A PROJECT REPORT ON

INTELLIGENT AGRICULTURAL CROP RECOMMENDATION SYSTEM BASED ON PRODUCTIVITY, SEASONAL PATTERNS, AND ENVIRONMENTAL DATA USING MACHINE LEARNING

Submitted in partial fulfillment of the requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY

In

CSE - ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

Submitted By

V Divya	**21MQ1A4231**
M Sai Ganesh	**21MQ1A4247**
Ch Sree Charan	**21MQ1A4240**
S Kusuma Pravallika	**22MQ5A4204**

Under the Esteemed Supervision of

T VEENA

Assistant Professor

DEPARTMENT OF

CSE – ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

SRI VASAVI INSTITUTE OF ENGINEERING & TECHNOLOGY

Accredited by NAAC with 'A' Grade & NBA (CSE, ECE & ME)

Approved by AICTE-New Delhi & Affiliated to JNTUK Kakinada

An ISO 9001:2015 Certified Institute

Nandamuru, Pedana(M)-521369, Krishna Dist, Andhra Pradesh.

2021-2025

DEPARTMENT OF
CSE – ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

SRI VASAVI INSTITUTE OF ENGINEERING & TECHNOLOGY

Accredited by NAAC with 'A' Grade & NBA (CSE, ECE & ME)

Approved by AICTE-New Delhi & Affiliated to JNTU, Kakinada

An ISO 9001:2015 Certified Institute

Nandamuru, Pedana(M)-521369, Krishna Dist, Andhra Pradesh.

2021-2025

CERTIFICATE

This is to certify that the project report entitled " **INTELLIGENT AGRICULTURAL CROP RECOMMENDATION SYSTEM BASED ON PRODUCTIVITY, SEASONAL PATTERNS AND ENVIRONMENTAL DATA USING MACHINE LEARNING** " is a bonafide work carried out by **V.Divya (21MQ1A4231), M.Sai Ganesh (21MQ1A4247**), **Ch.Sree Charan (21MQ1A4240**), **S.Kusuma Pravallika (22MQ5A4204).** Under the guidance and supervision in partial fulfillment of the requirements for the award of degree of **B.Tech** in **CSE - Artificial Intelligence and Machine Learning** from **Jawaharlal Nehru Technological University, Kakinada**. The results embodied in this project report have not been submitted to any other University or Institute for the award of any degree or diploma.

PROJECT GUIDE

T.Veena
MCA.,M.Tech.,(Ph.D).
Assistant Professor

HEAD OF THE DEPARTMENT

Dr.G.Syam Prasad
B.Tech.,M.Tech.,Ph.D.
Professor & HOD

EXTERNAL EXAMINER

DECLARATION

We certify that,

- The work contained in this report is original and has been done by us under the guidance of our supervisor.
- The work has not been submitted to any other institute for any degree or diploma. We have followed the guidelines provided by the institute in preparing the report.
- We confirm to the norms and guidelines given in the Ethical code of Conduct of the institute.
- Whenever we have used materials (data, theoretical analysis, figures and text) from the sources, we have given due credit to them by references.
- Further, we have taken permission from the copy right owners of the sources whenever necessary.

Signature(s) of the student(s):

V Divya	**21MQ1A4231**
M Sai Ganesh	**21MQ1A4247**
Ch Sree Charan	**21MQ1A4240**
S Kusuma Pravallika	**22MQ5A4204**

ACKNOWLEDGEMENT

We take great pleasure to express our deep sense of gratitude to our project guide **T.Veena, Assistant Professor** for her valuable guidance during the course of our project work.

We would like to thank **Dr.G.Syam Prasad, Professor & Head of the Department of C.S.E – Artificial Intelligence & Machine Learning** for his encouragement.

We would like to express our heart-felt thanks to **Dr.B.R.S.REDDY,** Principal, **Sri Vasavi Institute of Engineering & Technology**, Nandamuru for providing all the facilities for our project.

Our utmost thanks to all the Faculty members and Non-Teaching Staff of the Department of Computer Science and Engineering for their support throughout our project work.

Our Family Members and Friends receive our deepest gratitude and love for their support throughout our academic year.

Project Associates:

V Divya	**21MQ1A4231**
M Sai Ganesh	**21MQ1A4247**
Ch Sree Charan	**21MQ1A4240**
S Kusuma Pravallika	**22MQ5A4204**

ABSTRACT

Agriculture is one of the major and the least paid occupation in India. Machine learning can bring a boom in the agriculture field by changing the income scenario through growing the optimum crop. This paper focuses on predicting the yield of the crop by applying various machine learning techniques. The outcome of these techniques is compared on the basis of mean absolute error. The prediction made by machine learning algorithms will help the farmers to decide which crop to grow to get the maximum yield by considering factors like temperature, rainfall, area, etc. Agriculture is the foremost factor which is important for the survival of human beings. One of those is crop recommendation. Crop productivity is boosted as a result of accurate crop prediction. As crop production has already started to suffer from climate change, improving crop output is consequently desirable because agronomists are impotent to select the appropriate crop(s) depending on environmental and soil parameters, and the mechanism of forecasting the selection of the appropriate crops manually has failed. The system would help the farmers for the appropriate decision to be taken regarding the crop type.

Index terms : Crop Recommendation System, Crop Yield Prediction, Long Short-Term Memory (LSTM), K-Nearest Neighbors (KNN),

LIST OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATION	EXPLANATION
CNN	Convolutional Neural Network
SVM	Support Vector Machine
DT	Decision Tree
KNN	K Nearest Neighbour
ML	Machine Learning
SVC	Support Vector Classifier
LSTM	Long Term Short Memory

LIST OF TABLES

CHAPTER-1
INTRODUCTION

INTRODUCTION

1.1 Introduction

Agriculture in India has deep historical roots, dating back to the Indus Valley Civilization. Over the centuries, it has evolved into one of the most essential components of the Indian economy. Even today, India ranks second in the world in agricultural output and is the largest country by net cropped area, surpassing even the United States and China. Agriculture and allied sectors, including forestry and fisheries, account for approximately 15.4% of the country's GDP and employ nearly 31% of the workforce, making it the single largest source of livelihood for the majority of the population.

Despite this importance, the economic contribution of agriculture to India's GDP has been on the decline, largely due to the country's industrialization and rapid urban development. Farmers face numerous challenges: climate change, erratic rainfall, rising input costs, soil degradation, and lack of access to timely and accurate information. These issues affect not only crop productivity but also the livelihoods of millions who depend on farming. Traditional agricultural practices, while rooted in local wisdom, often lack the precision needed in today's rapidly changing environmental conditions. As weather patterns become increasingly unpredictable due to global warming and industrial pollution, farmers struggle to make informed decisions about what to grow, when to sow, and how to manage resources efficiently. This uncertainty leads to reduced crop yields and financial instability. In this scenario, Machine Learning (ML) emerges as a powerful solution. By analyzing historical data on weather, soil conditions, crop yields, and seasonal patterns, machine learning models can predict the most suitable crops for a given region and season. These intelligent systems offer data-driven recommendations, helping farmers choose the right crop, minimize risk, and maximize productivity. Ultimately, this intelligent system strives to enhance agricultural efficiency, promote sustainable farming, and empower Indian farmers with the tools they need to thrive in the 21st century.

1.2 Scope:

The scope of this project revolves around building an intelligent and automated crop recommendation **system** that leverages machine learning techniques to analyze and interpret various agricultural and environmental parameters. The project is designed to assist farmers, agricultural planners, and agronomists in making data-driven decisions regarding which crop to cultivate in a particular region and season for maximum yield and profitability.

1.3 Current Scope:

The current scope of this project focuses on developing an intelligent system that assists farmers in selecting the most suitable crop based on environmental conditions, seasonal patterns, and productivity factors. By leveraging machine learning algorithms such as Random Forest, SVM, KNN, RNN, LSTM, and CNN, the system analyzes key parameters like temperature, rainfall, and soil characteristics to provide accurate crop recommendations and yield predictions. The project includes collecting and preprocessing relevant datasets, training and validating predictive models, and implementing a user-friendly web interface for real-time crop suggestions. This system aims to enhance decision-making in agriculture, improve crop productivity, and support sustainable farming practices through data-driven insights.

1.4 Future Scope:

The proposed system has strong potential for future enhancements and real-world applications. In the future, this project can be expanded to integrate live data from IoT sensors placed in agricultural fields to monitor real-time soil conditions, temperature, and humidity, allowing for dynamic and more accurate crop recommendations. The system can also be enhanced to include market price forecasting, helping farmers choose crops based not only on yield but also on profitability. Integration with satellite imagery and remote sensing data can further improve prediction accuracy and support large-scale deployment. Additionally, multilingual mobile app development can make the system more accessible to farmers across different regions, promoting digital agriculture and supporting government initiatives like Smart Farming and Digital India.

1.5 Project Summary and Purpose:

The Indian Agriculture sector is facing is the integration of technology to bring the desired outputs. With the advent of new technologies and overuse of non-renewable energy resources patterns of rainfall and temperature are disturbed. The inconsistent trends developed from the side effects of global warming make it cumbersome for the farmers to clearly predict the temperature and rainfall patterns thus affecting their crop yield productivity. In order to perform accurate prediction and handle inconsistent trends in temperature and rainfall various machine learning algorithms like RNN, LSTM, etc can be applied to get a pattern. It will complement the agricultural growth in India and all together augment the ease of living for farmers. In past, many researchers have applied machine learning techniques to enhance agricultural growth of the country.

1.6 Objectives

- To develop a smart crop recommendation system using machine learning algorithms that analyze environmental and seasonal factors for better crop selection.To collect and process agricultural data including soil type, temperature, rainfall, humidity, and crop season to be used as input features for prediction.

- To apply machine learning models such as Random Forest, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), CNN, RNN, and LSTM for accurate crop prediction and recommendation.

- To compare the performance of different algorithms and identify the most efficient one based on accuracy, precision, and error metrics.

- To design a user-friendly interface for farmers that allows them to input their local data and receive clear, actionable crop recommendations.

- To promote precision agriculture by helping farmers choose crops based on scientific data rather than assumptions or traditional practices.

- To improve agricultural sustainability by aligning crop choices with environmental conditions, thus reducing the impact of over-farming or unsuitable cultivation.

