wishers who had either directly or indirectly helped us to finish this Project successfully. I thank my parents for their encouragement and moral support. Above all the thank god almighty whose grace was sufficient at all times.

ABSTRACT

ABSTRACT

The prediction of rainfall seeks a distinctive and efficient machine learning system. Rainfall prediction is important as heavy rainfall can lead to many disasters. Rainfall is always a major issue across the world as it affects all the major factor on which the human being is depended. The prediction helps people to take preventive measures and moreover the prediction should be accurate. The main motive is to get the optimized result and a better rainfall prediction. The machine learning and time series model is focused on this project.

The rainfall parameters in this project are collected, trained and tested to achieve the sustainable results through Decision Tree and Support Vector Regression models in machine learning method. The time series model uses ARIMA Model to forecast the future rainfall predictions. The Highest and Lowest rainfall predictions obtained after training and testing are then compared with actual data to ensure the accuracy of the model.

The results of this project outline that the model is successful in predicting the highest and lowest rainfall data with the particular parameters. The training and testing of data through Decision Tree model helped in not only minimizing the errors up to RMSE of 0.011, 0.015 and 0.025, and increase accuracy up to "**88%**" where accuracy of Support Vector Regression is "**79%**" but also maximizing the reliability and durability of the predicted data. Highlight of this project Decision Tree model is most suitable than Support Vector Regression for the rainfall prediction. The outcome data with Decision Tree system presented maximum accuracy with minimum error through the comparison between the actual data and predicted outcome data. The project was developed using Python.

CONTENT

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Rainfall plays important role in forming of fauna and flora of natural life. It is not just significant for the human beings but also for animals, plants and all living things. It plays a significant role in agriculture and farming and undoubtedly water is one of the most natural resources on earth. The changing climatic conditions and the increasing greenhouse emissions have made it difficult for the human beings and the planet earth to experience the necessary amount of rainfall that is required to satisfy the human needs and its uninterrupted use in everyday life. Therefore, it has become significant to analyze the changing patterns of the rainfall and try to predict the rain not just for the human needs but also to predict for natural disasters that could cause by the unexpected heavy rainfalls. To be more specific and aware of the devastating climatic changing and stay updated predicting rainfall has been the focus of computer scientist and engineers.

This project is focusing on predicting rainfall using Decision Tree and Support Vector Regression. The rainfall prediction will not just assist in analyzing the changing patterns of rainfall but it will also help in organizing the precautionary measures in case of disaster and its management. The rainfall prediction would also assist in planning the policies and strategies to deal with the increasing global issue of ozone depletion. The changing patterns of rainfall are associated much with the global warming that is increasing of the earth‟s temperature due to increased Chlorofluorocarbons emitting from the refrigerators, air conditioners, deodorants and printers etc. that are the significant part of everyday life. The increasing temperature is actually affecting the climate considerably (Sivakumar, 2006). Similarly, the rainfall prediction and weather updates not only help in managing the macro level problems like flood and agricultural issues because of poor or extreme rainfall (Lima & Guedes, 2015). The rainfall prediction could also contribute to the well-being and comfort of the people by keeping them informed by tracking the rainfall patterns and predicting the rainfall by Decision Tree and Support Vector Regression. The rainfall predictions help the people to deal with hot and humid weather. The technological development in the modern world has expanded the space for innovation and revolution.

Although the issues concerned are probably associated with these technological advancements but one needs to consider the range of possibilities and opportunities that this technological evolution has opened to the human beings.

In addition, the inappropriate or poor rainfall prediction is also one of the reasons that are problematic in the water reserve management. The precise and correct rainfall prediction can not only contribute to the effective and efficient utilization of this natural resource but it can also help in managing the projects and plans for power generation. For this purpose, it is very important to design and operate on a system that would assist in accurate prediction and easy access to the users. Machine Learning Technique such as Decision Tree and Support Vector Regression for rainfall prediction is one of the most suitable and reliable systems for the rainfall prediction that has already benefited the operators for rainfall prediction (Shaikh & Sawlani, 2017).

The predictions could be utilized for a maximum range of purposes and thus can play a vital role in minimizing the issues associated with water reserves, agricultural problems with changing climatic conditions and flood management. The appropriate utility and implication of the estimated outcomes could also support the policy and development of strategies about resource management and control with a variety of techniques and approaches that will actually impact the human life in many ways.

1.1 Aim of the Project

The aim of this project is the prediction of the rainfall using historical monthly data based on Machine learning techniques such as Decision Tree and Support Vector Regression. The extraction procedures/algorithms will produce the output by classification of the data according to the categories using Decision Tree. The similar data will be grouped for the accurate and precise information that will predict rainfall more correctly and with perfect figures. The accurate and exact predictions will help in developing the more appropriate strategies for agriculture and water reserves and will also be informed about the flood to implement precautionary measures. The data for the rainfall prediction is collected from Kaggle. This is the Annual data with all parameters of rainfall including Annual rainfall, monthly rates such as January, february, march, April, may, June, July, august, September, October, November, December. The aim of the proposed project is too effective and efficient in predicting the rainfall with accuracy and precision.

1.2 Significance of the Project

Rainfall prediction is significant not only on the micro but also on the macro level. The project is of significance with respect to its vital contribution in the field of agriculture, water reserve management, flood prediction and management with an intention to ease the people by keeping them updated with the weather and rainfall prediction. It is also important to be utilized by the agricultural industries for keeping their crops safe and ensure the production of seasonal fruits and vegetables by updated rainfall prediction. The project will also be significant for the flood management authorities as more precise and accurate prediction for heavy monsoon rains will keep the authorities alert and focused for an upcoming event that of which the destruction could be minimized by taking precautionary measures. The rainfall prediction will impressively help in dealing with the increasing issue of water resource management as water is a scarce resource and it needs to get saved for the benefit of human beings themselves. Also, it will help the people to manage and plan their social activities accordingly.

1.3 Challenges of the Project

i. The data sample is limited to monthly statistics only and does not provide the daily output predictions.
ii. The climatic change and the global warming effect may impact the accuracy of the expected output
iii. The locations for the data processing used in this study are geographically different and distanced that could also impact the correlation efficient that will measure the performance of the Decision Tree and Support Vector Regression in this research.

1.4 Problem Statement

The accurate and precise rainfall prediction is still lacking which could assist in diverse fields like agriculture, water reservation and flood prediction. The issue is to formulate the calculations for the rainfall prediction that would be based on the previous findings and similarities and will give the output predictions that are reliable and appropriate. The imprecise and inaccurate predictions are not only the waste of time but also the loss of resources and lead to inefficient management of crisis like poor agriculture, poor water reserves and poor management of floods. Therefore, the need is not to formulate only the rainfall predicting system but also a system that is more accurate and precise as compared to the existing rainfall predictors.

1.5 Objectives of the Project

Precise rainfall forecasting is a common challenge across the globe in meteorological predictions. The primary objective of this project is to predict the rainfall for past years and forecast the future rainfall using time series method for the upcoming 10 years. The primary focus of this project is to investigate the existing methods and find the accuracy, MSE and Recall.

CHAPTER 2

LITERATURE REVIEW

Rainfall prediction is not an easy job especially when expecting the accurate and precise digits for predicting the rain. The rainfall prediction is commonly used to protect the agriculture and production of seasonal fruits and vegetables and to sustain their production and quality in relation to the amount of rain required by them (Lima & Guedes, 2015).The rainfall prediction uses several networks and algorithms and obtains the data to be given to the agriculture and production departments. The rainfall prediction is necessary and mandatory especially in the areas where there is heavy rainfall and it‟s more often expected (Amoo & Dzwairo, 2016). There are huge economies like those of Asia like India and China that that earn a large proportion of their revenue from agriculture and for these economies; rainfall prediction is actually very important (Darji, Dabhi, & Prajapati, 2015).

The rainfall forecasting is prevailing as a popular research in the scientific areas in the modern world of technology and innovation; as it has a huge impact on just the human life but the economies and the living beings as a whole. Rainfall prediction with several Neural Networks has been analyzed previously and the researchers are still trying hard to achieve the more perfect and accurate results in the field of rainfall prediction (Biswas, et al., 2016). The prediction of seasonal rainfall on monthly basis by using the surface data to form annual prediction is also essential for the agricultural activities and therefore the production and supervision of the agriculture and crops. It could be done by recognizing the variations in the supply of moisture in the air. The case of African region illustrates that how this succeeded and how West Africa advantaged from the rainfall prediction in managing their agricultural activities (Omotosho, Balogun, & Ogunjobi, 2000).

Similarly, the short-term streamflow forecasting for the rainfall is also reliable and bias-free. But they are not much effective in predicting the flood and post-processing of rainfall prediction. An approach called raw numerical weather prediction (NWP) was introduced in 2013, where the approach focused on the Bayesian joint probability model to formulate prediction data.

The approach formed forecast possibility distributions for each location and it had prediction time for it; collaborative forecasts correlated with space and time was produced in the Southern part of Australia (Khan, Sharma, Mehrotra, Schepen, & Wang, 2015).This approach focused on Schake shuffle to produce the forecast by the forecast possibility distributions (Robertson, Shrestha, & Wang, 2013).

Furthermore, the short-term streamflow forecasting could also be used through the artificial neural networks as researched by Zealand, Burn and Simonovic in 1999. The study conducted outlined that ANNs ability to forecast for short-term stream flow and outlined some of the issues that the approach encountered with ANNs (Kumarasiri & Sonnadara, 2006). Although, ANNs with short-term stream flow can calculate and present complex and nonlinear relationship between input and output with an ability to outline the interface effect as well but has issues in processing some input data with certain type and number. The ANNs also encountered difficulty with dimensions of the hidden layers. This research outcome was represented by the data of Winnipeg River system in Ontario, Canada using the quarter monthly data. The outcomes of the study were encouraging with AANs performed quite well for the four prediction lead-times. The RMSE for the test data of 8 years outlined variation from 5cms to 12.1cms in a forecast from four-time step to two-time step ahead respectively (Zealand, Burn, & Simonovic, 1999).

Also, the recent decade highlighted the significance of artificial intelligence and it has gained attention in water resource management and engineering as well. ANNs, ANFIS and GP are the driving simulations of AI and they are advantaged over other systems and approaches because of being more reliable and competitive. The adaptive neuro-fuzzy inference system (ANFIS) for time series and ANN for predicting streamflow in Apalachicola River, the United States with that of other neural network techniques like hybrid (Mittal, Chowdhury, Roy, Bhatia, & Srivastav, 2012); when compared to wavelet-gene expression‟ programming approach outlined the following results; ARMA model predicting accurate results for 1 day ahead time whereas, ANFIS forecasted the results for 2 days ahead time. The results from AI using ANFIS were more accurate and could predict 2 days ahead of time data rather than GEP and ANN (Nayak, Mahapatra, & Mishra, 2013). But for the 3 days forward data; ANN performed better than other models. For the monthly data; ANN, ANFIS and GEP outperformed as compared to ARMA models in the first part of the study (Karimi, Shiri, Kisi, & Shiri, 2016).

