**A PROJECT REPORT**

**ON**

**WEATHER PREDICTION USING MACHINE LEARNING ALGORITHMS**

Submitted in partial fulfillment for the Award of Credits To

**PROJECT WORK**

## IN BACHELOR OF TECHNOLOGY

## IN

**ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**



Submitted By:

## **S. SESHA RATNAM [21BJ1A54B2]**

**K. SIRISHA [21BJ1A5461]**

## **M. RESHMA [21BJ1A5439]**

**B. GNANA JYOTHI [21BJ1A5404]**

**R. NAGA PRANATHI [21BJ1A5450]**

*Under the esteemed guidance of*

#### DR JAIDEEP GERA ,M TECH

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

## **ST.MARY’S GROUP OF INSTITUTIONS GUNTUR**

**(**Affiliated to JNTU KAKINADA, Approved by AICTE, NEWDELHI & Accredited by ‘NAAC’)

**CHEBROLU (V&M), GUNTUR DIST., ANDHRA PRADESH, INDIA, PIN: 522212,**

**2021-2025**

**ST.MARY’S GROUP OF INSTITUTIONS GUNTUR**

**(**Affiliated to JNTU Kakinada, Approved by AICTE, NEW DELHI & Accredited by ‘NAAC’)

**CHEBROLU (V&M), GUNTUR DIST., ANDHRA PRADESH, INDIA, PIN: 522212**

**DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

****

# CERTIFICATE

## This is to certify that the project report entitled “**WEATHER PREDICTION USING MACHINE LEARNING ALGORITHMS**”is the bonafiedproject work carried out by **S.SESHARATNAM[21BJ1A54B2],K.SIRISHA[21BJ1A5461],M.RESHMA[21BJ1A5439],B.GNANA JYOTHI[21BJ1A5404],R.NAGA PRANATHI[21BJ1A5450].** during the academic year **2024-2025**, in partial fulfillment of the requirements to the award of the Credits in IV-II of **Bachelor of Technology In ARTIFICIAL INTELLIGENCE AND DATA SCIENCE** from **St. Mary’s Group of Institutions Guntur** of **Jawaharlal Nehru Technological University, Kakinada.**

|  |  |
| --- | --- |
| PROJECT GUIDEDR JAIDEEP GERA , M TECHAssociate professor | HEAD OF THE DEPARTMENTDr. S S N ANJANEYULU M.Tech, Ph.D |

# PROJECT CO-ORDINATOR EXTERNAL EXAMINER

# ACKNOWLEDGEMENT

On the occasion of presenting the dissertation, we would like to thank the **Almighty** for providing strength and power by his enormous blessings to overcome all the hurdles and hindrances during this project work.

First & Foremost, We would like to express my sincere gratitude to our guide **DR JAIDEEP GERA**, M TECH Department of Artificial Intelligence and Data Science, St. Mary’s Group of Institutions Guntur, for the continuous support of our B. Tech study and research, for his patience, motivation, enthusiasm, and immense knowledge.His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for our B. Tech study.

We also thank Chairman Sir **Rev. Dr.** **K.V.K. RAO garu**, St. Mary’s Group of Institutions Guntur for providing necessary facilities to carry out this research work.

Besides our advisor, we would like to thank **Dr. SUKESH BABU CH, Principal** & **Dr. S S N ANJANEYULU, Head of The Department,** St. Mary’s Group of Institutions Guntur, for their guidance, support, encouragement and valuable suggestions.

We also express my deepest sense of gratitude to all Lab technicians of SMGG. We would like to tthank all the teaching and non-teaching staff of who were very helpful in completion of our project work successfully.

We would extend my heartfelt thanks to our family: our parents for giving birth to me at the first place and supporting me spiritually throughout my life.

We are fortunate to have friends and classmates**,** who constantly helped me throughout the period, this study was carried out. I extend my hearty gratitude to all those who have directly or indirectly helped me during this research. Thanks to one and all….

**ST.MARY’S GROUP OF INSTITUTIONS GUNTUR**

**(**Affiliated to JNTU, KAKINADA, Approved by AICTE, NEWDELHI & Accredited by ‘NAAC’)

**CHEBROLU (V&M), GUNTUR DIST., ANDHRA PRADESH, INDIA, PIN: 522212**

**DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**



# DECLARTION

We, **S.SESHA RATNAM [21BJ1A54B2], K.SIRISHA [21BJ1A5461], M.RESHMA [21BJ1A5439], B.GNANA JYOTHI [21BJ1A5405] , R.NAGA PRANATHI [21BJ1A5450]** Students of B. Tech, ST.MARY’S GROUP OF INSTITUTIONS GUNTUR, Jawaharlal Nehru Technological University Kakinada, A.P, do hereby Declare that the Project Report Entitled **“WEATHER PREDICTION USING MACHINE LEARNING ALGORITHMS”** is the Genuine Work carried out at ST.MARY’S GROUP OF INSTITUTIONS GUNTUR, under the Guidance of **DR JAIDEEP GERA ASSOCIATE PROFESSOR**, Department Of Computer Science and Engineering, ST.MARY’S GROUP OF INSTITUTIONS GUNTUR. We declare that the Work embodied in the thesis has not been submitted for the award of credits for Degree or Diploma of this or any other University.

**S. SESHA RATNAM [21BJ1A54B2]**

**K. SIRISHA [21BJ1A5461]**

**M. RESHMA [21BJ1A5439]**

**B. GNANA JYOTHI [21BJ1A5404]**

**R. NAGA PRANATHI [21BJ1A5450]**

***Index***

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**WEATHER PREDICTION USING MACHINE LEARNING ALGORITHMS**

**Abstract:**

Weather forecasts have grown increasingly significant in recent years since they can save us time, money, property, or even our lives. Despite the fact that India has a large number of weather stations, they are mainly located in inhabited regions such as cities, suburbs, or towns. This makes weather forecasting in isolated regions more imprecise, which can be inconvenient for individuals such as farmers who rely largely on weather reports in their daily work. In this paper, we are predicting the weather by analyzing features like temperature, apparent temperature, humidity, wind speed, wind bearing, visibility, cloud cover with Random Forest, Decision Tree, MLP classifier, Linear regression, and Gaussian naive Bayes are examples of machine learning methods. Based on the results obtained a comparative study is done concerning the accuracy.

**CHAPTER 1**

**INTRODUCTION**

A Weather prediction is used to predict the current weather situation. The application of physics principles, augmented by a range of statistical and analytical methods, to predict the weather is known as weather forecasting. Weather forecasting provides forecasts of shifts in the Earth's surface temperature in addition to predictions of atmospheric phenomena. These shifts are influenced by atmospheric conditions such as snow and ice cover. The foundation for weather prediction began with ancient Greek philosophers' theories and continued with Renaissance scientists. Any weather prediction requires a systematic compilation of weather records from different locations, as well as adequate data analysis and prediction. In this work, we propose a method to utilize surrounding city's historical weather data along with a particular city's data to predict its weather condition. We combine these data and use it to train simple machine learning models, which in turn, can predict correct weather conditions for the next few days

**Objective:**

Weather Forecasting is crucial since it helps to determine future climate changes. With the use of latitude, we can determine the probability of snow and hail reaching the surface. We are able to identify the thermal energy from the sun that is exposed to a region.

**Problem statement:**

Weather prediction is a useful tool for informing populations of expected weather conditions. Weather prediction is a complex topic and poses significant variation in practice. We will attempt to understand and implement a weather prediction application

**Overview of the project**

We have developed the system in such a way that it can be used efficiently by all the users. Let’s Assume all the webpages and logics related to admin and users as separate modules. We have two modules and each of these modules have their own unique functions. Important thing is that it is not designed for particular type peoples. This application can help all the users to access the weather report details through internet or intranet that means anytime and anywhere. We wanted to create a tool that fits into modern age, but still stays true to the “concepts of studying”. So, this process also helps in maintaining consistency and integrity. This system helps the user to generate the dynamic legwork.

**Thesis Outline:**

The thesis is structured as follows:

* **Chapter 1**: introduces the problem statement, research motivation, and objectives of the project.
* **Chapter 2**: provides a literature review
* **Chapter 3:** The system analysis recommendation behind the current recommended approaches.
* **Chapter 4 :** To explain system study of the project
* **Chapter 5:** Details the system research that was conducted under various circumstances. To includeUML diagrams are used to describe the system design.
* **Chapter 6:** To explain software description with machine learning
* **Chapter 7**:To explain simulative study and implementation of system module description**.**
* **Chapter 8:** To evaluatesource code experimental results and analysis
* **Chapter 9:** To conclude as per project Conclusions and Scope for Future Research

**CHAPTER 2:**

**LITERATURE REVIEW**

[1] **Sunita B. Aher and L. M. R. J. Lobo:**This paper likely compares different classification algorithms used in data mining or machine learning. It could explore algorithms such as Decision Trees, Support Vector Machines, K-Nearest Neighbors, etc., comparing their performance on various datasets, highlighting advantages and disadvantages in terms of accuracy, computational complexity, and suitability for different tasks.

