A

PROJECT REPORT

on

FARMEASY: An Environment Intelligence Project

Submitted in partial fulfillment for the award of degree of

BACHELOR OF TECHNOLOGY

In

Computer Science & Engineering

Submitted to:
Mr. Pradeep Jha
HOD-CSE/IT Dept

Submitted By:
Nishant Kumar(18EGJCS085)
Nitesh Kumar(18EGJCS086)
Himanshu Sharma(18EGJC051)

Department of Computer Science & Engineering
GLOBAL INSTITUTE OF TECHNOLOGY
JAIPUR (RAJASTHAN)-302022

Batch: 2018-22

GLOBAL INSTITUTE OF TECHNOLOGY

ITS-1 & 2, IT Park, EPIP, Sitapura, Jaipur – 302022, Rajasthan – INDIA

Certificate

This is to certify that the work, which is being presented in the project entitled **"FARMEASY: An Environment Intelligence Project"** submitted by Mr. **Himanshu Sharma, Nishant Kumar, Nitesh Kumar** a student of fourth year (VIII Sem), B.Tech. in Computer Science & Engineering, in partial fulfilment for the award of degree of Bachelor of Technology is a record of student's work carried out and found satisfactory for submission.

Himanshu Sharma (18EGJCS051)
Nishant Kumar (18EGJCS085)
Nitesh Kumar (18EGJCS086)

Guide:	Project Coordinator
(Mr. Loveleen Kumar)	Mr. Rajesh Rajaan.

Mr. Pradeep Jha
(HOD, CSE/IT Dept.)

Acknowledgement

We take this opportunity to express my deep sense of gratitude to project coordinator **Mr. Rajesh Rajaan**, Assistant Professor, Department of Computer Science and Engineering and our project guide **"Mr. Loveleen Kumar",** Assistant Professor Department of Computer Science and Engineering, Global Institute of Technology, Jaipur, for his valuable guidance and cooperation throughout the Project work. He provided constant encouragement and unceasing enthusiasm at every stage of the project work.

We are grateful to our respected **Dr. I. C. Sharma, Principal GIT,** for guiding us. We express our indebtedness to **Mr. Pradeep Jha**, Head of the Department of Computer Science and Engineering, Global Institute of Technology, Jaipur, for providing us with ample support.

Without their support and timely guidance, the completion of our project would have seemed a farfetched dream. We find ourselves lucky to have mentors of such great potential in this respect.

Place: GIT, Jaipur

NISHANT KUMAR (18EGJCS085)

HIMANSHU SHARMA (18EGJCS051)

NITESH KUMAR (18EGJCS086)

B.Tech. VIII Semester, IV Year

Computer Science &Engineering

ABSTRACT

Farming is one of the major sectors that influences a country's economic growth. In country like India, majority of the population is dependent on agriculture for their livelihood. Many new technologies, such as Machine Learning and Deep Learning, are being implemented into agriculture so that it is easier for farmers to grow and maximize their yield.

Many of the farmers don't know about which crop to grow in their field according to their field soil temperature, water levels, soil conditions, which of the fertilizers will use for better growing of their crop, if any disease occur in their crop so which of the solutions are available. Due to these problems this solution of crop recommendation, fertilizer recommendation and disease caught system predict the user which crop type, which fertilizer type would be the best suitable for selected area and also which of the disease is occur in crop by collecting the environmental factors for plant growth and processing them with the trained sub models of the main of the system.

In this project, we present a website in which the following applications are implemented Crop recommendation, Fertilizer recommendation and Plant disease prediction, respectively

In the crop recommendation application, the user can provide the soil data from their side and the application will predict which crop should the user grow.

For the fertilizer recommendation application, the user can input the soil data and the type of crop they are growing, and the application will predict whatthe soil lacks or has excess of and will recommend improvements.

For the last application, that is the plant disease prediction application, the user can input an image of a diseased plant leaf, and the application will predict what disease it is and will also give a little background about the disease and suggestions to cure it.