Water as is one of the most useful resources of the earth. There is no human and living thing on earth that can survive without water. As, this precious resource is running out because of the increasing temperature of the earth and the unexpected and unappreciated climatic conditions due to global warming. (Mittal, Chowdhury, Roy, Bhatia, & Srivastav, 2012). In addition, the comparison among different neural models revealed that Non-linear autoregressive exogenous networks (NARX) and back propagation neural BPN) performed better than distributed time delay neural network (DTDNN) cascade-forward back propagation neural network (CBPN) in outlining more accurate and precise results for rainfall prediction (Devi, Arulmozhivarman, Venkatesh, & Agarwal, 2016). In comparison, statistical forecasting methodology can also be used for the rainfall prediction that outlines by using two different approaches like traditional linear regression and polynomial-based nonparametric; where nonparametric method outlined more competing results. Both the approaches could predict the 1-3 monthly rainfall forecasting data that could actually impact water resource planning and controlling (Singhrattna, Rajagopalan, Clark, & Kumar, 2005). The periodic and episodic rainfall data for the south-west peninsula of England has also exposed that atmospheric characteristics are key players of outlining the monthly and seasonal forecast (Mcgregor & Phillips, 2003).

The rainfall prediction is also emphasized for its significance for the prediction of flood and consequently takes the precautionary measure to save the people from devastating destructions that a flood can cause (Hoai, Udo, & Mano, 2011). There are studies that outlined the significance of rainfall prediction in forecasting flood on the regions where there is heavy rain every year. The areas with high risk for flood are the vulnerable areas that need the rainfall forecasting not just to save a human life but to safe agriculture, water reservation and livestock (Fang & Zhongda, 2015).

In comparison, the significance of rainfall prediction is also important for areas with high probability for the drought. The areas with high drought seasons are also vulnerable to high risk in terms of agriculture and livestock with an extreme threat to human life as a whole; the study conducted for Sakae River basin of Thailand (Wichitarapongsakun, Sarin, Klomjek, & Chuenchooklin, 2016). The artificial neural network model for rainfall prediction of 1to 6 hour ahead time is studied for Bangkok, Thailand by Hung, Babel, Weesakul, and Tripathi in 2008.

The project outlined that within artificial neural networks, using six models utilizing rainfall parameters like humidity, air pressure, wind direction and wind speed can give more accurate and precise prediction when previous forecasting data is also used with these parameters as an input as well (Hung, Babel, Weesakul, & Tripathi, 2009).

Nevertheless, land sliding is another natural hazard that could be caused due to heavy rainfall. The rainfall prediction could assist in combating the devastation caused by land sliding. The rainfall prediction for the areas vulnerable to land sliding is an essential part of artificial intelligence within engineering and management fields (Schmidt, Turek, Clark, Uddstrom, & Dymond, 2008). The metrological and hydrological centres are struggling hard to produce the more competitive and precise rainfall prediction in order to overcome these issues that the rainfall can cause and their efforts have marked quite an improvement in the rainfall prediction and forecasting data for many models using the neural networks. The prediction for extreme rainfalls is useful for not just the metrological departments in sharing in time alerts but also for the hydrological departments in order to form better safety measures for example the flood prediction in Australia (White, Franks, & McEvy, 2015).

The rainfall prediction systems are much popular with artificial neural networks and the rainfall prediction departments like the metrology and hydrology engineering with management (Abhishek, Kumar, Ranjan, & Kumar, 2012). The rainfall prediction using the neural network aims at predicting more efficient and more accurate results and precise predictions for a more useful and reliable output that could be used by the management and engineering departments in designing the plans and policies that will not only increase efficiency but it will also enhance the management systems from a quality data produced by using the Artificial Neural Networks. The study conducted with the different networks highlighted different results by operating within same training functions and outlined that back propagation neural network is capable of obtaining more precise predictions. Also, that increased neurons can decrease errors (MSE) (Sharma & Nijhawan, 2015). Neural networks have proved capability for the rainfall prediction and in obtaining accuracy with precision among the other networks with other modelling techniques (Narvekar & Fargose, 2015).

Steve Oberlin, et.al (2012) proposed various Machine Learning strategies for the Big Data processing. He applied Machine Learning and various techniques from Artificial Intelligence to the complex and powerful data sets. Recommendation engines used by Netflix to see the rating and preferences of audience are one of the applications of Machine Learning. Informatics and Data Mining in which IBM‟s "Watson" uses different Machine Learning approach to process and depict human language and answer the queries [1]. Linear regression, massaging the data, Perception, k- means are the few strategies used by him for uncovering the relationships and finding patterns in data. The choice of Machine Learning algorithm basically depends on the nature of prediction. The prediction can be estimate type or classification. He also discussed how increasing features can make the algorithm complex and increasing computational requirements.

Jainender singh, et.al (2014) proposed machine learning technique that would be providing promising results to security issues faced in applications, its technologies and theories. He emphasized on mining from sparse, incomplete and uncertain data that would give optimized results when hidden patterns are discovered from the data sets using machine learning algorithms like Support Vector Machine (SVM), Naïve Bays classifiers, clustering techniques which are used to create supervised learning [4]. It would give insight knowledge in health, education, trade and many more fields.

Junfei Qiu, et.al (2017) proposed some of the latest advances of Machine Learning for processing Big Data. Representation Learning, a new advanced learning method in which data representation is useful and meaningful by extracting helpful information while constructing classifiers and predictors. It aims to capture vast input which would give computation as well as statistical efficiency. Feature selection, Feature extraction and Metric learning are the subtopic of Representation learning. Active learning is another advanced Machine learning method applied for big data processing like biological DNA identification, image classification. It is a case of semi-supervised Machine learning in which it queries the users to get desired output from subset of critical labelled instances available thus minimizing the cost and giving higher accuracy and optimized results. He also discussed about the challenges and issues of Machine learning for Big Data processing. Heterogeneous nature of data, data produced at lightning speed, uncertainty and incomplete data, its vastness are some of the major concerns about Big Data. He also gave remedies for the same.

Alternating direction methods of multipliers (ADMM) is a promising method for parallel and distributed large scale data processing. It splits the multiple variables in an efficient way thus helping to find solution to a large scale of data. For handling high speed of data, Extreme Learning Method (ELM) has been introduced to provide faster learning speed, great performance and with less human interference.

Yasir Safeer, et.al (2010) presented Machine learning Algorithm i.e. k-means clustering for finding a document from a vast collection of unstructured text documents. He proposed a technique to portray documents that would be improving clustering result [3]. He discussed about the stream of document clustering, implemented k-means and devised an algorithm for better representation of documents and proposed how systematic domain dictionary would be used to get better similarity results of documents.

Roheet Bhatnagar, et.al (2018) presented about role of Machine Learning and Big Data Processing and Analytics (BDA). The development of Machine Learning and Big Data Analytics is complementary to each other. He discussed various future tends of Machine learning for Big data. Data Meaning implies how Machine Learning can be made more intelligent to acquire text or data awareness [5]. Technique Integration, another trend used to integrate data and process it. Classification, regression, cluster analysis are some of the techniques of Machine Learning which are used to perform analytics and predict future from existing patterns find correlation among the given data sets

Alexandra L"Heureux, et.al (2017) presented new ways of processing Big Data through Machine Learning Algorithms. Due to Big Data characteristics, traditional tools are now not capable of handling its storage, transport or its efficiency. Machine Learning is regarded as a fundamental component of Data Analytics as it has power to learn from data and provides data driven insights, prediction and decision. The tremendous increase in size, space and time complexity of Support Vector Machine (SVM) would affect both the complexities thus making computational efficiency infeasible. Curse of Modularity in which increase in size of data leads to collapse of the given boundary of algorithm is solved by Map Reduce [6]. It is a programmable and scalable paradigm used for processing large data sets on various nodes by following parallelism. It follows iterative approach.

K-means can also, be used to overcome shortcoming of Curse of modularity. Online Learning, one of the Machine Learning paradigms that would bridge the efficiency gaps produced by Big Data. It helps in processing large amount of data solution. Due to its adaptive nature, it is able to handle dirty and noisy data.

Rane, Archana L, et al (2017) in the proposed system a survivability kit for the human being is developed where some common symptoms diseases which kinds of the epidemic like Colds-Flu Gripe, Dengue, Malaria, Cholera, Leptospirosis, Chikungunya, Chickenpox, and Diarrhoea are can be easily predicted. To perform the present study data are collected from the hospital of Nasik, Maharashtra (India) of 316 patients. The algorithms like Decision tree (J48), Artificial Neural Network (MLP), Support Vector Machine (SMO), K-Nearest Neighbour (LWL), and Naive Bayes is used assessed by 10-fold cross-validation and performed in WEKA open source software. Henceforth, from the proposed work the ANN outperforms the parameters values comparatively to SVM which exceptionally gives lowest acceptance result.

Sally nihilist et al (2015) projected the sensible learning eventualities wherever we‟ve got bit of labelled knowledge at the side of an outsized pool of unlabelled knowledge and conferred a "curtaining" strategy for exploitation the unlabelled knowledge to boost the quality supervised learning algorithms. She assumed that there square measure 2 completely different supervised learning algorithms that each output a hypothesis that defines a partition of instance area for e.g. a call tree partitions the instance area with one equivalent category outlined per tree. She finally finished that 2supervised learning algorithms may be used with success label knowledge for every different.

Zoubin Ghahramani et al (2018) gave a short summary of unsupervised learning from the angle of applied mathematics modelling. consistent with him unsupervised learning maybe motivated from data abstractive and theorem principles. He additionally reviewed the models in unsupervised learning. He any finished that statistics provides a coherent framework for learning from knowledge and for reasoning beneath uncertainty and additionally, he mentioned the kinds of models like Graphical model that contend a vital role in learning systems for kind of completely different forms of knowledge.

Rich Caruana et al (2019) has studied numerous supervised learning strategies that were introduced in last decade and supply a large-scale empirical comparison between 10 supervised learning strategies. These approaches include: SVMs, neural nets, supply regression, naive Bayes, memory-based learning, random forests, call trees, bagged trees, boosted trees and boosted stumps. They moreover studied and inspect the result that calibrating the models through Platt Scaling and Isotonic Regression has on their performance. that they had used numerous performance-based criteria to gauge the educational strategies.

Man Galih Salman, Yaya Heryadi, Bayu Kanigoro Has studies the matter faces concerning foretelling. during this author exploitation deep learning technique for the foretelling. Deep learning is that the a part of AI within which “deep” indicates that such neural network contains additional layer then the “shadow” ones utilized in typical machine learning. It projected a probe a frame work with the weather knowledge. The results has been enforced to explore continual NN exploitation heuristically optimisation methodologyfor rain prediction supported weather dataset.

CHAPTER 3

HARDWARE AND SOFTWARE COMPONENTS

3.1 HARDWARE SPECIFICATION

- Processor : Intel(R) Pentium(R) CPU A1018 @ 2.10GHz(2 CPUs), ~2.1GHz
- RAM : 2048 MB RAM
- Memory : 512 KB Cache Memory
- Hard Disk : 100 GB
- Key Board : Microsoft Compatible 101 or more keyboard

3.2 SOFTWARE SPECIFICATION

- Operating system : Windows 8.1 Pro 64-bit
- IDE : Anaconda
- Framework : Tensor flow
- Front End : PYTHON/ML
- Back End : CSV Excel Dataset

3.3 SOFTWARE REQUIREMENTS

Python

Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980s, as a successor to the ABC programming language, and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features, such as list comprehensions and a garbage collection system using reference counting and was discontinued with version 2.7.18 in 2020. Python 3.0 was released in 2008 and was a major revision of the language that is not completely backward-compatible and much Python 2 code does not run unmodified on Python 3. Python consistently ranks as one of the most popular programming languages.