[2] **C. Lakshmi Devasena:**  This study compares several popular classification algorithms, specifically Random Forest, REP Tree, and J48, with a focus on how they can be applied to credit risk prediction. The paper might examine which classifier performs best in terms of predicting whether a borrower will default on a loan, using datasets such as those from financial institutions.

[3] **Holmstrom Mark, Dylan Liu, and Christopher Vo,** This paper likely explores how machine learning algorithms (such as Random Forests, Neural Networks, etc.) can be used to improve weather forecasting models. The authors would have discussed how these techniques could predict weather patterns more accurately by processing historical weather data and potentially other factors like geographic features.

**[4]Kalmegh Sushilkumar,** The paper likely compares different classifiers implemented in Weka (a popular data mining tool), focusing on REPTree, SimpleCART, and RandomTree. The goal would be to classify Indian news articles into categories such as politics, sports, entertainment, etc., while analyzing the performance of each algorithm in terms of accuracy and computational cost.

[5] **Vrushali Y. Kulkarni and Pradeep K. Sinha,** This paper likely discusses the Random Forest algorithm, an ensemble learning method that combines multiple decision trees. The authors would have explored its effectiveness in solving classification problems, potentially comparing its performance to other algorithms and demonstrating its advantages in terms of accuracy and robustness to overfitting.

[6] **Nalluri Sravani, Ramasubbareddy Somula, and G Kannayaram:** The paper probably focuses on applying clustering algorithms, such as K-means or hierarchical clustering, to predict weather patterns. The study could show how unsupervised learning can be beneficial for analyzing and predicting weather, perhaps by grouping similar weather events together and identifying patterns.

**[7]K. Hanumantha Rao et al.:** This paper likely discusses different machine learning algorithms for anomaly detection, which involves identifying unusual patterns in data that deviate from the norm. Applications could range from fraud detection to system health monitoring.

**[8] Shashank Singh, Faraz Ahmed, Nagrami, and Pillai Aditya:** Similar to the paper by Holmstrom et al., this paper would also apply machine learning techniques to weather forecasting. It would describe how modern techniques like deep learning or other advanced models could be leveraged to predict weather conditions more accurately and potentially in real-time.

[9] **T R Prajwala:** The focus of this paper is likely on comparing two popular machine learning algorithms: Decision Trees and Random Forests. The study could demonstrate how these algorithms are implemented using the R programming language, comparing their accuracy, ease of implementation, and performance on different datasets.

**CHAPTER – 3**

**SYSTEM ANALYSIS**

Weather prediction using machine learning has been an area of growing interest, leveraging advanced algorithms like Random Forest and decision tree to improve accuracy and reliability. Random Forest, an ensemble learning technique, excels in handling large datasets and complex patterns by combining predictions from multiple decision trees. This robustness makes it effective for capturing nonlinear relationships in meteorological data. decision tree , a simple yet powerful algorithm, classifies data points based on proximity to their tree decisions, making it suitable for localized weather predictions. Existing studies have demonstrated the effectiveness of these algorithms in tasks such as temperature forecasting, rainfall prediction, and wind speed estimation, showcasing their potential to enhance traditional methods by reducing errors and providing more granular insights into weather patterns.

**Dataset**

Weather prediction relies on datasets that include variables such as temperature, humidity, wind speed, pressure, and precipitation, collected over time from weather stations and satellites. These datasets are analyzed using statistical models or machine learning algorithms to forecast future weather conditions. By examining patterns and trends in the data, predictions can be made for specific locations and time frames, helping to provide accurate weather forecasts for the coming hours, days, or weeks**.**

**Dataset pre processing**

Weather prediction dataset preprocessing involves cleaning and preparing the data for analysis. This includes handling missing values, outliers, and noise, as well as converting raw data into a usable format. Steps like normalization or standardization may be applied to scale variables like temperature and wind speed. Categorical data, such as wind direction, might be encoded numerically. Temporal features, like dates and times, are often extracted and transformed to capture seasonal patterns. After preprocessing, the dataset is ready for training machine learning models to make accurate weather predictions.

**Model evolution**

Model evaluation in weather prediction involves assessing the performance of predictive models using various metrics. Common evaluation techniques include splitting the dataset into training and testing sets to assess the model's ability to generalize. Metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared are typically used to measure prediction accuracy. Cross-validation may also be employed to ensure the model performs consistently across different data subsets. By evaluating the model on unseen data, it helps identify its strengths, weaknesses, and overall reliability for forecasting weather conditions.

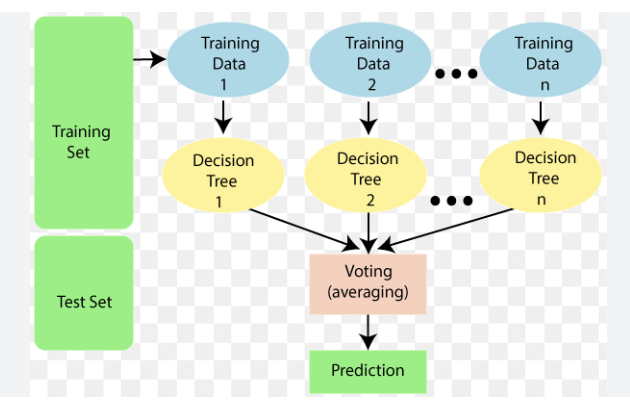
Random forest is a commonly-used machine learning algorithm, trademarked by Leo Breiman and Adele Cutler, that combines the output of multiple decision trees to reach a single result. Its ease of use and flexibility have fueled its adoption, as it handles both classification and regression problems. Since the random forest model is made up of multiple decision trees, it would be helpful to start by describing the decision tree algorithm briefly. Decision trees start with a basic question, such as, “Should I surf?” From there, you can ask a series of questions to determine an answer, such as, “Is it a long period swell?” or “Is the wind blowing offshore?”. These questions make up the decision nodes in the tree, acting as a means to split the [data](https://www.ibm.com/consulting/analytics" \t "_self). Each question helps an individual to arrive at a final decision, which would be denoted by the leaf node. Observations that fit the criteria will follow the “Yes” branch and those that don’t will follow the alternate path. Decision trees seek to find the best split to subset the data, and they are typically trained through the Classification and Regression Tree (CART) algorithm. Metrics, such as Gini impurity, information gain, or mean square error (MSE), can be used to evaluate the quality of the split.

This decision tree is an example of a classification problem, where the class labels are "surf" and "don't surf."

While decision trees are common supervised learning algorithms, they can be prone to problems, such as bias and overfitting. However, when multiple decision trees form an ensemble in the random forest algorithm, they predict more accurate results, particularly when the individual trees are uncorrelated with each other.

The random forest algorithm is an extension of the bagging method as it utilizes both bagging and feature randomness to create an uncorrelated forest of decision trees. Feature randomness, also known as feature bagging or “[the random subspace method](https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf" \t "_blank)”(link resides outside ibm.com), generates a random subset of features, which ensures low correlation among decision trees. This is a key difference between decision trees and random forests. While decision trees consider all the possible feature splits, random forests only select a subset of those features.

A random forest is an ensemble of decision trees. Each tree in the forest is trained on a random subset of the data and makes an independent prediction. The final prediction is based on the majority vote (classification) or the average (regression) of the predictions from individual trees.



**Fig:** Random Forest Structure

**Weather Prediction using Random Forest** involves utilizing an ensemble learning method where multiple decision trees are built and their outputs are aggregated to predict weather conditions. Each tree in the Random Forest model is trained on a random subset of features (such as temperature, humidity, wind speed, and atmospheric pressure), and the final prediction is based on majority voting (for classification) or averaging (for regression). This method effectively handles complex relationships between features and can capture non-linear patterns in the data.

**Node splits**: At each decision point, the tree splits the data based on which feature (e.g., batting average, team composition, match venue) results in the best division of the data. This is typically done by looking for the feature that minimizes

**Classification (Match Outcome Prediction)**: If the task is to predict match outcomes (win or loss), each decision tree in the Random Forest will predict a result. The final prediction is made by taking a **majority vote** among all the trees. For example, if 60 trees predict a win and 40 trees predict a loss, the final prediction will be a "win."

**Regression (Player Performance Prediction)**: If the task is predicting continuous variables like player scores or wickets, the final prediction will be the **average** of all tree predictions. For example, if individual trees predict different scores, their average will be the predicted score.

