

CROP PREDICTION SYSTEM BY USING MACHINE LEARNING

Submitted in partial of the required for the award of

Bachelor of Computer Science

By

MITHRAN S

(REGISTER NO 40290056)



DEPARTMENT OF COMPUTER SCIENCE

SCHOOL OF SCIENCE AND HUMANITIES

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAR NAGAR, RAJIV GANDHI SALAI, CHENNAI

600 199

MAY 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **MITHRAN S (40290056)** who carried out the project entitled "**CROP PREDICTION SYSTEM BY USING MACHINE LEARNING**" under our supervision from JANUARY 2023 TO MAY 2023.

Internal Guide

Dr. REKHA CHAKRAVARTHI M.E., Ph.D.

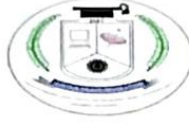
DEAN AND HEAD OF THE DEPARTMENT

Dr. REKHA CHAKRAVARTHI M.E., Ph.D.

Submitted for Viva voce examination held on 08/05/2023


Internal Examiner


External Examiner



DECLARATION

I am, **MITHRAN S (40290056)** here by declare that the Project Report entitled **"CROP PREDICTION SYSTEM BY USING MACHINE LEARNING"** done by us under the guidance of **Dr. REKHA CHAKRAVARTHI., M.E., Ph.D.** Dean of Arts and Science Department of Computer Science at **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY**, Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai-600 119 is submitted in partial fulfillment of the requirements for the award of Bachelor of Science degree in Computer Science.

MITHRAN S

DATE: 08-05-2023

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ABSTRACT

The impact of climate change in India, most of the agricultural crops are being badly affected in terms of their performance over a period of the last two decades. Predicting the crop yield in advance of its harvest would help the policy makers and farmers take appropriate measures for marketing and storage. This project will help the farmers to know the yield of their crop before cultivating onto the agricultural field and thus help them to make the appropriate decisions. It attempts to solve the issue by building a prototype of an interactive prediction system. Implementation of such a system with an easy-to-use web based graphic user interface and the machine learning algorithm will be carried out. The results of the prediction will be made available to the farmer. Thus, for such kind of data analytics in crop prediction, there are different techniques or algorithms, and with the help of those algorithms we can predict crop yield. Decision Tree classification algorithm is used. By analyzing all these issues and problems like weather, temperature, humidity, rainfall, moisture, there is no proper solution and technologies to overcome the situation faced by us. In India, there are many ways to increase the economic growth in the field of agriculture. Data mining is also useful for predicting crop yield production. Generally, data mining is the process of analyzing data from various viewpoints and summarizing it into important information. Decision Tree is the most popular and powerful supervised machine learning algorithm capable of performing both classification and regression tasks, that operate by constructing a multitude of decision trees during training time and generating output of the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

TABLE CONTENT

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	5
1	INTRODUCTION	8-13
	1.1 DOMAIN INTRODUCTION	8
	1.2 PROJECT INTRODUCTION	10
	1.3 OVERVIEW	10
	1.4 DESCRIPTION	10
	1.5 REVIEW ON LITERATURE WORK	11
2	SYSTEM ANALYSIS	14-16
	2.1 FEASIBILITY STUDY	14
	2.2 EXISTING SYSTEM	15
	2.3 PROPOSED SYSTEM	15
3	SYSTEM REQUIREMENTS	17-22
	3.1 REQUIREMENT ANALYSIS	17
	3.2 HARDWARE ENVIRONMENT	17
	3.3 SOFTWARE REQUIREMENT	17
	3.4 PYTHON	18
	3.5 PYCHARM COMMUNITY EDITION	19
	3.6 PANDAS	20
	3.7 TKINTER	20
	3.8 SCIKIT-LEARN	21
	3.9 NUMPY	21

4	SYSTEM DESIGN	23-25
	4.2 ARCHITECTURE DIAGRAM	24
	4.3 USECASE DIAGRAM	24
	4.4 ENTITY RELATIONSHIP DIAGRAM	25
	IMPLEMENTATION	26-30
5	5.1 SYSTEM ALGORITHM	26
	5.2 MODULES DESCRIPTION	28
	5.3 DATASET COLLECTION	29
	5.4 SOFTWARE IMPLEMENTATION	29
6	CONCLUSION AND FURTHER WORK	31
	6.1 CONCLUSION	31
	6.2 FURTHER WORK	31
7	APPENDICES	32-37
	7.1 SOURCE CODE	32
	7.2 OUTPUT SCREENSHOT	36
8	RESULT AND DISCUSSION	38
	8.1 RESULT	38
	8.2 USES OF THE EXTRACTED INFORMATION	38
9	REFERNCE	39-40

CHAPTER - 1

INTRODUCTION

Agriculture is the practice of cultivating plants and livestock to provide facilities the human beings. In the rise of the sedentary human lifestyle agriculture was the key development. The cultivation of plants and food grains began years ago to provide food to the city population. Agriculture is the main need for people to live in society. Agriculture is the main source of livelihood; it provides a source for the people to earn. Most of the population in the rural areas is dependent on agriculture as their main source of income. Agriculture contributes significantly to a country's GDP, that is the Gross Domestic Production of a country. With the passing of time, there are several revolutions that take place to improve agriculture throughout the world or in a country. If we talk about agriculture, India has witnessed several revolutions, that is, the green revolution, yellow revolution, blue revolution, agriculture. Agriculture affects the biodiversity of a country depending upon agricultural activities. The major agricultural products can be widely grouped into categories of food grains, fibers and raw materials. Foodgrains included the grains or cereals that have been used for eating. Fiber crops are completely Commercial, they cannot be eaten and are completely grown for making money. Raw materials are a category of crops that are completely grown for use as raw materials in industries to prepare other items.

1.1 DOMAIN INTRODUCTION

Machine learning (ML) is a branch of artificial intelligence (AI) that enables computers to “self-learn” from training data and improve over time, without being explicitly programmed. Machine learning algorithms can detect patterns in data and learn from them, to make their own predictions. In short, machine learning algorithms and models are learned through experience.

Types of Machine Learning:

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Supervised Learning:

Supervised learning algorithms and supervised learning models make predictions based on labeled training data. Each training sample includes an input and a desired output. A supervised learning algorithm analyzes this sample data and makes an inference – basically, an educated guess when determining the labels for unseen data.

Unsupervised Learning:

Unsupervised learning algorithms uncover insights and relationships in unlabeled data. In this case, models are fed input data, but the desired outcomes are unknown, so they must make inferences based on circumstantial evidence, without any guidance or training. The models are not trained with the “right answer,” so they must find patterns on their own.

Reinforcement Learning:

Reinforcement learning (RL) is concerned with how a software agent (or computer program) ought to act in a situation to maximize the reward. In short, reinforced machine learning models attempt to determine the best possible path they should take in each situation. They do this through trial and error. Since there is no training data, machines learn from their own mistakes and choose the actions that lead to the best solution or maximum reward.