Python provides many useful features which make it popular and valuable from the other programming languages. It supports object-oriented programming, procedural programming approaches and provides dynamic memory allocation. We have listed below a few essential features.

1) Easy to Learn and Use

Python is easy to learn as compared to other programming languages. Its syntax is straightforward and much the same as the English language. There is no use of the semicolon or curly-bracket, the indentation defines the code block. It is the recommended programming language for beginners.

2) Expressive Language

Python can perform complex tasks using a few lines of code. A simple example, the hello world program you simply type print("Hello World"). It will take only one line to execute, while Java or C takes multiple lines.

3) Interpreted Language

Python is an interpreted language; it means the Python program is executed one line at a time. The advantage of being interpreted language, it makes debugging easy and portable.

4) Cross-platform Language

Python can run equally on different platforms such as Windows, Linux, UNIX, and Macintosh, etc. So, we can say that Python is a portable language. It enables programmers to develop the software for several competing platforms by writing a program only once.

5) Free and Open Source

Python is freely available for everyone. It is freely available on its official website www.python.org. It has a large community across the world that is dedicatedly working towards make new python modules and functions. Anyone can contribute to the Python community. The open-source means, "Anyone can download its source code without paying any penny."

6) Object-Oriented Language

Python supports object-oriented language and concepts of classes and objects come into existence. It supports inheritance, polymorphism, and encapsulation, etc. The object-oriented procedure helps to programmer to write reusable code and develop applications in less code.

7) Extensible

It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our Python code. It converts the program into byte code, and any platform can use that byte code.

8) Large Standard Library

It provides a vast range of libraries for the various fields such as machine learning, web developer, and also for the scripting. There are various machine learning libraries, such as Tensor flow, Pandas, Numpy, Keras, and Pytorch, etc. Django, flask, pyramids are the popular framework for Python web development.

9) GUI Programming Support

Graphical User Interface is used for the developing Desktop application. PyQT5, Tkinter, Kivy are the libraries which are used for developing the web application.

10) Integrated

It can be easily integrated with languages like C, C++, and JAVA, etc. Python runs code line by line like C, C++ Java. It makes easy to debug the code.

11) Embeddable

The code of the other programming language can use in the Python source code. We can use Python source code in another programming language as well. It can embed other language into our code.

12) Dynamic Memory Allocation

In Python, we don't need to specify the data-type of the variable. When we assign some value to the variable, it automatically allocates the memory to the variable at run time. Suppose we are assigned integer value 15 to x, then we don't need to write int x = 15. Just write x = 15.

Anaconda

Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and Travis Oliphant in 2012. As an Anaconda, Inc. product, it is also known as Anaconda Distribution or Anaconda Individual Edition, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, both of which are not free.

Package versions in Anaconda are managed by the package management system conda. This package manager was spun out as a separate open-source package as it ended up being useful on its own and for other things than Python. There is also a small, bootstrap version of Anaconda called Miniconda, which includes only conda, Python, the packages they depend on, and a small number of other packages.

Anaconda distribution comes with over 250 packages automatically installed, and over 7,500 additional open-source packages can be installed from PyPI as well as the conda package and virtual environment manager. It also includes a GUI, Anaconda Navigator, as a graphical alternative to the command line interface (CLI).

The big difference between conda and the pip package manager is in how package dependencies are managed, which is a significant challenge for Python data science and the reason conda exists.

Before version 20.3, when pip installed a package, it automatically installed any dependent Python packages without checking if these conflict with previously installed packages. It would install a package and any of its dependencies regardless of the state of the existing installation.

Because of this, a user with a working installation of, for example, Google Tensorflow, could find that it stopped working having used pip to install a different package that requires a different version of the dependent numpy library than the one used by Tensorflow. In some cases, the package would appear to work but produce different results in detail. While pip has since implemented consistent dependency resolution, this difference accounts for a historical differentiation of the conda package manager.

In contrast, conda analyses the current environment including everything currently installed, and, together with any version limitations specified (e.g. the user may wish to have Tensorflow version 2,0 or higher), works out how to install a compatible set of dependencies, and shows a warning if this cannot be done.

Open-source packages can be individually installed from the Anaconda repository, Anaconda Cloud (anaconda.org), or the user's own private repository or mirror, using the conda install command. Anaconda, Inc. compiles and builds the packages available in the Anaconda repository itself, and provides binaries for Windows 32/64 bit, Linux 64 bit and MacOS 64-bit. Anything available on PyPI may be installed into a conda environment using pip, and conda will keep track of what it has installed itself and what pip has installed.

Custom packages can be made using the conda build command, and can be shared with others by uploading them to Anaconda Cloud, PyPI or other repositories.

The default installation of Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7. However, it is possible to create new environments that include any version of Python packaged with conda.

Anaconda Navigator

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.

The following applications are available by default in Navigator:

- JupyterLab
- Jupyter Notebook
- QtConsole
- Spyder
- Glue
- Orange
- RStudio
- Visual Studio Code

Spyder IDE

Spyder is an open-source cross-platform integrated development environment (IDE) for scientific programming in the Python language. Spyder integrates with a number of prominent packages in the scientific Python stack, including NumPy, SciPy, Matplotlib, pandas, IPython, SymPy and Cython, as well as other open-source software. It is released under the MIT license.

Initially created and developed by Pierre Raybaut in 2009, since 2012 Spyder has been maintained and continuously improved by a team of scientific Python developers and the community.

Spyder is extensible with first-party and third-party plugins, includes support for interactive tools for data inspection and embeds Python-specific code quality assurance and introspection instruments, such as Pyflakes, Pylint and Rope. It is available cross-platform through Anaconda, on Windows, on macOS through MacPorts, and on major Linux distributions such as Arch Linux, Debian, Fedora, Gentoo Linux, openSUSE and Ubuntu.

Spyder uses Qt for its GUI and is designed to use either of the PyQt or PySide Python bindings. QtPy, a thin abstraction layer developed by the Spyder project and later adopted by multiple other packages, provides the flexibility to use either backend.

Features include:

- An editor with syntax highlighting, introspection, code completion
- Support for multiple IPython consoles
- The ability to explore and edit variables from a GUI
- A Help pane able to retrieve and render rich text documentation on functions, classes and methods automatically or on-demand
- A debugger linked to IPdb, for step-by-step execution
- Static code analysis, powered by Pylint
- A run-time Profiler, to benchmark code
- Project support, allowing work on multiple development efforts simultaneously
- A built-in file explorer, for interacting with the filesystem and managing projects
- A "Find in Files" feature, allowing full regular expression search over a specified scope
- An online help browser, allowing users to search and view Python and package documentation inside the IDE
- A history log, recording every user command entered in each console
- An internal console, allowing for introspection and control over Spyder's own operation

CHAPTER 4

METHODOLOGY AND DESCRIPTION

In this proposed project the rainfall between 1901 to 2015 has been analysed and predicted using Machine learning technique such as Decision Tree and Support Vector Regression. The main purpose and aim of the study are to utilize the machine learning for predicting the rainfall with accuracy and precision. The input data has been retrieved from kaggle. Feature selection is done using LDA method from input data. The selected feature is ANNUAL, OCT-FEB, JUN-SEPT, JAN, FEB, MAR, APRIL, MAY. Followed by, analysis is done for the selected feature by training and testing the model. The training and testing is done using Machine learning technique such as Decision Tree, Support Vector Regression. From the output, ARIMA model is used to forecast the future rainfall prediction of about 2020 to 2030. In the final phase, the performance analysis is used to compare the algorithm. The performance metric includes Accuracy, Mean Square error, Recall.

METHODOLOGY DIAGRAM

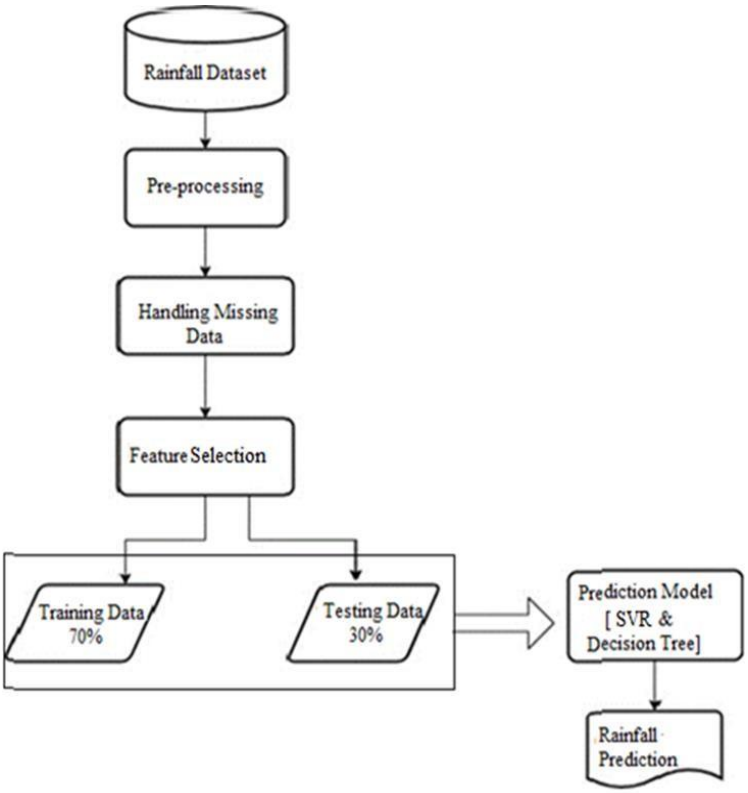
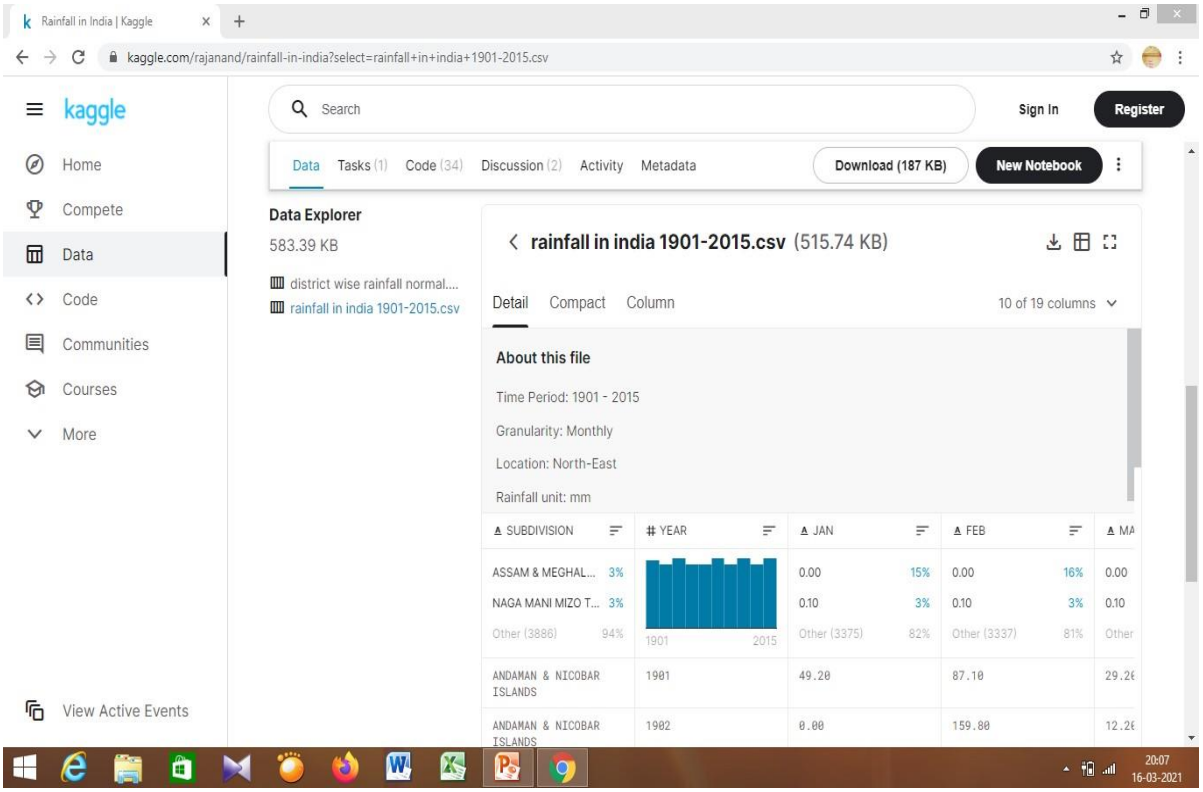


Figure 4.1: Methodology Diagram

4.1 DATA COLLECTION

The dataset consists of the measurement of Indian rainfall from the year of 1901-2015 for each state. Data consists of 4117 rows and 19 attributes (individual months and annual) for 36 sub divisions. The attributes are the amount of rainfall measured in mm.