Table of Contents

List Of Figures

Chapter-1

INTRODUCTION AND OBJECTIVE

A farmer’s decision about which crop to grow is generally clouded by his intuition and other irrelevant factors like making instant profits, lack of awareness about market demand, overestimating a soil’s potential to support a particular crop, and so on. A very misguided decision on the part of the farmer could place a significant strain on his family’s financial condition. Perhaps this could be one of the many reasons contributing to the countless suicide cases of farmers that we hear from media on a daily basis. In a country like India, where agriculture and related sector contributes to approximately 20.4 per cent of its Gross Value Added, such an erroneous judgment would have negative implications on not just the farmer’s family, but the entire economy of a region.

For this reason, we have identified a farmer’s dilemma about which crop to grow during a particular season, as a very grave one. The need of the hour is to design a system that could provide predictive insights to the Indian farmers, thereby helping them make an informed decision about which crop to grow. With this in mind, we propose a system, an intelligent system that would consider environmental parameters (temperature, rainfall, geographical location in terms of state) and soil characteristics (pH value, soil type and nutrients concentration) before recommending the most suitable crop to the user.

The designed system will recommend the most suitable crop for particular land. Based on weather parameter and soil content such as Rainfall, Temperature, Humidity and pH. They are collected from, Government website and weather department websites. The system takes the required input from the farmers such as Temperature, Humidity and pH. This all inputs data applies to machine learning predictive algorithms like Support Vector Machine (SVM) and Decision tree to identify the pattern among data and then process it as per input conditions. The system recommends the crop for the farmer and also recommends the amount of nutrients to be add for the predicted crop.

1.1 Existing System:

More and more researchers have begun to identify this problem in Indian agriculture and are increasingly dedicating their time and efforts to help alleviate the issue. Different works include the use of Regularized Greedy Forest to determine an appropriate crop sequence at a given time stamp.

Another approach proposes a model that makes use of historical records of meteorological data as training set. Model is trained to identify weather conditions that are deterrent for the production of apples. It then efficiently predicts the yield of apples on the basis of monthly weather patterns.

The use of several algorithms like Artificial Neural Network, K Nearest Neighbors, and Regularized Greedy Forest is demonstrated into select a crop based on the prediction yield rate, which, in turn, is influenced by multiple parameters. Additional features included in the system are pesticide prediction and online trading based on agricultural commodities

1.2 Proposed System:

we propose an Intelligent Crop Recommendation system- which takes into consideration all the appropriate parameters, including temperature, rainfall, location and soil condition, to predict crop suitability.

This system is fundamentally concerned with performing the primary function of agriculture consultant, which is, providing crop recommendations, fertilizer recommendation to farmers using algorithms like decision tree, guassian naïve bayes, support vector machine, logistic regression, random forest.

For the last application, that is the plant disease prediction application, the user can input an image of a diseased plant leaf, and the application will predict what disease it is and will also give a little background about the disease and suggestions to cure it.

1.3 System Implementation:

1.3.1 Data Collection:

The dataset have been obtained from different official government websites:

- Data.gov.in(details regarding area, production, crop name)
- Indianwaterportal.org(rainfall details)
- Power.iarc.nasa.in(temperature, humidity, pH details)

1.3.2 Data Preprocessing:

After collecting datasets from various resources. Dataset must be preprocessing before training to the model. The data preprocessing can be done by various stages, begins with reading the collected dataset the process continues to data cleaning.

In data cleaning the datasets contain some redundant attributes, those attributes are not considering for crop prediction. So, we have to drop unwanted attributes and datasets containing some missing values we need to drop these missing values or fill with unwanted nan values in order to get better accuracy.

Then define the target for a model. After data cleaning the dataset will be split into training and test set by using sklearn library

1.3.3 Training Model:

After the preprocessing step we used the data-set to train different machine learning models like support vector machine and logistic regression to attain accuracy as high as possible in crop and fertilizer recommendation. Neural network for disease prediction.

1.4 Objective Of The Project:

- To build a robust model to give correct and accurate prediction of crop sustainability in a given state for the particular soil type and climatic conditions.
- Provide recommendation of the best suitable crops in the area so that the farmer does not incur any losses.

Chapter-2

SYSTEM REQUIREMENTS SPECIFICATION

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide.