1.2 PROJECT INTRODUCTION

We have implemented a simple crop prediction tool that uses a decision tree classifier to predict the crop based on input values for temperature, rainfall, soil type, and soil depth. The program loads a dataset of crops and their corresponding features from a CSV file, filters the data based on the input values, trains a decision tree model on the filtered data, and outputs the predicted crop in a GUI window.

1.3 OVERVIEW

The program uses the panda's library to load and manipulate the dataset, the tkinter library to create the GUI window, and the scikit-learn library to create and train the decision tree classifier. The GUI window contains input fields for temperature, rainfall, soil type, and soil depth, and a button to trigger the prediction. The predicted crop is displayed on a label on the window.

1.4 DESCRIPTION

The program first imports the required libraries, including pandas, tkinter, scikit-learn, and the necessary modules from each library. The dataset is loaded from a CSV file using the pandas `read_csv()` function and stored in a variable named `crop_data`.

The program defines a function named `predict_crop()` that gets called when the predict button on the GUI window is clicked. This function retrieves the input values for temperature, rainfall, soil type, and soil depth from the GUI input fields, filters the crop data based on the soil type and depth, and splits the data into features and target variable columns.

If there are not enough crops to train the model, or if there are no crops for the given soil type and depth, the function displays an appropriate message in the crop label on the window. If there are enough crops to train the model, the function splits the data into training and testing sets, creates a decision tree classifier using scikit-learn's `DecisionTreeClassifier()` class, trains the model on the training data, predicts the crop based on the input values, and displays the predicted crop in the crop label on the window.

The GUI window is created using tkinter and contains input fields for temperature, rainfall, soil type, and soil depth, a button to trigger the prediction, and a label to display the predicted

crop. The window is set to a size of 400x300 pixels using the geometry() method, and the mainloop() method is called to start the window and keep it running until the user closes it.

1.5 REVIEW ON LITERATURE WORK

Paper 1: Crop Yield Prediction and Efficient use of Fertilizers

Author Name: Bhanumathi. S, Vineeth. M and Rohit. N

Year of Publish: 2019

Abstract:

India being an agriculture country, its economy predominantly depends on agriculture yield growth and agroindustry products. Data Mining is an emerging research field in crop yield analysis. Yield prediction is a very important issue in agriculture. Any farmer is interested in knowing how much yield he is about to expect. Analyze the various related attributes like location, pH value from which alkalinity of the soil is determined. Along with its type of soil, nutrient value of the soil in that region, amount of rainfall in the region, soil composition can be determined. All these attributes of data will be analyzed, train the data with various suitable machine learning algorithms for creating a model. The system comes with a model to be precise and accurate in predicting crop yield and deliver the end user with proper recommendations about required fertilizer ratio based on atmospheric and soil parameters of the land which enhance to increase the crop yield and increase farmer revenue.

Paper 2: Crop Yield Forecasting Methods

Author Name: 1R.K. Singh, 2T.R. Singh and 2U. Kaushal

Year of Publish: 2019

Abstract:

Increasing the accuracy of agricultural forecasting on the important application of earth observation. The study on review to aware about the ability to reliably forecast crop Predicting yield and quality is valuable for economic planning and commodities forecasting as well as ensuring global food security. Study regarding the overview of the current crop yield forecasting methods, which include ways to use crop yield forecasting method to improve

agriculture to yield forecasting have been discussed in this paper.

Paper 3: Crop yield predictions - high resolution statistical model for intra-season forecasts applied to corn in the US

Author Name: Yiqing Cai, Kristen Moore, Adam Pellegrini, Aymn, Elhaddad

Year of Publish: 2017

Abstract:

Accurately forecasting crop yields has broad implications for economic trading, food production monitoring, and global food security. However, the variety of environmental variables across time and space presents challenges and has limited the precision of yield models. Here, we develop a machine-learning based model forecasting end-offseason, or final, corn yields, using data from the United States from 2001- 2016. We combine several algorithms in a multi-level model, and use understanding of physiological processes in temporal feature selection, to achieve high precision in our intra-season forecasts, including in very anomalous seasons.

Paper 4: Recommendation System for Crop Identification and Pest Control Technique in Agriculture

Author Name: Avinash Kumar, Subhangi Sarkar and Chittaranjan Pradhan

Year of Publish: 2019

Abstract:

In Agriculture, mostly crops are lost due to the erroneous selection of the crop to be grown in a particular land holding. The farmers are generally not aware of the requirements of the crop, i.e., the minerals, soil moisture and other soil requirements. This can cause distress to farmers both mentally and financially. One more problem that a farmer generally encounters is the pests and diseases that can affect the crops they grow, which they are generally unaware of in an early stage. This problem of farming is addressed in our paper, and we have

tried to solve it with the help of a Recommendation System. With the help of our model, we predict the most suitable crop for the farmer and detect the pest that may affect it as well as suggest pest control techniques. In this paper, we have applied SVM classification algorithm, Decision Tree algorithm and Logistic Regression algorithm and we have found that the SVM classification model gives better accuracy as compared to other algorithms.

Paper 5: Soil Classification using Machine Learning Methods and Crop

Suggestion Based on Soil Series

Author Name: Sk. Al. Zaminur Rahman, Kaushik Chandra Mitra, S.M. Mohidul Islam

Year of Publish: 2018

Abstract:

Soil is an important ingredient in agriculture. There are several kinds of soil. Each type of soil can have different kinds of features and different kinds of crops grow on different types of soils. We need to know the features and characteristics of various soil types to understand which crops grow better in certain soil types. Machine learning techniques can be helpful in this case. In recent years, it has progressed a lot. Machine learning is still an emerging and challenging research field in agricultural data analysis. In this paper, we have proposed a model that can predict soil series with land type and according to prediction it can suggest suitable crops. Several machine learning algorithms such as weighted k-Nearest Neighbor (k-NN), Bagged Trees, and Gaussian Kernel-based Support Vector Machines (SVM) are used for soil classification. Experimental results show that the proposed SVM based method performs better than many existing methods.

CHAPTER – 2

SYSTEM ANALYSIS

2.1 FEASIBILITY STUDY

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 funds that the company can pour into the research and development of the system is limited. The expenditure must be justified. Thus, the developed system is 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 for the available technical resources. This will lead to high demand for 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.

2.2 EXISTING SYSTEM

Nowadays, dilettante farmers find it hard to understand the cultivation process, crop type, climate change, etc. Farming is the spine for every nation's economy. Future agriculture depends on dilettante farmers. But the new farmers have low level knowledge in this field. So, Machine learning help to solve this type of problems. In Existing system, they provide soil type and crop using Random Forest algorithm. But everyone can be able to find a soil type easily. So, we need to predict the crop type and predict the crop price based on Machine learning technology.