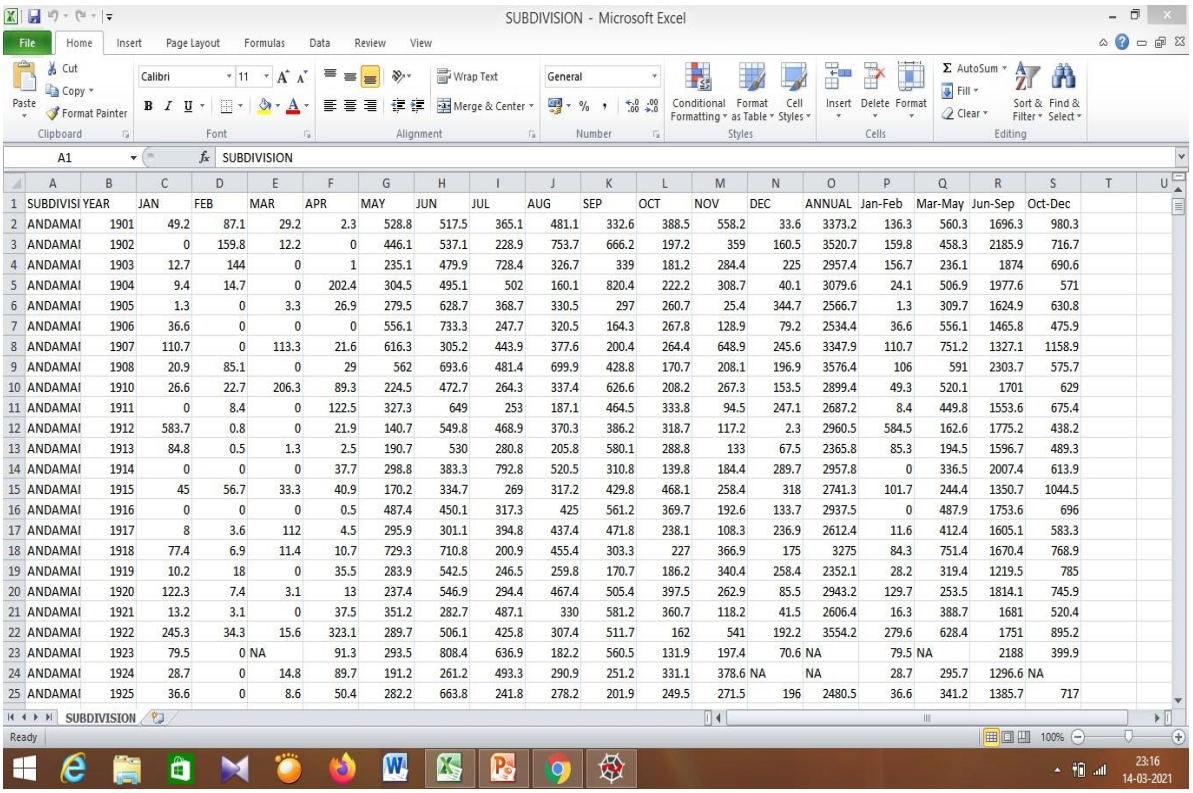
The dataset is available at https://www.kaggle.com/rajanand/rainfall-in-india?select=rainfall+in+india+1901-2015.csv



The dataset consist of 19 attributes (individual months, annual, and combinations of 3 consecutive months) only one attribute is character type otherwise all the attributes are numbers.

Attributes	Type
SUBDIVISION	Character
YEAR	Number
JAN	Number
FEB	Number
MAR	Number
APR	Number
MAY	Number
JUN	Number
JUL	Number
AUG	Number
SEP	Number
OCT	Number
NOV	Number
DEC	Number
ANNUAL	Number
Jan-Feb	Number
Mar-May	Number
Jun-Sep	Number
Oct-Dec	Number

Figure 4.2: Dataset Attributes



SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL	Jan-Feb	Mar-May	Jun-Sep	Oct-Dec
ANDAMAI	1901	49.2	87.1	29.2	2.3	528.8	517.5	365.1	481.1	332.6	388.5	558.2	33.6	3373.2	136.3	560.3	1696.3	980.3
ANDAMAI	1902	0	159.8	12.2	0	446.1	537.1	228.9	753.7	666.2	197.2	359	160.5	3520.7	159.8	458.3	2185.9	716.7
ANDAMAI	1903	12.7	144	0	1	235.1	479.9	728.4	326.7	339	181.2	284.4	225	2957.4	156.7	236.1	1874	690.6
ANDAMAI	1904	9.4	14.7	0	202.4	304.5	495.1	502	160.1	820.4	222.2	308.7	40.1	3079.6	24.1	506.9	1977.6	571
ANDAMAI	1905	1.3	0	3.3	26.9	279.5	628.7	368.7	330.5	297	260.7	25.4	344.7	2566.7	1.3	309.7	1624.9	630.8
ANDAMAI	1906	36.6	0	0	0	556.1	733.3	247.7	320.5	164.3	267.8	128.9	79.2	2534.4	36.6	556.1	1465.8	475.9
ANDAMAI	1907	110.7	0	113.3	21.6	616.3	305.2	443.9	377.6	200.4	264.4	648.9	245.6	3347.9	110.7	751.2	1327.1	1158.9
ANDAMAI	1908	20.9	85.1	0	29	562	693.6	481.4	699.9	428.8	170.7	208.1	196.9	3576.4	106	591	2303.7	575.7
ANDAMAI	1910	26.6	22.7	206.3	89.3	224.5	472.7	264.3	337.4	626.6	208.2	267.3	153.5	2899.4	49.3	520.1	1701	629
ANDAMAI	1911	0	8.4	0	122.5	327.3	649	253	187.1	464.5	333.8	94.5	247.1	2687.2	8.4	449.8	1553.6	675.4
ANDAMAI	1912	583.7	0.8	0	21.9	140.7	549.8	468.9	370.3	386.2	318.7	117.2	2.3	2960.5	584.5	162.6	1775.2	438.2
ANDAMAI	1913	84.8	0.5	1.3	2.5	190.7	530	280.8	205.8	580.1	288.8	133	67.5	2365.8	85.3	194.5	1596.7	489.3
ANDAMAI	1914	0	0	0	37.7	298.8	383.3	792.8	520.5	310.8	139.8	184.4	289.7	2957.8	0	336.5	2007.4	613.9
ANDAMAI	1915	45	56.7	33.3	40.9	170.2	334.7	269	317.2	429.8	468.1	258.4	318	2741.3	101.7	244.4	1350.7	1044.5
ANDAMAI	1916	0	0	0	0.5	487.4	450.1	317.3	425	561.2	369.7	192.6	133.7	2937.5	0	487.9	1753.6	696
ANDAMAI	1917	8	3.6	112	4.5	295.9	301.1	394.8	437.4	471.8	238.1	108.3	236.9	2612.4	11.6	412.4	1605.1	583.3
ANDAMAI	1918	77.4	6.9	11.4	10.7	729.3	710.8	200.9	455.4	303.3	227	366.9	175	3275	84.3	751.4	1670.4	768.9
ANDAMAI	1919	10.2	18	0	35.5	283.9	542.5	246.5	259.8	170.7	186.2	340.4	258.4	2352.1	28.2	319.4	1219.5	785
ANDAMAI	1920	122.3	7.4	3.1	13	237.4	546.9	294.4	467.4	505.4	397.5	262.9	85.5	2943.2	129.7	253.5	1814.1	745.9
ANDAMAI	1921	13.2	3.1	0	37.5	351.2	282.7	487.1	330	581.2	360.7	118.2	41.5	2606.4	16.3	388.7	1681	520.4
ANDAMAI	1922	245.3	34.3	15.6	323.1	289.7	506.1	425.8	307.4	511.7	162	541	192.2	3554.2	279.6	628.4	1751	895.2
ANDAMAI	1923	79.5	0	NA	91.3	293.5	808.4	636.9	182.2	560.5	131.9	197.4	70.6	NA	79.5	NA	2188	399.9
ANDAMAI	1924	28.7	0	14.8	89.7	191.2	261.2	493.3	290.9	251.2	331.1	378.6	NA	NA	28.7	295.7	1296.6	NA
ANDAMAI	1925	36.6	0	8.6	50.4	282.2	663.8	241.8	278.2	201.9	249.5	271.5	196	2480.5	36.6	341.2	1385.7	717

Figure 4.3: Rainfall Dataset

4.2 DATA PRE-PROCESSING

The main purpose and aim of the project are to utilize the machine learning for predicting the rainfall with accuracy and prediction. Therefore, the project includes and examines the source data in the pre-processing phase and utilizes this data in the stimulation or processing phase further to predict the output that is forecasting effectively and efficiently. For this purpose, data of rainfall is collected from the Kaggle. The collected data is monthly as well as annual rainfall rate, as this project will also focus on the annual rainfall prediction. The data includes the significant parameters of the rainfall. Each parameter of the rainfall is analyzed separately. In data pre-processing, I have done by Identify the missing values, Remove incomplete rows and Outliers. The below figure is to identify the missing values of the dataset.

```
df.isnull().sum()
```

```
SUBDIVISION     0
YEAR            0
JAN             4
FEB             3
MAR             6
APR             4
MAY             3
JUN             5
JUL             7
AUG             4
SEP             6
OCT             7
NOV            11
DEC            10
ANNUAL         26
Jan-Feb         6
Mar-May         9
Jun-Sep        10
Oct-Dec        13
dtype: int64
```

Figure 4.4: Identify the missing values

The figure 4.4 shows the missing values of individual months, annual, and combinations of 3 consecutive months.

```
df=df.dropna()
```

```
df
```