In order to fully understand one's project, it is very important that they come up with a SRS listing out their requirements, how are they going to meet it and how will they complete the project. It helps the team to save upon their time as they are able to comprehend how are going to go about the project.

Doing this also enables the team to find out about the limitations and risks early on. Requirement is a condition or capability to which the system must conform. Requirement Management is a systematic approach towards eliciting, organizing and documenting the requirements of the system clearly along with the applicable attributes. The elusive difficulties of requirements are not always obvious and can come from any number of sources.

2.1 Functional Requirements:

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality.

Following are the functional requirements on the system:

- All the data must be in the same format as a structured data.
- The data collected will be vectorized and sent across to the classifier

2.2 Non-Functional Requirements:

Non-functional requirements are the requirements which are not directly concerned with the specific function delivered by the system.
They specify the criteria that can be used to judge the operation of a system rather than specific behaviors.

They may relate to emergent system properties such as reliability, response time and store occupancy.
Non functional requirements arise through the user needs, because of budget constraints, organizational policies and the need for interoperability with other software and hardware systems.

2.2.1 Software Requirement:

2.2.1.1 Hardware Configuration:

CPU: Processors above **Intel Core i5 8th Generation** is advised as it is more powerful and delivers High Performance.

GPU: Minimum NVIDIA Geforce 130 / AMD Radeon 530

RAM: Minimum 8 Gb or above**.**

Storage: A minimum of **1TB HDD / SSD** a minimum of **256 GB** is advised

2.2.1.2 Software Configuration:

Operating System: Windows XP/7/8/8.1/10, Linux and Mac

Coding Language: Python

Tools:

1. Vs Code
2. Jupyter Notebook
3. Google Collab

4. Anaconda Prompt

2.2.2 Product Requirements:

The complete product is broken up into many modules and well-defined interfaces are developed to explore the benefit of flexibility of the product. This software is being developed in such a way that the overall performance is optimized and the user can expect the results within a limited time with utmost relevancy and correctness.

Non functional requirements are also called the qualities of a system. These qualities can be divided into execution quality and evolution quality. Execution qualities are security and usability of the system which are observed during run time, whereas evolution quality involves testability, maintainability, extensibility or scalability.

Chapter-3

METHEDOLOGY

3.1 Overview On Machine Learning:

Machine learning is an application of artificial intelligence (AI) that gives systems the ability to automatically learn and evolve from experience without being specially programmed by the programmer. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide.

The main aim of machine learning is to allow computers to learn automatically and adjust their actions to improve the accuracy and usefulness of the program, without any human intervention or assistance.

Traditional writing of programs for a computer can be defined as automating the procedures to be performed on input data in order to create output artifacts. Almost always, they are linear, procedural and logical. A traditional program is written in a programming language to some specification, and it has properties like:
We know or can control the inputs to the program.

- We can specify how the program will achieve its goal.
- We can map out what decisions the program will make and under what conditions it makes them.
- Since we know the inputs as well as the expected outputs, we can be con dent that the program will achieve its goal.

Traditional programming works on the premise that, as long as we can define what a program needs to do, we are confident we can define how a program can achieve that goal. This is not always the case as sometimes, however, there are problems that you can represent in a computer that you cannot write a traditional program to solve. Such problems resist a procedural and logical solution.

They have properties such as:

- The scope of all possible inputs is not known beforehand.

- You cannot specify how to achieve the goal of the program, only what that goal is.
- You cannot map out all the decisions the program will need to make to achieve its goal.
- You can collect only sample input data but not all possible input data for the program.

3.2 Supervised And Unsupervised Machine Learning:

Machine learning techniques can be broadly categorized into the following types:

Supervised learning takes a set of feature/label pairs, called the training set. From this training set the system creates a generalized model of the relationship between the set of descriptive features and the target features in the form of a program that contains a set of rules. The objective is to use the output program produced to predict the label for a previously unseen, un labelled input set of features, i.e. to predict the outcome for some new data. Data with known labels, which have not been included in the training set, are classified by the generated model and the results are compared to the known labels.

This dataset is called the test set. The accuracy of the predictive model can then be calculated as the proportion of the correct predictions the model labeled out of the total number of instances in the test set.

