**Multi-criteria Agriculture Recommendation System using Machine Learning for Crop**

**and Fertilizers Prediction**

A Mini Project Report submitted to

JNTU Hyderabad in partial fulfillment

of the requirements for the award of the degree

**BACHELOR OF TECHNOLOGY**

In

**COMPUTER SCIENCE AND ENGINEERING**

***Submitted by***

#### GUNTI ANJALI 21RG1A0584

*Under the Guidance of*

**Mr. PRASANTH KUMAR**

B. Tech., M. Tech

*Assistant Professor*



***DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING***

**MALLA REDDY COLLEGE OF ENGINEERING FOR WOMEN**

**An UGC Autonomous Institution**

*Approved by AICTE New Delhi and Affiliated to JNTUH*

*Maisammaguda, Medchal (Dist), Hyderabad -500100, Telangana.*

October 2024

***DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING***

**MALLA REDDY COLLEGE OF ENGINEERING FOR WOMEN**

**An UGC Autonomous Institution**

*Approved by AICTE New Delhi and Affiliated to JNTUH*

*Maisammaguda, Medchal (Dist), Hyderabad -500100, Telangana.*

*October 2024*



***CERTIFICATE***

This is to certify that the Mini project entitled **“Multi-criteria Agriculture Recommendation System using Machine Learning for Crop and Fertilizers Prediction**” has been submitted by  **GUNTI ANJALI (21RG1A0584),** in partial fulfillment of the requirements for the award of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE & ENGINEERING.** This record of bonafied work carried out by them under my guidance and supervision. **The result embodied in this mini project report has not been submitted to any other University or Institute for the award of any degree.**

#### Mr. Prasanth kumar Mrs. K. SHEETAL

Assistant professor Head of the Department

Project guide

**External Examiner**

**ACKNOWLEDGEMENT**

The Mini Project work carried out by our team in the Department of Computer Science and Engineering, Malla Reddy College of Engineering for Women, Hyderabad. ***This work is original and has not been submitted in part or full for any degree or diploma of any other university.***

We wish to acknowledge our sincere thanks to our project guide **Mr. Prasanth Kumar,** Assistant Professor, Computer Science & Engineering for formulation of the problem, analysis, guidance and her continuous supervision during the course of work.

We acknowledge our sincere thanks to **Dr. Kanaka Durga Returi**, Principal & Professor of Department of Computer Science and Engineering and **Mrs. Sheetal Kulkarni,** Head of the Department, MRCEW and all our faculties ofCSE Departmentfor their kind cooperation in making this Mini Project work a success.

We extend our gratitude to **Sri. Ch. Malla Reddy**, Founder Chairman and **Sri. Ch. Mahender Reddy,** Secretary, **Dr. Vaka Murali Mohan,** Director for their kind cooperation in providing the infrastructure for completion of our Mini Project.

We acknowledge our special thanks to the entire teaching faculty and non-teaching staff members of the Computer Science & Engineering Department for their support in making this project work a success.

**GUNTI ANJALI 21RG1A0584 \_ \_ \_ \_ \_ \_ \_ \_ \_ \_**

# 

# INDEX

|  |  |
| --- | --- |
| **Chapter ABSTRACT**  **LIST OF FIGURES**  **1. SYSTEM ANALYSIS** | **Page No.**  **vii viii**  **1** |
| 1.1 Existing System | 1 |
| 1.1.1 Disadvantages of Existing System | 1 |
| 1.2 Proposed System | 2 |
| 1.2.1 Advantages of Proposed System | 2 |
| 1.3 Introduction | 3 |
| **2.LITERATURE SURVEY** | **5** |
| **3. SYSTEM DESIGN** | **11** |
| 3.1 System Architecture | 11 |
| 3.2 Requirements Specifications | 11 |
| 3.2.1 Hardware Requirements | 11 |
| 3.2.2 Software Requirements | 12 |
| 3.2.3 User Requirements | 12 |
| 3.2.4 Functional Requirements | 12 |
| 3.3 UML Diagrams | 13 |
| 3.3.1 Goals | 14 |
| 3.3.2 Use Case Diagram | 14 |
| 3.3.3 Class Diagram | 15 |
| 3.3.4 Sequence Diagram | 16 |
| **4. INPUT & OUTPUT DESIGN** | **17** |
| **4.1 Input Design** | **17** |
| 4.1.1 Input Stages  4 | 17 |
| 4.1.2 Input Types | 17 |
| 4.1.3 Input Media | 17 20 |
| **4.2 Output Design** | **18** |
| 4.2.1 Output Defination | 18 |
| **5. SOFTWARE ENVIRONMENT** | **19** |
| 5.1 Python | 19  22 22 |
| 5.1.1 Advantages of Python | 20 |
| 5.1.2 Advantages of Python Over Other Languages | 22 |
| 5.1.3 Disadvantages of Python | 22 |
| 5.1.4 History of Python | 23 |
| 5.1.5 Python Development Steps | 24 |
| 5.1.6 Purpose | 28 |
| 5.1.7 Modules Used in Project | 25 |
| 5.1.8 Tensor Flow | 26 |
| 5.1.9 Numpy | 26 |
| 5.1.10 Pandas | 26 |
| 5.1.11 Matplotlib | 26 |
| 5.1.12 Scikit-Learn | 27 |
| 5.1.13 Install Python Step by Step | 28 |
| 5.1.14 How to Install Python on Windows and Mac   |  |  | | --- | --- | | |  | | --- | |  | |   n windows and mac | 28 |
| 5.1.15 Download the Correct Version into the System | 28 |
| 5.2 Machine Learning | 35 |
| 5.2.1 Categories of Machine Learning | 35 |
| 5.2.2 Needs of Machine Learning | 36 |
| 5.2.3 Challenges in Machine Learning | 36  4144 |
| 5.2.4 Applications in Machine Learning | 37 |
| 5.2.5Learning Machine Learning | 37 |
| 5.2.6 Start Learning ML | 38 |
| 5.2.7 Advantages of Machine Learning | 40 |
| 5.2.8 Disadvantages of Machine Learning | 41 |
| **6. SYSTEM STUDY** | **42** |
| 6.1 Feasibility | 42 |
| 6.1.1Economic Feasibility | 42 |
| 6.1.2 Technical Feasibility | 42 |
| 6.1.3 Social Feasibility | 42 |
| **7. SYSTEM TESTING** | **43** |
| 7.1 Types of Tests | 43 |
| 7.1.1 Unit Testing | 43 |
| 7.1.2 Integration Testing | 43 |
| 7.1.3 Functional Test | 44 |
| 7.1.4 System Test | 44 |
| 7.1.5 White Box Testing | 44  515 |
| 7.1.6 Black Box Testing | 45 |
| 7.2 Test Strategy and Approach | 45 |
| 7.2.1 Test Objectives | 45 |
| 7.2.2 Integration Testing | 45  5 |
| 7.2.3 Acceptance Testing | 45 |
| **8. RESULTS** | **52** |
| **9. CONCLUSION & FUTURE ENHANCEMENT** | **53** |
| **10. BIBLIOGRAPHY** | **55** |