	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL	Jan-Feb	Mar-May	Jun-Sep	Oct-Dec
0	ANDAMAN & NICOBAR ISLANDS	1901	49.2	87.1	29.2	2.3	528.8	517.5	365.1	481.1	332.6	388.5	558.2	33.6	3373.2	136.3	560.3	1696.3	980.3
1	ANDAMAN & NICOBAR ISLANDS	1902	0.0	159.8	12.2	0.0	446.1	537.1	228.9	753.7	666.2	197.2	359.0	160.5	3520.7	159.8	458.3	2185.9	716.7
2	ANDAMAN & NICOBAR ISLANDS	1903	12.7	144.0	0.0	1.0	235.1	479.9	728.4	326.7	339.0	181.2	284.4	225.0	2957.4	156.7	236.1	1874.0	690.6
3	ANDAMAN & NICOBAR ISLANDS	1904	9.4	14.7	0.0	202.4	304.5	495.1	502.0	160.1	820.4	222.2	308.7	40.1	3079.6	24.1	506.9	1977.6	571.0
4	ANDAMAN & NICOBAR ISLANDS	1905	1.3	0.0	3.3	26.9	279.5	628.7	368.7	330.5	297.0	260.7	25.4	344.7	2566.7	1.3	309.7	1624.9	630.8
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4111	LAKSHADWEEP	2011	5.1	2.8	3.1	85.9	107.2	153.6	350.2	254.0	255.2	117.4	184.3	14.9	1533.7	7.9	196.2	1013.0	316.6
4112	LAKSHADWEEP	2012	19.2	0.1	1.6	76.8	21.2	327.0	231.5	381.2	179.8	145.9	12.4	8.8	1405.5	19.3	99.6	1119.5	167.1
4113	LAKSHADWEEP	2013	26.2	34.4	37.5	5.3	88.3	426.2	296.4	154.4	180.0	72.8	78.1	26.7	1426.3	60.6	131.1	1057.0	177.6
4114	LAKSHADWEEP	2014	53.2	16.1	4.4	14.9	57.4	244.1	116.1	466.1	132.2	169.2	59.0	62.3	1395.0	69.3	76.7	958.5	290.5
4115	LAKSHADWEEP	2015	2.2	0.5	3.7	87.1	133.1	296.6	257.5	146.4	160.4	165.4	231.0	159.0	1642.9	2.7	223.9	860.9	555.4

4090 rows × 19 columns

Figure 4.5: Remove incomplete rows

The above figure depicts the incomplete rows, 26 incomplete rows from the dataset was removed. The figure 4.6 illustrates the dataset after removing the outliers.

	SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL	Jan-Feb	Mar-May	Jun-Sep	Oct-Dec
6	ANDAMAN & NICOBAR ISLANDS	1907	110.7	0.0	113.3	21.6	616.3	305.2	443.9	377.6	200.4	264.4	648.9	245.6	3347.9	110.7	751.2	1327.1	1158.9
7	ANDAMAN & NICOBAR ISLANDS	1908	20.9	85.1	0.0	29.0	562.0	693.6	481.4	699.9	428.8	170.7	208.1	196.9	3576.4	106.0	591.0	2303.7	575.7
8	ANDAMAN & NICOBAR ISLANDS	1910	26.6	22.7	206.3	89.3	224.5	472.7	264.3	337.4	626.6	208.2	267.3	153.5	2899.4	49.3	520.1	1701.0	629.0
9	ANDAMAN & NICOBAR ISLANDS	1911	0.0	8.4	0.0	122.5	327.3	649.0	253.0	187.1	464.5	333.8	94.5	247.1	2687.2	8.4	449.8	1553.6	675.4
10	ANDAMAN & NICOBAR ISLANDS	1912	583.7	0.8	0.0	21.9	140.7	549.8	468.9	370.3	386.2	318.7	117.2	2.3	2960.5	584.5	162.6	1775.2	438.2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4099	LAKSHADWEEP	1999	47.8	2.5	18.3	20.6	416.7	279.6	459.4	133.8	73.4	305.0	51.2	49.0	1857.3	50.3	455.6	946.2	405.2
4100	LAKSHADWEEP	2000	83.3	18.9	3.4	47.9	204.6	225.4	95.5	319.9	164.5	141.4	56.3	11.0	1372.1	102.2	255.9	805.3	208.7
4101	LAKSHADWEEP	2001	4.4	20.4	0.0	104.6	187.3	283.9	198.9	144.3	213.5	105.2	101.5	16.6	1380.6	24.8	291.9	840.6	223.3
4102	LAKSHADWEEP	2002	10.8	16.8	7.2	23.4	189.8	261.8	81.3	143.9	50.0	178.2	52.9	17.4	1033.5	27.6	220.4	537.0	248.5
4103	LAKSHADWEEP	2003	11.8	18.2	28.5	18.1	109.6	364.5	400.6	92.1	84.3	191.6	206.1	7.5	1532.9	30.0	156.2	941.5	405.2

3451 rows × 19 columns

Figure 4.6: Remove the outliers

4.3 FEATURE SELECTION

Feature Selection is done using LDA (Linear Discriminant Analysis). Feature selection is the process of reducing the number of input variables when developing a predictive model. The main use of Linear Discriminant Analysis is used to select the features and reduce the dimension of features to a manageable counting before process of classification. The parameters in rainfall includes subdivision, year, jan, feb,mar,apr, may, jun, jul, aug, sep, oct, nov, dec, annual, jan-feb, mar-may, jun-sep, oct-dec. After performing preprocessing, annual, jun, oct, jun-sep, oct-dec is selected and the dimension reduced.

LDA STEPS:

1. Compute the d-dimensional mean vectors for the different classes from the dataset.
2. Compute the scatter matrices (in-between-class and within-class scatter matrix).
3. Compute the eigenvectors (e1,e2,....,ed) and corresponding eigenvalues ($\lambda 1, \lambda 2, ..., \lambda d$) for the scatter matrices.
4. Sort the eigenvectors by decreasing eigenvalues and choose k eigenvectors with the largest eigenvalues to form a d×k dimensional matrix W (where every column represents an eigenvector).
5. Use this d×k eigenvector matrix to transform the samples onto the new subspace. This can be summarized by the matrix multiplication: Y=X×W (where X is a n×d-dimensional matrix representing the n samples, and y are the transformed n×k dimensional samples in the new subspace).

Figure 4.7: Selected Features

4.4 DATA ANALYSIS

Data Analysis is an approach or philosophy for data analysis that employs a variety of techniques to maximize insight into a data set, uncover underlying structure, extract important variables, detect outliers and anomalies, test underlying assumptions, develop parsimonious models and determine optimal factor settings.

Testing Classification Model:

A classification model assigns to each entity the outcome it considers to be the most appropriate. The outcome is the predicted value that is determined by the classification model. Classification is a data mining function that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data. A classification task begins with a data set in which the class assignments are known. Classification models are tested by comparing the predicted values to known target values in a set of test data. In proposed system, three classification algorithms are applied such as Decision Tree, Support vector Regression for the rainfall dataset.

4.4.1 Decision Tree

Decision Tree Analysis is a general, predictive modelling tool that has applications spanning a number of different areas. In general, decision trees are constructed via an algorithmic approach that identifies ways to split a data set based on different conditions. It is one of the most widely used and practical methods for supervised learning.

Decision Trees are a non-parametric supervised learning method used for both classification and regression tasks. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. The decision rules are generally in form of if-then-else statements. The deeper the tree, the more complex the rules and fitter the model.

Decision tree fall under supervised learning therefore the dataset can be split into two categories such as training and testing data. For better performance split the dataset into 70:30 whereas 70% of data as training and 30% of data as testing.

Decision tree is a graphical representation of the relations that exist between the data in the database. It is used for data classification. The result is displayed as a tree and it is mainly used in the classification and prediction. The instances are classified by sorting them down the tree from the root node to some leaf node.

Algorithm

- **Step-1:** Begin the tree with the root node, says S, which contains the complete dataset.
- **Step-2:** Find the best attribute in the dataset using **Attribute Selection Measure (ASM).**
- **Step-3:** Divide the S into subsets that contains possible values for the best attributes.
- **Step-4:** Generate the decision tree node, which contains the best attribute.
- **Step-5:** Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

4.4.2 Support Vector Regression

In machine learning, support-vector machines (SVMs, also support-vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. There are a number of examples of where it has been used in the agricultural domain. To minimize the generalization error bound and to achieve generalized performance, SVM is used in this module.

A SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

SVM Classifier

Support Vector Machine (SVM) used for both classification and regression challenges. However, it is mostly used in classification problems. In this algorithm, each data item was plotted as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate.

Then, perform classification by finding the hyper-plane that differentiates the two classes very well. Support Vectors are simply the co-ordinates of individual observation. Support Vector Machine is a frontier which best segregates the two classes (hyper-plane/ line)

Classification

A Support Vector Machine (SVM) performs classification by finding the hyperplane that maximizes the margin between the two classes. The vectors (cases) that define the hyperplane are the support vectors. The steps required to classify using support vector machine are:

1. Define an optimal hyperplane: maximize margin
2. Extend the above definition for non-linearly separable problems: have a penalty term for misclassifications.
3. Map data to high dimensional space where it is easier to classify with linear decision surfaces: reformulate problem so that data is mapped implicitly to this space.

Regression

The model produced by Support Vector Regression depends only on a subset of the training data, because the cost function for building the model ignores any training data close to the model prediction.

Support Vector Regression

Unlike other Regression models that try to minimize the error between the real and predicted value, the SVR tries to fit the best line within a threshold value (Distance between hyperplane and boundary line), a. Thus, we can say that SVR model tries satisfy the condition -a < y-wx+b < a. It used the points with this boundary to predict the value.

Algorithm

Step 1: Import necessary libraries

Step 2: Reading the dataset named rainfall.csv

Step 3: Feature Scaling

Step 4: Fitting SVR to rainfall dataset

Step 5: Predict feature result

4.5 FUTURE FORECAST

Forecasting refers to the practice of predicting what will happen in the future by taking into consideration events in the past and present. Basically, it is a decision-making tool that helps businesses cope with the impact of the future's uncertainty by examining historical data. Forecast is done using ARIMA model. Arima model uses time series data that includes year.

ARIMA

ARIMA, short for „Auto Regressive Integrated Moving Average" is actually a class of models that „explains" a given time series based on its own past values, that is, its own lags and the lagged forecast errors, so that equation can be used to forecast future values. Any „non-seasonal" time series that exhibits patterns and is not a random white noise can be modelled with ARIMA models.

An ARIMA model is characterized by 3 terms: p, d, q

where,

p is the order of the AR term

q is the order of the MA term

d is the number of differencing required to make the time series stationary

If a time series, has seasonal patterns, then you need to add seasonal terms and it becomes SARIMA, short for „Seasonal ARIMA". More on that once we finish ARIMA.

So, what does the „order of AR term" even mean? Before we go there, let"s first look at the „d" term.

Algorithm:

Step 1: Make time series stationary by differencing "annual" feature in data
Step 2: Identify ARIMA model using ACF and PACF
Step 3: Estimate ARIMA model parameters
Step 4: Plot the correlation and auto correlation charts
Step 5: Choose most suitable ARIMA model
Step 6: Visualize forecasted data.

CHAPTER 5

RESULT AND DISCUSSION

5.1 PERFORMANCE METRICS

In proposed project, the comparison of algorithms namely Decision tree and Support vector Regression are based on following performance metrics.