Crop Problem Statement:

- Crop yield prediction is an important agricultural problem. The Agricultural yield primarily depends on weather conditions (rain, temperature, etc.), pesticides.
- Accurate information about the history of crop yield is important for making decisions related to agricultural risk management and future predictions.
- Now, some new formers or students are willing to work in agriculture, which means they don't know about that field.
- So, we collected the past data and used the machine learning algorithm to predict the crop and soil parameters based on our previous model.

2.3 PROPOSED SYSTEM

The system prepared predict major crops yield in a particular district in Tamil Nadu And any other states. After launching the program, the system takes inputs for temperature, rainfall, soil type, and soil depth, and predicts the crop that is best suited for those conditions. The system uses a decision tree classifier to predict the crop based on the input values. The dataset used for training the model contains information about various crops, their preferred temperature, rainfall, soil type, and soil depth. The system filters the dataset based on the soil type and depth input values and trains the model on the remaining data. The system also checks if there are enough crops to train the model

and if there are any crops to predict based on the input values. If there are not enough crops, the system informs the user that there is insufficient data to make a prediction. After submitting the inputs, it's redirected into the GUI Window. The GUI window is an intermediate between user input and trained data set. The input goes to the trained data, where it processes decision tree classifier algorithm to predict crop. After the prediction, the predicted value passes to the GUI. That GUI gives the predicted value to the user on the same GUI window. The system then displays the predicted crop in the output label.

CHAPTER – 3

SYSTEM REQUIREMENTS

3.1 REQUIREMENT ANALYSIS

Requirement analysis determines the requirements of a new system. This project analyses product and resource requirements, which are required for this successful system. The product requirement includes input and output requirements it gives the wants in term of input to produce the required output. The resource requirements given in brief are about the software and hardware that are needed to achieve the required functionality.

3.2 HARDWARE ENVIRONMENT

- Hard Disk : 500 GB and Above
- RAM : 4 GB and Above
- Processor : 13 and Above

3.3 SOFTWARE REQUIREMENTS

- Operating System : Windows 7,8,10 (64 bit)
- Software : Python and PY Charm
- Tools : Pandas, Tkinter, Scikit-learn

3.4 PYTHON

Python is a high-level, interpreted programming language that was first introduced in 1991. The language's versatility enables it to cater to various applications, including web development, scientific computing, data analysis, artificial intelligence, and machine learning. One of the unique aspects of Python is its syntax, which is designed to be easily readable and writable, making it an excellent choice for developers of all skill levels. Python supports several programming paradigms, including object-oriented, functional, and procedural programming, providing developers with multiple options to tackle various programming tasks. The language has a vast and active community of developers who contribute to its development, resulting in a plethora of libraries, frameworks, and tools that streamline the creation of complex programs with minimal effort. Python's standard library contains a vast range of modules that simplify various tasks, such as file management, networking, and data processing. As an open-source language, Python is free to use, distribute, and modify. It runs seamlessly on multiple platforms, including Windows, macOS, and Linux, which makes it an excellent choice for cross-platform development. Furthermore, Python has several implementations, including CPython, Jython, Iron Python, and PyPy, each with unique features and advantages. Overall, Python is a powerful and easy-to-learn programming language with numerous applications. Its ability to cater to various programming paradigms, vast community support, and extensive standard library make it a top choice for developers, data scientists, and researchers worldwide.

Easy to learn: Python has a simple and intuitive syntax that is easy to learn and understand, making it an ideal language for beginners.

High-level language: Python is a high-level language, which means that it abstracts away low-level details such as memory management, allowing developers to focus on the problem at hand.

Large standard library: Python has a large and comprehensive standard library that provides built-in modules and functions for a wide range of tasks, such as data processing, networking, and file I/O.

Cross-platform compatibility: Python is a cross-platform language, which means that it can run on different operating systems such as Windows, Linux, and macOS.

Open-source: Python is an open-source language, which means that it is free to use, distribute, and modify.

Extensible: Python is a highly extensible language that supports the integration of other programming languages such as C and C++, making it suitable for writing high-performance applications.

Data analysis and visualization: Python has several powerful libraries such as NumPy, Pandas, and Matplotlib that make it an ideal language for data analysis and visualization.

Community: Python has a large and active community of developers and users who provide support, documentation, and code examples.

Machine learning and AI: Python has several powerful libraries such as TensorFlow, Keras, and Scikit-Learn that make it an ideal language for developing machine learning and AI applications.

Web development: Python has several powerful web frameworks such as Django and Flask that make it an ideal language for web development.

3.5 PYCHARM COMMUNITY EDITION

PyCharm Community Edition is an integrated development environment (IDE) for Python programming language is both free and open source. JetBrains, the developer of PyCharm, designed it to provide Python developers with a range of features that make it easier to create Python applications. The IDE has a user-friendly interface that simplifies coding and debugging processes. PyCharm Community Edition supports various Python frameworks and technologies such as Django, Flask, and Pyramid, among others. The IDE provides several productivity features like code completion, code inspections, and quick fixes that help developers write high-quality code. PyCharm Community Edition comes with a built-in debugger that makes it easier for developers to debug their code. The debugger provides features like breakpoints, watches, and step-by-step code execution. Additionally, PyCharm integrates with popular version control systems like Git, Mercurial, and Subversion, making it easy to manage project source code. Other notable features of PyCharm Community Edition include support for multiple languages, integration with databases, support for unit testing frameworks, and a powerful refactoring

system. Overall, PyCharm Community Edition is an excellent IDE for Python developers who want to work with a feature-rich and intuitive tool without any financial cost. The IDE has many features that make it easier to write, test, and debug Python code. Its integration with popular tools and frameworks makes it a top choice for many developers.

3.6 PANDAS

Pandas is a popular open-source data manipulation and analysis library for Python. It provides easy-to-use data structures for data manipulation, cleaning, exploration, and analysis. Pandas is built on top of the NumPy package and is often used in conjunction with other libraries like Matplotlib for data visualization and Scikit-learn for machine learning. Pandas are used in various industries like finance, healthcare, social sciences, and many more. Some of the key features of Pandas are:

- Efficient handling of structured data in various formats like CSV, Excel, SQL databases, and more
- Data cleaning and transformation capabilities
- Powerful indexing and selection methods for data filtering and querying
- Integration with other libraries like Matplotlib for data visualization and Scikit-learn for machine learning

Pandas is a versatile and powerful library that can be used for various data-related tasks in Python.

3.7 TKINTER

Tkinter is a standard Python library for creating graphical user interfaces (GUIs). It provides a set of tools to create windows, widgets, and menus to build desktop applications. Tkinter is included with most Python installations, making it easy to use and widely available. Tkinter provides a variety of widgets that can be used to create GUIs, such as buttons, labels, text boxes, and more. These widgets can be customized with various options to change their appearance and behavior. Tkinter also includes a layout manager, which is used to position and resize widgets within a window. There are several layout managers available in Tkinter, such

as grid, pack, and place. Tkinter is a powerful and flexible library for creating GUIs in Python, and it is a good option for those who want to build desktop applications with Python. Tkinter is a Python built-in library for creating graphical user interfaces (GUIs) and is a part of the standard Python library. It provides a set of tools for creating windows, frames, buttons, labels, etc. for creating interactive desktop applications.