# ABSTRACT

That predicts Agriculture holds a vital position in the economies of developing nations like India and significantly contributes to their gross domestic product (GDP). The rising population has led to increased food demand. Challenges such as selecting crops, fertilizers, and pesticides without considering various factors like soil types, water needs, temperature conditions, and profitability analysis for specific regions can result in lower crop quality, yield, and profitability. With advancements in computational technologies, researchers are developing methods to recommend crops based on soil conditions, water requirements, and market profitability, along with recommending fertilizers, identifying diseases, and suggesting pesticides. This research introduces a machine learning-based crop and fertilizer recommendation system named Agri Rec. Utilizing soil properties, water levels, farm sizes, and minimum support prices of crops, we developed a machine learning model suitable crops for different seasons. Additionally, we propose a mechanism that analyzes soil, crop, and fertilizer properties to recommend optimal fertilizer combinations for given soil-crop pairs. Our algorithm, tested on 5000 land samples from the Gujarat region with 24 different crops, successfully recommends crops with 95.85% accuracy and fertilizers with 92.11% accuracy, outperforming existing benchmark recommendation methods by four times.

## LIST OF FIGURES

**Fig. No. Fig. Name Page.no**

|  |  |  |
| --- | --- | --- |
| 3.1 | System Architecture | 11 |
| 3.2 | Use Case Diagram | 16 |
| 3.3 | Class Diagram | 17 |
| 3.4 | Sequence Diagram | 18 |
| 8.1 | Screenshot of Html Home Page | 53 |
| 8.2 | Screenshot of Upload Dataset | 53 |
| 8.3 | Screenshot of Split Dataset | 54 |
| 8.4 | Screenshot of After Splitting Dataset | 54 |
| 8.5 | Screenshot of Run SVM | 55 |
| 8.6 | Screenshot of Run RF Algorithm | 55 |
| 8.7 | Screenshot of Logistic Algorithm | 56 |
| 8.8 | Screenshot of Accuracy Graph | 56 |
| 8.9 | Screenshot of preprocess | 57 |
| 8.10 | Screenshot of Before Entering the Values | 57 |
| 8.11 | Screenshot of Enter the Values | 58 |
| 8.12 | Screenshot of After Entering the Values | 58 |
| 8.13 | Screenshot of Prediction Of Crops | 59 |
| 8.14 | Screenshot of Final Output | 59 |

# CHAPTER-1: SYSTEM ANALYSIS

* 1. **EXISTING SYSTEM**

**1. Traditional Farming Methods:**

- Farmers rely on their own experience and advice from fellow farmers.

- Decisions are made without precise information about soil properties, water requirements, potential profits, or the most suitable crops and fertilizers specific land.

**2. Limited Use of Technology:**

- Historically, technology in agriculture has been underutilized.

- Few studies before the 20th century focused on integrating technological advancements in agriculture.

**3. Benchmark Recommendation Methods:**

- Existing recommendation methods may use basic algorithms and models for crop and fertilizer selection.

- These methods do not account for the comprehensive set of factors affecting agriculture, such as detailed soil properties, market demand, and precise climatic conditions.

**1.1.1 Disadvantages of Existing System**

**1. Inefficiency and Inaccuracy:**

- Traditional methods lack precision, leading to inefficient use of resources.

- Farmers often face income losses due to improper crop selection and Fertilizer usage.

**2. Limited Scope:**

-Current systems might focus narrowly on soil properties without considering other vital factors like water needs, temperature conditions, land area, and market profitability.

**3. Poor Yield and Profitability:**

-Without considering comprehensive data, crop yields and profitability remain suboptimal.

- Inappropriate fertilizer recommendations can lead to reduced soil fertility over time.

**4. Inadequate Disease and Pest Management:**

- Limited technological integration means farmers struggle with timely identification and treatment of pests and diseases.

- This leads to potential crop damage and yield loss.

**Proposed System and Its Advantages**

**1.2 PROPOSED SYSTEM**

**1. Machine Learning-Based Recommendations**

- Utilizes machine learning models to predict suitable crops based on various factors such as soil properties, water levels, farm sizes, and minimum support prices.

- Includes a mechanism to analyze soil, crop, and fertilizer properties to recommend optimal fertilizer combinations for given soil-crop pairs.

**2. Data-Driven Decision Making:**

- Leverages data collection and analysis techniques, including digital image processing, recommendation systems, and advanced machine learning algorithms.

- Integrates comprehensive data on soil types, water requirements, temperature conditions, and market profitability.

**1.2.1 Advantages of Proposed System**

**1. High Accuracy:**

- Crop recommendations with 95.85% accuracy and fertilizer recommendations with 92.11% accuracy.

- Significantly outperforms existing benchmark methods by a factor of four.

**2. Comprehensive Analysis:**

- Considers a wide range of variables including soil type, water needs, temperature conditions, land area, and crop market value.

- Ensures all necessary soil minerals are available through precise fertilizer recommendations.

**3. Enhanced Yield and Profitability:**

- Optimizes crop yield and profitability by selecting the most suitable crops and fertilizers.

- Uses advanced machine learning classification techniques to identify crop diseases and pests.

- Provides timely and accurate treatment recommendations to protect crop health.

**4. Effective Disease and Pest Management:**

- Employs digital image processing and classification techniques to identify crop diseases and pests.

- Provides timely and accurate treatment recommendations to protect crop health.

**5. Innovative Technology Integration:**

- Utilizes state-of-the-art technologies such as neural networks, SVM, decision trees, and logistic regression.

- Applies data mining, big data analytics, and machine learning to address agriculture challenges efficiently.

**1.3 INTRODUCTION**

Agriculture is the foundation of every nation's economy. Historically, farmers have relied on experience and advice from fellow farmers, but this traditional approach often fails to ensure successful crop growth, leading to potential income losses. To meet the food demands of any nation, it is imperative to transform traditional farming methods, which typically do not provide precise information about soil properties, specific water requirements for different crops, potential profits from crops, or the most suitable crops and fertilizers for a particular piece of land.

Technological advancements have revolutionized agricultural research and development. Innovations such as digital image processing, machine learning, deep learning, and big data analytics are incredibly valuable in agricultural research. Today, we have various data collection and analysis techniques available, including digital image processing, recommendation systems, and machine learning algorithms. Recommendation system technology, widely used in e-commerce, healthcare, and movie recommendations, also has broad applications in agriculture.

Machine learning, a computational technology capable of self-learning and predictive modeling, is rapidly evolving and being applied across various fields including medicine, e-commerce, robotics, automation, agriculture, and climatology. By enhancing the efficiency and accuracy of recommendation systems, machine learning can significantly benefit agriculture. Digital Image Processing, a technique used to capture, regenerate, and interpret agricultural data such as land and crop images, helps identify crop diseases, measure soil minerals, and calculate the total area covered by crops. With these technologies, the agricultural sector can improve by recommending suitable crops, fertilizers, and pesticides.