Random Forest also handles missing values and outliers well and reduces the risk of over fitting for interactions between multiple weather features, and provide reliable, accurate forecasts. compared to individual decision trees. It is particularly useful for weather prediction, as it can manage large datasets, account

**Draw backs**

* The atmospheric system has an intrinsic limit that represents a natural and ultimate boundary beyond which prediction is no longer possible.

**Proposed System:**

The proposed system leverages the Naïve Bayes algorithm for efficient and accurate weather prediction by modelling meteorological data using probabilistic principles. The system processes historical weather datasets, such as temperature, humidity, pressure, and wind speed, to calculate conditional probabilities and predict weather outcomes. By assuming feature independence, Naïve Bayes simplifies the computational process, making it highly efficient for real-time forecasting. The system incorporates preprocessing steps like normalization and feature selection to enhance model performance. Additionally, it integrates user-friendly interfaces for visualization and real-time updates, aiming to provide accurate, interpretable predictions for applications such as agriculture, disaster management, and urban planning**.**

**Dataset**

Weather prediction relies on datasets that include variables such as temperature, humidity, wind speed, pressure, and precipitation, collected over time from weather stations and satellites. These datasets are analyzed using statistical models or machine learning algorithms to forecast future weather conditions. By examining patterns and trends in the data, predictions can be made for specific locations and time frames, helping to provide accurate weather forecasts for the coming hours, days, or weeks

**Dataset pre processing**

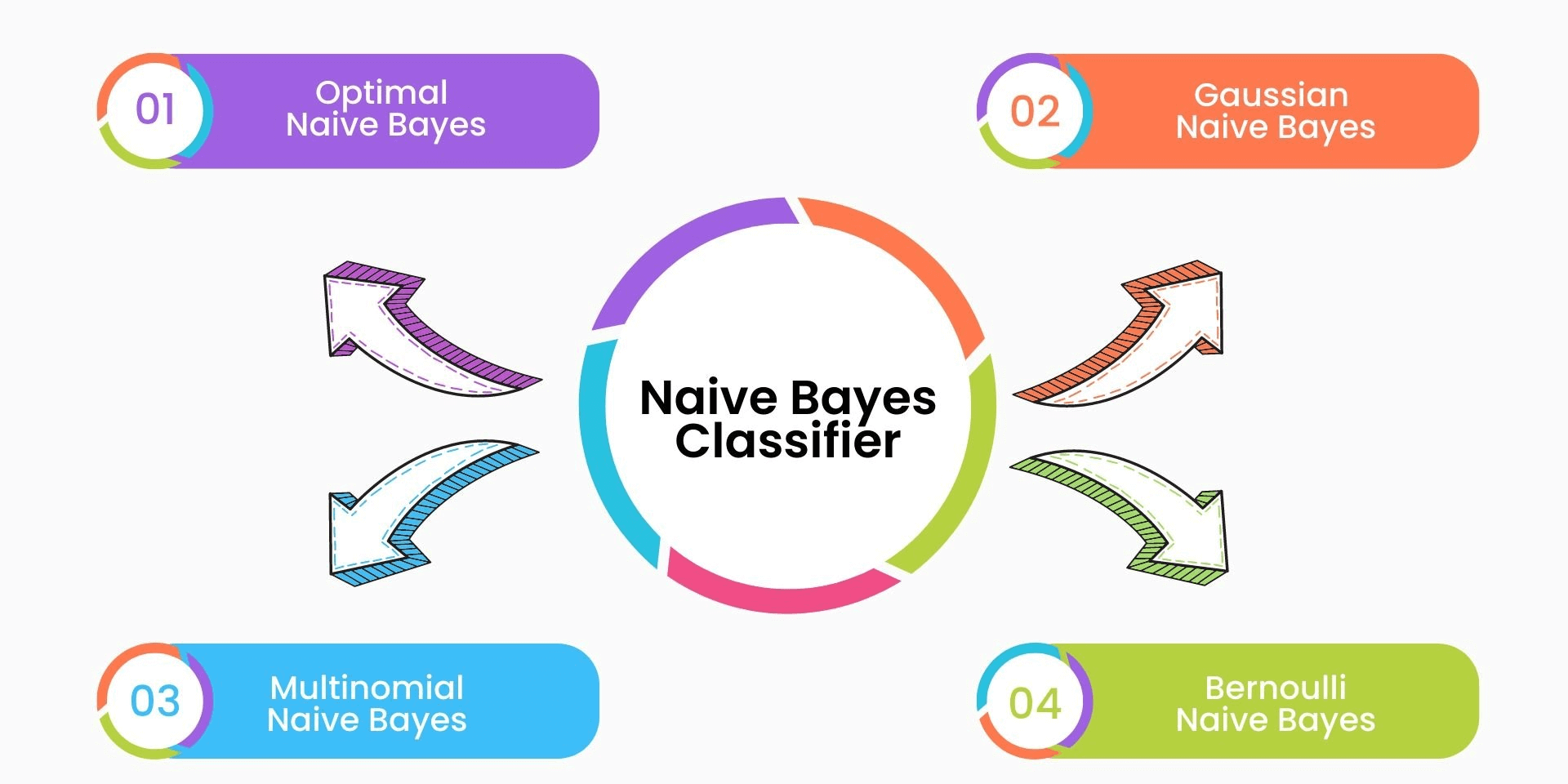
Weather prediction dataset preprocessing involves cleaning and preparing the data for analysis. This includes handling missing values, outliers, and noise, as well as converting raw data into a usable format. Steps like normalization or standardization may be applied to scale variables like temperature and wind speed. Categorical data, such as wind direction, might be encoded numerically. Temporal features, like dates and times, are often extracted and transformed to capture seasonal patterns. After preprocessing, the dataset is ready for training machine learning models to make accurate weather predictions.

**Model evolution**

Model evaluation in weather prediction involves assessing the performance of predictive models using various metrics. Common evaluation techniques include splitting the dataset into training and testing sets to assess the model's ability to generalize. Metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared are typically used to measure prediction accuracy. Cross-validation may also be employed to ensure the model performs consistently across different data subsets. By evaluating the model on unseen data, it helps identify its strengths, weaknesses, and overall reliability for forecasting weather conditions.

Naive Bayes is a machine learning algorithm that is used by data scientists for classification. The naive Bayes algorithm works based on the Bayes theorem. Before explaining Naive Bayes, first, we should discuss Bayes Theorem. Bayes theorem is used to find the probability of a hypothesis with given evidence. This beginner-level article intends to introduce you to the Naive Bayes algorithm and explain its underlying concept and implementation.

Naive Bayes being fast and simple, its assumption of feature independence can limit its effectiveness in detecting more complex, correlated patterns in crime data. In practice, Naive Bayes may perform well in scenarios with clear, separable features but might require further tuning

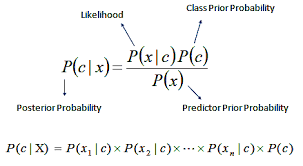


**Fig:** Navie Bias

**Naive Bayes (NB) for Spam Detection:**

Naive Bayes is a probabilistic classifier that operates on Bayes’ Theorem, assuming that the features (such as words or attributes in an email) are conditionally independent. Here's how it works in spam detection:

1. **Training**: During training, the model learns the probability distribution of words for both spam and non-spam (ham) emails. For example, certain words like "offer", "free", and "discount" might be more likely to appear in spam emails, while words like "meeting" or "schedule" might be more likely in ham emails.
2. **Classification**: When a new email comes in, Naive Bayes calculates the likelihood that the email is spam or ham based on the observed words in the email. It does this by applying Bayes' Theorem:



**Weather Prediction using Naive Bayes** is a probabilistic approach that classifies weather conditions based on historical data such as temperature, humidity, wind speed, and atmospheric pressure. The Naive Bayes algorithm calculates the probability of each weather condition occurring given the features, assuming that the features are independent. After training on historical weather data, it can predict future weather conditions. Despite its simplicity and the assumption of feature independence, Naive Bayes can perform well for basic weather prediction tasks, especially in situations where the data is clean and the relationships between features are not too complex. However, for more accurate and complex weather forecasting, more advanced models like decision trees, neural networks, or ensemble methods are generally preferred. Overall, Naive Bayes provides a lightweight, fast, and interpretable solution for weather prediction, suitable for real-time applications where speed is critical

**Advantages**

* Naive Bayes is computationally light, making it suitable for real-time weather prediction applications.
* It is easy to implement and requires fewer computational resources compared to more complex models.