Unsupervised learning takes a dataset of descriptive features without labels as a train- ing set. In unsupervised learning, the algorithms are left to themselves to discover interesting structures in the data.

The goal now is to create a model that finds some hidden structure in the dataset, such as natural clusters or associations. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data.

The system does not figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.

Unsupervised learning can be used for clustering, which is used to discover any inherent grouping that are already present in the data. It can also be used for association problems, by creating rules based on the data and finding relationships or associations between them.

Semi-supervised machine learning falls somewhere in between supervised and unsupervised learning, since they use both labeled and unlabeled data for training typically a small amount of labeled data and a large amount of unlabeled data.

The systems that use this method are able to considerably improve learning accuracy. Usually, semi supervised learning is chosen when the acquired labeled data requires skilled and relevant resources in order to train it / learn from it. Otherwise, acquiring labeled data generally does not require additional resources.

Reinforcement machine learning algorithms is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Machine learning algorithms are tools to automatically make decisions from data in order to achieve some over-arching goal or requirement.

The promise of machine learning is that it can solve complex problems automatically, faster and more accurately than a manually specified solution, and at a larger scale. Over the past few decades, many machine learning algorithms have been developed by researchers, and new ones continue to emerge and old ones modified.

3.3 Dataset:

For the system, we are using various datasets all downloaded for government website.
A brief description of the datasets:

Yield Dataset: This dataset contains yield for major crops grown across all the states in kg and also the hectares in which the crops are grown.
Fertilizer Dataset: This dataset contains amount of some important fertilizers according to the crop.

Rainfall Temperature dataset: This dataset contains rainfall, temperature and pH values data according to the crop.

3.4 Data Preprocessing:

After collecting datasets from various resources. Dataset must be preprocessing before training to the model. The data preprocessing can be done by various stages, begins with reading the collected dataset the process continues to data cleaning.

In data cleaning the datasets contain some redundant attributes, those attributes are not considering for crop prediction. So, we have to drop unwanted attributes and datasets containing some missing values we need to drop these missing values or fill with unwanted nan values in order to get better accuracy.

Then define the target for a model. After data cleaning the dataset will be split into training and test set by using sklearn library.

3.5 Machine Learning Algorithms:

3.5.1 Support Vector Machine:

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data

point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane.