3.8 SCIKIT-LEARN

Scikit-learn (or sklearn) is a free and open-source machine learning library for Python programming language. It provides a range of supervised and unsupervised learning algorithms in Python. Some of the commonly used algorithms in scikit-learn are classification, regression, clustering, and dimensionality reduction.

Sub-modules of scikit-learn include:

- **sklearn.datasets:** Provides functionality to load and access standard datasets.
- **sklearn.linear_model:** Includes algorithms for linear regression and logistic regression.
- **sklearn.model_selection:** Provides tools for model selection and evaluation such as cross-validation.
- **sklearn.neighbors:** Implements nearest neighbors-based learning methods.
- **sklearn.tree:** Implements decision tree-based learning methods.
- **sklearn.cluster:** Implements clustering algorithms such as k-means and hierarchical clustering.
- **sklearn.metrics:** Implements various metrics to evaluate model performance such as accuracy, precision, recall, and F1-score.

scikit-learn is a powerful and popular library for machine learning in Python, providing a wide range of tools for various tasks related to supervised and unsupervised learning.

3.9 NUMPY

NumPy is a Python library that provides support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these

arrays. It is one of the fundamental packages used for scientific computing in Python. Some of the key features of NumPy include:

- **Powerful N-dimensional arrays:** NumPy provides an array object that is faster and more efficient than traditional Python lists, especially for large arrays.
- **Broadcasting functions:** NumPy provides broadcasting functions that allow mathematical operations to be performed on arrays of different sizes and shapes.
- **Linear algebra and random number generation:** NumPy provides a comprehensive set of functions for linear algebra operations, including matrix multiplication, inversion, and eigenvalue problems. It also provides tools for working with random number generation.
- **Integration with other libraries:** NumPy is often used in conjunction with other libraries in the scientific Python ecosystem, such as SciPy, Pandas, and Matplotlib.

Some of the key sub-modules of NumPy include:

- **np.array:** This is the primary NumPy array object that supports multi-dimensional arrays and many array operations.
- **np.random:** This sub-module provides tools for working with random number generation.
- **np.linalg:** This sub-module provides tools for linear algebra operations, such as matrix multiplication and eigenvalue problems.
- **np.fft:** This sub-module provides tools for working with Fast Fourier Transforms.

CHAPTER – 4

SYSTEM DESIGN

Input data: The system takes input data from a CSV file that contains information about a different set of data such as temperature, rainfall, soil depth and so on.

Preprocessing: The input data is preprocessed using Pandas to remove any missing or invalid data, as well as any unnecessary columns that do not contribute to the classification task.

Feature engineering: The preprocessed data is then transformed into a set of features that can be used by the Decision Tree Classifier algorithm. This is done using NumPy to perform mathematical operations and transformations on the data.

Model training: The Decision Tree Classifier algorithm is trained on the preprocessed data to create a model that can classify new customers based on their features.

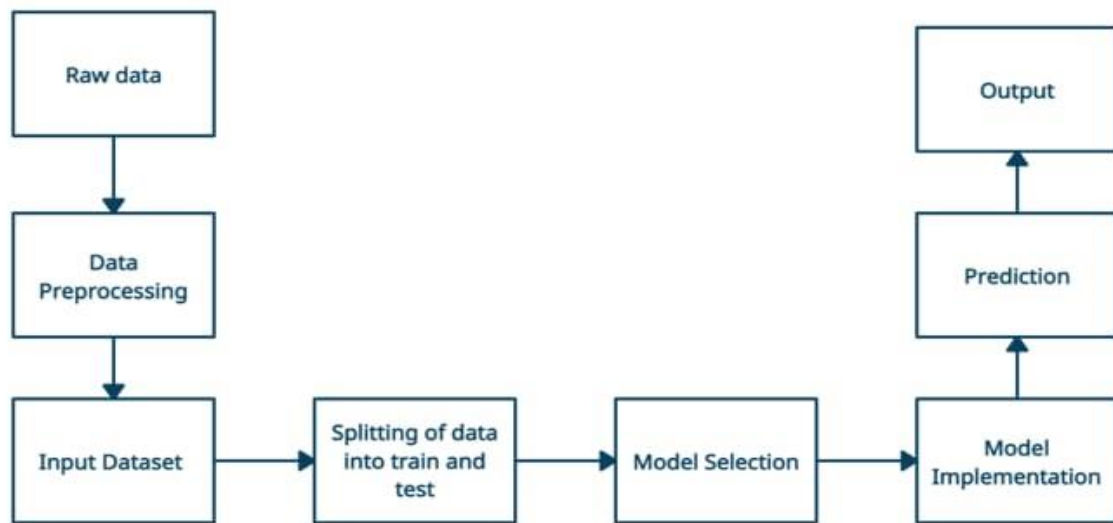
Model evaluation: The performance of the model is evaluated using a test dataset to determine its accuracy and ability to generalize to new data.

Model deployment: Once the model has been evaluated and deemed to be accurate and effective, it can be deployed in a production environment to classify new customers based on their features.

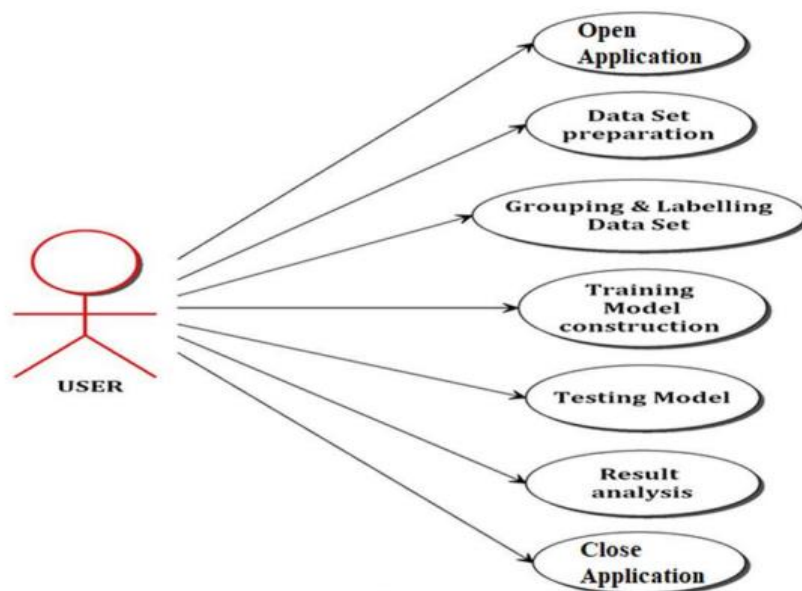
User interface: A user interface (UI) refers to the visual and interactive elements of a software application, website, or any digital device that enables users to interact with the system. It is the point of interaction between the user and the software system, allowing the user to communicate with the system and access its functionality. This system provides a user interface built with Tkinter that allows users to input data for a new customer and receive a classification result based on the trained model.