CHAPTER-2

OBJECTIVE OF THE PROJECT

2. OBJECTIVE OF THE PROJECT

2.1 Existing System:

Due to the revolution in industrialization, the economic contribution of agriculture to India's GDP is steadily declining with the country's broad-based economic growth. The problem that the Indian Agriculture sector is facing is the integration of technology to bring the desired outputs. With the advent of new technologies and overuse of non-renewable energy resources patterns of rainfall and temperature are disturbed. The inconsistent trends developed from the side effects of global warming make it cumbersome for the farmers to clearly predict the temperature and rainfall patterns thus affecting their crop yield productivity. In order to perform accurate prediction and handle inconsistent trends in temperature and rainfall various machine learning algorithms like RNN, LSTM, etc can be applied to get a pattern. It will complement the agricultural growth in India and all together augment the ease of living for farmers. In past, many researchers have applied machine learning techniques to enhance agricultural growth of the country.

LIMITATIONS-

Declining Agricultural Contribution

The economic contribution of agriculture to India's GDP is steadily declining due to industrialization.

Lack of Data-Driven Decisions

Farmers primarily depend on traditional knowledge and experience rather than scientific data. There is no proper analysis of historical crop data, weather patterns, or soil information.

Inability to Handle Climate Variability

Modern technologies like Machine Learning, IoT, or predictive analytics are rarely used. Most farmers are not exposed to digital tools that can assist in crop planning.

Lack of Personalization

Crop selection is generalized and not tailored to specific regions or environmental conditions. Local soil health, seasonal data, and farmer-specific needs are often ignored.

Low Awareness and Accessibility

Many farmers, especially in remote areas, are unaware of or lack access to smart farming solutions. There is limited training or educational support on using modern agricultural technologies.

Inconsistent Weather Trends

Global warming has caused inconsistent temperature and rainfall patterns, making it hard for farmers to plan.

Lack of Advanced Tools

Traditional methods aren't sufficient; there's a need for machine learning algorithms to analyze patterns in rainfall and temperature.

2.2 Proposed system:

This paper focuses on the practical application of machine learning algorithms and its quantification. The work presented here also takes into account the inconsistent data from rainfall and temperature datasets to get a consistent trend. Crop yield prediction is determined by considering all the features in contrast with the usual trend of determining the prediction considering one feature at a time. The Proposed System for enhancing prediction accuracy in agriculture using machine learning (ML). It describes a user-friendly web interface designed to make predictions more accessible to farmers. The system aims to provide comprehensive recommendations by analyzing multiple factors such as crop yield, market prices, and farmer preferences.

ADVANTAGES

Enhanced Prediction Accuracy

The system uses machine learning (ML) algorithms that analyze all features together rather than one at a time. This leads to more accurate and reliable crop yield predictions.

Multi-Factor Analysis

Unlike traditional systems that rely on a single feature, this system considers multiple important factors (like temperature, rainfall, soil conditions, etc.), leading to better decision-making.

.

User-Friendly Web Interface

The system includes a simple and interactive web platform, making it easy for farmers and users with minimal technical knowledge to use.

Better Decision Support for Farmers

By analyzing various factors such as market prices, crop yield potential, and farmer preferences, the system offers recommendations that help farmers make better decisions.

Improved Resource Utilization

With better predictions, farmers can plan the use of resources like water, fertilizers, and labor more efficiently.

2.3 Workflow of the Proposed System

The proposed system follows a structured workflow to recommend the most suitable crop based on environmental, seasonal, and productivity data. Below is the step-by-step process

1. Data Collection

- Gather historical data related to crop yield, rainfall, temperature, humidity, soil type, and region.
- Sources can include government agricultural databases, weather APIs, and sensor devices (if applicable).

2. Data Preprocessing

- Clean and normalize the collected data to remove noise and handle missing values.
- Encode categorical variables (e.g., soil type) and scale numerical data for model input.

3. Feature Selection

- Select the most relevant features that influence crop yield, such as rainfall, temperature, and soil pH.
- Use correlation analysis or feature importance methods for better model performance.

4. Model Training

- Train multiple machine learning models such as Random Forest, SVM, KNN, CNN, RNN, and LSTM using the preprocessed data.
- Split the data into training and testing sets to evaluate model accuracy.

5. Model Evaluation

- Compare model performances using metrics like Mean Absolute Error (MAE), Accuracy, and F1-Score.
- Select the best-performing model for final deployment.

6. Crop Recommendation

- Input current environmental conditions and location into the system.
- The trained model predicts and recommends the most suitable crop for that region and season.

7. Output Display

- The system displays the recommended crop along with supporting data such as expected yield, suitable sowing period, and climatic compatibility.

CHAPTER-3

LITERATURE SURVEY

3 LITERATURE SURVEY

3.1 PREDICTING YIELD OF THE CROP USING MACHINE LEARNING ALGORITHM

The agriculture plays a dominant role in the growth of the country's economy Climate and other environmental changes has become a major threat in the agriculture field. Machine learning (ML) is an essential approach for achieving practical and effective solutions for this problem. Crop Yield Prediction involves predicting yield of the crop from available historical available data like weather parameter, soil parameter and historic crop yield. This paper focus on predicting the yield of the crop based on the existing data by using Random Forest algorithm. Real data of Tamilnadu were used for building the models and the models were tested with samples. The prediction will helps to the farmer to predict the yield of the crop before cultivating onto the agriculture field. To predict the crop yield in future accurately Random Forest, a most powerful and popular supervised machine learning algorithm is used.

3.2 APPLICATIONS OF MACHINE LEARNING TECHNIQUES IN AGRICULTURAL CROP PRODUCTION: A REVIEW

This paper has been prepared as an effort to reassess the research studies on the relevance of machine learning techniques in the domain of agricultural crop production. Methods/Statistical Analysis: This method is a new approach for production of agricultural crop management. Accurate and timely forecasts of crop production are necessary for important policy decisions like import-export, pricing marketing distribution etc. which are issued by the directorate of economics and statistics. However one has understand that these prior estimates are not the objective estimates as these estimate requires lots of descriptive assessment based on many different qualitative factors. Hence there is a requirement to develop statistically sound objective prediction of crop production. That development in computing and information storage has provided large amount of data. Findings: The problem has

been to intricate knowledge from this raw data, this has lead to the development of new approach and techniques such as machine learning that can be used to unite the knowledge of the data with crop yield evaluation. This research has been intended to evaluate these innovative techniques such that significant relationship can be found by their applications to the various variables present in the data base. Application/Improvement: The few techniques like artificial neural networks, Information Fuzzy Network, Decision Tree, Regression Analysis, Bayesian belief network. Time series analysis, Markov chain model, k-means clustering, k nearest neighbor, and support vector machine are applied in the domain of agriculture were presented.

3.3 A Model for Prediction of Crop Yield.

Data Mining is emerging research field in crop yield analysis. Yield prediction is a very important issue in agricultural. Any farmer is interested in knowing how much yield he is about to expect. In the past, yield prediction was performed by considering farmer's experience on particular field and crop. The yield prediction is a major issue that remains to be solved based on available data. Data mining techniques are the better choice for this purpose. Different Data Mining techniques are used and evaluated in agriculture for estimating the future year's crop production. This research proposes and implements a system to predict crop yield from previous data. This is achieved by applying association rule mining on agriculture data. This research focuses on creation of a prediction model which may be used to future prediction of crop yield. This paper presents a brief analysis of crop yield prediction using data mining technique based on association rules for the selected region i.e. district of Tamil Nadu in India. The experimental results shows that the proposed work efficiently predict the crop yield production.

3.4 AGRICULTURAL CROP YIELD PREDICTION USING ARTIFICIAL NEURAL NETWORK APPROACH

By considering various situations of climatologically phenomena affecting local weather conditions in various parts of the world. These weather conditions have a direct effect on crop yield. Various researches have been done exploring the connections between large-scale climatologically phenomena and crop yield. Artificial neural networks have been demonstrated

to be powerful tools for modeling and prediction, to increase their effectiveness. Crop prediction methodology is used to predict the suitable crop by sensing various parameter of soil and also parameter related to atmosphere. Parameters like type of soil, PH, nitrogen, phosphate, potassium, organic carbon, calcium, magnesium, sulphur, manganese, copper, iron, depth, temperature, rainfall, humidity. For that purpose we are used artificial neural network (ANN).

3.5 PREDICTIVE ABILITY OF MACHINE LEARNING METHODS FOR MASSIVECROP YIELD PREDICTION.

An important issue for agricultural planning purposes is the accurate yield estimation for the numerous crops involved in the planning. Machine learning (ML) is an essential approach for achieving practical and effective solutions for this problem. Many comparisons of ML methods for yield prediction have been made, seeking for the most accurate technique. Generally, the number of evaluated crops and techniques is too low and does not provide enough information for agricultural planning purposes. This paper compares the predictive accuracy of ML and linear regression techniques for crop yield prediction in ten crop datasets. Multiple linear regression, M5-Prime regression trees, perceptron multilayer neural networks, support vector regression and k-nearest neighbor methods were ranked. Four accuracy metrics were used to validate the models: the root mean square error (RMS), root relative square error (RRSE), normalized mean absolute error (MAE), and correlation factor (R). Real data of an irrigation zone of Mexico were used for building the models. Model were tested with samples of two consecutive years. The results show that M5- Prime and k-nearest neighbor techniques obtain the lowest average RMSE errors (5.14 and 4.91), the lowest RRSE errors (79.46% and 79.78%), the lowest average MAE errors (18.12% and 19.42%), and the highest average correlation factors (0.41 and 0.42). Since M5-Prime achieves the largest number of crop yield models with the lowest errors, it is a very suitable tool for massive crop yield prediction in agricultural planning.

CHAPTER-4

SYSTEM ANALYSIS

SYSTEM ANALYSIS

Feasibility Report:

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 REQUIREMENTS SPECIFICATION

SYSTEM REQUIREMENTS SPECIFICATIONS

A Software Requirements Specification (SRS) – a requirements specification for a software system – is a complete description of the behavior of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

System requirements specification: A structured collection of information that embodies the requirements of a system. A business analyst, sometimes titled system analyst, is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the systems development life cycle domain, typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers.