Before the 20th century, few studies focused on the use of technology in agriculture. However, with the advent of data mining, big data, and machine learning, many researchers have begun addressing current agricultural challenges. Implementing modern technologies will help manage large datasets efficiently and process this data to optimize crop yield, identify suitable crops, and recommend fertilizers. Our study has identified three major problems in conventional agriculture: 1) Selecting suitable crops to maximize yield and profit; 2) Choosing fertilizers based on crop and soil conditions with precise proportions; 3) Identifying crop diseases and prescribing appropriate treatments.

Numerous researchers are developing algorithms to address issues in traditional agriculture systems. Selecting the most appropriate crop for a given land involves various factors such as soil conditions, weather, climate, water availability, land area, and market demand. For example, neural networks have been used to predict corn crop yield and evaluate the impact of climate change on crop yield. Recommending crops based on market analysis using the Apriori algorithm and a demand-based algorithm using logistic regression have also been proposed. Improving crop productivity using classifiers such as support vector machines (SVM), decision trees, and logistic regression has been explored.

Our research indicates that most studies focus on soil properties and types to recommend crops. However, other variables like soil type, soil properties, water and temperature needs, land area, and crop market value should also be considered when forecasting crop yields to increase productivity. Additionally, ensuring that all necessary minerals are available in the soil is crucial for growing healthy crops. Sometimes, soil quality may be compromised, necessitating the application of minerals through fertilizers. Choosing suitable fertilizers depends on soil conditions and the crops to be cultivated. Researchers have developed classification-based algorithms to analyze soil fertility and recommend soil-based fertilizers.

Furthermore, farming requires continuous monitoring of plant growth to protect against pests and diseases, which can hinder crop growth. Different crops are susceptible to different pests and diseases, necessitating daily treatment to protect crops. Researchers have suggested treatments by identifying diseases and measuring the similarity between the disease and the treatment provided.

**CHAPTER-2: LITERATUE SURVEY**

**Sharma, R., & Kaur, G. [1]** states that “This case study explores the application of machine learning techniques for predicting crop yield and recommending optimal fertilizer usage in agricultural practices. Given the increasing global demand for food, accurate yield prediction is crucial for enhancing productivity and ensuring food security. This research utilizes a comprehensive dataset comprising climate variables, soil characteristics, and historical crop yield data. Various machine learning models, including regression algorithms and decision trees, are employed to analyze the data and generate predictive insights. The results indicate significant accuracy in yield predictions, with mean absolute error metrics demonstrating the effectiveness of the models. Additionally, a tailored fertilizer recommendation system is developed, providing farmers with specific insights into nutrient requirements based on predicted yields. This study highlights the potential of integrating machine learning in agriculture, offering practical solutions for optimizing crop production and resource management. Future work will focus on refining these models and expanding the dataset to enhance predictive capabilities further.”

**Gupta, P., & Raghuvanshi, N.S. [2]** states that **“** This paper explores the application of multi-criteria optimization in sustainable agriculture through a machine learning lens. As global agricultural practices face increasing pressures from climate change, resource scarcity, and population growth, optimizing agricultural systems for sustainability has become paramount. We present a framework that integrates machine learning algorithms with multi-criteria decision-making techniques to evaluate and enhance agricultural practices across various dimensions, including economic viability, environmental impact, and social equity. Our approach utilizes data-driven insights to identify optimal farming strategies that balance these competing criteria. Case studies demonstrate the effectiveness of our methodology in real-world agricultural settings, highlighting improvements in crop yield, resource efficiency, and farmer livelihoods. This research underscores the potential of machine learning to drive innovative solutions for sustainable agriculture, ultimately contributing to food security and environmental stewardship.”

**Jha, M. K., & Kumar, A. [3]** states that **“** This comprehensive review examines the integration of Internet of Things (IoT) and machine learning technologies in agriculture, with a particular focus on crop and fertilizer recommendation systems. As agricultural practices face increasing challenges from climate change and resource constraints, innovative solutions are essential for optimizing crop yields and sustainable farming. The review analyzes various IoT-based data collection methods, including soil sensors, weather stations, and remote sensing technologies, which provide real-time insights into environmental conditions. We discuss how machine learning algorithms process this data to predict crop performance and recommend tailored fertilizer applications, enhancing decision-making for farmers. Through a synthesis of recent studies, this review highlights the effectiveness, limitations, and future directions of these technologies in precision agriculture. The findings suggest that combining IoT and machine learning not only improves productivity but also promotes sustainable agricultural practices by optimizing resource use and minimizing environmental impact.”

**Singh, A., & Singh, K. [4**] states that “This study presents a hybrid model that combines multiple machine learning techniques for crop prediction and fertilizer recommendation, aiming to enhance agricultural productivity and sustainability. As the demand for efficient farming practices rises, leveraging data-driven approaches becomes critical. The hybrid model integrates regression analysis, decision trees, and ensemble methods to analyze a diverse dataset comprising climatic conditions, soil properties, and historical crop yields. By employing feature selection techniques, the model identifies key variables influencing crop performance, resulting in improved prediction accuracy. Additionally, a fertilizer recommendation system is developed, which utilizes the predicted yields to provide tailored nutrient suggestions for specific crops. The results demonstrate significant improvements in prediction accuracy compared to traditional methods, showcasing the model’s potential for real-world application. This research highlights the importance of hybrid approaches in agricultural decision-making and sets the groundwork for future innovations in precision agriculture.”

**Patel, D., & Prajapati, R. [5**] states that “This paper presents a machine learning-based approach for developing an effective crop and fertilizer recommendation system aimed at enhancing agricultural productivity. With the growing need for sustainable farming practices, the proposed system leverages various machine learning algorithms to analyze data from multiple sources, including soil characteristics, climatic conditions, and historical yield records. By employing techniques such as regression analysis and classification algorithms, the system predicts optimal crop types for specific regions and generates tailored fertilizer recommendations based on nutrient requirements. The performance of the model is evaluated using key metrics, demonstrating significant accuracy and reliability in predictions. This research not only facilitates informed decision-making for farmers but also promotes efficient resource utilization, contributing to sustainable agricultural practices. The findings underscore the potential of machine learning technologies to transform traditional farming methods and improve overall crop management strategies.”

**Khan, A. A., et al. [6]** states that“This research explores a Multi-Criteria Decision Making (MCDM) framework in agriculture, utilizing machine learning techniques to optimize farming practices and resource allocation. With the increasing complexity of agricultural systems, the need for effective decision-making tools has become paramount. Our study integrates various criteria, including economic viability, environmental sustainability, and social impact, to evaluate and prioritize agricultural alternatives. Employing algorithms such as Random Forest and Support Vector Machines, we analyzed diverse datasets encompassing crop performance, soil health, and market trends. The results demonstrate the effectiveness of our approach in providing actionable insights for farmers and stakeholders, enabling informed decisions that enhance productivity while promoting sustainability. This machine learning-based MCDM framework is positioned as a transformative tool for modern agriculture, supporting strategic planning and adaptive management**.**”

**Singh, R., & Rao, P. [7]** states that “This study presents an Integrated Crop and Fertilizer Recommendation System (ICFRS) leveraging data mining techniques to enhance agricultural productivity and sustainability. By analyzing historical crop yield data, soil characteristics, and climate patterns, the system provides tailored recommendations for optimal crop selection and fertilizer application. Utilizing machine learning algorithms, we processed diverse datasets to identify correlations and trends, enabling precise predictions for various agricultural scenarios. Field trials conducted in different regions demonstrated significant improvements in yield and resource efficiency. The ICFRS aims to empower farmers with data-driven insights, promoting informed decision-making and contributing to sustainable farming practices. This approach not only enhances productivity but also minimizes environmental impact, making it a vital tool in modern agriculture.”