The system design involves reading input data, preprocessing it, extracting relevant features, training a model, evaluating its performance, deploying it, and providing a user interface for interaction.

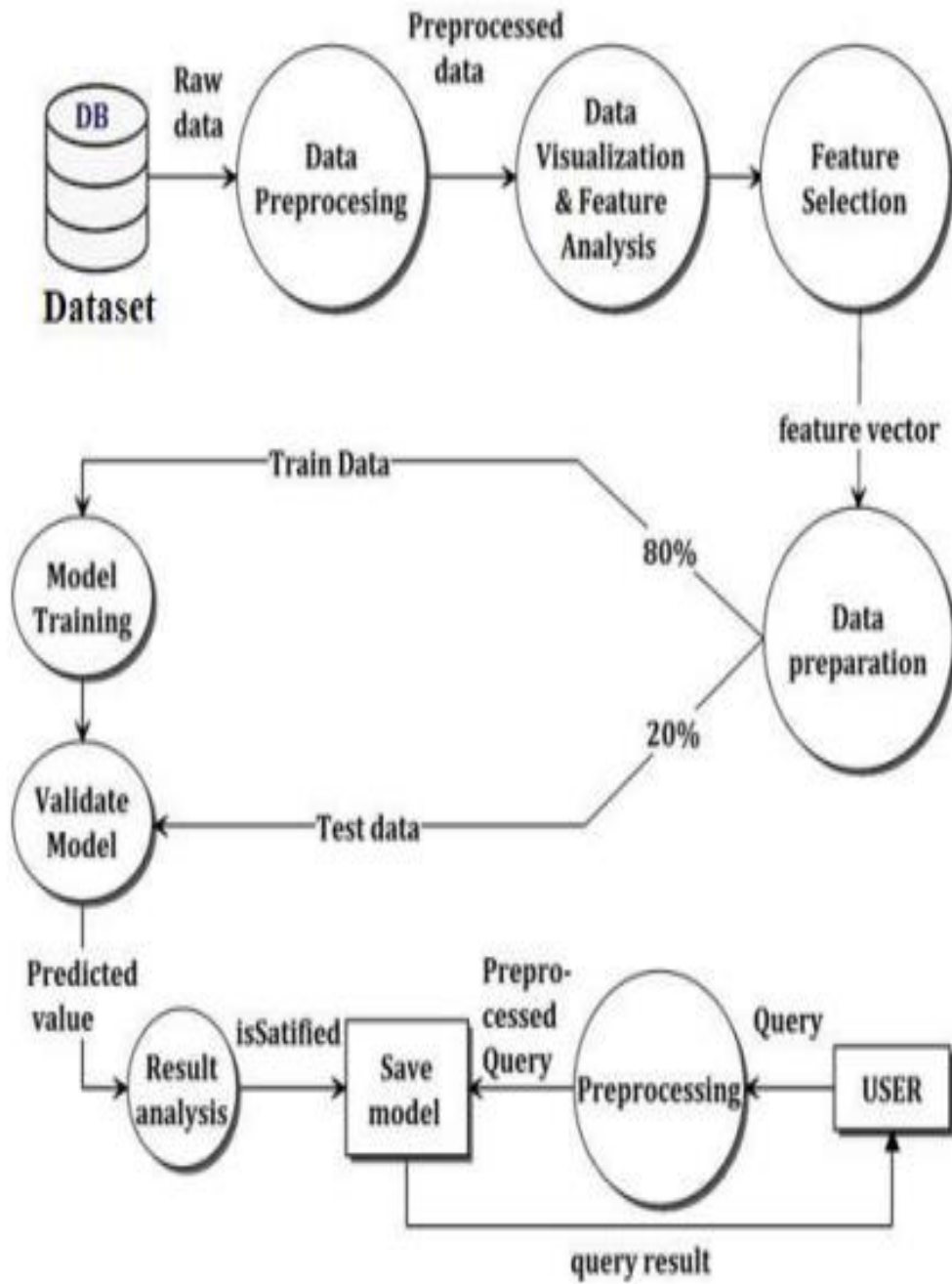
4.1 ARCHITECTURE DIAGRAM



4.2 USECASE DIAGRAM



4.3 ENTITY RELATIONSHIP DIAGRAM



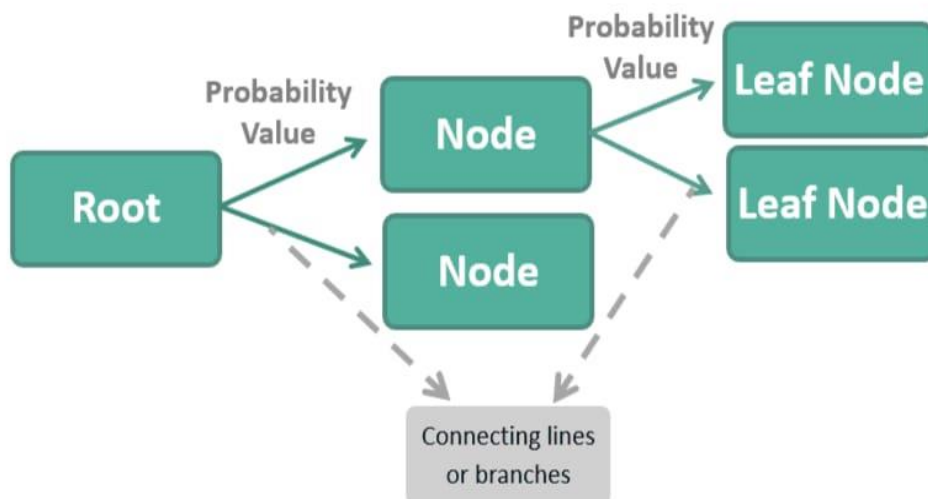
CHAPTER – 5

IMPLEMENTATION

5.1 SYSTEM ALGORITHM

DECISION TREE CLASSIFIER

Decision Tree Classifier is a machine learning algorithm that belongs to the family of supervised learning algorithms. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. It is a predictive modeling tool that can be used for both regression and classification tasks. The Decision Tree Classifier algorithm follows a top-down approach known as the recursive partitioning process. It begins by selecting the best feature as the root node that splits the dataset into two or more subsets. Then, it continues to split each subset into smaller subsets using the same process, recursively, until a stopping criterion is reached. Some of the advantages of using the Decision Tree Classifier algorithm are that it can handle both categorical and numerical data, it is easy to understand and interpret, and it can handle high dimensional data with high accuracy. However, it is susceptible to overfitting, especially when dealing with noisy data.



Data Preparation: The first step is to prepare the dataset for use with the decision tree algorithm. This includes cleaning the data, removing missing values, and converting categorical variables into numerical ones. The dataset should be split into training and testing sets.