5.1 TOOLS USED:

Python 3.10:

Python is a high-level, interpreted programming language known for its simplicity and readability. It is widely used in machine learning due to its extensive library support and developer-friendly syntax.

Scikit-learn:

A powerful Python library used for implementing traditional machine learning algorithms like Random Forest, SVM, and KNN. It provides simple tools for classification, regression, and clustering.

TensorFlow / Keras:

TensorFlow is a deep learning framework developed by Google. Keras is a user-friendly API built on top of TensorFlow for building neural networks like CNN, RNN, and LSTM models.

Pandas:

A data analysis library in Python that provides data structures like DataFrames, which help in organizing and manipulating large datasets effectively.

NumPy:

A library used for numerical computing. It supports powerful operations on arrays and matrices, which are the core data structures in machine learning.

Matplotlib / Seaborn:

These are data visualization libraries used to create charts, graphs, and plots. Seaborn is built on top of Matplotlib and provides more advanced and aesthetically pleasing visualizations.

Flask:

A lightweight web application framework in Python used to build the backend of your project. It allows you to create APIs and serve machine learning models to the web interface.

MySQL:

MySQL is a relational database management system used to store and manage structured data. It helps in storing user input, environmental data, and the recommended crop results securely and efficiently.

Jupyter Notebook:

An interactive development environment that allows you to write code, visualize data, and document your work in a single notebook interface. Ideal for testing and presenting machine learning workflows.

VS Code / PyCharm:

These are Integrated Development Environments (IDEs) that offer code editing, debugging, and version control integration. They improve productivity while writing and managing code.

Heroku / AWS (Amazon Web Services):

These platforms are used to deploy web applications online. Heroku is easy to use for small projects, while AWS provides cloud computing infrastructure for scalable applications.

GitHub:

A version control platform based on Git. It allows developers to store project code, collaborate with team members, and track changes over time.

5.2 Functional Requirements:

5. 2.1 Hardware Requirements:

Processor: Intel i5 or above / AMD Ryzen 5 or above

RAM: 8 GB or more (for handling larger datasets and training ML models)

Hard Disk: 100 GB SSD (for faster read/write and data processing)

Graphics Card (Optional): NVIDIA GPU with CUDA support (for deep learning model acceleration)

Internet Connection: Stable broadband connection (for cloud services like Google Colab, AWS, etc.)

Monitor: 15" LED or LCD Display

Keyboard: Standard Windows-compatible keyboard

Mouse: Optical mouse with two or three buttons

5.2.1 Software Requirements:

Operating System: Windows 10 / 11 (64-bit)

Python 3.7 or above: Primary programming language

Jupyter Notebook / Google Colab: For model training and experimentation

VS Code / PyCharm: IDEs for code development

Scikit-learn: Traditional ML algorithms (Random Forest, SVM, KNN)

TensorFlow / Keras: Deep learning models (CNN, RNN, LSTM)

Pandas: Data handling and preprocessing

NumPy: Numerical operations and matrix computations

Matplotlib / Seaborn: Data visualization and graphing

Flask: For building the backend and connecting ML models with the front-end interface

MySQL: For storing user data, environmental inputs, and prediction results

CHAPTER-6

SYSTEM DESIGN

SYSTEM DESIGN

6.1 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.

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.

6.1.1 SYSTEM ARCHITECTURE:

The system architecture of the **Intelligent Agricultural Crop Recommendation System** is designed to integrate data processing, machine learning, and user interaction seamlessly. It follows a modular, layered approach that ensures scalability, maintainability, and efficient performance**.**

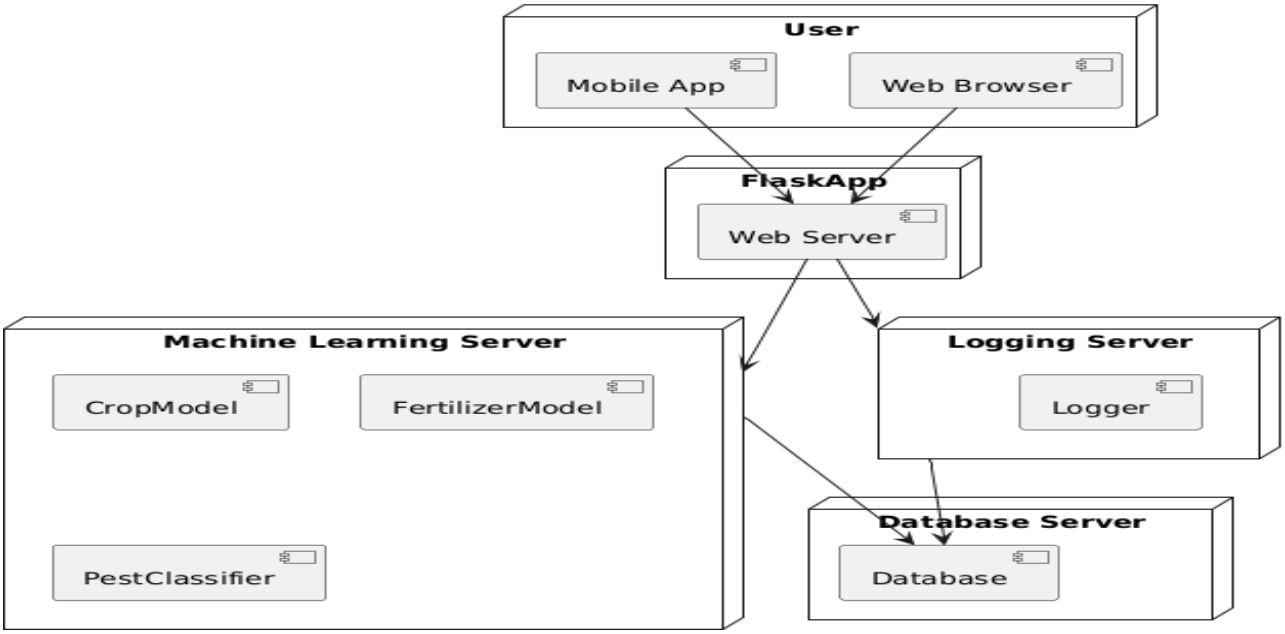


Fig: system architecture

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.

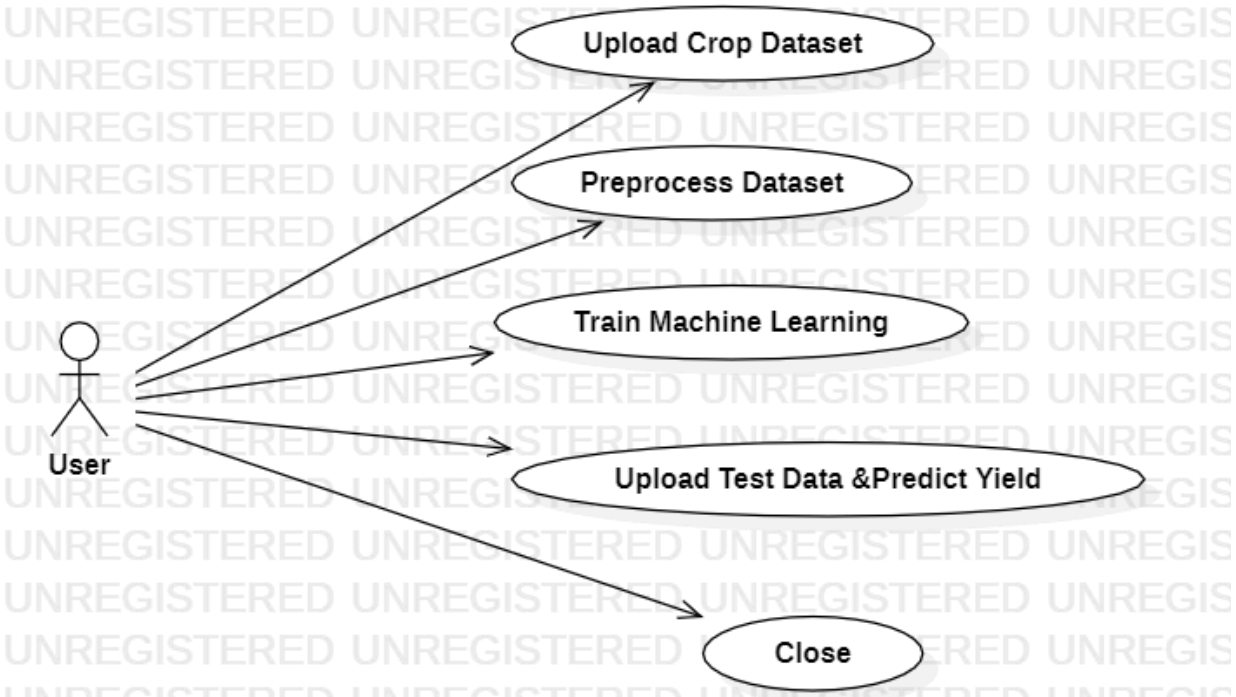


Fig: use case diagram

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.