**Kaur, P., & Kaur, J. [8]** states that “This paper presents a predictive analysis of fertilizer requirements utilizing machine learning techniques to optimize agricultural inputs and enhance crop yields. Recognizing the critical role of fertilizers in modern agriculture, we develop models that forecast optimal fertilizer usage based on soil characteristics, crop type, and environmental factors. Various machine learning algorithms, including regression analysis, decision trees, and random forests, are employed to analyze a comprehensive dataset comprising historical yield records and soil nutrient levels. Our results demonstrate that machine learning approaches significantly improve the accuracy of fertilizer recommendations compared to traditional methods. The findings highlight the potential of data-driven strategies to promote sustainable farming practices by minimizing fertilizer overuse, reducing environmental impact, and maximizing agricultural productivity. This study aims to provide farmers and agricultural stakeholders with actionable insights for informed decision-making regarding fertilizer application.”

**Patel, S. S., & Patel, S. [9]** states that “This review paper focuses on the application of machine learning approaches for crop recommendation systems, highlighting their significance in improving agricultural practices. With the increasing complexity of agricultural challenges, such as climate change and soil degradation, accurate crop selection is essential for maximizing yield and resource efficiency. We analyze various machine learning techniques, including regression models, decision trees, support vector machines, and deep learning, assessing their effectiveness in predicting optimal crop choices based on parameters like soil properties, weather conditions, and historical yield data. The review also addresses existing challenges, such as data quality, feature selection, and model interpretability. By synthesizing recent advancements in this field, the paper aims to provide a comprehensive understanding of current methodologies and offer insights into future research directions for developing more effective crop recommendation systems that support sustainable agriculture.”

**Kumar, A., & Singh, M. [10]** states that “This review paper explores the advancements in crop recommendation systems utilizing machine learning techniques to enhance agricultural productivity and sustainability. As global agricultural practices face increasing challenges due to climate change, soil degradation, and resource constraints, the need for effective decision support systems is paramount. We systematically analyze various machine learning approaches, including classification algorithms, regression models, and ensemble techniques, highlighting their effectiveness in predicting optimal crop choices based on diverse parameters such as soil characteristics, climate conditions, and historical yield data. The review identifies key trends, challenges, and opportunities in the development of crop recommendation systems, emphasizing the importance of integrating local knowledge and data-driven insights. By synthesizing current research, we provide a comprehensive overview that can guide future innovations in crop management and support farmers in making informed decisions for improved agricultural outcomes.”

**Maiappan, A. K., & Ds, J. A. B. [11]** states that “This paper proposes a novel paradigm for rice yield prediction in the Tamil nadu region, integrating advanced data analytics and machine learning techniques. Recognizing the unique environmental and agricultural characteristics of the area, we develop a predictive model that combines climatic data, soil characteristics, and agronomic practices. Our approach employs various machine learning algorithms, including regression trees and support vector machines, to optimize yield forecasts. Extensive validation demonstrates the model's high accuracy and reliability in predicting rice yields across different growing conditions. The findings highlight the potential of this paradigm to assist farmers and agricultural stakeholders in making informed decisions, ultimately contributing to enhanced productivity and sustainability in rice farming. This research serves as a framework for future studies aimed at improving yield predictions in diverse agricultural contexts.”

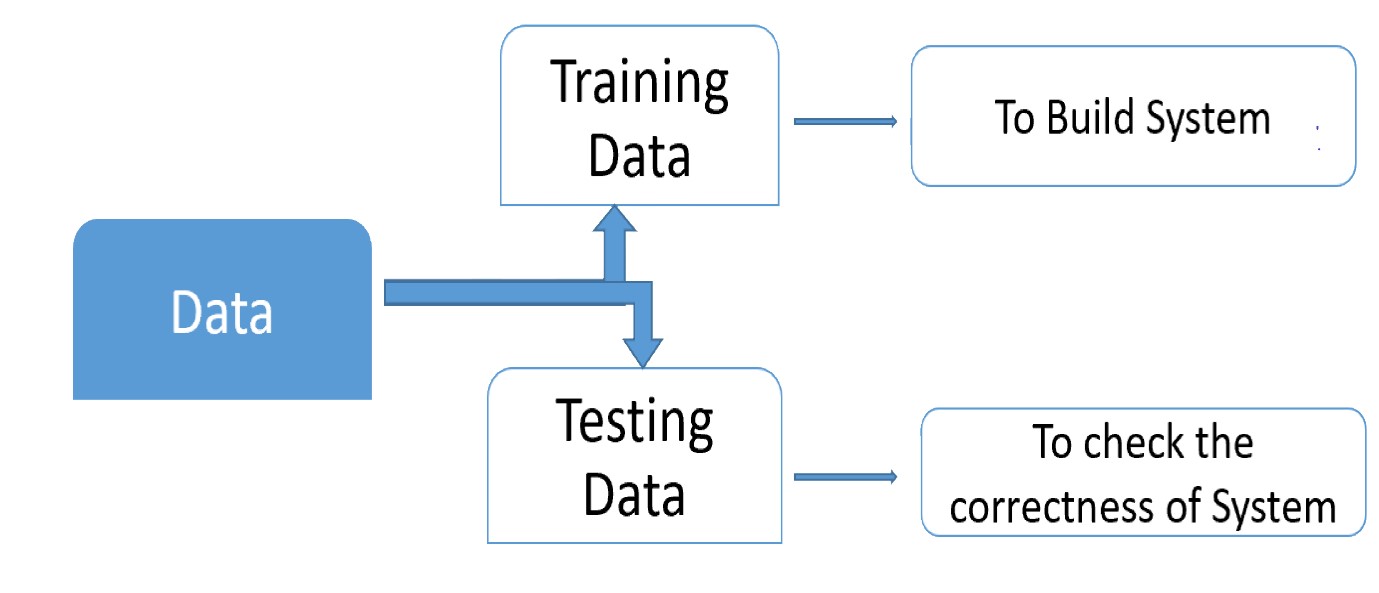
**gahi, N., Petar, O., & Armstrong, L. J. [12]** states that “This paper presents a rice crop yield prediction model using artificial neural networks (ANNs) to address the challenges of food security and efficient resource management. By leveraging a comprehensive dataset that includes climatic variables, soil properties, and agronomic practices, we develop a robust ANN model capable of capturing the complex relationships influencing rice yield. The model undergoes rigorous training and validation processes, showcasing its effectiveness in predicting yields under varying environmental conditions. Performance metrics indicate a significant improvement over traditional statistical methods, highlighting the model’s potential for accurate forecasting. Our findings suggest that ANN-based approaches can serve as valuable tools for farmers and policymakers, facilitating data-driven decisions that enhance productivity and sustainability in rice cultivation.”