**CHAPTER-4**

**SYSTEM STUDY**

**FEASIBILITY STUDY**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

**ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### **TECHNICAL FEASIBILITY**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**SOCIAL FEASIBILITY**

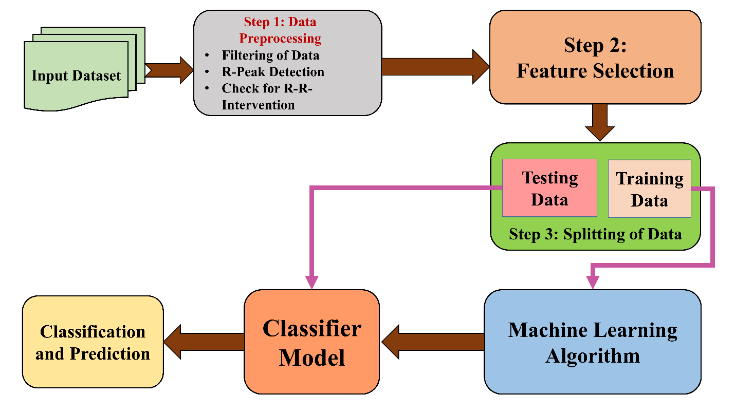
The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**CHAPTER-5**

**SYSTEM DESIGN**

**5.1. System Architecture:**

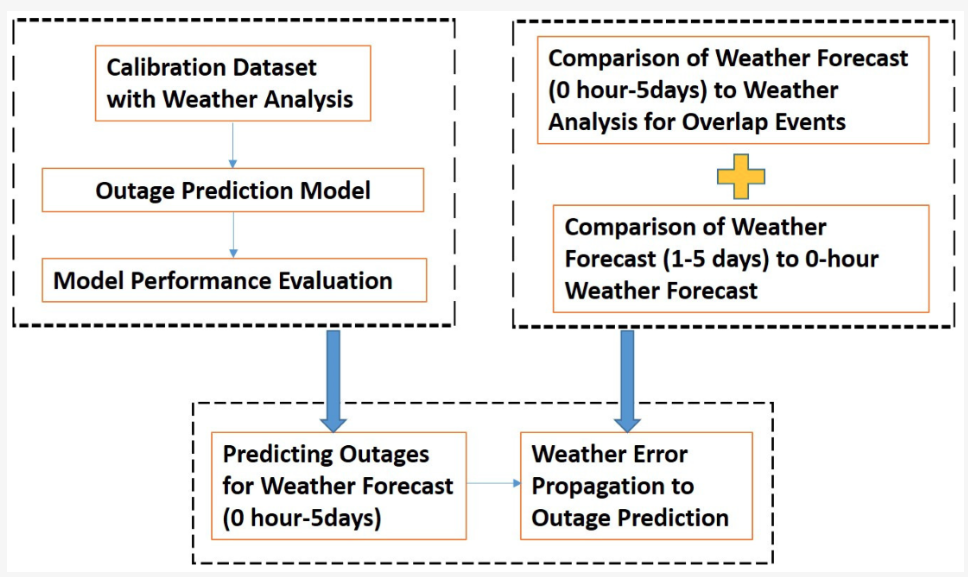
A weather prediction system using machine learning typically consists of several key modules designed to collect, process, and predict weather patterns. The **Data Collection Module** gathers historical and real-time weather data from various sources such as weather stations, satellites, and IoT sensors, which include features like temperature, humidity, wind speed, and atmospheric pressure. This data is stored in the **Data Storage Module**, which uses cloud or local databases for easy retrieval. The **Data Preprocessing Module** cleans and transforms the data, handling missing values, outliers, and normalizing features for model training.



**Fig: System Architecture**

**5.2** **DATA FLOW DIAGRAM:**

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.



**Fig: Data Flow Diagram**

**5.3 UML DIAGRAMS**

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

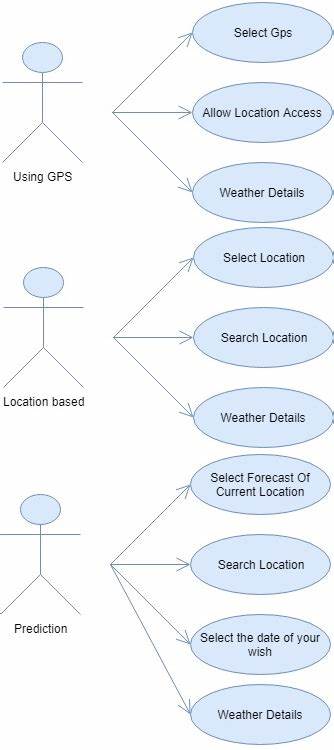
**GOALS:**

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

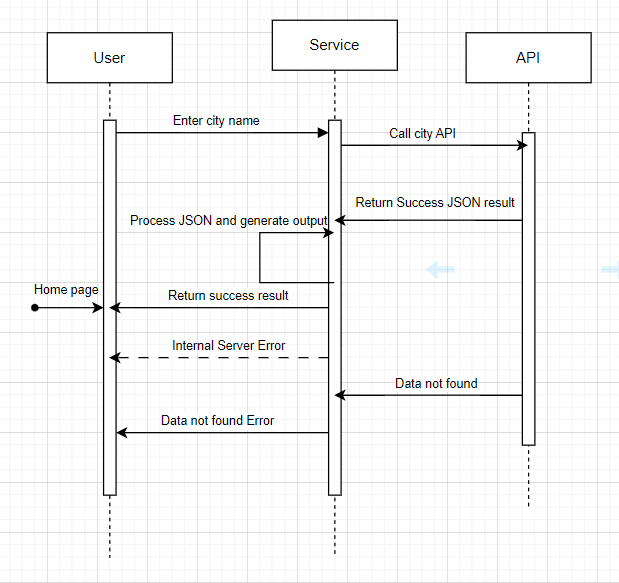
**5.3.1 USE CASE DIAGRAM:**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



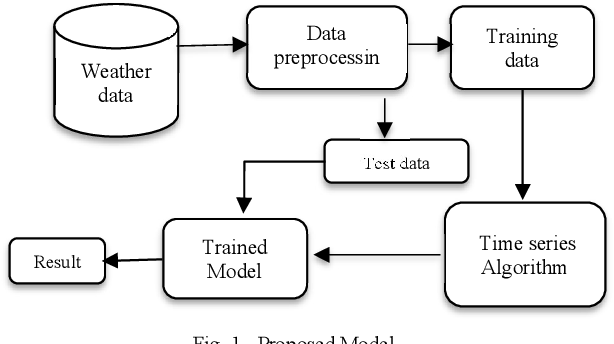
**Fig: USE CASE DIAGRAM**

**Sequence diagram**



**CLASS DIAGRAM:**

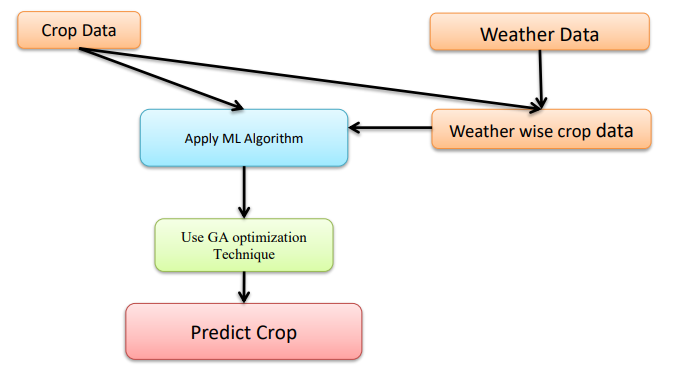
The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes can be either an "is-a" or "has-a" relationship. Each class in the class diagram may be capable of providing certain functionalities. These functionalities provided by the class are termed "methods" of the class. Apart from this, each class may have certain "attributes" that uniquely identify the class.



**Fig:** class diagram

**STATE DIAGRAM:**

A state diagram, as the name suggests, represents the different states that objects in the system undergo during their life cycle. Objects in the system change states in response to events. In addition to this, a state diagram also captures the transition of the object's state from an initial state to a final state in response to events affecting the system.



**CHAPTER-6**

**SOFTWARE ENVIRONMENT**

**6.1 INTRODUCTION TO MACHINE LEARNING:**

Arthur Samuel coined the term “Machine Learning” in 1959 and defined it as a “Field of study that gives computers the capability to learn without being explicitly programmed”. And that was the beginning of Machine Learning! In modern times, Machine Learning is one of the most popular (if not the most!) career choices. According to Indeed, Machine Learning Engineer is The Best Job of 2019 with a 344% growth and an average base salary of $146,085 per year. But there is still a lot of doubt about what exactly is Machine Learning and how to start learning it? So this article deals with the Basics of Machine Learning and also the path you can follow to eventually become a full-fledged Machine Learning Engineer.

Now let’s get started!!! Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data. Fundamentally, machine learning involves building mathematical models to help understand data.