Decision Tree Construction: The next step is to construct the decision tree. This involves selecting the best attribute to split the data and recursively splitting the dataset based on the values of the selected attribute. This process continues until all the data is classified or a stopping criterion is met.

Tree Pruning: After constructing the decision tree, it may be necessary to prune the tree to improve its accuracy and reduce overfitting. This involves removing nodes that do not significantly improve the accuracy of the tree.

Tree Evaluation: Once the decision tree has been constructed and pruned, it is evaluated using the testing dataset to determine its accuracy. The accuracy of the tree can be improved by adjusting the decision rules and selecting different attributes for splitting.

Prediction: Finally, the decision tree can be used to make predictions on new data. The input features are evaluated against the decision rules in the tree, and the predicted class is returned.

Overall, the decision tree classifier algorithm is a simple and powerful machine learning technique that can be used for a wide range of classification tasks. With careful preparation of the dataset and selection of the decision rules, it can produce accurate and reliable predictions.

ADVANTAGES DECISION TREE CLASSIFIER

- It is the most understandable Machine Learning algorithm, and it can be easily interpreted.
- It can be used for classification and regression problems.
- Unlike most Machine Learning algorithms, it works effectively with non-linear data.
- Constructing a Decision Tree is a speedy process since it uses only one feature per node to split the data.

DISADVANTAGES OF DECISION TREE CLASSIFIER

- Sometimes decision tree learners may over complicate the tree and not generalize the data well. This problem is known as overfitting.
- They don't have very good stability.
- Decision trees are not good at extrapolating data because they are not smooth or continuous.
- Decision trees have issues learning XOR, parity, or multiplexer problems.

5.2 MODULES DESCRIPTION

- **pandas:** Pandas are an open-source data manipulation and data analysis library used for data manipulation and data analysis tasks. It is used to read and manipulate data in a tabular form.
- **tkinter:** Tkinter is a standard Python library used for creating GUI (Graphical User Interface) applications. It provides several widgets and functions to create a GUI application.
- **ttk:** ttk is a module in tkinter library, which provides access to the Tk themed widget set. It provides a more modern look and feel to the GUI applications.
- **sklearn.tree:** This module contains the DecisionTreeClassifier class, which is used for building decision tree models. Decision trees are a popular machine learning algorithm that can be used for both classification and regression problems.

- **sklearn.model_selection:** This module contains the `train_test_split` function, which is used for splitting the dataset into training and testing sets. It is a very important step in machine learning model building process.

5.3 DATASET COLLECTION

Our crop prediction project dataset is collected from [kaggle.com](https://www.kaggle.com). Data is pre-processed after collection of various records. The dataset contains a greater number of records, where some records are with some missing values. Those missing records have been removed from the dataset and filled in the data using python package techniques (Pandas, NumPy). The AI is used to collect some of the trained data for the libraries and it has provided the unique data's as follows.

5.4 SOFTWARE IMPLEMENTATION

- Import the required libraries:
- Python (if not already installed)
- Pandas: “**pip install pandas**”
- Tkinter: “**pip install tk**”
- scikit-learn: “**pip install -U scikit-learn**”
- NumPy: “**pip install numpy**”
- Load the dataset (`crop_data.csv`) using `pandas read_csv` function and store it in a variable.
- Define a function which will predict the crop based on the input values. Inside the function, do the following steps:
- Get the input values for temperature, rainfall, soil type, and soil depth using the **`get()`** method on the respective entry and combobox widgets.
- Filter the crop data based on the soil type and depth selected using `pandas` and store it in a new variable
- Define the feature columns and the target variable column)

- Check if there are any crops found for the given soil type and depth. If not, display "No crops found for the given soil type and depth."
- If there are crops found, split them into features (**X**) and target variable (**y**) using pandas indexing.
- Check if there are enough crops to train the model (at least 2). If not, display "Not enough crops found for the given soil type and depth".
- If there are enough crops, split the **X** and **y** into training and testing sets using function from scikit-learn.
- Create a decision tree classifier model using the `DecisionTreeClassifier` class from scikit-learn.
- Train the model on the training set using the `fit` method of the classifier.
- Predict the crop based on the input values using the `predict` method of the classifier.
- Display the predicted crop
- Create a GUI window using the `Tk` class from `Tkinter` and set its geometry and title.
- Create labels and entry/combo boxes for temperature, rainfall, soil type, and soil depth.
- Create a button called "Predict" which will call the function when clicked.
- Create a label to display the predicted crop.
- Call the `function` method on the window object to run the GUI.

CHAPTER – 6

CONCLUSION AND FURTHER WORK

6.1 CONCLUSION

Based on the climatic input parameters the present study provided the demonstration of the potential use of data mining techniques in predicting the crop yield based. The developed interface is user friendly, and the accuracy of predictions are above 75 per cent in all the crops and districts selected in the study indicating higher accuracy of prediction. By providing climatic data of that place the user-friendly GUI interface developed for predicting crop yield can be used by any user their choice of crop. The present study showed the practical use of data mining techniques in predicting crop yield based on climate input parameters. The built GUI interface is user-friendly, and that reliability of prediction in all the other grains and regions chosen in the analysis should be above 75 percent, indicating greater predictive performance. The net page is developed to predict crop yield by providing data from that area. Any user uses their crop option.

6.2 FURTHER WORK

Yield prediction is an important agricultural problem. In the past farmers used to predict their yield from previous year yield experiences. Thus, for this kind of data analytics in crop prediction, there are different techniques or algorithms, and with the help of those algorithms we can predict crop yield. We can make It has a website and maintained by the server and make .it as an android application to make useful for every user to make easy use of it. The decision tree classifier algorithm is used. Using all these algorithms and with the help of inter-relation between them, there is a growing range of applications and the role of big data analytics techniques in agriculture.

CHAPTER - 7

APPENDICES

7.1 SOURCE CODE

Import required libraries

```
import pandas as pd

import tkinter as tk

from tkinter import ttk

from sklearn.tree import DecisionTreeClassifier

from sklearn.model_selection import train_test_split
```

Load dataset

```
crop_data = pd.read_csv('crop_data.csv')
```

Define function to predict the crop

```
def predict_crop():
```

Get input values for temperature, rainfall, soil type, and soil depth

```
    temp = int(temperature_entry.get())
```

```
    rain = int(rainfall_entry.get())
```

```
    soil_type = soil_type_combobox.get()
```

```
    soil_depth = int(soil_depth_entry.get())
```

Filter crops based on soil type and depth

```
    filtered_crop_data = crop_data.loc[(crop_data['Soil Type'] == soil_type) & (crop_data['Soil Depth'] == soil_depth)]
```