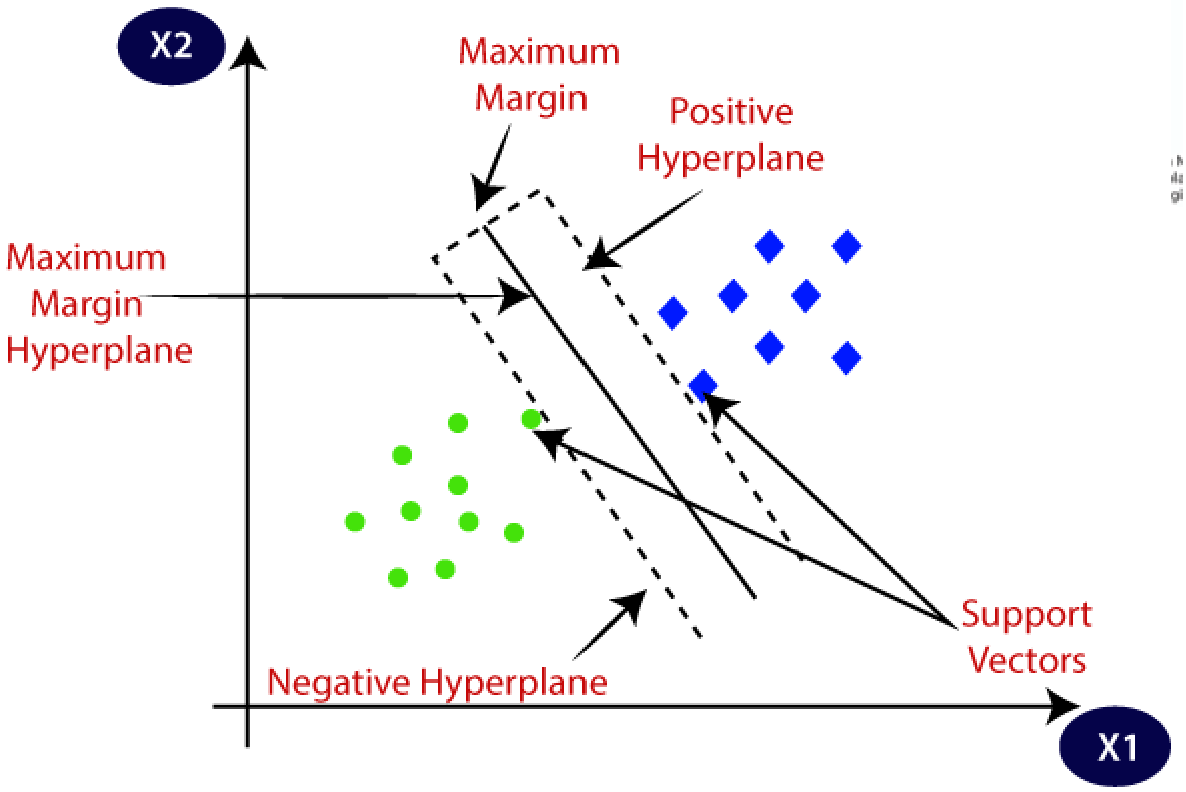

Fig 3.5.1 SVM Model Graph Representation

SVM can be of two types:

Linear SVM: Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

Non-linear SVM: Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

3.5.2 Gaussian Naïve Bayes:

Naïve Bayes is a probabilistic machine learning algorithm used for many classification functions and is based on the Bayes theorem. Gaussian Naïve Bayes is the extension of naïve Bayes. While other functions are used to estimate data distribution, Gaussian or normal distribution is the simplest to implement as you will need to calculate the mean and standard deviation for the training data.

Naive Bayes is a probabilistic machine learning algorithm that can be used in several classification tasks. Typical applications of Naive Bayes are classification of documents, filtering spam, prediction and so on. This algorithm is based on the discoveries of Thomas Bayes and hence its name.