"Learning" enters the fray when we give these models tunable parameters that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model based "learning" is similar to the "learning" exhibited by the human brain. Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

Machine learning uses data to detect various patterns in a given dataset.

* It can learn from past data and improve automatically.
* It is a data-driven technology.
* Machine learning is much similar to data mining as it also deals with the huge amount of the data.

**6.2 HOW DOES MACHINE LEARNING WORK?**

A Machine Learning system learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately. Machine learning tasks are classified into several broad categories. In supervised learning, the algorithm builds a mathematical model from a set of data that contains both the inputs and the desired outputs.

For example, if the task were determining whether an image contained a certain object, the training data for a supervised learning algorithm would include images with and without that object (the input), and each image would have a label (the output) designating whether it contained the object. In special cases, the input may be only partially available, or restricted to special feedback Semi-supervised learning algorithms develop mathematical models from incomplete training data, where a portion of the sample input doesn't have labels.

Classification algorithms and regression algorithms are types of supervised learning. Classification algorithms are used when the outputs are restricted to a limited set of values. For a classification algorithm that filters emails, the input would be an incoming email, and the output would be the name of the folder in which to file the email. For an algorithm that identifies spam emails, the output would be the prediction of either "spam" or "not spam", represented by the Boolean values true and false.

Regression algorithms are named for their continuous outputs, meaning they may have any value within a range. Examples of a continuous value are the temperature, length, or price of an object. 17 In unsupervised learning, the algorithm builds a mathematical model from a set of data that contains only inputs and no desired output labels. Unsupervised learning algorithms are used to find structure in the data, like grouping or clustering of data points. Unsupervised learning can discover patterns in the data, and can group the inputs into categories, as in feature learning.

Dimensionality reduction is the process of reducing the number of features, or inputs, in a set of data. Active learning algorithms access the desired outputs (training labels) for a limited set of inputs based on a budget and optimize the choice of inputs for which it will acquire training labels. When used interactively, these can be presented to a human user for labeling. Reinforcement learning algorithms are given feedback in the form of positive or negative reinforcement in a dynamic environment and are used in autonomous vehicles or in learning play a game against a human opponent Other specialized algorithms in machine learning include topic modeling, where the computer program is given a set of natural language documents and finds other documents to that cover similar topics.

Machine learning algorithms can be used to find the unobservable probability density function in density estimation problems. Meta learning algorithms learn their own inductive bias based on previous experience. In developmental robotics, robot learning algorithms generate their own sequences of learning experiences, also known as a curriculum, to cumulatively acquire new skills through self-guided exploration and social interaction with humans. These robots use guidance mechanisms such as active learning, maturation, motor synergies, and imitation.

**Applications of Machines Learning:**

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach.

Following are some real-world applications of ML –

➢ Emotion analysis

➢ Sentiment analysis

➢ Error detection and prevention

➢ Weather forecasting and prediction

➢ Stock market analysis and forecasting

➢ Speech synthesis

➢ Speech recognition

➢ Customer segmentation

➢ Object recognition

➢ Fraud detection

➢ Fraud prevention

➢ Recommendation of products to customer in online shopping

**Advantages of Machine learning:**

* Easily identifies trends and patterns - Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.
* No human intervention needed - With ML, you don’t need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.
* Continuous Improvement As ML algorithms gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.
* Handling multi-dimensional and multi-variety data Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.
* Wide Applications You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

**Disadvantages of Machine Learning:**

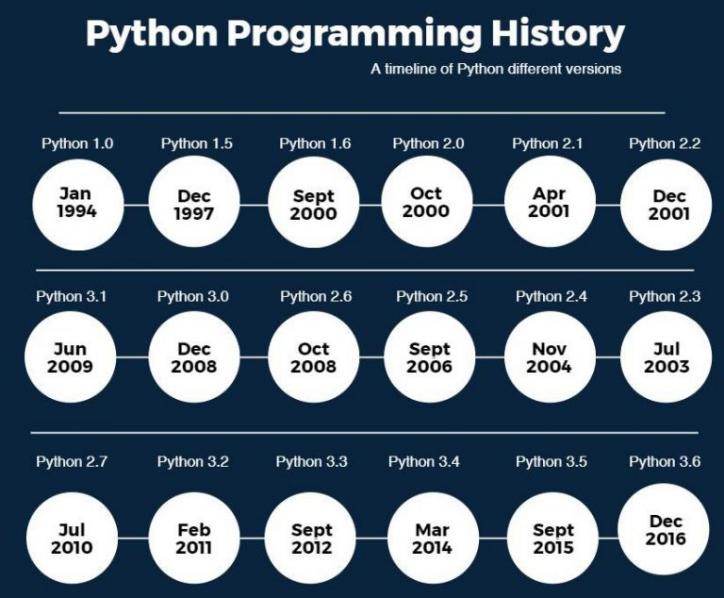
* Data Acquisition Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.
* Time and Resources ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.
* Interpretation of Results Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.
* High error-susceptibility Machine Learning is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

**6.3 PYTHON**

Guido Van Rossum published the first version of Python code (version 0.9.0) at outsources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others. It was also object oriented and had a module system. Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked. Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting Unicode. Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released.

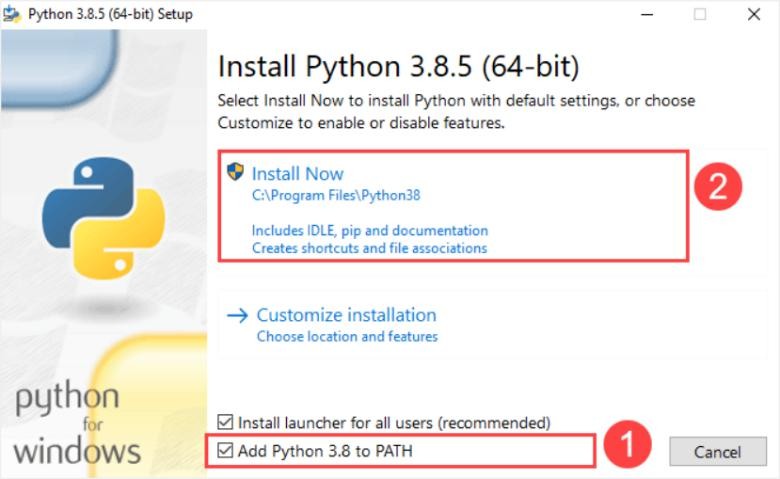
**Why the name Python?**

No. It wasn't named after a dangerous snake. Rossum was fan of a comedy series from late 70s. The name "Python" was adopted from the same series "Monty Python's Flying Circus". Python Version HistoryImplementationstarted-December1989 Internal releases – 1990

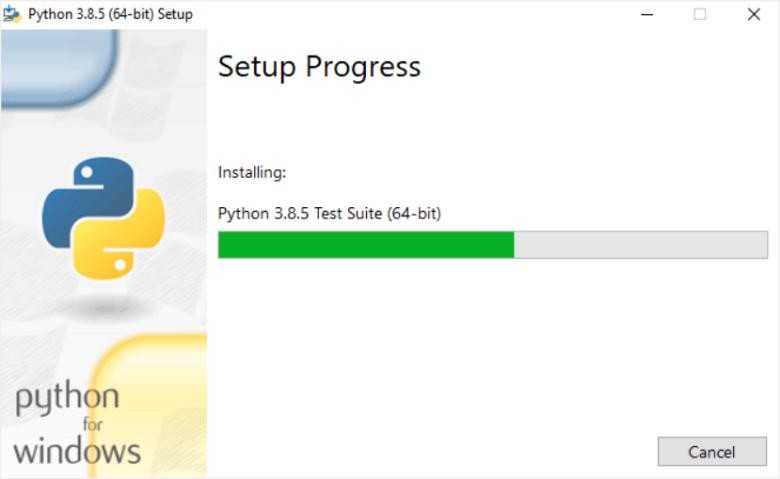


**Install Python on Windows**

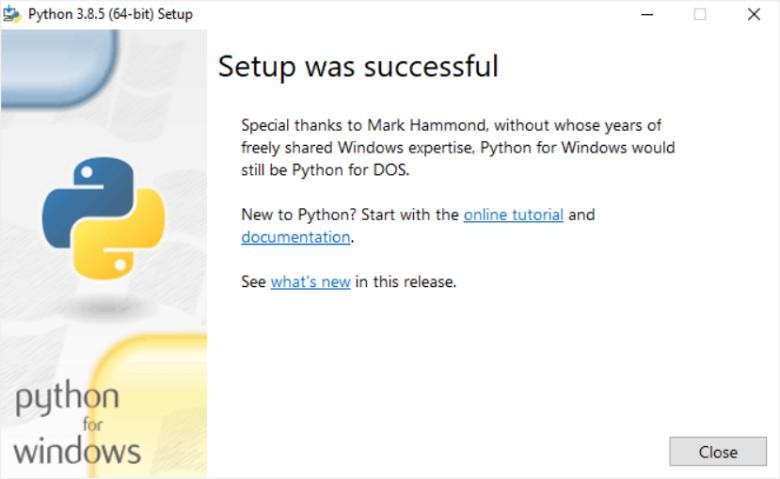
* First, download from the download page. Second, double-click the installer file to launch the setup wizard.
* In the setup window, you need to check the **Add Python3.8 to PATH** and click Install Now to begin the installation.