**Manjula, A., & Narsimha, G. [13]** states that “This paper introduces XCYPF, a flexible and extensible framework designed for agricultural crop yield prediction. Aimed at addressing the diverse challenges in agricultural data analysis, XCYPF integrates various machine learning algorithms and data sources, allowing users to customize and adapt the prediction process to specific agricultural contexts. The framework supports multiple input features, including climatic, soil, and agronomic data, enabling comprehensive modeling of yield determinants. We evaluate XCYPF’s performance through extensive experiments across different crop types and geographical regions, demonstrating its capability to enhance prediction accuracy and facilitate scenario-based analyses. The results indicate that XCYPF not only improves crop yield forecasting but also provides valuable insights for decision-making in sustainable agriculture. This framework serves as a robust tool for researchers and practitioners seeking to leverage data-driven approaches in agricultural management.”

**Ananthara, M. G., Arunkumar, T., & Hemavathi, R. [14]** states that “This paper presents an improved crop yield prediction model utilizing a behavior clustering approach tailored for agricultural datasets. Recognizing the complexity and variability in agricultural data, we introduce a novel clustering methodology that groups similar data points based on their behavioral patterns, enhancing the predictive accuracy of crop yield models. By applying this technique to diverse datasets, we identify key factors influencing yield outcomes, allowing for more targeted and effective predictions. Experimental results demonstrate significant improvements in prediction accuracy and robustness, highlighting the model's potential to support decision-making in agriculture. This approach offers valuable insights for farmers and agricultural stakeholders, paving the way for data-driven strategies in crop management and optimization.

**CHAPTER-3: SYSTEM DESIGN**

**3.1 System Architecture**



**Fig.3.1: System Architecture**

**3.2 Requirements Specifications**

**3.2.1 Hardware Requirements:**

System : Intel I-3, 5, 7 Processor.

Hard Disk : 500 GB.

Floppy Drive : 1.44 Mb.

Monitor : 14’ Colour Monitor.

Mouse : Optical Mouse.

Ram : 2Gb.

**3.2.2 Software Requirements****:**

Operating system : Windows 7,8,10 Ultimate, Linux, Mac.

Front-End : Python

Coding Language : Python.

Software Environment : Anaconda (jupyter or spyder).

* + 1. **User Requirements:**
* User has to load the application before using it
* User needs to have data (text, audio, image, video) which is to be hidden
* User needs to have a master file in which he/she wants to hide the data
* User needs to have a stego-key in order to encrypt or decrypt the data
  + 1. **Functional Requirements:**

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provide 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.
* Understanding user’s preferences, expertise level and his business requirements

through a friendly questionnaire.

* Input data can be in four different forms - Relational DB, text files, xl and xml files. For testing and demo you can choose data from any domain. User-B can provide business data as input.

## 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 object oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

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 object oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

* + 1. **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.

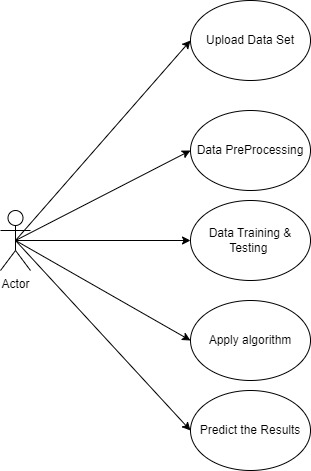
5. Encourage the growth of OO tools market.

6. Support higher level development concepts such as collaborations, frameworks, patterns and components.

7. Integrate best practices.

**3.3.2 USE CASE DIAGRAM:**

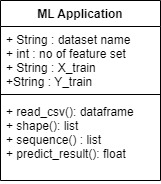
A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies 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. 3.2: Use Case Diagram**

**3.3.3 Class Diagram**

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

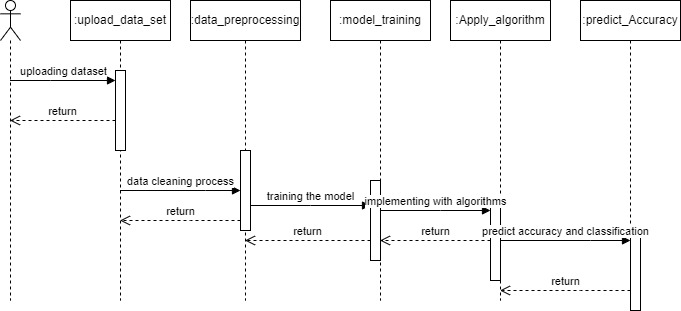


## 

## Fig. 3.3: Class Diagram

**3.4.4 Sequence Diagram**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



## Fig. 3.4: Sequence Diagram

# CHAPTER-4:INPUT AND OUTPUT DESIGN

**4.1 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.
  + 1. **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

**4.1.2 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.

**4.1.3 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.

**4.2 OUTPUT DESIGN**

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provides apermanent 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.

**4.2.1 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

# CHAPTER-5: SOFTWARE ENVIRONMENT

* 1. **Python**

Below are some facts about Python.

* Python is currently the most widely used multi-purpose, high-level programming language.
* Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.
* Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.
* Python language is being used by almost all tech-giant companies like

Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

Machine Learning

* + GUI Applications (like Kivy, Tkinter, PyQt etc.)
  + Web frameworks like Django (used by YouTube, Instagram, Dropbox)
* Image processing (like Opencv, Pillow)
* Web scraping (like Scrapy, Beautiful Soup, Selenium)
* Test frameworks
* Multimedia

**5.1.1 Advantages of Python**

Let’s see how Python dominates over other languages.

**1. Extensive Libraries:**

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don’t have to write the complete code for that manually.

**2. Extensible:**

As we have seen earlier, Python can be extended to other languages. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

**3. Embeddable:**

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add scripting capabilities to our code in the other language.

**4. Improved Productivity:**

The language’s simplicity and extensive libraries render programmers more productive than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

**5. IOT Opportunities:**

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

**6. Simple and Easy:**

When working with Java, you may have to create a class to print ‘Hello World’. But in Python, just a print statement will do. It is also quite easy to learn, understand, and code. This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

**7. Readable:**

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and indentation is mandatory. This further aids the readability of the code.

**8. Object-Oriented:**

This language supports both the procedural and object-oriented programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the encapsulation of data and functions into one.

**9. Free and Open-Source:**

Like we said earlier, Python is freely available. But not only can you download Python for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

**10. Portable:**

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn’t the same with Python. Here, you need to code only once, and you can run it anywhere. This is called Write Once Run Anywhere (WORA). However, you need to be careful enough not to include any system-dependent features.

**11. Interpreted:**

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, debugging is easier than in compiled languages.

Any doubts till now in the advantages of Python? Mention in the comment section.

**5.1.2 Advantages of Python Over Other Languages**

**1. Less Coding:**

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python

also has an awesome standard library support, so you don’t have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

**2. Affordable:**

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.

**3. Python is for Everyone:**

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and machine learning, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

**5.1.3 Disadvantages of Python**

So far, we’ve seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let’s now see the downsides of choosing Python over another language.