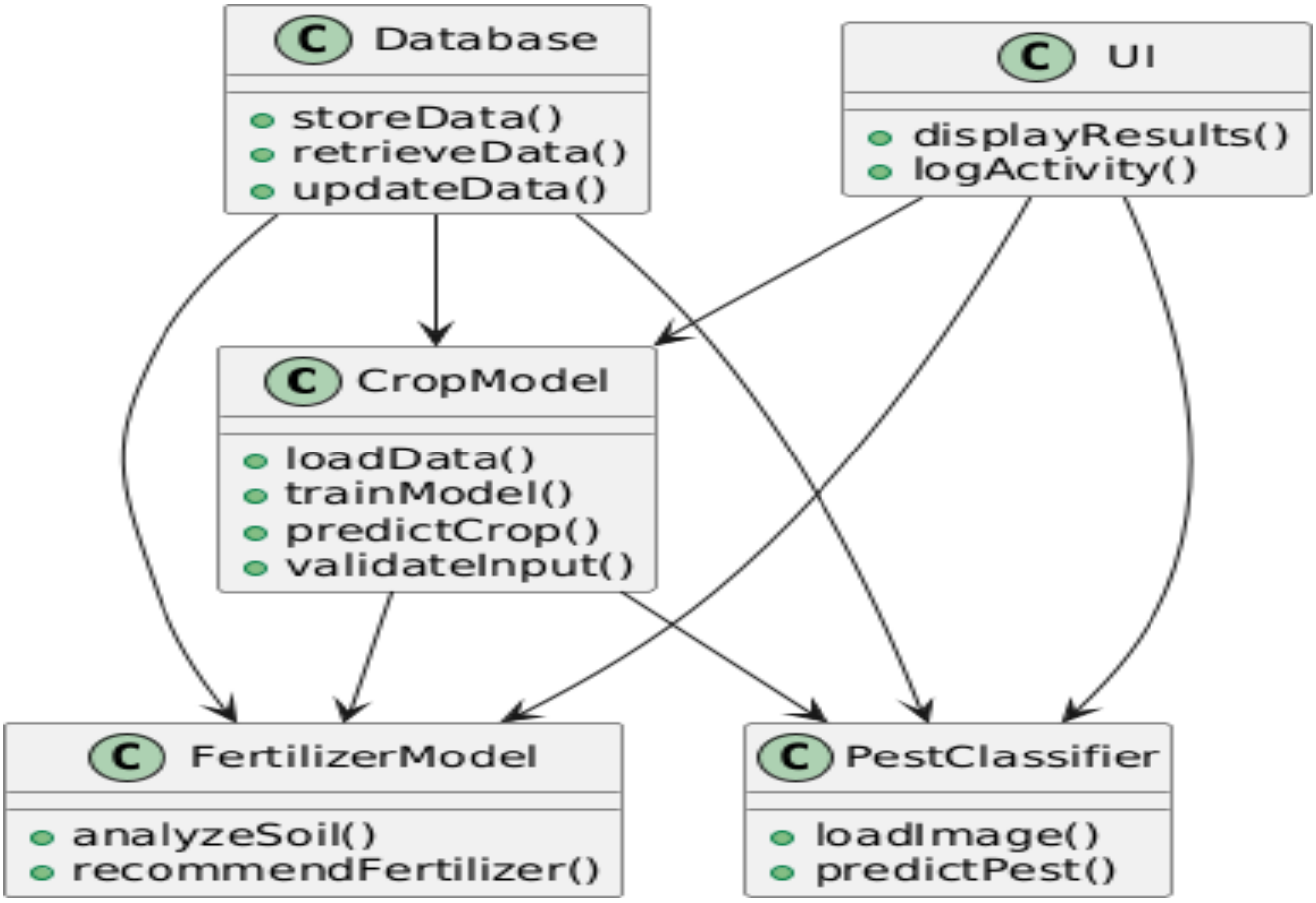


Fig: class diagram

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.

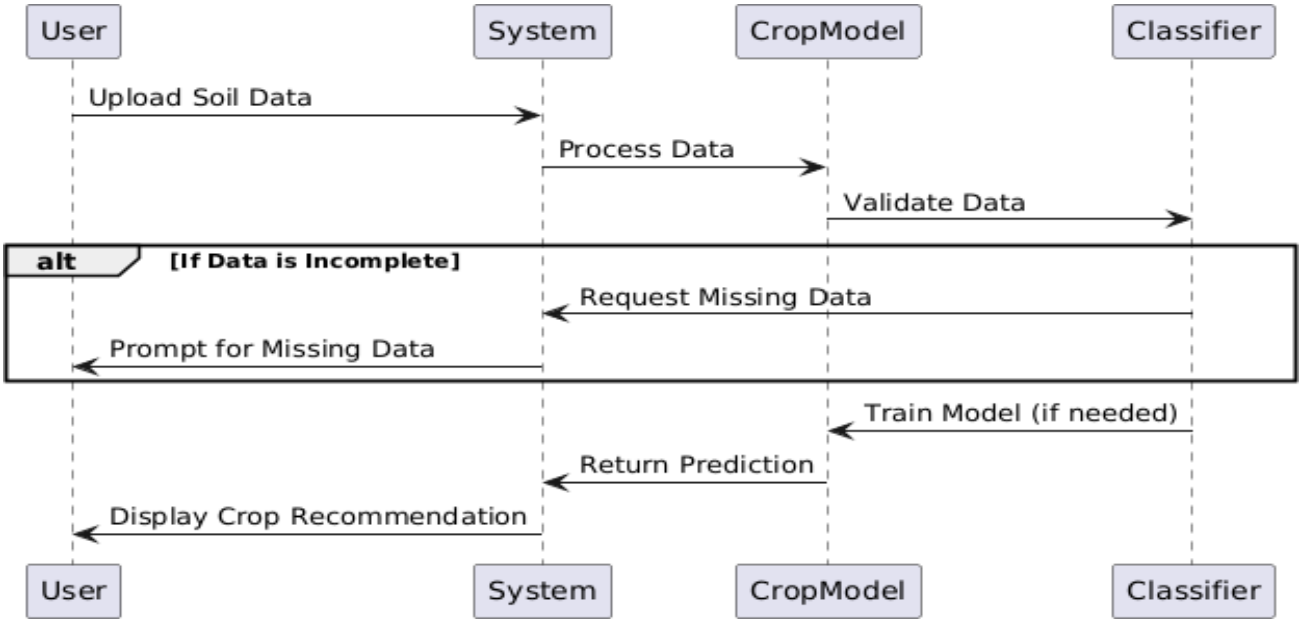
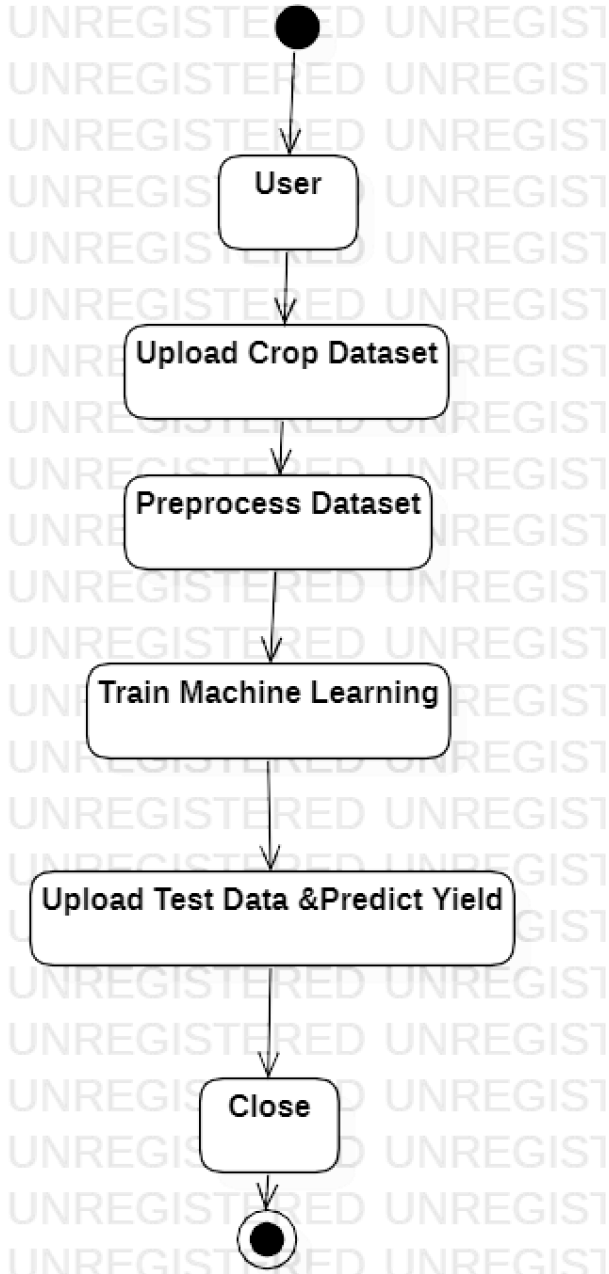


Fig: sequence diagram

ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



CHAPTER-7

TECHNOLOGY DESCRIPTION AND IMPLEMENTATION

7 DESCRIPTION AND IMPLEMENTATION

PYTHON

What is Python: -

Below are some facts about Python.

1. Python is currently the most widely used multi-purpose, high-level programming language.

2. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

3. Programmers have to type relatively less and the indentation requirement of the language, makes them readable all the time.

4. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

Advantages of Python: -

Let's see how Python dominates over other languages.

Extensive Libraries:

Python downloads with an extensive library and it contain code for various purposes like regular expres sions, 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.

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.

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.

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.

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.

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. These further aids the readability of the code.

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 encapsu lation of data and functions into one.

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.

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.

Interpreted:

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

Advantages of Python Over Other Languages

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.

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**.

.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.

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.

Speed Limitations:

We have seen that Python code is executed line by line. But since Python is interpreted, it often results in slow execution. This, however, isn't a problem 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.

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.

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.

Underdeveloped Database Access Layers:

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

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.

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 group

ing 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."

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. Fundament ally, 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.

Categories Of Machine Leaning: -

At the most fundamental level, machine learning can be categorized into two main types:

1. Supervised learning
2. Unsupervised learning.

Supervised learning involves somehow modelling 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 modelling 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.

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 haven't 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.

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 – **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.

Time-Consuming task: -

Another challenge faced by ML models is the consumption of time especially for data acquisition, feature

extraction and retrieval.

Lack of specialist persons: -

As ML technology is still in its infancy stage, availability of expert resources is a tough job.

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.

Issue of overfitting & underfitting: -

If the model is overfitting or underfitting, it cannot be represented well for the problem.

Curse of dimensionality: -

Another challenge ML model faces is too many features of data points. This can be a real hindrance.

Difficulty in deployment: -

Complexity of the ML model makes it quite difficult to be deployed in real life.

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

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!!!

How to start learning ML?

This is a rough road map 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 GeeksforGeeks.

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:

(a) 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 colour, 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 labelled 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 labelled data. Using labelled 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 behaviours that are based on the current state and that will maximize the reward in the future.

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 behaviours 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 (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 com mon example of this is anti-virus software's; 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.

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 lists, 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. 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"

behavior. Text Vs. Data Instead Of Unicode Vs. 8-bit

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.

Modules Used in Project :

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.

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

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.

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 pyplot module provides a MATLAB-like interface, particularly when combined with IPython. 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.

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.

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.

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 3.

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 Cheat sheet here. The steps on how to install Python on Windows 10, 8 and 7 are divided into 4 parts to help understand better.