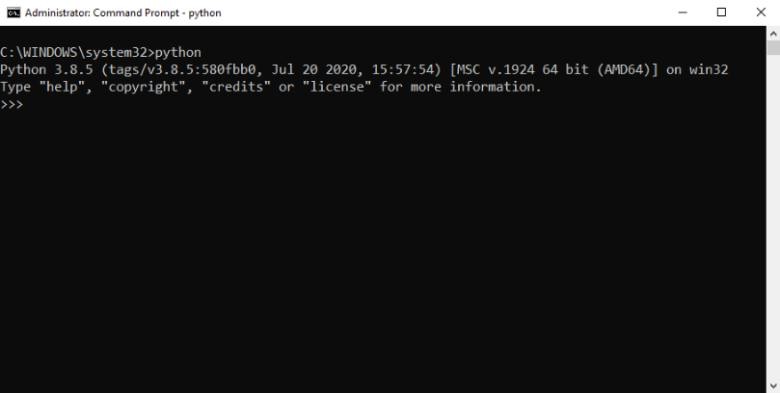
It’ will take a few minutes to complete the set up.



Once the setup completes, you’ll see the following window:



In the Command Prompt, type python command as follows:



If you see the output like the above screenshot, you’ve successfully installed Python on your computer.

To exit the program, you type Ctrl-Z and press Enter.

If you see the following output from the Command Prompt after typing the python command:

'python'isnotrecognizedasaninternalorexternalcommand,operable program or batch file

Likely, you didn’t check he **Add Python3.8 to PATH** check box when you install Python

Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it." Some changes in Python 7.3:

* Print is now a function
* Views and iterators instead of lists
* The rules for ordering comparisons have been simplified.
* E.g: a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.
* There is only one integer type left, i.e. int. long is int as well.
* The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behaviour.
* Text Vs. Data Instead Of Unicode Vs. 8-bit Purpose :- We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with lowquality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.
* Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. Python features a dynamic type system and automatic memory management.
  + It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

**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 Interactive:**

You can actually sit at a Python prompt and interact with the interpreter directly to write your programs. Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code.

**Maintainability:**

It also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

**Open Source :**

Python is an open-source software. Anyone can freely distribute it, read the source code, edit it easily.

**High-level Language:**

When writing programs in python, the programmers don’t have to worry aboutthe low-level details like managing the memory used by the program. Python is a High-level language that needed to concentrate on writing solutions above problems.

**Interactive:**

Python programs work in an interactive mode which allows us interactive testing and debugging of a piece of code. Most of the Programmers can easily interact with the interpreter directly.

**Prerequisites**

Before learning machine learning, you must have the basic knowledge of followings so that you can easily understand the concepts of machine learning:

* Fundamental knowledge of probability and linear algebra.
* The ability to code in any computer language, especially in Python language.
* Knowledge of Calculus, especially derivatives of single variable and multivariate functions.

Linear Regression in Machine Learning Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as sales, salary, age, product price, etc.

**6.4 PACKAGES USED IN PROJECT:**

**NumPy**

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python.

It contains various features including these important ones:

▪ A powerful N-dimensional array object

▪ Sophisticated (broadcasting) functions

▪ Tools for integrating C/C++ and Fortran code

▪ Useful linear algebra, Fourier transform, and random number capabilities.

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using NumPy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases. Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis.

**Pandas**

Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Pandas is an open-source, BSD-licensed Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc. In this tutorial, we will learn the various features of Python Pandas and how to use them in practice.

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data.In 2008, developer Wes McKinney started developing pandas when in need of high performance, flexible tool for analysis of data.

Prior to Pandas, Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, wecan accomplish five typical steps in the processing and analysis of data, regardless of the origin of data — load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

**Key Features of Pandas:**

• Fast and efficient Data Frame object with default and customized indexing.

• Tools for loading data into in-memory data objects from different file formats.

• Data alignment and integrated handling of missing data

**Tensor flow**

Tensor Flow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google. Tensor Flow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

**Matplotlib**

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and I Python shells, the Jupiter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery. For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with I Python. For the power user, you have full control of line styles, font properties, axes properties, etc., via an object oriented interface or via a set of functions familiar to MATLAB users.

**Scikit – learn**

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

**Classification:** samples belong to two or more classes and we want to learn from already labeled data how to predict the class of unlabeled data. An example of a classification problem would be handwritten digit recognition, in which the aim is to assign each input vector to one of a finite number of discrete categories. Another way to think of classification is as a discrete (as opposed to continuous) form of supervised learning where one has a limited number of categories and for each of the n samples provided, one is to try to label them with the correct category or class.

**Regression:** If the desired output consists of one or more continuous variables, then the task is called regression. An example of a regression problem would be the prediction of the length of a salmon as a function of its age and weight. Unsupervised learning, in which the training data consists of a set of input vectors x without any corresponding target values. The goal in such problems may be to discover groups of similar examples within the data, where it is called clustering, or to determine the distribution of data within the input space, known as density estimation, or to project the data from a high-dimensional space down to two or three dimensions for the purpose of visualization.

scikit-learn comes with a few standard datasets, for instance the iris and digits datasets for

classification and the boston house prices dataset for regression.

In the following, we start a Python interpreter from our shell and then load the iris and digits

datasets. Our notational convention is that $ denotes the shell prompt while >>> denotes the

Python interpreter prompt:

$ python

>>> from sklearn import datasets

>>> iris = datasets.load\_iris()

>>> digits = datasets.load\_digits()

A dataset is a dictionary-like object that holds all the data and some metadata about the data.

This data is stored in the .data member, which is a n\_samples, n\_features array. In the case ofsupervised problem, one or more response variables are stored in the target member. More details on the different datasets can be found in the dedicated section.For instance, in the case of the digits dataset, digits.data gives access to the features that can be used to classify the digits samples:

>>>print(digits.data)

[[ 0. 0. 5. ... 0. 0. 0.]

[ 0. 0. 0. ... 10. 0. 0.]

[ 0. 0. 0. ... 16. 9. 0.]

...

[ 0. 0. 1. ... 6. 0. 0.]

[ 0. 0. 2. ... 12. 0. 0.]

[ 0. 0. 10. ... 12. 1. 0.]]

Target gives the ground truth for the digit dataset, that is the number corresponding to each digit image that we are trying to learn.

**SciKit-learn Genetic:**

Sklearn-genetic-opt uses evolutionary algorithms to fine-tune scikit-learn machine learning algorithms and perform feature selection. It is designed to accept a scikit-learn regression or classification model (or a pipeline containing on of those). The idea behind this package is to define the set of hyper parameters we want to tune and what are their lower and uppers bounds on the values they can take. It is possible to define different optimization algorithms, call-backs and build-in parameters to control how the optimization is taken. To get started, we’ll use only the most basic features and options.

The optimization is made by evolutionary algorithms with the help of the deep package. It works by defining the set of hyper parameters to tune, it starts with a randomly sampled set of options (population). Then by using evolutionary operators as the mating, mutation, selection and evaluation, it generates new candidates looking to improve the cross-validation score in each generation. It’ll continue with this process until a number of generations is reached or until a callback criterion is met.

**Keras:**

Two of the top numerical platforms in Python that provide the basis for Deep Learning research and development are Theano and Tensor Flow.Both are very powerful libraries, but both can be difficult to use directly for creating deep learning models.In this post, you will discover the Keras Python library that provides a clean and convenient way to create a range of deep learning models on top of Theano or Tensor Flow.Keras is a minimalist Python library for deep learning that can run on top of Theano or Tensor Flow.

It was developed to make implementing deep learning models as fast and easy as possible for research and development.It runs on Python 2.7 or 3.5 and can seamlessly execute on GPUs and CPUs given the underlying frameworks. It is released under the permissive MIT license.Keras was developed and maintained by François Chollet, a Google engineer using four guiding principles:

**Modularity:** A model can be understood as a sequence or a graph alone. All the concerns of a deep learning model are discrete components that can be combined in arbitrary ways.