**1. Speed Limitations:**

interpreted, it often results in slow execution. This, however, isn’t a problem We have seen that Python code is executed line by line. But since Python is unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

**2. Weak in Mobile Computing and Browsers:**

While it serves as an excellent server-side language, Python is much rarely seen on the client-side. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called Carbonnelle.

The reason it is not so famous despite the existence of Brython is that it isn’t that secure.

**3. Design Restrictions:**

As you know, Python is dynamically-typed. This means that you don’t need to declare the type of variable while writing the code. It uses duck-typing. But wait, what’s that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can raise run-time errors.

**4. Underdeveloped Database Access Layers:**

Compared to more widely used technologies like JDBC (Java Data Base Connectivity) and ODBC (Open Data Base Connectivity), Python’s database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

**5. Simple:**

No, we’re not kidding. Python’s simplicity can indeed be a problem. Take my example. I don’t do Java, I’m more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

**5.1.4 History of Python**

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python. Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners1, Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it. "Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

**5.1.5 Python Development Steps**

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt. sources 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. 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

**5.1.6 Purpose**

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

**Python**

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 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.

**5.1.7 Modules Used in Project**

**5.1.8 TensorFlow:**

TensorFlow 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.‍

TensorFlow 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.

**5.1.9 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 datatypes can be defined using NumPy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

**5.1.10 Pandas:**

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 solved this problem. 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.

**5.1.11 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 IPython shells, the Jupyter 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 python 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.

**5.1.12 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. Python

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 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.

**5.1.13 Install Python Step-by-Step in Windows and Mac:**

Python a versatile programming language doesn’t come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

**5.1.14 How to Install Python on Windows and Mac:**

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python

Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your System Requirements. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a Windows 64-bit operating system. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. Download the Python Cheatsheet here. The steps on how to install Python on Windows 10, 8 and 7 are divided into 4 parts to help understand better.

**5.1.15 Download the Correct version into the system:**

Step 1: Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: [https://www.python.org](https://www.python.org/)

A screenshot of a computer

Description automatically generated with medium confidence

Now, check for the latest and the correct version for your operating system.

Step 2: Click on the Download Tab.

Graphical user interface, application

Description automatically generated

Step 3: You can either select the Download Python for windows 3.7.4 button in Yellow Colour or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for

Graphical user interface, application

Description automatically generated

Step 4: Scroll down the page until you find the Files option.

Step 5: Here you see a different version of python along with the operating system.

Graphical user interface, text

Description automatically generated

* To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.
* To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

Note: To know the changes or updates that are made in the version you can click on the Release Note Option.

Installation of Python

Step 1: Go to Download and Open the downloaded python version to carry out the installation process.Graphical user interface, text, application

Description automatically generated

Step 2: Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.

Graphical user interface, text, application, chat or text message

Description automatically generated

Step 3: Click on Install NOW After the installation is successful. Click on Close.

Graphical user interface, text, application, chat or text message

Description automatically generated

With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Note: The installation process might take a couple of minutes.

Verify the Python Installation

Step 1: Click on Start

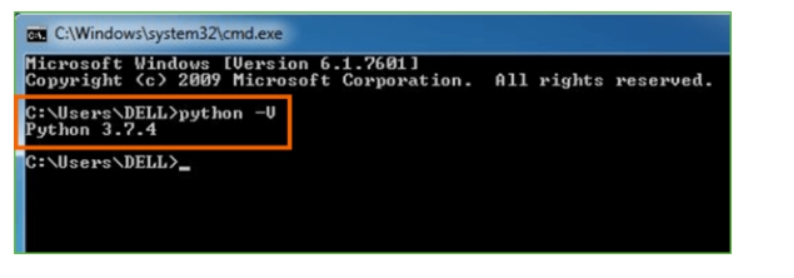
Step 2: In the Windows Run Command, type “cmd”.

Graphical user interface, application

Description automatically generated

Step 3: Open the Command prompt option.

Step 4: Let us test whether the python is correctly installed. Type python –V and press Enter.



Step 5: You will get the answer as 3.7.4

Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

Check how the Python IDLE works

Step 1: Click on Start

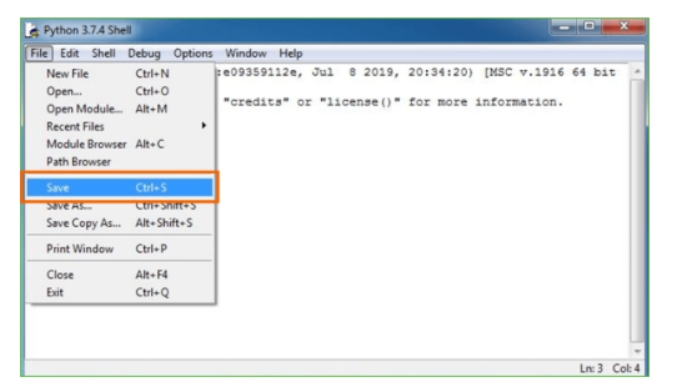
Step 2: In the Windows Run command, type “python idle”.

Application

Description automatically generated with low confidence

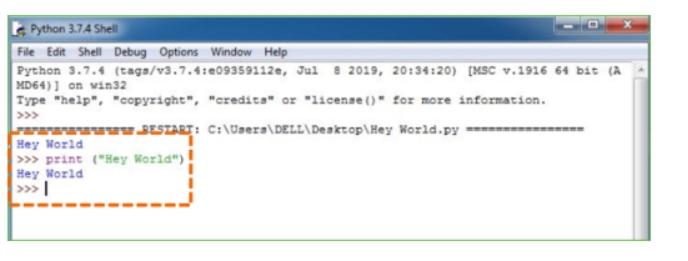
Step 3: Click on IDLE (Python 3.7 64-bit) and launch the program

Step 4: To go ahead with working in IDLE you must first save the file. Click on File > Click on Save



Step 5: Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

Step 6: Now for e.g. enter print (“Hey World”) and Press Enter.



You will see that the command given is launched. With this, we end our tutorial on how to install Python. You have learned how to download python for windows into your respective operating system.

Note: Unlike Java, Python does not need semicolons at the end of the statements otherwise it won’t work.

# 5.2 What is Machine Learning

# 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.

# 5.2.1 Categories of Machine Leaning:

# At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

# Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into classification tasks and regression tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

# Unsupervised learning involves modeling the features of a dataset without reference to any label and is often described as "letting the dataset speak for itself." These models include tasks such as clustering and dimensionality reduction. Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

# 5.2.2 Need for Machine Learning:

# Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate, and solve complex problems. On the other side, AI is still in its initial stage and have not surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, “to make decisions, based on data, with efficiency and scale”.

# Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programming logic, in the problems that cannot be programmed inherently. The fact is that we can’t do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

# 5.2.3 Challenges in Machines Learning:

# While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are −

# 1.Quality of data − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

# 2.Time-Consuming task − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

# 3.Lack of specialist persons − As ML technology is still in its infancy stage, availability of expert resources is a tough job.

# 4.No clear objective for formulating business problems − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

# 5.Issue of overfitting & underfitting − If the model is overfitting or underfitting, it cannot be represented well for the problem.

# 6.Curse of dimensionality − Another challenge ML model faces is too many features of data points. This can be a real hindrance.