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

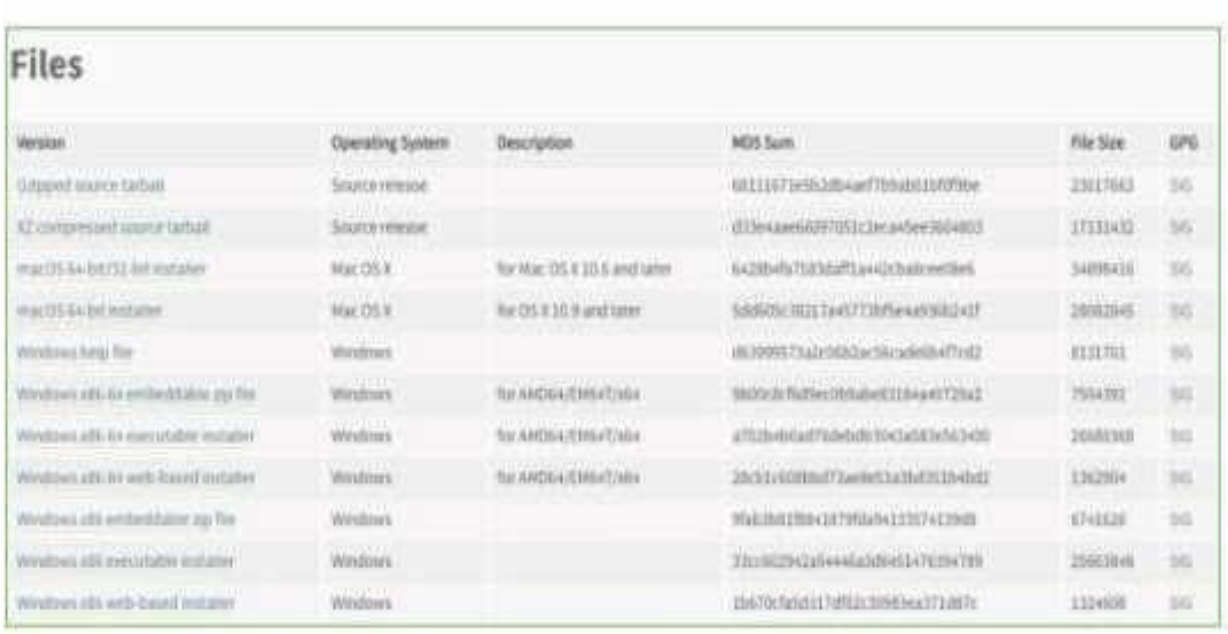
Step 2: Click on the Download Tab.

Step 3: You can either select the Download Python for windows 3.7.4 button in Yellow Color 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 windows 3.7.4

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.



- 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.

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

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

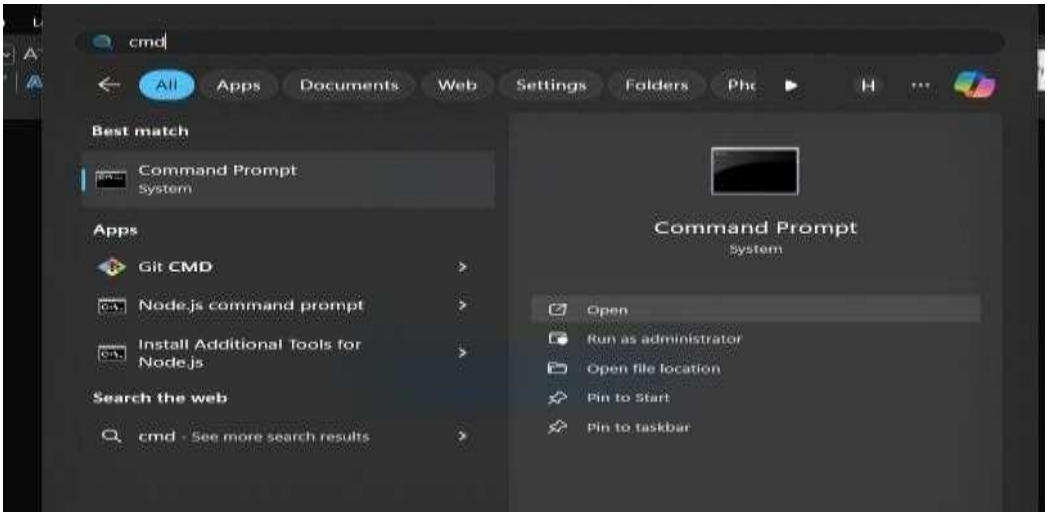
With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

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".



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

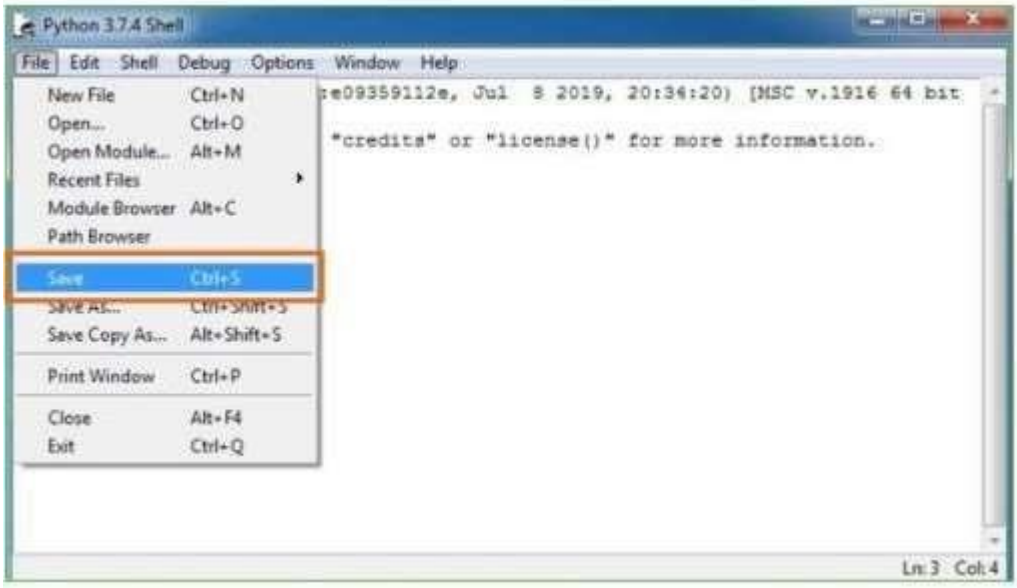
Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

Check how the Python IDLE works Step 1: Click on Start

Step 2: In the Windows Run command, type "python idle

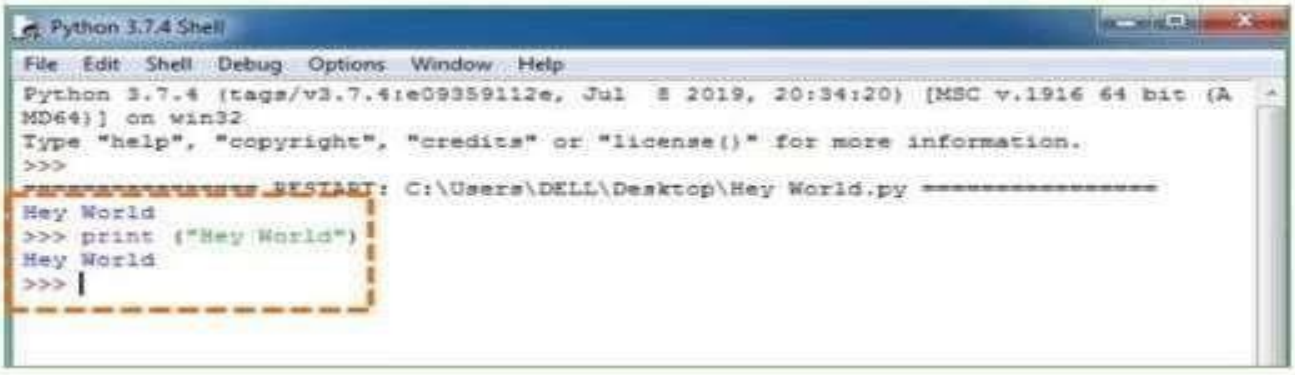
Step 3: Click on IDLE (Python 3.7 64-bit) and launch the program

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 doesn't need semicolons at the end of the statements otherwise it won't work.