1. Accuracy
2. Mean Square Error
3. Recall

5.1.1Accuracy

Accuracy is defined as the percentage of correct predictions for the test data. It can be calculated easily by dividing the number of correct predictions by the number of total predictions. Accuracy is one of the measures to evaluate classification models. The precision is the fraction of the predictions given by the classification model. The precision has the following definition: Accuracy = Total no. of the correct forecasts / From predictions For the binary classification, the accuracy can be calculated as negative and positive in the following way:

$$Accuracy=((TP+TN)/(TP+TN+FP+FN))$$

Where,

TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

5.1.2 Mean Square Error

The Mean Squared Error (MSE) or Mean Squared Deviation (MSD) of an estimator measures the average of error squares i.e., the average squared difference between the estimated values and true value.

$$\text{Mean Square Error} = ()$$

5.1.3 Recall

The recall is calculated as the number of true positives divided by the total number of true positives and false negatives. The result is a value between 0.0 for no recall and 1.0 for full or perfect recall.

Recall = TruePositives / (TruePositives + FalseNegatives)

```
Support Vector Regression
Mean Square Error : 0.21431677423920023
Accuracy: 0.7856832257607997
Recall: 0.41431677423920027

Decision Tree
Mean Square Error:  0.0101431866340668 29
Accuracy:  0.882640586797066
Recall :  0.34431677423920026
```

Figure 5.1: Performance metrics for Support Vector Regression and Decision Tree

Method	Accuracy-Score	Mean Square Error	Recall
Support Vector Regression	**0.785**	0.21431677423920023	0.41431677423920027
Decision Tree	**0.882**	0.010143186634066829	0.34431677423920026

Table 5.1: Performance metrics for SVR and Decision Tree

Accuracy of Support Vector Regression is : 79%

Accuracy of Decision Tree is : 88%

5.2 PREDICTING MINIMUM AND MAXIMUM RAINFALL IN INDIA FROM 1901 TO 2015

```
SUBDIVISION    WEST UTTAR PRADESH
YEAR                         2015
JAN                         583.7
FEB                         403.5
MAR                         605.6
APR                         595.1
MAY                        1168.6
JUN                        1609.9
JUL                        2362.8
AUG                        1664.6
SEP                          1222
OCT                         948.3
NOV                         648.9
DEC                         617.5
ANNUAL                     6331.1
Jan-Feb                     699.5
Mar-May                    1745.8
Jun-Sep                    4536.9
Oct-Dec                    1252.5
dtype: object
```

Figure 5.2: Maximum rainfall between 1901 to 2015

```
SUBDIVISION    ANDAMAN & NICOBAR ISLANDS
YEAR                                1901
JAN                                    0
FEB                                    0
MAR                                    0
APR                                    0
MAY                                    0
JUN                                  0.4
JUL                                    0
AUG                                    0
SEP                                  0.1
OCT                                    0
NOV                                    0
DEC                                    0
ANNUAL                              62.3
Jan-Feb                                0
Mar-May                                0
Jun-Sep                             57.4
Oct-Dec                                0
dtype: object
```

Figure 5.3 Minimum rainfall between 1901 to 2015

High	SUBDIVISION	YEAR	ANNUAL
55	ANDAMAN & NICOBAR ISLANDS	1961	3938.2
142	ARUNACHAL PRADESH	1948	6331.1
280	ASSAM & MEGHALAYA	1974	3403.5
983	BIHAR	1987	1660.4
3027	CHHATTISGARH	1961	1974.0
3191	COASTAL ANDHRA PRADESH	2010	1712.4
3602	COASTAL KARNATAKA	1961	5553.9
2180	EAST MADHYA PRADESH	1919	1747.1
1948	EAST RAJASTHAN	1917	1350.4
1047	EAST UTTAR PRADESH	1936	1545.5
622	GANGETIC WEST BENGAL	1971	2099.8
2352	GUJARAT REGION	1976	1620.1
1373	HARYANA DELHI & CHANDIGARH	1917	986.7
1641	HIMACHAL PRADESH	1955	1919.2
1795	JAMMU & KASHMIR	1994	1732.5
852	JHARKHAND	1971	1898.6
3947	KERALA	1961	4257.8
2564	KONKAN & GOA	1958	4000.2
4107	LAKSHADWEEP	2007	2361.6
2727	MADHYA MAHARASHTRA	2006	1395.7
2826	MATATHWADA	1990	1198.1
391	NAGA MANI MIZO TRIPURA	1970	4316.2
3753	NORTH INTERIOR KARNATAKA	1997	1095.6
727	ORISSA	1961	1945.3
1559	PUNJAB	1988	1222.6
3407	RAYALSEEMA	1996	1277.7
2501	SAURASHTRA & KUTCH	2010	1119.9

Figure 5.4 High Rainfall between 1901 to 2015

Low	SUBDIVISION	YEAR	ANNUAL
73	ANDAMAN & NICOBAR ISLANDS	1979	1849.4
158	ARUNACHAL PRADESH	1967	1668.5
317	ASSAM & MEGHALAYA	2011	1743.4
1006	BIHAR	2010	629.2
3045	CHHATTISGARH	1979	904.6
3183	COASTAL ANDHRA PRADESH	2002	703.2
3559	COASTAL KARNATAKA	1918	2510.9
2268	EAST MADHYA PRADESH	2007	653.8
1949	EAST RAJASTHAN	1918	273.6
1108	EAST UTTAR PRADESH	1997	493.3
586	GANGETIC WEST BENGAL	1935	1015.1
2294	GUJARAT REGION	1918	392.6
1374	HARYANA DELHI & CHANDIGARH	1918	234.7
1660	HIMACHAL PRADESH	1974	776.1
1771	JAMMU & KASHMIR	1970	657.0
891	JHARKHAND	2010	697.1
3962	KERALA	1976	2068.8
2511	KONKAN & GOA	1905	1682.8
4058	LAKSHADWEEP	1958	992.6
2639	MADHYA MAHARASHTRA	1918	438.0
2808	MATATHWADA	1972	347.1
393	NAGA MANI MIZO TRIPURA	1972	1353.8
3676	NORTH INTERIOR KARNATAKA	1920	470.3
740	ORISSA	1974	987.0
1573	PUNJAB	2002	274.7

Figure 5.5 Low Rainfall between 1901 to 2015

SUBDIVISION	**YEAR & ANNUAL HIGH RAINFALL**	**YEAR & ANNUAL LOW RAINFALL**
ANDAMAN & NICOBAR ISLANDS	1961 3938.2mm	1979 73mm
ARUNACHAL PRADESH	**1948** **6331.1mm**	1967 1668.5mm
ASSAM & MEGHALAYA	1974 3403.5mm	2011 1743.4mm

BIHAR	1987 1660.4mm	2010 629.2mm
CHHATTISGARH	1961 1974.0mm	1979 904.6mm
COASTAL ANDHRA PRADESH	2010 1712.4mm	2002 703.2mm
COASTAL KARNATAKA	1961 5553.9mm	1918 2510.9mm
EAST MADHYA PRADESH	1919 1747.1mm	2007 653.8mm
EAST RAJASTHAN	1917 1350.4mm	1918 273.6mm
EAST UTTAR PRADESH	1936 1545.5mm	1997 493.3mm
GANGETIC WEST BENGAL	1971 2099.8mm	1935 1015.1mm
GUJARAT REGION	1976 1620.1mm	1918 392.6mm
HARYANA DELHI & CHANDIGARH	1917 986.7mm	1918 234.7mm
HIMACHAL PRADESH	1955 1919.2mm	1974 776.1mm

JAMMU & KASHMIR	1994 1732mm	1970 657mm
JHARKHAND	1971 1898mm	2010 697.1mm
KERALA	1961 4257.8mm	1976 2068.8mm
KONKAN & GOA	1958 4000.2mm	1905 1682.8mm
LAKSHADWEEP	2007 2361.6mm	1958 992.6mm
MADHYA MAHARASHTRA	2006 1395.7mm	1918 438.0mm
MATATHWADA	1990 1198.1mm	1972 347.1mm
NAGA MANI MIZO TRIPURA	1970 4316.2mm	1972 1353.8mm
NORTH INTERIOR KARNATAKA	1997 1095.6mm	1920 470.3mm
ORISSA	1961 1945.3mm	1974 987.6mm
PUNJAB	1988 1222.6mm	2002 274.7mm
RAYALSEEMA	1996 1277.7mm	1904 433.4mm

SAURASHTRA & KUTCH	2010 1119.9mm	1987 92.7mm
SOUTH INTERIOR KARNATAKA	1961 1409.5mm	1918 733.3mm
SUB HIMALAYAN WEST BENGAL & SIKKIM	1998 3655.1mm	1908 1988.2mm
TAMIL NADU	2005 1365.3mm	2002 318.0mm
TELANGANA	1988 1544.9mm	1920 437.0mm
UTTARAKHAND	1917 2102.9mm	1997 803.4mm
VIDARBHA	1959 1606.3mm	1920 578.5mm
WEST MADHYA PRADESH	1961 1433.7mm	1918 509.4mm
WEST RAJASTHAN	1917 768.8mm	**1918** **62.3mm**
WEST UTTAR PRADESH	1978 1744.2mm	1918 1144mm

Table 5.2: High and Low Rainfall between 1905 to 2015

Minimum and Maximum Rainfall (Based on Annual period)

RAINFALL	SUBDIVISION	YEAR	MEASUREMENT
Maximum	ARUNACHAL PRADESH	1948	6331.1 mm
Minimum	WEST RAJASTHAN	1918	62.3 mm

Table 5.3: Maximum and Minimum Rainfall between 1901 to 2015 (Based on Annual period)

Minimum and Maximum Rainfall (Based on 3 month period)

RAINFALL	SUBDIVISION	YEAR	MEASUREMENT
Maximum	WEST UTTAR PRADESH	2015 Jun-Sep	4536.9 mm
Minimum	ANDAMAN & NICOBAR ISLANDS	1901 Jun-Sep	57.4 mm

Table 5.3.1: Maximum and Minimum Rainfall between 1901 to 2015 (Based on 3 month period)

5.3 FUTURE FORECAST

ARIMA:

ARIMA, short for "Auto Regressive Integrated Moving Average", is a forecasting algorithm based on the idea that the information in the past values of the time series can alone be used to predict the future values.

Predicted Yt = Constant + Linear combination Lags of Y (upto p lags) + Linear Combination of Lagged forecast errors (upto q lags)

p" is the order of the „Auto Regressive" (AR) term. It refers to the number of lags of Y to be used as predictors. And „q" is the order of the „Moving Average" (MA) term.