# 7.Difficulty in deployment − Complexity of the ML model makes it quite difficult to be deployed in real life.

# 5.2.4 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

# 5.2.5 How to Start Learning 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!!!

# 5.2.6 How to start learning ML:

# This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer. Of course, you can always modify the steps according to your needs to reach your desired end-goal!

# Step 1 – Understand the Prerequisites

# In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python. And if you don’t know these, never fear! You don’t need a Ph.D. degree in these topics to get started but you do need a basic understanding.

# (a)Learn Linear Algebra and Multivariate Calculus

# Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on maths as there are many common libraries available. But if you want to focus on R&D in Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

# (b)Learn Statistics

# Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data. So it is no surprise that you need to learn it!!! Some of the key concepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc.

# (c) Learn Python

# Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is Python! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as Keras, TensorFlow, Scikit-learn, etc.

# So if you want to learn ML, it’s best if you learn Python! You can do that using various online resources and courses such as Fork Python available Free on Geeks for Geeks.

# Step 2 – Learn Various ML Concepts

# Now that you are done with the prerequisites, you can move on to actually learning ML (Which is the fun part!!!) It’s best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

# Terminologies of Machine Learning:

# Model – A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.

# Feature – A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.

# Target (Label) – A target variable or label is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.

# Training – The idea is to give a set of inputs(features) and it’s expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.

# Prediction – Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

# (b) Types of Machine Learning:

# Supervised Learning – This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.

# Unsupervised Learning – This involves using unlabelled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.

# Semi-supervised Learning – This involves using unlabelled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.

# Reinforcement Learning – This involves learning optimal actions through trial and error. So the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

# 5.2.7 Advantages of Machine learning:

# 1. 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.

# 2. No human intervention needed (automation):

# 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 software; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

# 3. 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.

# 4. 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.

# 5. 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.

# 5.2.8 Disadvantages of Machine Learning

# 1. 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.

# 2. 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.

# 3. 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.

# 4. 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.

# CHAPTER6:SYSTEM STUDY

# 6.1 FEASIBILITY STUDY

The feasibility of the project is analayz 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

1. ECONOMICAL FEASIBILITY
2. TECHNICAL FEASIBILITY
3. SOCIAL FEASIBILITY

**6.1.1 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.

**6.1.2 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.

**6.1.3 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.

**CHAPTER****-7**: **SYSTEM TEST**

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, Subassemblies, 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.

### **7.1 TYPES OF TESTS**

### **7.1.1 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.

**7.1.2 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.

**7.1.3 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/procedure : 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.

**7.1.4 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.

**7.1.5 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.

**7.1.6 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.

**7.2 Test strategy and approach**

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

**7.2.1 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.

# 7.2.2 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.

**7.2.3 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.

**CHAPTER-8: RESULTS**

**8.1 Html Home page:**



Fig.8.1: Screenshot of HTML Home Page

**8.2 Upload data:**

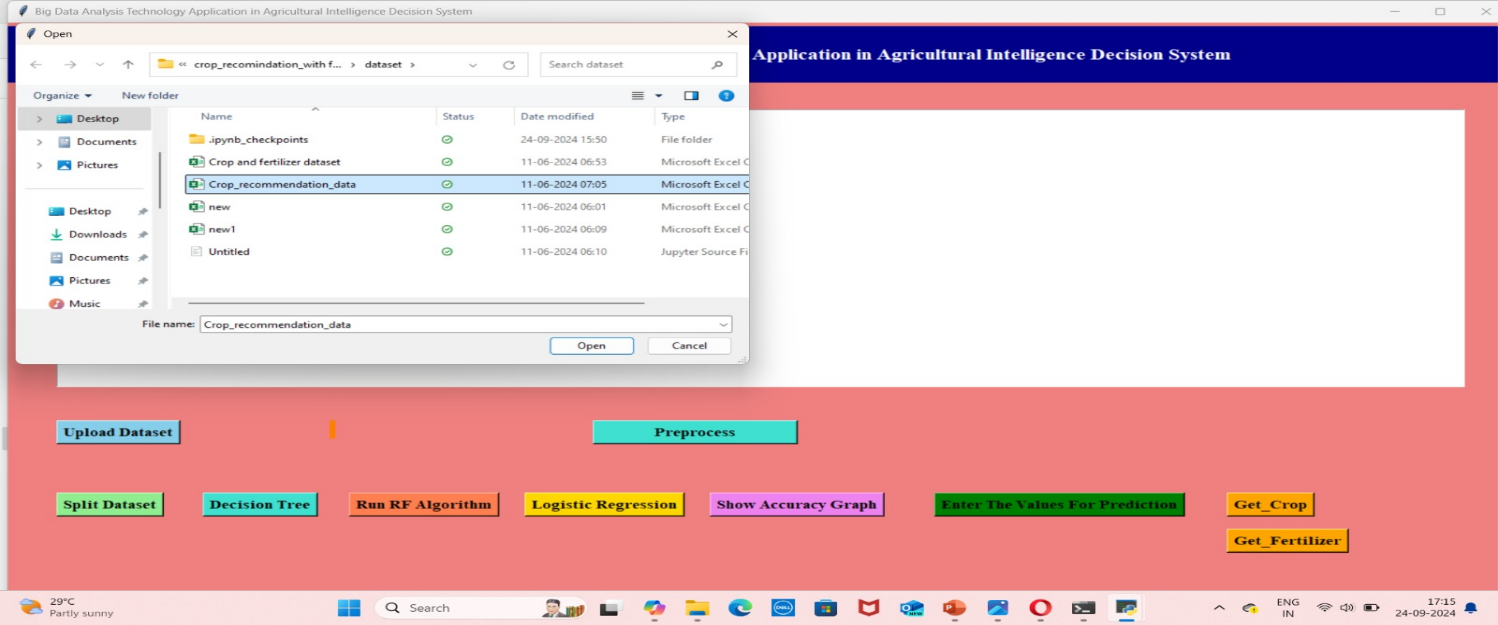
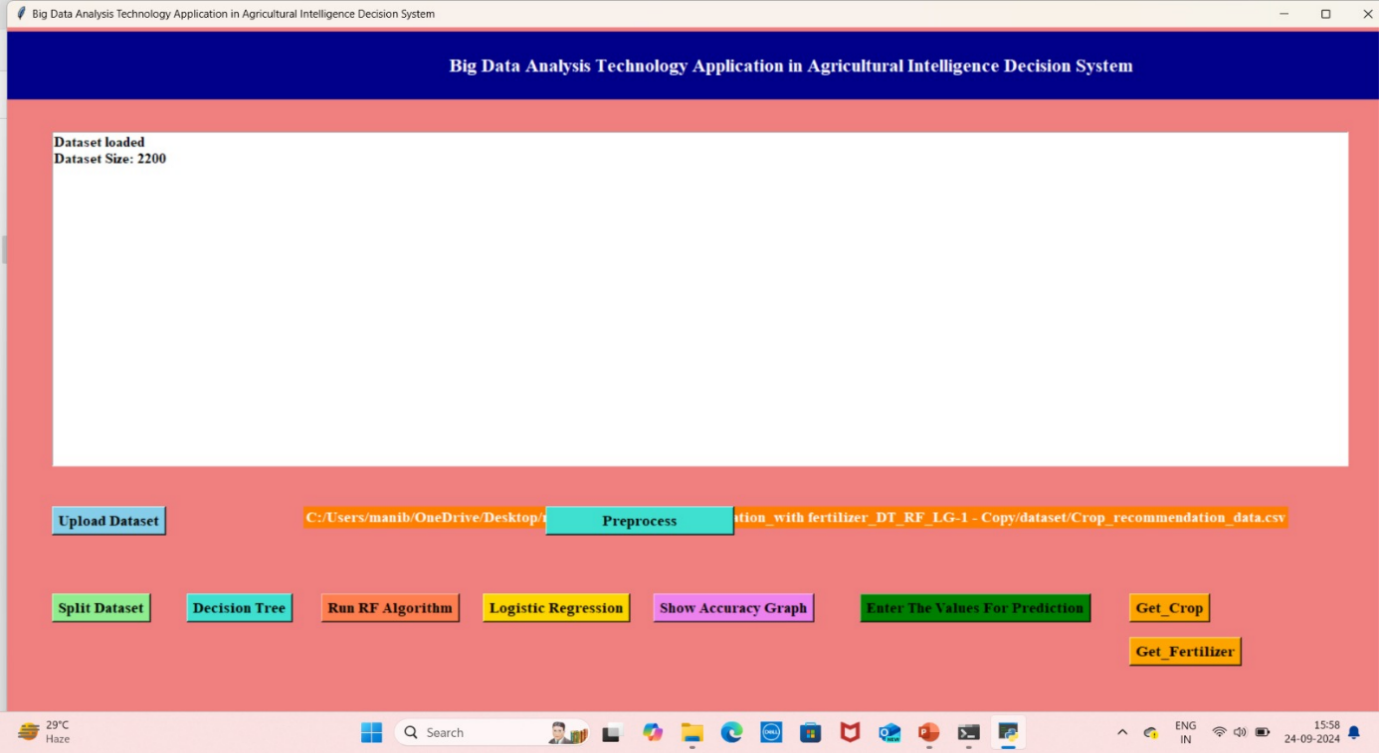
****