**Minimalism:** The library provides just enough to achieve an outcome, no frills and maximizing readability.

**Extensibility:** New components are intentionally easy to add and use within the framework, intended for researchers to trial and explore new ideas.

**Python:** No separate model files with custom file formats. Everything is native Python. Keras is relatively straightforward to install if you already have a working Python and SciPy Environment.

**6.5 INSTALLATION OF PACKAGES:**

Syntax for installation of packages via cmd terminal using the basic

Step:1- First check pip cmd, If ok then

Step:2- pip list

Check the list of packages installed and then install required by following cmds.

Step:3- pip install package name.

The package name should be based on our requirement.

**CHAPTER-7**

**IMPLEMENTATION MODULES**

**Modules Description**

The implementation of a weather prediction system using machine learning involves several key modules.

**Data Collection Module** gathers historical and real-time weather data from sources like weather stations, satellites, and IoT sensors. This data is stored in ensuring easy retrieval for analysis.

**Data Pre-processing Module** cleans the data by handling missing values, normalizing features, and removing outliers. relevant features such as seasonal trends and geographical patterns are extracted to improve model accuracy.

**Modelling Module** applies machine learning algorithms like Linear Regression, Decision Trees, or Neural Networks to predict weather conditions. The uses the trained model to generate forecasts for variables such as temperature, precipitation, and wind speed.

The **Evaluation Module** assesses the model’s performance using metrics like Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). **The**  presents the forecast results in an accessible format through dashboards or mobile apps. This architecture enables accurate and real-time weather predictions.

**Sample Code**

**SYSTEM REQUIREMENTS**

**Software Requirements**

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation.

The appropriation of requirements and implementation constraints gives the general overview of the project in regard to what the areas of strength and deficit are and how to tackle them.

* Python IDLE 3.7 or 3.10 version
* PYcharm
* Jupiter (or) HTML or CSS
* Django or Flask or Starlit or tinkter

**Hardware Requirements**

Minimum hardware requirements are very dependent on the particular software being developed by a given Enthought Python / Canopy / VS Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

* Operating system : Windows, Linux
* Processor : minimum intel i3
* Ram : minimum 4 GB
* Hard disk : minimum 250GB

**FUNCTIONAL REQUIREMENTS**

**Output Design**

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provides a permanent copy of the results for later consultation. The various types of outputs in general are:

* External Outputs, whose destination is outside the organization
* Internal Outputs whose destination is within organization and they are the
* User’s main interface with the computer.
* Operational outputs whose use is purely within the computer department.
* Interface outputs, which involve the user in communicating directly.

**Output Definition**

The outputs should be defined in terms of the following points:

* Type of the output
* Content of the output
* Format of the output
* Location of the output
* Frequency of the output
* Volume of the output
* Sequence of the output

It is not always desirable to print or display data as it is held on a computer. It should be decided as which form of the output is the most suitable.

**Input Design**

Input design is a part of overall system design. The main objective during the input design is as given below:

* To produce a cost-effective method of input.
* To achieve the highest possible level of accuracy.
* To ensure that the input is acceptable and understood by the user.

**Input Stages**

The main input stages can be listed as below:

* Data recording
* Data transcription
* Data conversion
* Data verification
* Data control
* Data transmission
* Data validation
* Data correction

**Input Types**

It is necessary to determine the various types of inputs. Inputs can be categorized as follows:

* External inputs, which are prime inputs for the system.
* Internal inputs, which are user communications with the system.
* Operational, which are computer department’s communications to the system?
* Interactive, which are inputs entered during a dialogue.

**Input Media**

At this stage choice has to be made about the input media. To conclude about the input media consideration has to be given to;

* Type of input
* Flexibility of format
* Speed
* Accuracy
* Verification methods
* Rejection rates
* Ease of correction
* Storage and handling requirements
* Security
* Easy to use
* Portability

Keeping in view the above description of the input types and input media, it can be said that most of the inputs are of the form of internal and interactive. As

Input data is to be the directly keyed in by the user, the keyboard can be considered to be the most suitable input device.

**Error Avoidance**

At this stage care is to be taken to ensure that input data remains accurate form the stage at which it is recorded up to the stage in which the data is accepted by the system. This can be achieved only by means of careful control each time the data is handled.

**Error Detection**

Even though every effort is make to avoid the occurrence of errors, still a small proportion of errors is always likely to occur, these types of errors can be discovered by using validations to check the input data.

**Data Validation**

Procedures are designed to detect errors in data at a lower level of detail. Data validations have been included in the system in almost every area where there is a possibility for the user to commit errors. The system will not accept invalid data. Whenever an invalid data is keyed in, the system immediately prompts the user and the user has to again key in the data and the system will accept the data only if the data is correct. Validations have been included where necessary.

The system is designed to be a user friendly one. In other words the system has been designed to communicate effectively with the user. The system has been designed with popup menus.

**User Interface Design**

It is essential to consult the system users and discuss their needs while designing the user interface:

**User Interface Systems Can Be Broadly Clasified As:**

* User initiated interface the user is in charge, controlling the progress of the user/computer dialogue. In the computer-initiated interface, the computer selects the next stage in the interaction.
* Computer initiated interfaces

In the computer-initiated interfaces the computer guides the progress of the user/computer dialogue. Information is displayed and the user response of the computer takes action or displays further information.

**User Initiated Interfaces**

User initiated interfaces fall into two approximate classes:

* Command driven interfaces: In this type of interface the user inputs commands or queries which are interpreted by the computer.
* Forms oriented interface: The user calls up an image of the form to his/her screen and fills in the form. The forms-oriented interface is chosen because it is the best choice.

**Computer-Initiated Interfaces**

The following computer – initiated interfaces were used:

* The menu system for the user is presented with a list of alternatives and the user chooses one; of alternatives.
* Questions – answer type dialog system where the computer asks question and takes action based on the basis of the users reply.

Right from the start the system is going to be menu driven, the opening menu displays the available options. Choosing one option gives another popup menu with more options. In this way every option leads the users to data entry form where the user can key in the data.

**Error Message Design**

The design of error messages is an important part of the user interface design. As user is bound to commit some errors or other while designing a system the system should be designed to be helpful by providing the user with information regarding the error he/she has committed.

This application must be able to produce output at different modules for different inputs.

**Performance Requirements**

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely in the part of the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly stated as given below:

* The system should be able to interface with the existing system
* The system should be accurate
* The system should be better than the existing system
* The existing system is completely dependent on the user to perform all the duties.

**RESULTS AND DISCUSSION**

In weather prediction using machine learning algorithms, the results typically showcase the accuracy and effectiveness of the model in forecasting weather conditions such as temperature, humidity, and precipitation. Commonly used algorithms like Decision Trees, Random Forest, Support Vector Machines (SVM), and Neural Networks are evaluated based on metrics like Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). The discussion often highlights how well the model generalizes to new data, its strengths in predicting short-term weather, and limitations such as difficulty in predicting extreme weather events or long-term forecasts. Comparing different algorithms helps in identifying the most suitable approach for specific weather prediction tasks, considering factors like accuracy, computational efficiency, and data requirements.

**Add output screens**

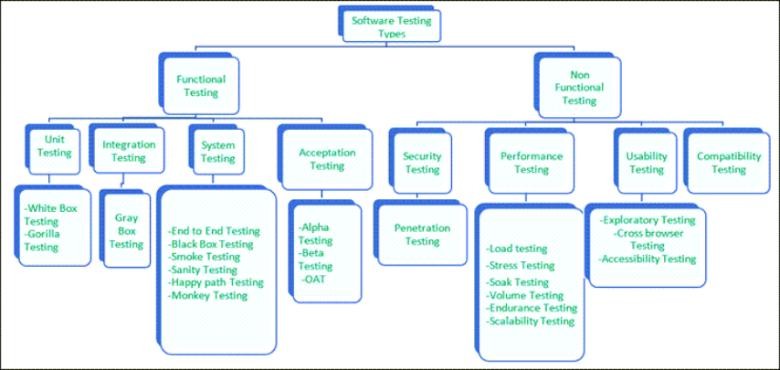
**CHAPTER-9**

**SYSTEM TESTING**

## **9.1 Different Testing Types with Details**

We, as testers, are aware of the various types of Software Testing like Functional Testing, Non-Functional Testing, Automation Testing, Agile Testing, and their sub- types, etc. Each type of testing has its own features, advantages, and disadvantages as well. However, in this tutorial, we have covered mostly each and every type of software testing which we usually use in our day-to-day testing life.

## Different Types of Software Testing



## **Performance Testing**

Performance testing is testing of an application’s stability and response time by applying load.

The word stability means the ability of the application to withstand in the presence of load. Response time is how quickly an application is available to users. Performance testing is done with the help of tools. Loader.IO, JMeter, Load Runner, etc. are good tools available in the market.