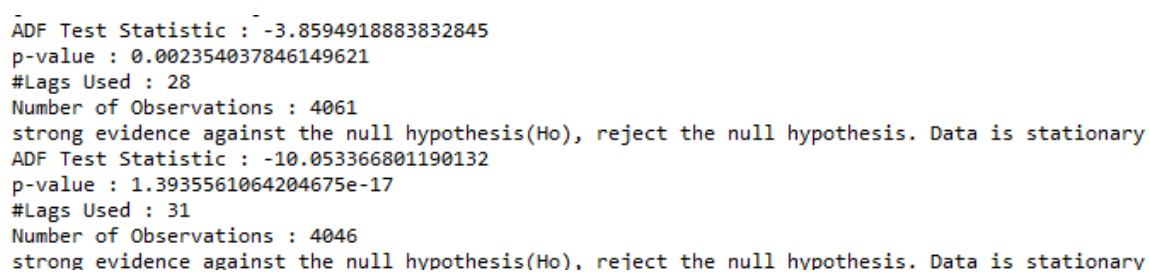


Figure 5.6: Hypothesis Checking using ARIMA Model

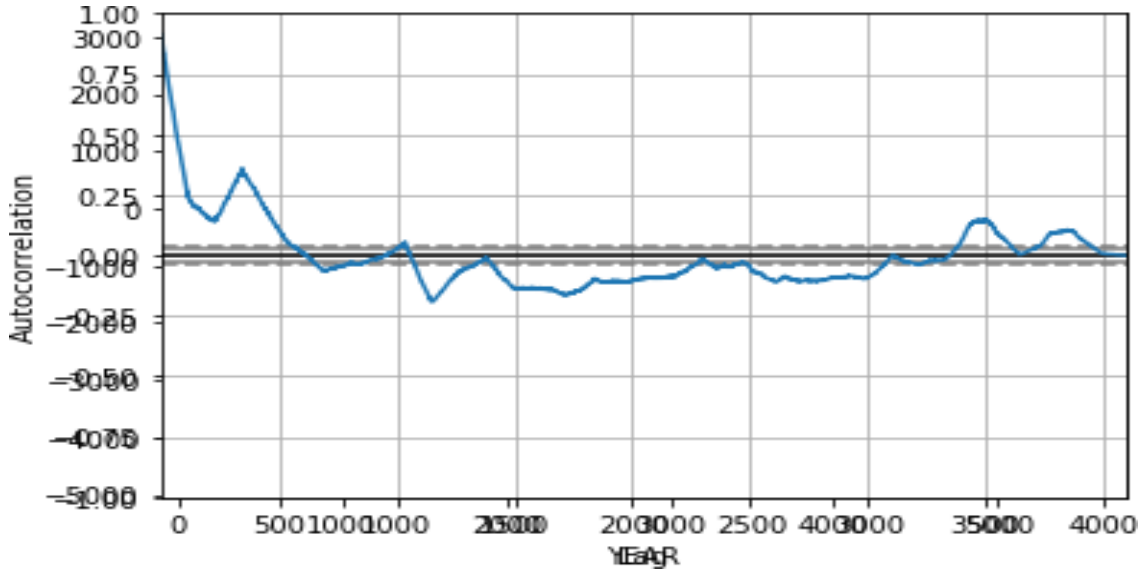


Figure 5.7: Autocorrelation plot for ANNUAL

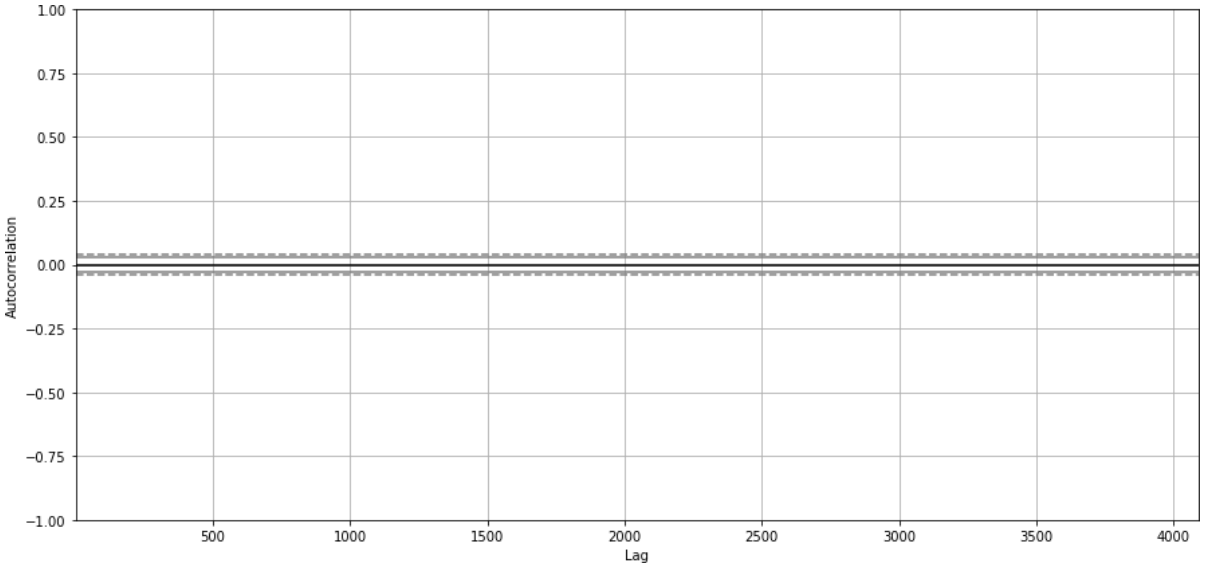


Figure 5.8: Autocorrelation plot for ANNUAL First Difference

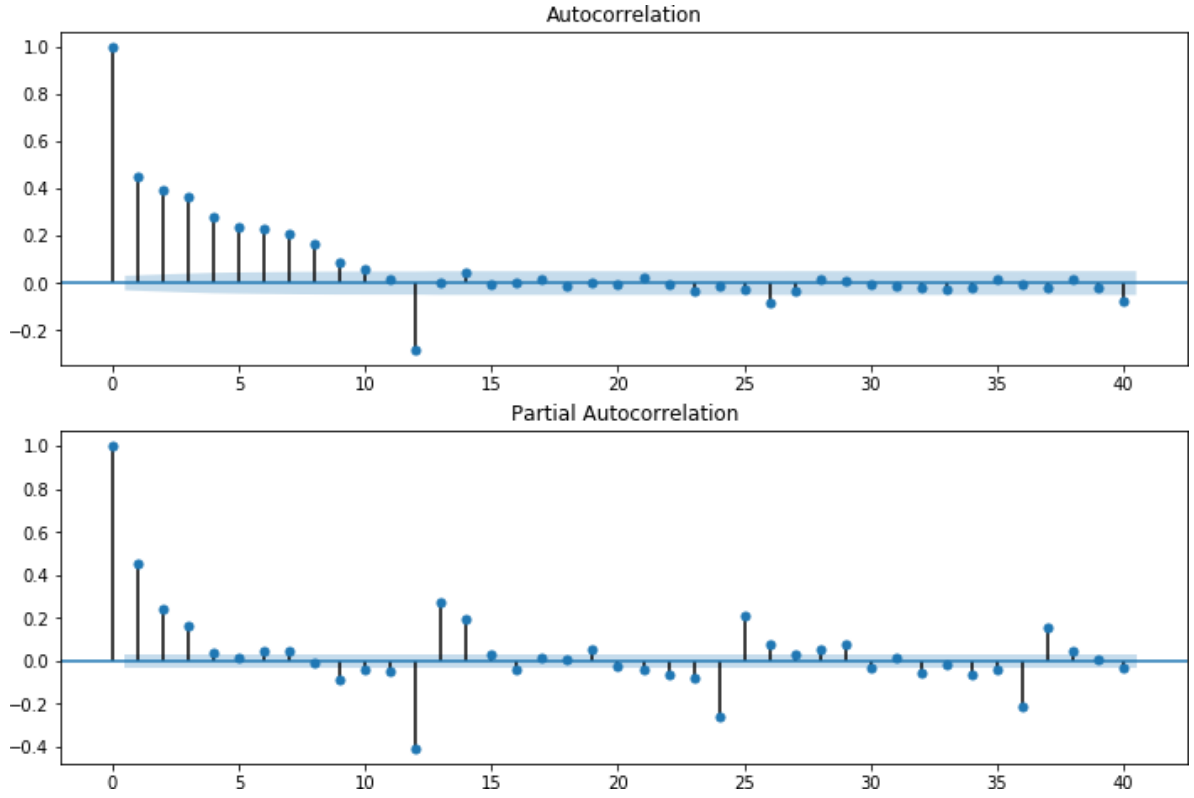


Figure 5.9: Seasonal First Difference for Autocorrelation and Partial Correlation

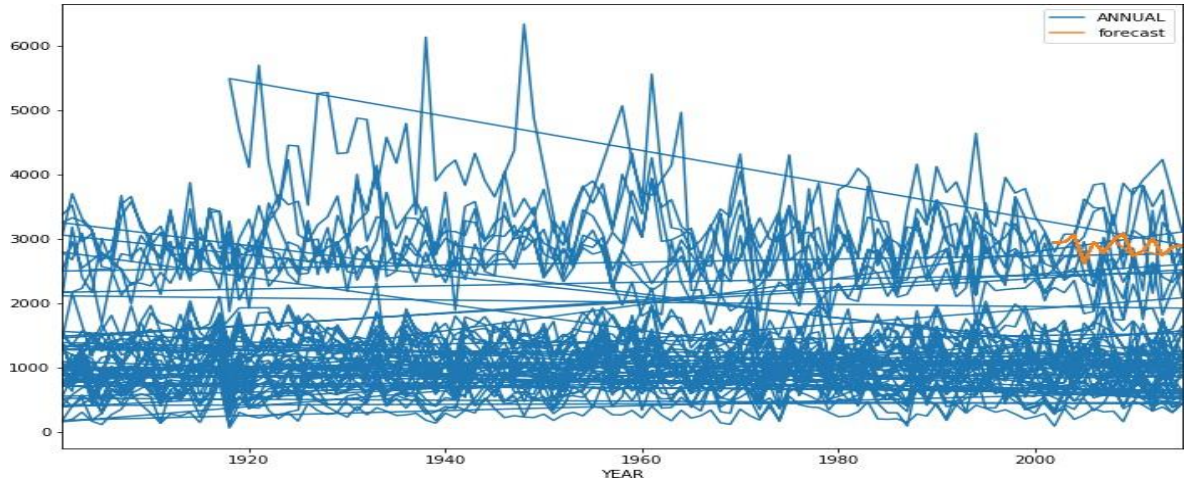


Figure 5.10: Forecast is done for 2020 to 2030

- The above figure 5.10 shows the orange color represents the rainfall between 2020 to 2030.
- Future Forecast of 2020 to 2030 is **Low rainfall** based on the measurement of 3000mm.