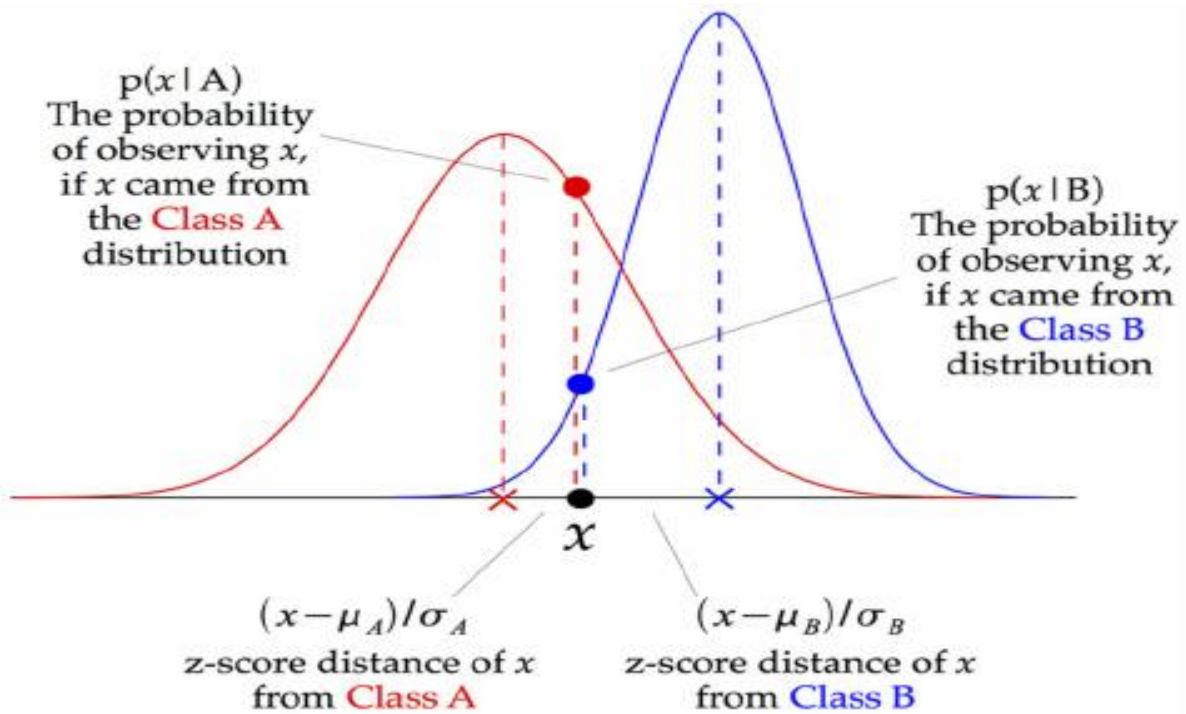

Fig 3.5.2 Gaussian Naïve Bayes Model Graph Representation

The name "Naïve" is used because the algorithm incorporates features in its model that are independent of each other. Any modifications in the value of one feature do not directly impact the value of any other feature of the algorithm. The main advantage of the Naïve Bayes algorithm is that it is a simple yet powerful algorithm.

It is based on the probabilistic model where the algorithm can be coded easily, and predictions did quickly in real-time. Hence this algorithm is the typical choice to solve real world problems as it can be tuned to respond to user requests instantly. But before we dive deep into Naïve Bayes and Gaussian Naïve Bayes, we must know what is meant by conditional probability.

3.5.3 Random Forest:

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning,** which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The below diagram explains the working of the Random Forest algorithm:

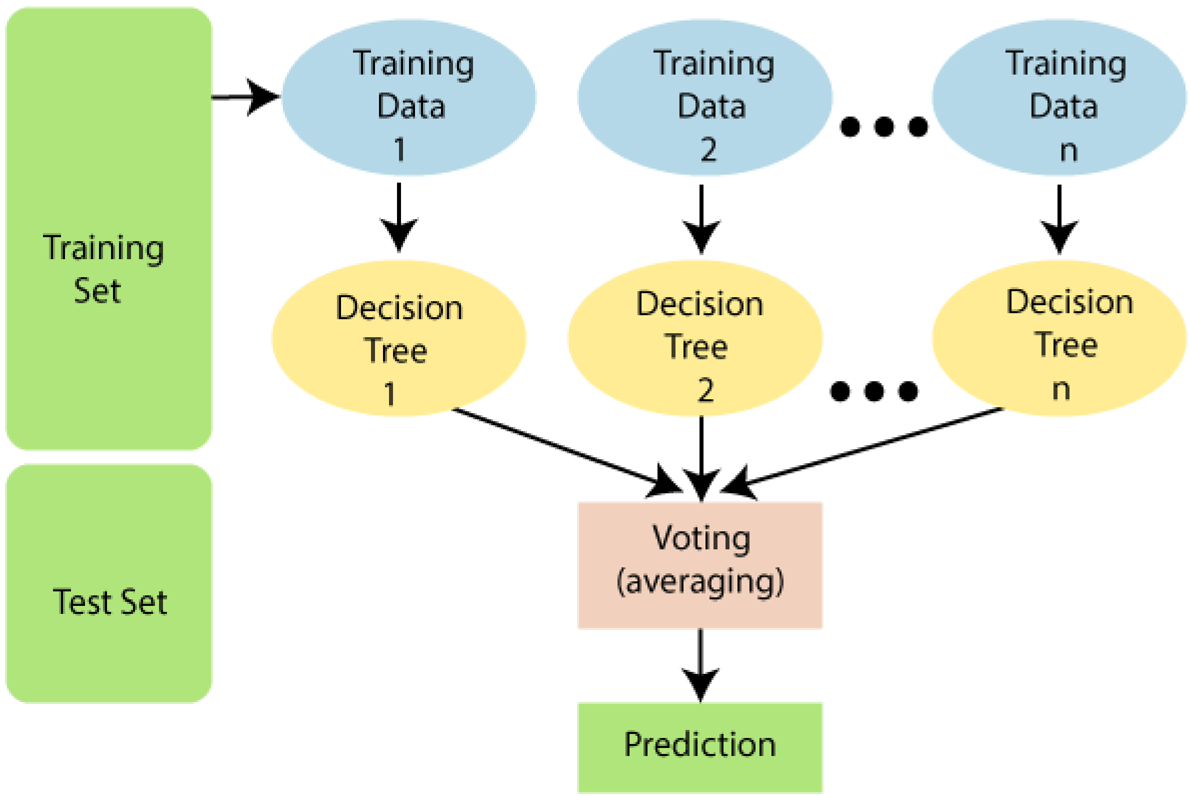

Fig 3.5.3 Random Forest Model Node Representation

3.5.4 Decision Tree:

Decision Tree is a **Supervised learning technique** that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. 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.