Fig.8.2: Screenshot of Upload dataset

**8.3 Split dataset:**

Fig.8.3: screenshot of split dataset

# 8.4 After splitting dataset:

# 

Fig.8.4: screenshot of After Splitting Dataset

# 8.5 RUN\_SVM:

# 

# Fig.8.5: Screenshot of Run SVM Algorithm

# 8.6 Run RF Algorithm:

# 

# Fig.8.6: screenshot of Run RF Algorithm

# 8.7 Logistic Algorithm:

# 

# Fig.8.7: screenshot of Logistic Algorithm

# 8.8 Accuracy Graph:

# 

# Fig.8.8 Screenshot of accuracy graph

# 8.9 Preprocess:

# 

# Fig.8.9: Screenshot of Preprocess

# 8.10 Before entering the values:

# 

# Fig.8.10: screenshot of Before entering the values

# 8.11 Enter the values:

# 

# Fig.8.11: Screenshot of Enter the Values

# 8.12 After entering the values:

# 

# Fig.8.12: Screenshot of After Entering the Values

# 8.13 Prediction of crop:

# 

# Fig.8.13: Screenshot of Prediction of Crops

# 8.14 Final output

# 

# Fig.8.14: Screenshot of Final Output

# CHAPTER-9: CONCLUSION & FUTURE ENHANCEMENT

Machine learning holds tremendous potential to revolutionize the agriculture sector. Crop recommendation systems can aid farmers in optimizing crop yield, minimizing waste, and reducing resource usage. Numerous recent studies, as discussed in our introduction, demonstrate improvements in the agriculture sector through the adoption of machine learning technology. However, widespread adoption still faces several challenges. The availability and quality of data, cost, and the need for technical skills are significant obstacles. Through this research, we aim to help farmers increase crop yields and thereby boost their income. This, in turn, benefits the economy by enhancing the country's agricultural output. Our research focuses on automating the process of crop selection and fertilization for farming.

# CHAPTER10: BIBLIOGRAPHY

[1]Sharma, R., & Kaur, G. (2024). "Crop Yield Prediction and Fertilizer Recommendation Using Machine Learning: A Case Study." Agricultural Informatics, 5(1), 45-60.

[2]Gupta, P., & Raghuvanshi, N. S. (2023). "Multi-Criteria Optimization for Sustainable Agriculture: A Machine Learning Perspective." Sustainability, 15(1), 12-34.

[3]Jha, M. K., & Kumar, A. (2023). "IoT and Machine Learning in Agriculture: A Comprehensive Review on Crop and Fertilizer Recommendation Systems." Sensors, 23(4), 2005.

[4]Singh, A., & Singh, K. (2023). "A Hybrid Model for Crop Prediction and Fertilizer Recommendation Using Machine Learning." Computers and Electronics in Agriculture, 210, 107786.

[5]Patel, D., & Prajapati, R. (2022). "Machine Learning-Based Approach for Crop and Fertilizer Recommendation System." Journal of Ambient Intelligence and Humanized Computing, 13(7), 3523-3533.

[6]Khan, A. A., et al. (2021). "Multi-Criteria Decision Making in Agriculture: A Machine Learning Approach." Agricultural Systems, 191, 103159.

[7]Singh, R., & Rao, P. (2021). "Integrated Crop and Fertilizer Recommendation System Using Data Mining." Computers and Electronics in Agriculture, 181, 105964.

[8]Kaur, P., & Kaur, J. (2020). "Predictive Analysis of Fertilizer Requirement Using Machine Learning Techniques." Journal of Agricultural Science and Technology, 22(3), 1-15.

[9]Patel, S. S., & Patel, S. (2019). "Machine Learning Approaches for Crop Recommendation: A Review." International Journal of Advanced Research in Computer Science, 10(1), 20-25.

[10]Kumar, A., & Singh, M. (2018). "A Review on Crop Recommendation System Using Machine Learning Techniques." International Journal of Computer Applications, 182(15), 1-7.

[11]Maiappan, A. K., & Ds, J. A. B. (2017, April). A paradigm for rice yield prediction in Tamil Nadu. In 2017 IEEE Technological Innovations in ICT for Agriculture and Rural Development (TIAR) (pp.18- 21). IEEE

[12]Gahi, N., Petar, O., & Armstrong, L. J. (2016, July). Rice crop yield prediction using artificial neural networks. In 2016 IEEE Technological Innovations in ICT for Agriculture and Rural Development (TIAR) (pp.105-110). IEEE

[13]Manjula, A., & Narsimha, G. (2015, January). XCYPF: A flexible and extensible framework for agricultural Crop Yield Prediction. In 2015IEEE 9th International Conference on Intelligent Systems and Control (ISCO)(pp.1-5). IEEE.

[14]Ananthara, M. G., Arunkumar, T., &Hemavathi, R. (2013, February). CRI—an improved crop yield prediction model using behave clustering approach for agricultural datasets. In 2013 International Conference Pattern Recognition, Informatic Mobile Engineering (pp. 473-478). IEEE.