Define feature columns and target variable column

```
    feature_cols = ['Temperature', 'Rainfall']
```

```

target_col = 'Crop'

# Check if there are any crops to predict

if filtered_crop_data.empty:

    crop_label.configure(text="No crops found for the given soil type and depth.")

else:

# Split dataset into features and target variable

    X = filtered_crop_data[feature_cols]

    y = filtered_crop_data[target_col]

# Check if there are enough crops to train the model

    if len(X) < 2:

        crop_label.configure(text="Not enough crops found for the given soil type and depth")

    else:

# Split dataset into training and testing sets

        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


# Create decision tree model

        clf = DecisionTreeClassifier()

# Train model on training set

        clf.fit(X_train, y_train)

# Predict the crop based on input values

        predicted_crop = clf.predict([[temp, rain]])

# Display the predicted crop in the label

        crop_label.configure(text="Predicted crop: " + predicted_crop[0])

#To show the output

# Create GUI window

```

```
window = tk.Tk()window.geometry('400x300') # for resize the o/p window
window.title("Crop Prediction")
```

Create temperature label and entry

```
temperature_label = ttk.Label(window, text="Temperature")
temperature_label.grid(row=0, column=0, padx=10, pady=10)
temperature_entry = ttk.Entry(window)
temperature_entry.grid(row=0, column=1, padx=10, pady=10)
```

Create rainfall label and entry

```
rainfall_label = ttk.Label(window, text="Rainfall")
rainfall_label.grid(row=1, column=0, padx=10, pady=10)
rainfall_entry = ttk.Entry(window)
rainfall_entry.grid(row=1, column=1, padx=10, pady=10)
```

Create soil type label and combobox

```
soil_type_label = ttk.Label(window, text="Soil Type")
soil_type_label.grid(row=2, column=0, padx=10, pady=10)
soil_type_combobox = ttk.Combobox(window, values=crop_data['Soil Type']. unique())
soil_type_combobox.grid(row=2, column=1, padx=10, pady=10)
```

Create soil depth label and entry

```
soil_depth_label = ttk.Label(window, text="Soil Depth")
soil_depth_label.grid(row=3, column=0, padx=10, pady=10)
soil_depth_entry = ttk.Entry(window)
soil_depth_entry.grid(row=3, column=1, padx=10, pady=10)
```

Create predict button

```
predict_button = ttk.Button(window, text="Predict", command=predict_crop)
predict_button.grid(row=4, column=0, columnspan=2, padx=10, pady=10)
```

Create predicted crop label

```
crop_label = ttk.Label(window)
```

```
crop_label.grid(row=5, column=0, columnspan=2, padx=10, pady=10)
```

Run the GUI window

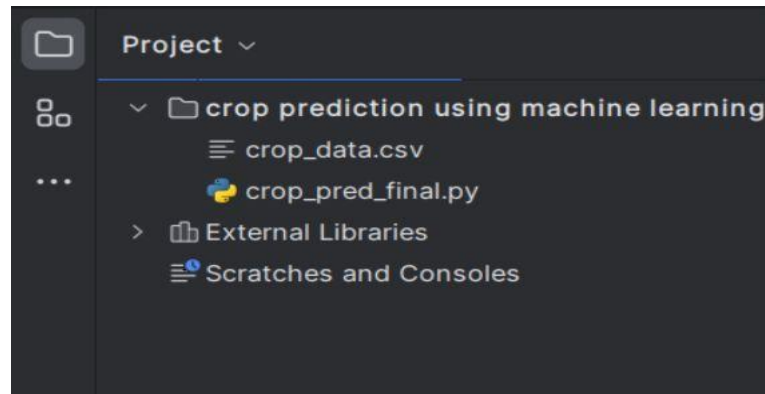
```
window.mainloop()
```

#Some of the sample trained data sets named as ('**crop_data.csv**')

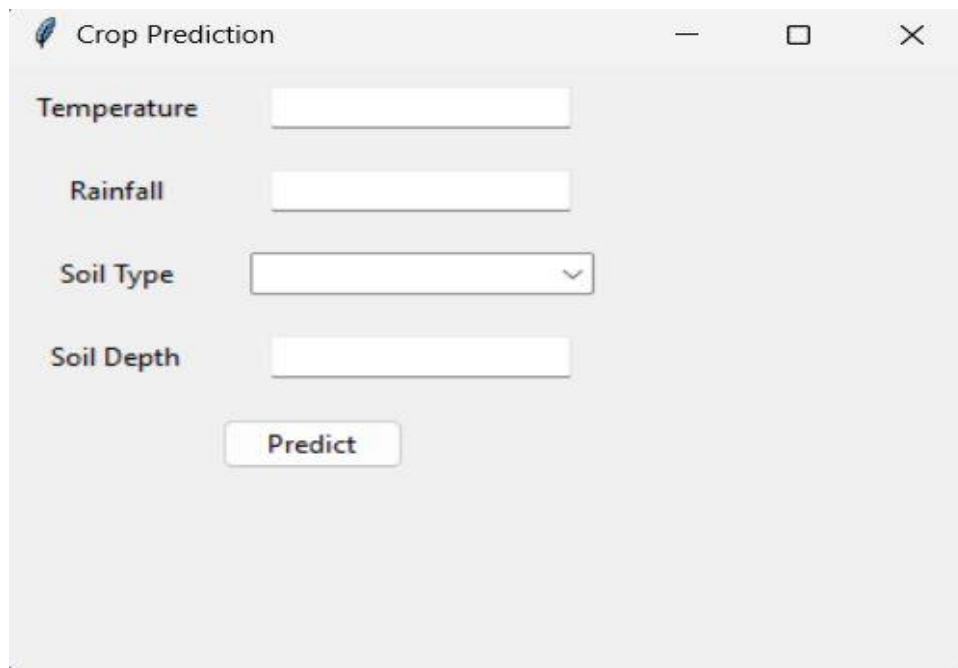
Temperature	Rainfall	Soil Type	Soil Depth	Crop
20	100	Sandy Loam	50	Wheat
25	150	Clay Loam	70	Rice
30	80	Sandy Loam	60	Maize
25	120	Clay Loam	80	Sugarcane
28	90	Sandy Loam	50	Cotton
22	110	Clay Loam	70	Soyabean
23	20	Red Soil	80	Millets

7.2 OUTPUT SCREENSHOT

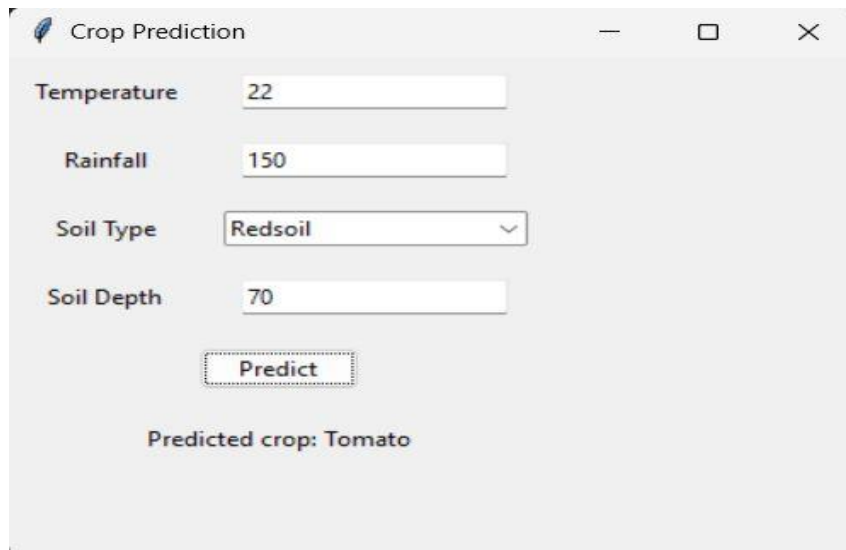
When the python program gets executed first it gets loaded or goes to the CSV file. The CSV file contains the number of pre trained data sets, and it has named as a ('crop_data.csv').