CHAPTER-8

SOURCE CODE

SOURCE CODE

```
#App.py
from flask import Flask,
render_template, request from
markupsafe import Markup
import pandas
as pd import os
import numpy as np
from tensorflow.keras.preprocessing
import image from
tensorflow.keras.models import
load_model import pickle
from werkzeug.utils import secure_filename

# Load trained models
classifier=load_model('Trained_model.h5')
crop_recommendation_model_path =
'Crop_Recommendation.pkl'

# Load crop recommendation model safely
withopen(crop_recommendation_model_path,'rb') as f:
    crop_recommendation_model = pickle.load(f)

# Load fertilizer dictionary
from fertilizer import fertilizer_dict
```

```
app = Flask(_name_)


# Ensure the upload folder exists

upload_folder = 'static/user_uploaded'

os.makedirs(upload_folder, exist_ok=True)
# Route for Fertilizer Prediction

@app.route('/fertilizer-predict',

methods=['POST']) def

fertilizer_recommend():

    crop_name =

    request.form['cropname'] N_filled

    = int(request.form['nitrogen'])

    P_filled =

    int(request.form['phosphorous'])

    K_filled =

    int(request.form['potassium'])


    # Read crop NPK data

    df =

    pd.read_csv('Data/Crop_NPK.csv')

    crop_data = df[df['Crop'] ==

    crop_name]


    if crop_data.empty:

        return "Invalid crop name", 400


    N_desired, P_desired, K_desired = crop_data.iloc[0][['N', 'P', 'K']]

    n, p, k = N_desired - N_filled, P_desired - P_filled, K_desired - K_filled
```

```
    key1 = "NHigh" if n < 0 else "Nlow" if n > 0 else
    "NNo" key2 = "PHigh" if p < 0 else "Plow" if p
    > 0 else "PNo" key3 = "KHigh" if k < 0 else
    "Klow" if k > 0 else "KNo"

return render_template('Fertilizer-
Result.html',
recommendation1=Markup(fertilizer_dict
[key1]),
recommendation2=Markup(fertilizer_dict
[key2]),
recommendation3=Markup(fertilizer_dict
[key3]), diff_n=abs(n), diff_p=abs(p),
diff_k=abs(k)) # Function to predict pests
using trained classifier def
pred_pest(pest):
    try:
        test_image = image.load_img(pest,
        target_size=(64, 64)) test_image =
        image.img_to_array(test_image) test_image
        = np.expand_dims(test_image, axis=0) result
        = classifier.predict(test_image)
        return np.argmax(result,
    axis=1) except Exception as e:
        print(f"Error in prediction:
        {e}") return 'x'
```

```
# Home route
@app.route('/')
@app.route('/in
dex.html') def
index():
    return render_template('index.html')

# Routes for different pages
@app.route('/CropRecommenda
tion.html') def crop():
    return render_template('CropRecommendation.html')

@app.route('/FertilizerRecommendati
on.html') def fertilizer():
    return render_template('FertilizerRecommendation.html')

@app.route('/PesticideRecommendation.html')
def pesticide():
    return render_template('PesticideRecommendation.html')

# Route for pest image upload and
prediction @app.route('/predict',
methods=['POST']) def predict():
    file =
    request.files.get('im
    age') if not file or
    file.filename == '':
        return "No file selected", 400
```

```
filename =
secure_filename(file.filename)
file_path = os.path.join(upload_folder,
filename) file.save(file_path)

pred =
pred_pest(pest=file_pa
th) if pred == 'x':
    return render_template('unaptfile.html')

pest_identified = {
    0: 'aphids', 1: 'armyworm', 2: 'beetle', 3: 'bollworm',
    4: 'earthworm', 5: 'grasshopper', 6: 'mites', 7: 'mosquito',
    8: 'sawfly', 9: 'stem borer'
}.get(pred[0], 'unknown')

return render_template(f'{pest_identified}.html', pred=pest_identified)

# Crop Prediction route
@app.route('/crop_prediction',
methods=['POST']) def
crop_prediction():
try:
    features = [
        int(request.form['nitrogen']),
        int(request.form['phosphorous']),
        int(request.form['potassium']),
        float(request.form['temperature']),
```

```
            float(request.form['humidity']),
            float(request.form['ph']), float(request.form['rainfall'])
        ]

        data = np.array([features])
        my_prediction =
        crop_recommendation_model.predict(data)
        final_prediction = my_prediction[0]

        return render_template('crop-result.html',
                               prediction=final_prediction,
                               pred=f'img/crop/{final_prediction}.jpg')
    except Exception as e:
        return f"Error occurred: {str(e)}", 500

if _name_ ==
    '_main_':
    app.run(deb=
    True)
```

CHAPTER-9

SYSTEM TESTING

SYSTEM TESTING

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.

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.
Function: 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.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

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.

TO RUN THE PROJECT FOLLOW THE BELOW COMMANDS:

- To run project double click on 'run.bat' file to get below screen
- In above screen click on 'Upload Crop Dataset' button to upload dataset
- In above screen selecting and uploading 'Dataset.csv' file and then click on 'Open' button to load dataset and to get below screen
- In above screen dataset loaded and we can see dataset contains some non-numeric values and ML will not take non-numeric values so we need to preprocess dataset to convert non-numeric values to numeric values by assigning ID to each non-numeric value. So click on 'Preprocess Dataset' button to process dataset.

In above screen all non-numeric values converted to numeric format and in below lines we can see dataset contains total 246091 records and application using (80%) 196872 records to train ML and using (20%) 49219 records to test ML prediction error rate (RMSE (root mean square error)). Now click on 'Train Machine Learning Algorithm' button to train Decision Tree Machine learning algorithm on above dataset and then calculate prediction error rate

In above screen ML is trained and we got prediction error rate as 0.067% and now Decision Tree model is ready and now click on 'Upload Test Data & Predict Yield' button to upload test data and then application will predict production

In above screen selecting and uploading 'test.csv' file and then click on 'Open' button to load test data and then application will give below prediction result

In above screen each test record is separated with newline and in above screen in square bracket we can see test data values and after square bracket we can see predicted production and after that we can see predicted YIELD per acre. So each test record and its prediction is separated with newline.

TEST CASES

SNO	Test case Title	Pre-requisites	Action	Expected Result	Test Result(pass/fail)
Test case 1	Software requirements	Python version 3.6.5	python --version (Checking the version)	Python 3.8.5present in your system	Pass
Test case 2	Packages need	pandas, numpy , sklearn	ls(list of packages)	all packages should import	Pass
Test case 3	Import the dataset	Import the dataset	Datset found	show the dataset in text format	Pass
Test case 4	Data cleaning	All data should be read from dataset	Apply pre processing, feature selection on the data	It will give clean data suitable for machine learning algorithms.	Pass
Test case 5	Prediction	Model should be trained with a training dataset	Use trained model on test data for prediction	It will display the predicted results	Pass

9.1 Test cases

CHAPTER-10
INPUT AND OUTPUT DESIGN

INPUT AND OUTPUT DESIGN

Inputs:

The system requires various inputs from the user or data sources in order to accurately predict and recom mend the most suitable crop. These inputs are categorized as environmental, seasonal, and location-based.

Outputs:

The outputs of the system are designed to assist the user (especially farmers) in making informed and actionable decisions based on the inputs provided. These outputs are generated using trained machine learning models that analyze environmental, seasonal, and productivity data.

INPUT DESIGN

Input design is one of the most critical stages in the development of an intelligent system. It determines how users interact with the system and how accurately data is collected. The goal is to ensure that the input data is valid, well-structured, and entered easily without confusion or complexity**.**

- To ensure accurate and efficient data entry.
- To minimize user input errors through validations.
- To enhance user experience with a simple, intuitive interface.
- To collect all required parameters essential for crop recommendation.

OUTPUT DESIGN

Output design focuses on how the results of the system's processing are presented to the user. Effective output design ensures the information is clear, actionable, and helpful for decision-making. For farmers, the output must be simple, easy to interpret, and in a format they can readily use.

- To provide meaningful and accurate crop recommendations.

- To present information in an organized and user-friendly way.

- To support decision-making with additional insights like expected yield or sowing time.
- To maintain consistency in format and clarity.