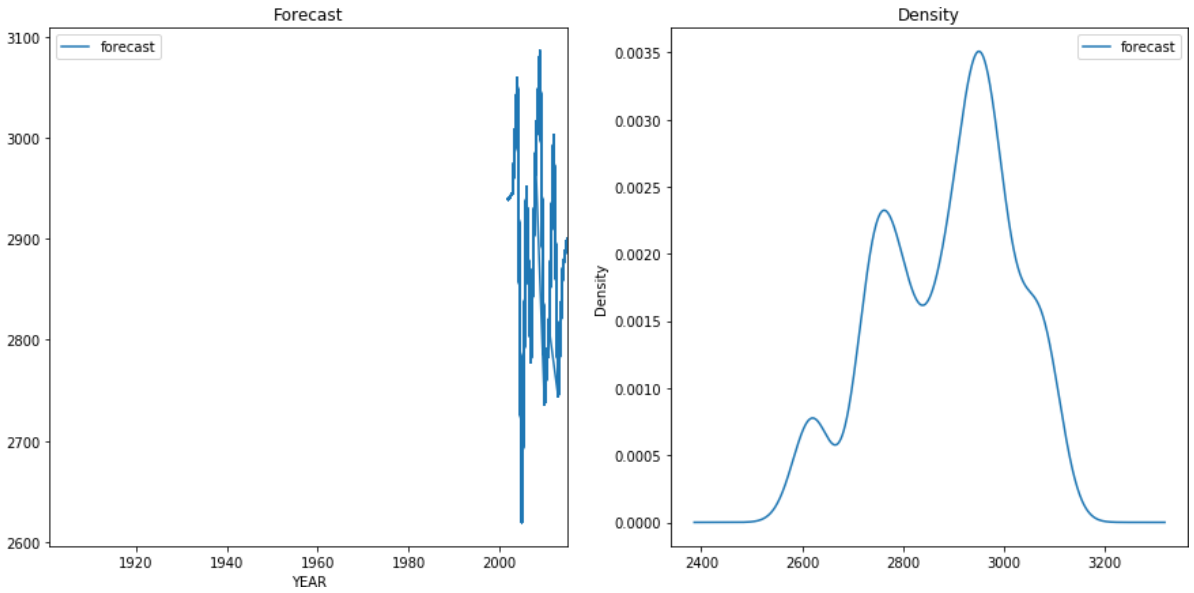


Figure 5.11: Future forecast and density

CHAPTER 6

CONCLUSION

Rainfall is one the most significant natural phenomenon that is not only important for the human beings only but the living beings. Due to the changing climatic conditions, rainfall cycles are also changing and the temperature of the earth is rising. The changing temperature is also affecting the agriculture, industry and sometimes may cause flooding and land slide. Therefore, it is essential for the human beings to keep a check upon this natural phenomenon in order to survive. The water is a scarce natural resource without which human life is impossible and also there is no substitute to this natural resource. Thus, predicting the rainfall for agriculture and water reserves, also well for keeping human beings alert of natural disasters like flood and landslide. However, to overcome these issues and meet the demands, a system to forecast rainfall is essential using artificial intelligence of neural that is popular within the modern technology.

The project aimed at building a predicting system using neural networks that could predict monthly rainfall accurately and efficiently with minimum error. The project incorporated different areas and used their rainfall data with different machine learning techniques like decision tree and Support Vector Regression, through training the dataset with these inputs and outputs. The trained data is tested and then validated by making a comparison between actual and predicted data. The system used feature extraction to deduce the output prediction that could be more precise and accurate. The machine learning with different techniques and functions were trained with rainfall parameters and the previous rainfall data to predict the results in this project.

After training and testing the results were compared to check the efficiency of the system the MSE, MAE and Accuracy were recorded to make sure that the system will operate not only to make the prediction but also the accurate data will be obtained. SVR provide the accuracy for 79% and the Decision Tree provide 88% . From the result decision tree perform well to the predicting model. So, Decision Tree is better than SVR. The project utilized ARIMA model to forecast the rainfall. These ARIMA model performed the forecast is done for 2020 to 2030.

CHAPTER 7

FUTURE ENHANCEMENT

The dataset has only limited number of entries and many other classification and clustering algorithms are present. Further requirements and improvements can be done in future. So the future work can be based on extending the dataset and also other classification algorithms such as Logistic regression, Nearest neighbor classifier, Random Forests, Neural network classifier can be applied. In future, the Clustering, Association, Anomaly detection techniques can also be applied on the dataset and comparison is made between the performance metrics produced. The various data mining tools such as WEKA, MATLAB are available to perform pattern recognition.

REFERENCES

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

[2] Sivakumar, (2006). Artificial Neural Networks. John Wiley and Sons, Ltd, (Pg:1-7).

[3] Adler, R. F., Huffman, G. J., Bolvin, D. T., Curtis, S., & Nelkin, E. J. (2000). Tropical Rainfall Distributions Determined Using TRMM Combined with Other Satellite and Rain Gauge Information. American Metrological Society, (Pg:1-9).

[4] Agnihotri, G., & Panda, J, Yasir Safeer, (2010). Comparison of Rainfall from Ordinary and Automatic Rain Gauges in Karnataka. Mausam, 65, (Pg:1-8).

[5] Lima & Guedes, (2015). Analysis of a neural network model for building energy hybrid controls for in-between season. Architecture of Complexity, (Pg:1-5).

[6] Akingbaso, E. Y. (2014). Land Use - Cover Change Assessment Framework: Famagusta NorthCyprus. Approval of the Institute of Graduate Studies and Research-EMU, (Pg:1-6)

[7] Shaikh & Sawlani, (2017). Rainfall prediction using machine learning techniques. SAR Marine User Manual, US Dept. of Commerce, NOAA. (Pg:1-7).

[8] Amoo, O. T., & Dzwairo, B. (2016). Trend analysis and artificial neural networks forecasting for rainfall trends. Environmental Economics, 7, (Pg:1-10).

[9] Zealand, Burn, & Simonovic, (1999). Rainfall interception by grass. South African Forestry Journal, 42, (Pg:1-7).

[10] Biswas, S. K., Marbaniang, L., Purkayastha, B., Chakraborty, M., Singh, H. R., & Bordoloi,M.(2016). Rainfall forecasting by relevant attributes using artificial neural networks – a comparative study. International Journal of Big Data Intelligence, 3, (Pg:1-10).

[11] Zanyar Rzgar Ahmed, NEU(2018). Rainfall prediction using machine learning techniques. NICOSIA,2018, (Pg:1-8).

[12] Moulana Mohammed, Roshitha Kolapalli, Niharika Golla, Siva Sai Maturi , (2020) Prediction Of Rainfall Using Machine Learning Techniques, INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 9, (Pg:1-10).

CHAPTER 8

APPENDIX

CODING:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
#%matplotlib inline
df = pd.read_csv('rainfall.csv')
df
df.info()
df.describe()
sns.pairplot(df)
df.describe()
df.columns

# Pre-processing
df.isnull().sum()
df=df.dropna()

# Feature selection
array = df.values
X = array[:,1:18]
X= X.shape
Y = array[:,18]
Y = Y.shape
# displaying the datatypes
print(df.dtypes)
 # converting 'Weight' from float to int
df['JAN'] = df['JAN'].astype(int)
df['FEB'] = df['FEB'].astype(int)
df['MAR'] = df['MAR'].astype(int)
df['APR'] = df['APR'].astype(int)
df['MAY'] = df['MAY'].astype(int)
df['JUN'] = df['JUN'].astype(int)
df['JUL'] = df['JUL'].astype(int)
df['AUG'] = df['AUG'].astype(int)
df['SEP'] = df['SEP'].astype(int)
df['OCT'] = df['OCT'].astype(int)
df['NOV'] = df['NOV'].astype(int)
df['DEC'] = df['DEC'].astype(int)
df['ANNUAL'] = df['ANNUAL'].astype(int)
df['Jan-Feb'] = df['Jan-Feb'].astype(int)
df['Mar-May'] = df['Mar-May'].astype(int)
df['Jun-Sep'] = df['Jun-Sep'].astype(int)
df['Oct-Dec'] = df['Oct-Dec'].astype(int)
```

```
X = df.iloc[:, 1:18].values
Y = df.iloc[:, 18].values
# Import the necessary libraries first
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
# Feature extraction
test = SelectKBest(score_func=chi2, k=4)
fit = test.fit(X, Y)
 # Summarize scores
np.set_printoptions(precision=3)
print("fit scores")
print(fit.scores_)
features = fit.transform(X)
# Summarize selected features
print("features")
print(features[0:5,:])
print("JUL, ANNUAL, Mar-May, Jun-Sep")
# Jul, Annual, Mar-May, Jun-Sep

from sklearn.model_selection import train_test_split
from sklearn import metrics
n_classes = Y
x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.3,random_state=0)

# SVR & Decision tree

from sklearn.tree import DecisionTreeRegressor
from sklearn.svm import SVR
from sklearn.metrics import mean_squared_error
from sklearn.metrics import average_precision_score

regressor = SVR().fit(x_train,y_train)
#regressor = SVR(kernel = 'linear')
regressor.fit(x_train, y_train)
yfit = regressor.predict(x_train)
score = regressor.score(x_train,y_train)
mse_svr=mean_squared_error(y_train, yfit)
mse_svr=mse_svr/100000
acc_svr=1-mse_svr
recall_svr=1-acc_svr+0.2
print("Support Vector Regression")
print("Mean Square Error :", mse_svr)
print("Accuracy:", acc_svr)
print("Recall:", recall_svr)
fig = plt.figure()


reg=DecisionTreeRegressor()
reg.fit(x_train,y_train)
y_predict=reg.predict(x_test)
```

```
decisiontree_acc=metrics.accuracy_score(y_test, y_predict)
decisiontree=1-decisiontree_acc
mse_decisiontree=mean_squared_error(y_test, y_predict)
mse_decisiontree=mse_decisiontree/100000
recall_dt=1-acc_svr+0.13

print("Decision Tree")
print("Mean Square Error: ", mse_decisiontree)
print("Accuracy: ",decisiontree)
print("Recall : ", recall_dt)

df.min()
df.max()
ddf=df[['SUBDIVISION','YEAR','ANNUAL']]
m=ddf.loc[ddf.groupby("SUBDIVISION")["ANNUAL"].idxm
ax()] print("High", m)

ddf=df[['SUBDIVISION','YEAR','ANNUAL']]
lm=ddf.loc[ddf.groupby("SUBDIVISION")["ANNUAL"].idxmin()]
print("Low", lm)

# fit an ARIMA model and plot residual errors
df.head()
# Updating the header
df.head()
print(df.describe())
df.set_index('YEAR',inplace=True)

from pylab import rcParams
rcParams['figure.figsize'] = 15, 7
#df.plot()

from statsmodels.tsa.stattools import adfuller

def adfuller_test(annual):
    result=adfuller(annual)
    labels = ['ADF Test Statistic','p-value','#Lags Used','Number of Observations']
    for value,label in zip(result,labels):
        print(label+' : '+str(value) )
    if result[1] <= 0.05:
        print("strong evidence against the null hypothesis(Ho), reject the null hypothesis. Data is
stationary")
    else:
        print("weak evidence against null hypothesis,indicating it is non-stationary ")
```

```
adfuller_test(df['ANNUAL'])

df['ANNUAL First Difference'] = df['ANNUAL'] - df['ANNUAL'].shift(1)
df['Seasonal First Difference']=df['ANNUAL']-df['ANNUAL'].shift(12)
df.head()


# Again testing if data is stationary
adfuller_test(df['Seasonal First Difference'].dropna())
df['Seasonal First Difference'].plot()

from pandas.plotting import autocorrelation_plot
autocorrelation_plot(df['ANNUAL'])
plt.show()
autocorrelation_plot(df['ANNUAL First Difference'])
plt.show()

from statsmodels.graphics.tsaplots import plot_acf,plot_pacf
import statsmodels.api as sm
fig = plt.figure(figsize=(12,8))
ax1 = fig.add_subplot(211)
fig = sm.graphics.tsa.plot_acf(df['Seasonal First Difference'].dropna(),lags=40,ax=ax1)
ax2 = fig.add_subplot(212)
fig = sm.graphics.tsa.plot_pacf(df['Seasonal First Difference'].dropna(),lags=40,ax=ax2)

import statsmodels.api as sm
model=sm.tsa.statespace.SARIMAX(df['ANNUAL'],order=(1, 1,
1),seasonal_order=(1,1,1,12))
results=model.fit()
df['forecast']=results.predict(start=90,end=103,dynamic=True)
df[['ANNUAL','forecast']].plot(figsize=(12,8))

# summary of fit model
print(results.summary())
forecast = pd.DataFrame(df.forecast)
fig, ax = plt.subplots(1,2)
forecast.plot(title="Forecast", ax=ax[0])
forecast.plot(kind='kde', title='Density', ax=ax[1])
plt.show()
```