## **Load testing**

Load testing is testing of an application’s stability and response time by applying load, which is equal to or less than the designed number of users for an application.

**For example,** your application handles 100 users at a time with a response time of 3 seconds, then load testing can be done by applying a load of the maximum of 100 or less than 100 users. The goal is to verify that the application is responding within 3 seconds for all the users.

## **Stress Testing**

Stress testing is testing an application’s stability and response time by applying load, which is more than the designed number of users for an application.

**For example,** your application handles 1000 users at a time with a response time of 4 seconds, and then stress testing can be done by applying a load of more than 1000 users. Test the application with1100, 1200, 1300users and notice the response time. The goal is to verify the stability of an application under stress.

## **Scalability Testing**

Scalability testing is testing an application’s stability and response time by applying load, which is more than the designed number of users for an application.

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**9.2 TYPES OF TESTS**

**Unit Testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

**Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

# Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**SYSTEM TESTING**

### TESTING METHODOLOGIES

The following are the Testing Methodologies:

* **Unit Testing.**
* **Integration Testing.**
* **User Acceptance Testing.**
* **Output Testing.**
* **Validation Testing.**

**Unit Testing**

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module’s control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing.

During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All important processing path are tested for the expected results. All error handling paths are also tested.

**Integration Testing**

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

**The following are the types of Integration Testing:**

**1)Top Down Integration**

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are incorporated into the structure in either a depth first or breadth first manner.

In this method, the software is tested from main module and individual stubs are replaced when the test proceeds downwards.

**2. Bottom-up Integration**

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated. The bottom up integration strategy may be implemented with the following steps:

* The low-level modules are combined into clusters into clusters that perform a specific Software sub-function.
* A driver (i.e.) the control program for testing is written to coordinate test case input and output.
* The cluster is tested.
* Drivers are removed and clusters are combined moving upward in the program structure

The bottom up approaches tests each module individually and then each module is module is integrated with a main module and tested for functionality.

**OTHER TESTING METHODOLOGIES**

**User Acceptance Testing**

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

**Output Testing**

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

**Validation Checking**

Validation checks are performed on the following fields.

**Text Field:**

The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables. Incorrect entry always flashes and error message.

**Numeric Field:**

The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error messages. The individual modules are checked for accuracy and what it has to perform. Each module is subjected to test run along with sample data. The individually tested modules are integrated into a single system. Testing involves executing the real data information is used in the program the existence of any program defect is inferred from the output. The testing should be planned so that all the requirements are individually tested.

A successful test is one that gives out the defects for the inappropriate data and produces and output revealing the errors in the system.

**Preparation of Test Data**

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using that test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

**Using Live Test Data:**

Live test data are those that are actually extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the systems person uses this data as a way to partially test the system. In other instances, programmers or analysts extract a set of live data from the files and have them entered themselves.

It is difficult to obtain live data in sufficient amounts to conduct extensive testing. And, although it is realistic data that will show how the system will perform for the typical processing requirement, assuming that the live data entered are in fact typical, such data generally will not test all combinations or formats that can enter the system. This bias toward typical values then does not provide a true systems test and in fact ignores the cases most likely to cause system failure.

**Using Artificial Test Data:**

Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department, make possible the testing of all login and control paths through the program.

The most effective test programs use artificial test data generated by persons other than those who wrote the programs. Often, an independent team of testers formulates a testing plan, using the systems specifications.

The package “Virtual Private Network” has satisfied all the requirements specified as per software requirement specification and was accepted.

**USER TRAINING**

Whenever a new system is developed, user training is required to educate them about the working of the system so that it can be put to efficient use by those for whom the system has been primarily designed. For this purpose the normal working of the project was demonstrated to the prospective users. Its working is easily understandable and since the expected users are people who have good knowledge of computers, the use of this system is very easy.

**MAINTAINENCE**

This covers a wide range of activities including correcting code and design errors. To reduce the need for maintenance in the long run, we have more accurately defined the user’s requirements during the process of system development. Depending on the requirements, this system has been developed to satisfy the needs to the largest possible extent. With development in technology, it may be possible to add many more features based on the requirements in future. The coding and designing is simple and easy to understand which will make maintenance easier.

**TESTING STRATEGY :**

A strategy for system testing integrates system test cases and design techniques into a well-planned series of steps that results in the successful construction of software. The testing strategy must co-operate test planning, test case design, test execution, and the resultant data collection and evaluation. A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high level tests that validate major system functions against user requirements.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing represents an interesting anomaly for the software. Thus, a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

**CHAPTER-10**

**CONCLUSION AND FUTURE SCOPE**

Weather forecasting using the linear regression algorithm and the Naïve Bayes algorithm is critical for improving people’s future results. The linear regression algorithm and the Naïve Bayes algorithm were used to forecast the weather using weather datasets. Using some selected input variables obtained from kaggle, GitHub we created a model to predict the weather. The issue with the current weather situation is that we are unable to organize ourselves and complete essential tasks. As a result, this model was developed in order to know the weather scenario with high precision while taking into account all of the factors that influence the weather scenario.

**FUTURE WORK**

A weather prediction technology can stipulate university growth and development. The scope magnitude of change that are occurring in department today are both exciting and daunting, very particularly we are contemplating how we will manage the many streams of technological innovations pouring into our department and networked information world. All the social medias combine to create one. This application should need to know to avoid natural disaster. To overcome that problem we have improve this to advanced features included. And then using speech recognition we need to pronounce correctly or else it will not work correctly. So that need to improve without this critical.

**REFERENCES**

[1] Aher, Sunita B., and L. M. R. J. Lobo. "Comparative study of classification algorithms." International Journal of Information Technology 5.2 (2012): 239 243.

[2] Bellaachia, Abdelghani, and Erhan Guven. "Predicting breast cancer survivability using data mining techniques." Age 58.13 (2006): 10-110.

[3] Devasena, C. Lakshmi. "Comparative analysis of random forest, REP tree and J48 classifiers for credit risk prediction." International Journal of Computer Applications (2014): 0975-8887.

[4] Grover, Aditya; Kapoor, Ashish; Horvitz, Eric (2015). [ACM Press the 21th ACM SIGKDD International Conference - Sydney, NSW, Australia (2015.08.10-2015.08.13)] Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15 - A Deep Hybrid Model for Weather Forecasting. , (), 379–386.

[5] Holmstrom, Mark, Dylan Liu, and Christopher Vo. "Machine learning applied to weather forecasting." Meteorol. Appl (2016): 1-5.

[6] Kalmegh, Sushilkumar. "Analysis of weka data mining algorithm reptree, simple cart and randomtree for classification of indian news." International Journal of Innovative Science, Engineering & Technology 2.2 (2015): 438-446.

[7] Kharche, Deepali, K. Rajeswari, and Deepa Abin. "Comparison of different datasets using various classification techniques with weka." International Journal of Computer Science and Mobile Computing 3.4 (2014): 389-393.

[8] Kulkarni, Vrushali Y. and Pradeep K. Sinha. “Effective Learning and Classification using Random Forest Algorithm.” (2014).

[9] Latinne, P., Debeir, O., & Decaestecker, C. (2001). Limiting the Number of Trees in Random Forests. Lecture Notes in Computer Science, 178 187

[10] Mishra, Ajay Kumar, and Bikram Kesari Ratha. "Study of random tree and random forest data mining algorithms for microarray data analysis." International Journal on Advanced Electrical and Computer Engineering 3.4 (2016): 5 7.

[11] Nalluri, Sravani; Ramasubbareddy, Somula; Kannayaram, G (2019). Weather Prediction Using Clustering Strategies in Machine Learning. Journal of Computational and Theoretical Nanoscience, 16(5), 1977–1981

[12] Pourdarab, Sanaz, Ahmad Nadali, and Hamid Eslami Nosratabadi. "A hybrid method for credit risk assessment of bank customers." International Journal of Trade, Economics and Finance 2.2 (2011): 125-131.

[13] Radhika, Y., and M. Shashi. "Atmospheric temperature prediction using support vector machines." International journal of computer theory and engineering 1.1 (2009): 55.

[14] Rao, K. Hanumantha, et al. "Implementation of anomaly detection technique using machine learning algorithms." International journal of computer science and telecommunications 2.3 (2011): 25-31.

[15] Rich Caruana and Alexandru Niculescu-Mizil. 2006. An empirical comparison of supervised learning algorithms. In Proceedings of the 23rd international conference on Machine learning. Association for Computing Machinery, New York, NY, USA, 161–168.

[16] Sharma, Aman Kumar, and Suruchi Sahni. "A comparative study of classification algorithms for spam email data analysis." International Journal on Computer Science and Engineering 3.5 (2011): 1890-1895.