CHAPTER -11
RESULTS

RESULTS

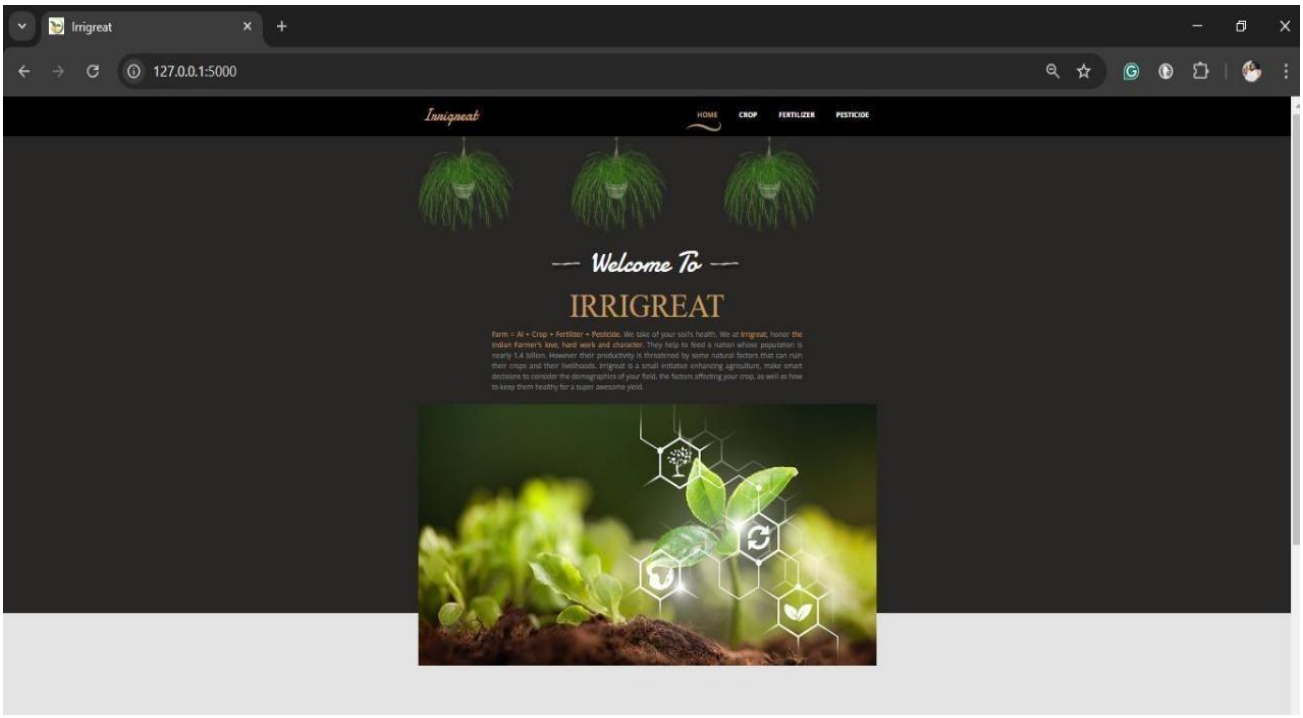


Fig: homepage

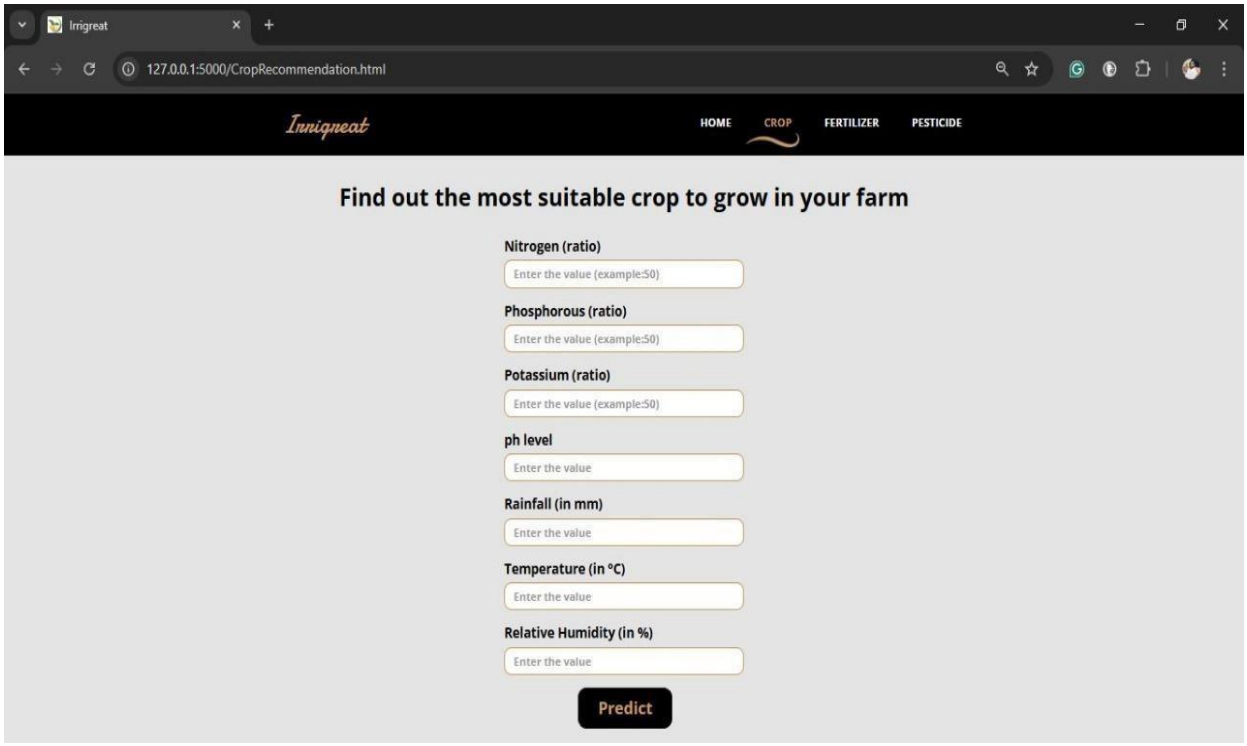


Fig: crop prediction

Fig: Predicted Output

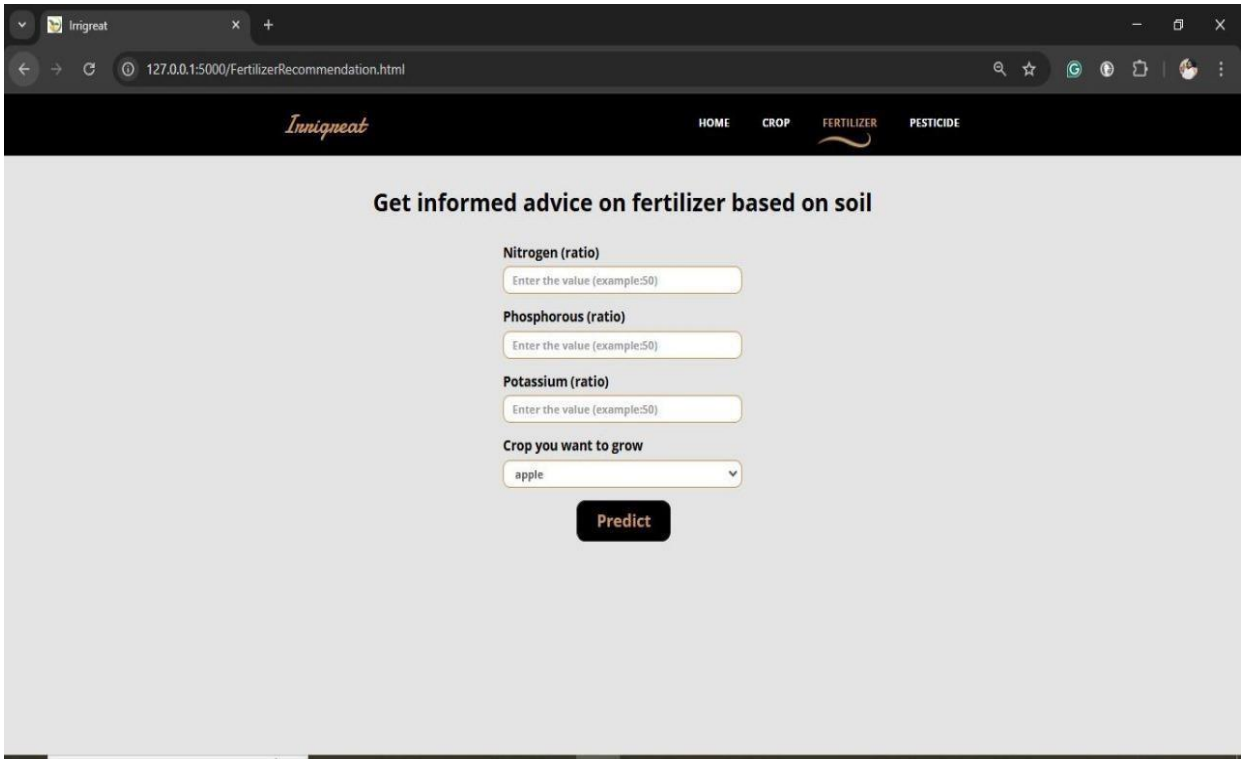
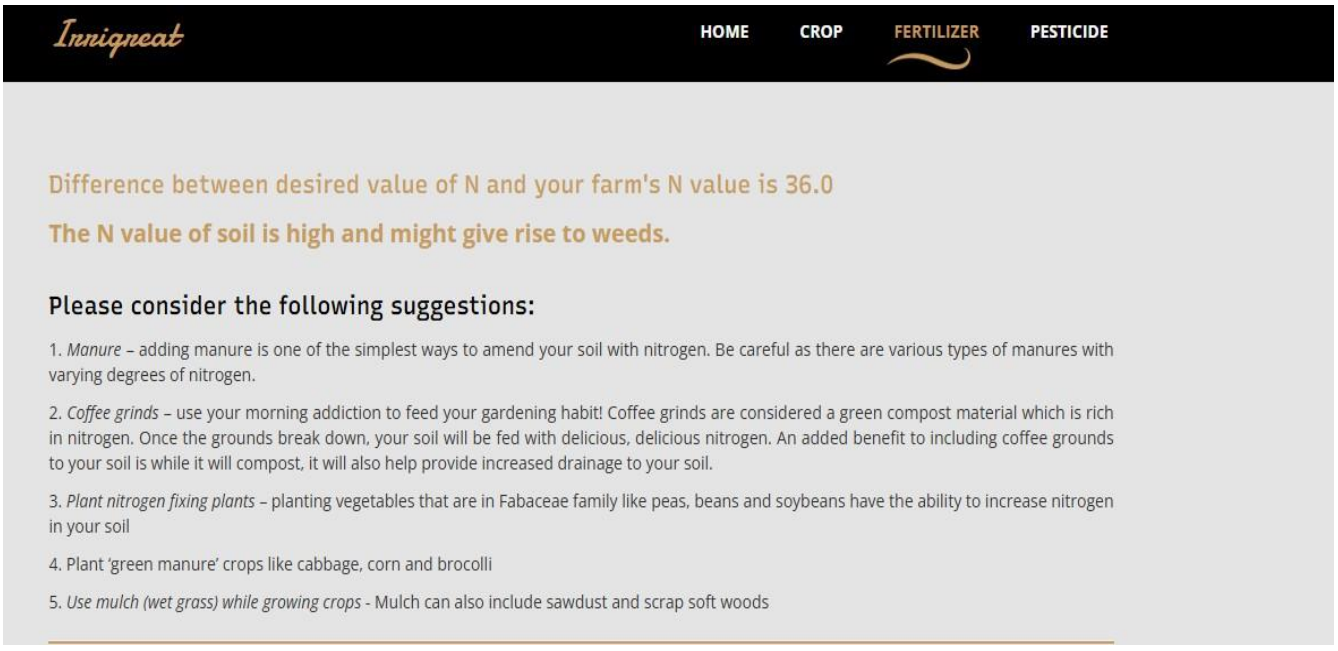


Fig: Fertilize Predicition



Difference between desired value of P and your farm's P value is 103.0

The P value of your soil is low.

Please consider the following suggestions:

1. *Bone meal* – a fast acting source that is made from ground animal bones which is rich in phosphorous.
2. *Rock phosphate* – a slower acting source where the soil needs to convert the rock phosphate into phosphorous that the plants can use.
3. *Phosphorus Fertilizers* – applying a fertilizer with a high phosphorous content in the NPK ratio (example: 10-20-10, 20 being phosphorous percentage).
4. *Organic compost* – adding quality organic compost to your soil will help increase phosphorous content.
5. *Manure* – as with compost, manure can be an excellent source of phosphorous for your plants.
6. *Clay soil* – introducing clay particles into your soil can help retain & fix phosphorus deficiencies.
7. *Ensure proper soil pH* – having a pH in the 6.0 to 7.0 range has been scientifically proven to have the optimal phosphorus uptake in plants.
8. If soil pH is low, add lime or potassium carbonate to the soil as fertilizers. Pure calcium carbonate is very effective in increasing the pH value of the soil.
9. If pH is high, addition of appreciable amount of organic matter will help acidify the soil. Application of acidifying fertilizers, such as ammonium sulfate, can help lower soil pH

Difference between desired value of K and your farm's K value is 190.0

The K value of your soil is low.