When the above sample code gets executed, we get the output window as shown below.

A screenshot of a 'Crop Prediction' application window. The window has a title bar with a feather icon and the text 'Crop Prediction'. It contains four input fields: 'Temperature', 'Rainfall', 'Soil Type' (a dropdown menu), and 'Soil Depth'. Below these fields is a 'Predict' button.

When the user gets the output window, they need to enter the following data and click the predict button now we get output for the predicted crop as shown below.



Crop Prediction

Temperature: 22

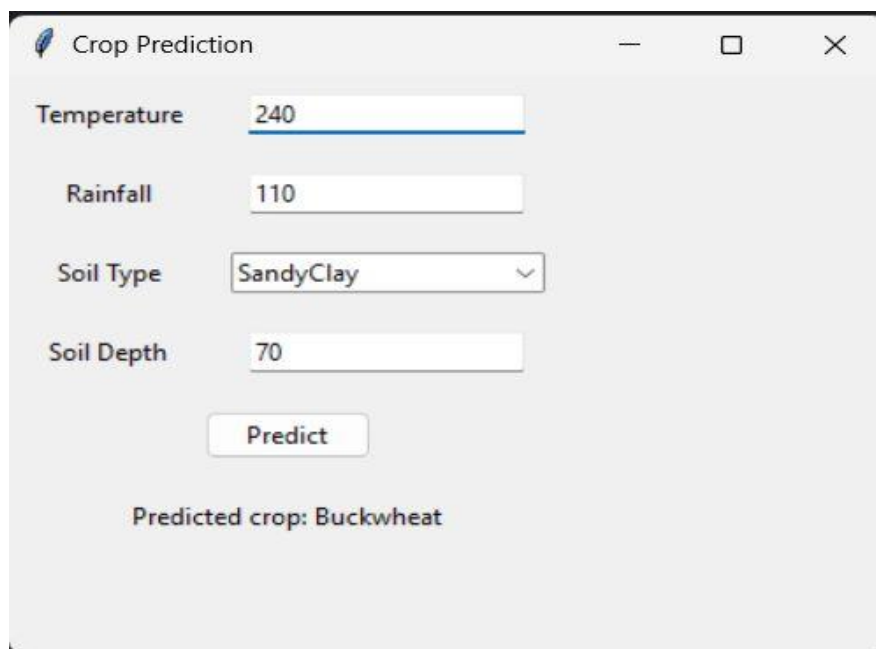
Rainfall: 150

Soil Type: Redsoil

Soil Depth: 70

Predict

Predicted crop: Tomato



Crop Prediction

Temperature: 240

Rainfall: 110

Soil Type: SandyClay

Soil Depth: 70

Predict

Predicted crop: Buckwheat

CHAPTER – 8

RESULT AND DISCUSSION

8.1 RESULT

The program provides a GUI interface for the user to input the temperature, rainfall, soil type, and soil depth values. When the user clicks on the "Predict" button, the program predicts the most likely crop to grow based on the input values and displays the result in the GUI label.

8.2 USES OF THE EXTRACTED INFORMATION

The extracted information from the above program can be used in various ways, including:

- **Crop prediction:** The program can be used to predict the type of crop that is suitable for a particular combination of temperature, rainfall, soil type, and soil depth. This information can be useful for farmers who want to maximize their yield by choosing the right crop to grow on their land.
- **Decision tree classification:** The program uses a decision tree classifier to predict the type of crop based on the input variables. The decision tree can be visualized and analyzed to gain insights into the factors that are most important in determining the type of crop that can be grown in a particular region.
- **Data analysis:** The program loads a dataset of crop data and filters it based on soil type and depth. This data can be used for further analysis to gain insights into the relationship between soil characteristics and crop growth.
- **GUI development:** The program also demonstrates how to create a simple GUI interface using tkinter, which can be useful for developing other applications that require user input and output.

Overall, the information extracted from this program can be useful for farmers, researchers, and developers interested in agriculture and data analysis.

CHAPTER - 9

REFERENCES

- [1] . S.Veenadhari, Dr Bharat Misra, Dr CD Singh.2019."Machine learning approach for forecasting crop yield based on climatic parameters.". International Conference on Computer Communication and Informatics (ICCCI), 2014.
- [2] . P.Priya, U.Muthaiah M.Balamurugan.Predicting yield of the crop using machine learning algorithm. International Journal of Engineering Science Research Technology (IJESRT),2018.
- [3] . D Ramesh, B Vishnu Vardhan. Analysis Of Crop Yield Prediction Using Data Mining Techniques. IJRET: International Journal of Research in Engineering and Technology. (IJERT),2015.
- [4] . F. Baheri, F. Davardoost and V. Ahmadzadeh, "Data mining with learning decision tree and Bayesian network for data replication in Data Grid", Proceedings of the Third International Conference on Contemporary Issues in Computer and Information Sciences, pp. 51- 51, 2012.
- [5] . Abdullah, A., Brobst, S, Pervaiz. I., Umer M.,and A. Nisar.2004. Learning dynamics of pesticide abuse through data mining. Proceedings of Australian Workshop on Data mining and Web Intelligence, New Zealand, January,2004.
- [6] . C.T. Dhanya and D. Nagesh Kumar, "Data mining for evolution of association rules for droughts and floods in India using climate inputs", J. of Geo. Phy. Res.114: 1-15, 2009.
- [7] . VS Boddu, BNK Reddy, MK Kumar, "Low-power and area efficient N-bit parallel processors on a chip," 2016 IEEE annual India conference (INDICON), pp. 1-4, 2016.
- [8] . M. Kaur, H. Gulati, and H. Kundra, "Data Mining in Agriculture on Crop Price Prediction: Techniques and Applications," International Journal of Computer Applications, vol. 99, no. 12, pp.1-3, August.2014.

[9] . F. Baheri, F. Davardoost and V. Ahmadzadeh, "Data mining with learning decision tree and Bayesian network for data replication in Data Grid", Proceedings of the Third International Conference on Contemporary Issues in Computer and Information Sciences, pp. 51- 51, 2012