Please consider the following suggestions:

1. Mix in muricate of potash or sulphate of potash
2. Try kelp meal or seaweed
3. Try Sul-Po-Mag
4. Bury banana peels an inch below the soils surface
5. Use Potash fertilizers since they contain high values potassium

Fig: Fertilizer Prediction Output

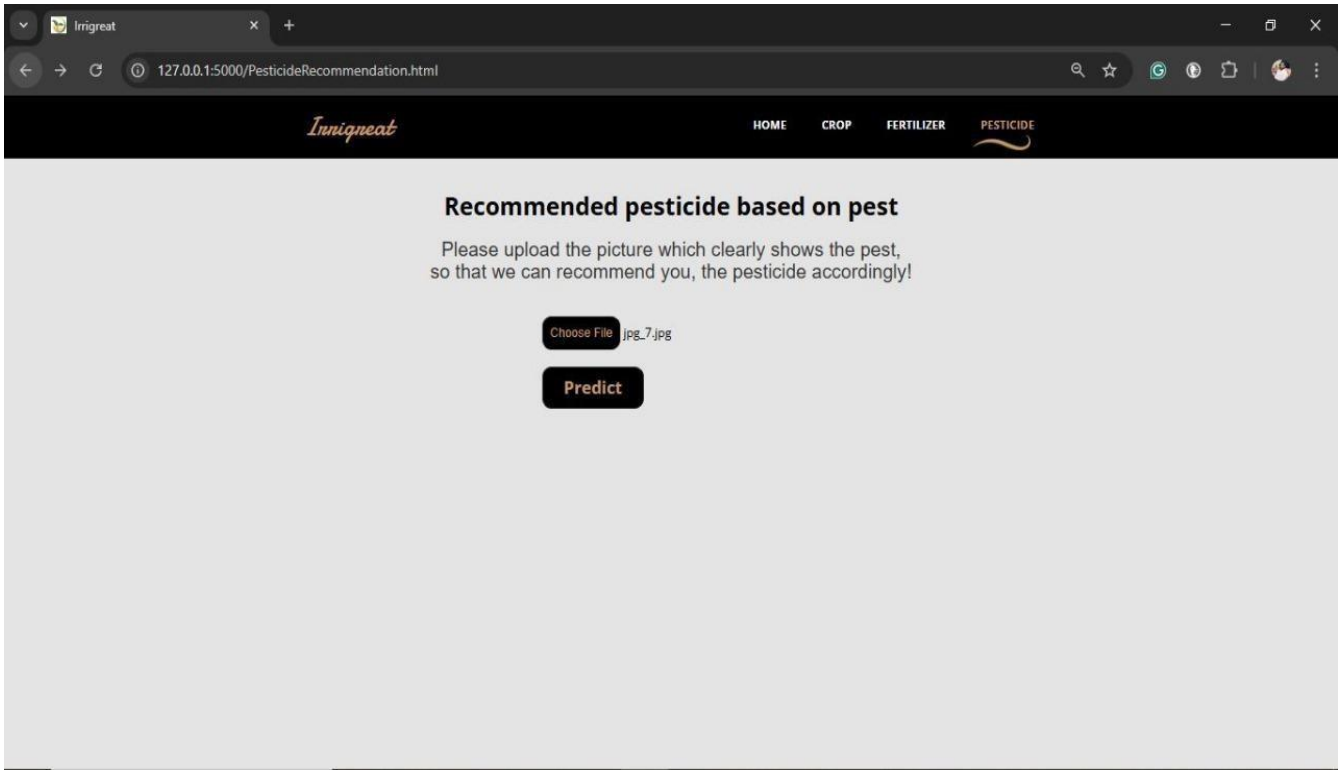


Fig: Pesticide Prediction Page

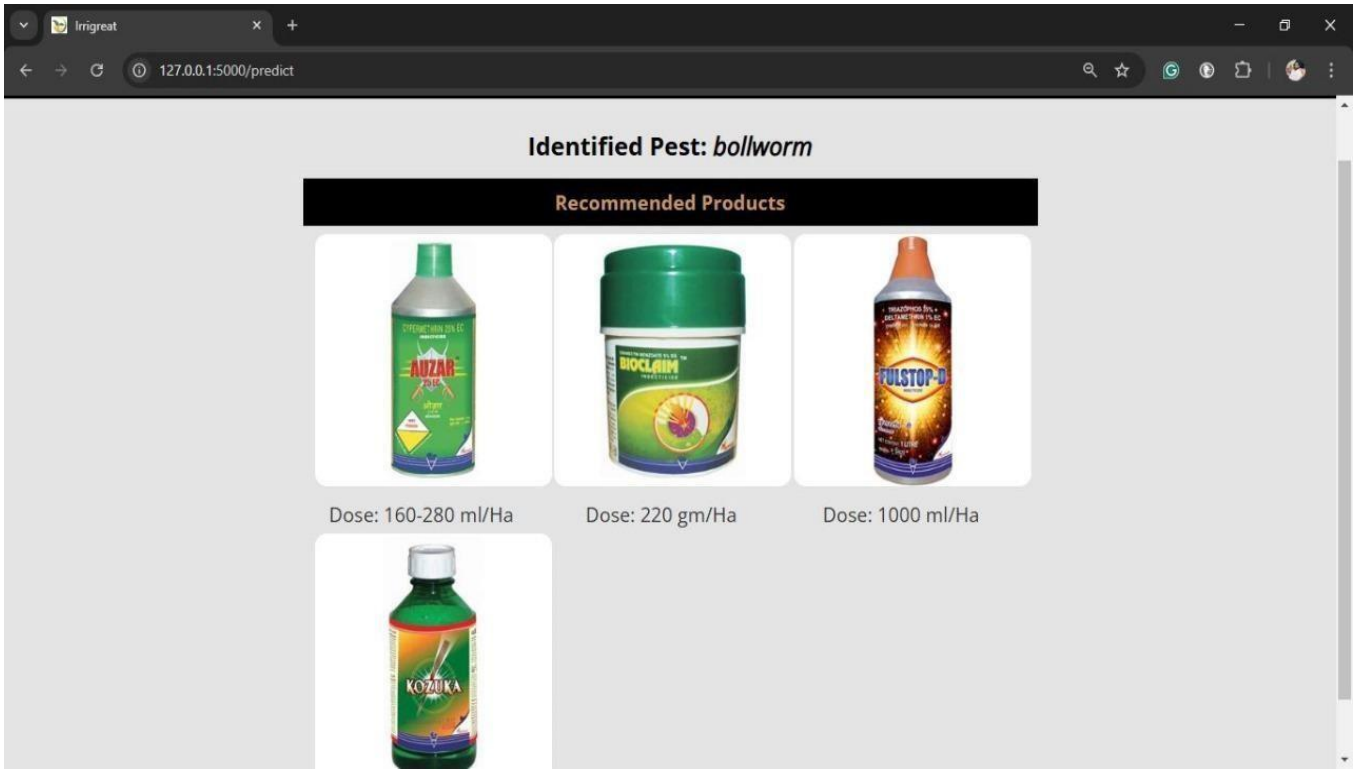


Fig: Pesticide Prediction Output

.

CHAPTER -12

CONCLUSION & FUTURE WORK

CONCLUSION & FUTURE WORK

CONCLUSION:

This paper presented a comprehensive study on the application of various machine learning algorithms for predicting crop yield based on several influential factors such as temperature, rainfall, season, and geographical area. The experiments were carried out using an authentic Indian government dataset, and the outcomes clearly demonstrated that the Random Forest Regressor algorithm outperforms other models in terms of yield prediction accuracy. Among the sequential models explored, the Simple Recurrent Neural Network (RNN) was found to be more effective in predicting rainfall patterns, while the Long Short-Term Memory (LSTM) model showed superior performance in predicting temperature trends. These findings indicate that different machine learning models may excel in specific domains of agricultural parameter prediction.

By integrating multiple parameters such as rainfall, temperature, season, and area, a holistic and more accurate yield prediction can be achieved for specific districts. The results revealed that Random Forest stands out as the most reliable and accurate classifier when all parameters are combined, indicating its robustness in handling complex and multidimensional data. This study not only aids farmers by providing insights into selecting the most suitable crops for upcoming seasons based on data-driven predictions, but also contributes to bridging the gap between advanced technologies and the agricultural sector.

By empowering farmers with predictive tools, this research paves the way for smarter farming practices, improved resource utilization, and ultimately, enhanced agricultural productivity.

FUTURE WORK:

The proposed system holds great potential for future development and expansion. With continuous advancements in technology and data availability, the system can evolve in the following ways

1. Integration with IoT Devices:

Real-time data from smart sensors (for soil moisture, temperature, humidity, etc.) can further enhance prediction accuracy. Automated data collection from the field will reduce manual effort and increase efficiency.

2. Mobile Application Development:

A dedicated mobile app can make the system more accessible to farmers in remote areas. Offline mode and local language support can improve usability and adoption.

3. Weather-Based Smart Alerts:

Real-time weather analysis and alert generation for droughts, floods, or pests will help farmers take preventive measures in advance.

5. Market Analysis & Price Forecasting:

By analyzing market trends, supply-demand patterns, and historical price data, the system can provide future price predictions to help farmers decide the best time to sell crops.

6. Automated Crop Monitoring Using Drones:

In future versions, drone integration can be added to monitor large-scale farms, detect issues like diseases or nutrient deficiency, and provide aerial insights.

7. Government & NGO Collaboration:

The system can be integrated with government agricultural schemes and subsidy programs to provide relevant information and benefits to farmers.

8. Scalability for Other Regions and Crops:

The system can be expanded to support multiple geographic regions, diverse crop types, and different languages, making it globally applicable.

CHAPTER-13
REFERENCES

REFERENCES

[1]. Priya, P., Muthaiah, U., Balamurugan, M. International Journal of Engineering Sciences Research Technology Predicting Yield of the Crop Using Machine Learning Algorithm.

[2]. Mishra, S., Mishra, D., Santra, G. H. (2016). Applications of machine learning techniques in agricultural crop production: a review paper.Indian J. Sci. Technol,9(38), 1-14.

[3]. Manjula, E., Djodiltachoumy, S. (2017). A Model for Prediction of Crop Yield.International Journal of Computational Intelligence and Informatics,6(4), 2349-6363.

[4]. Dahikar, S. S., Rode, S. V. (2014). Agricultural crop yield prediction using artificial neural network approach.International journal of innovative research in electrical, electronics, instrumentation and control engineering,2(1), 683-686.

[5]. Gonzlez Snchez, A., Frausto Sols, J., Ojeda Bustamante, W. (2014). Predictive ability of machine learning methods for massive crop yield prediction.

[6]. Mandic, D. P., Chambers, J. (2001). Recurrent neural networks for prediction: learning algorithms, architectures and stability. John Wiley Sons, Inc..

[7]. Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[8]. Sak, H., Senior, A., Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In Fifteenth annual conference of the international speech communication association.

[9]. Liaw, A., Wiener, M. (2002). Classification and regression by randomForest. R news, 2(3), 18-22.