In a Decision tree, there are two nodes, which are the **Decision Node** and **Leaf Node.** Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

Below diagram explains the general structure of a decision tree:

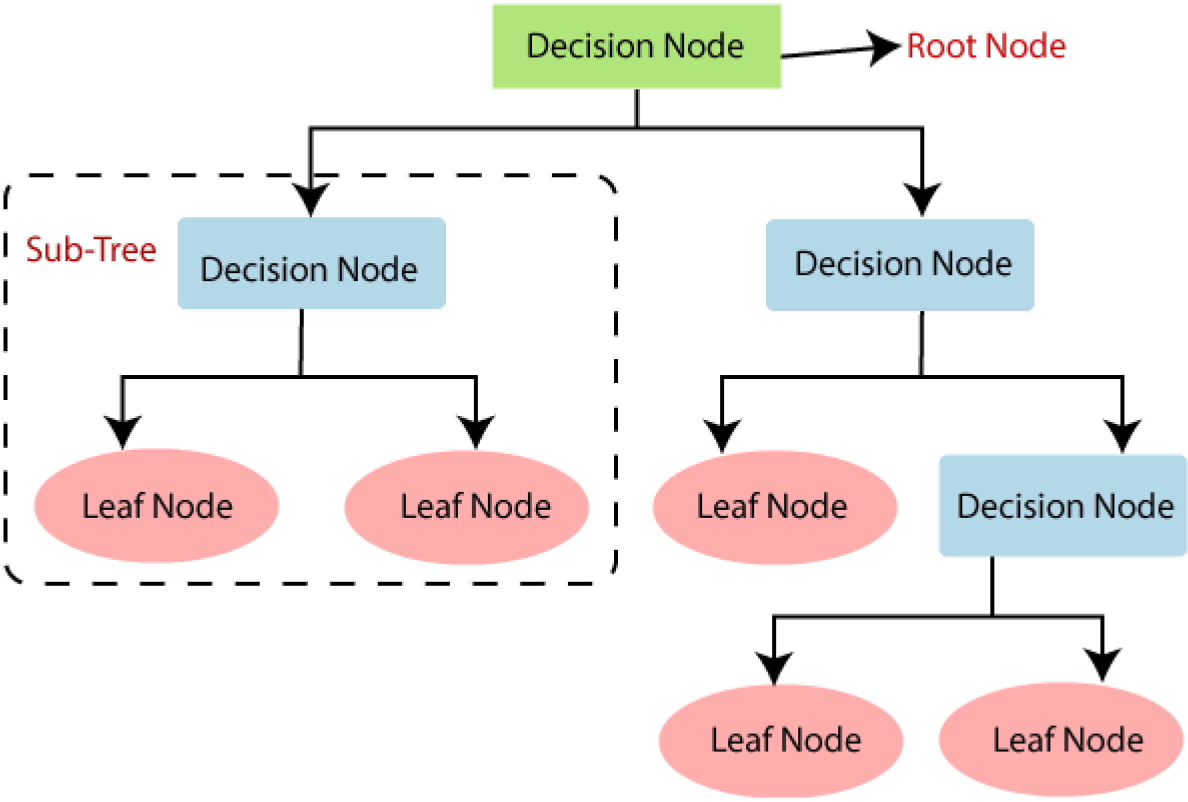

Fig 3.5.4 Decision Tree Model Node Representation

The decisions or the test are performed on the basis of features of the given dataset. It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.

It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.

In order to build a tree, we use the CART algorithm**,** which stands for **Classification and Regression Tree algorithm.** A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.

3.5.5 Logistic Regression:

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1**.**

Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems**.**

The below image is showing the logistic function:

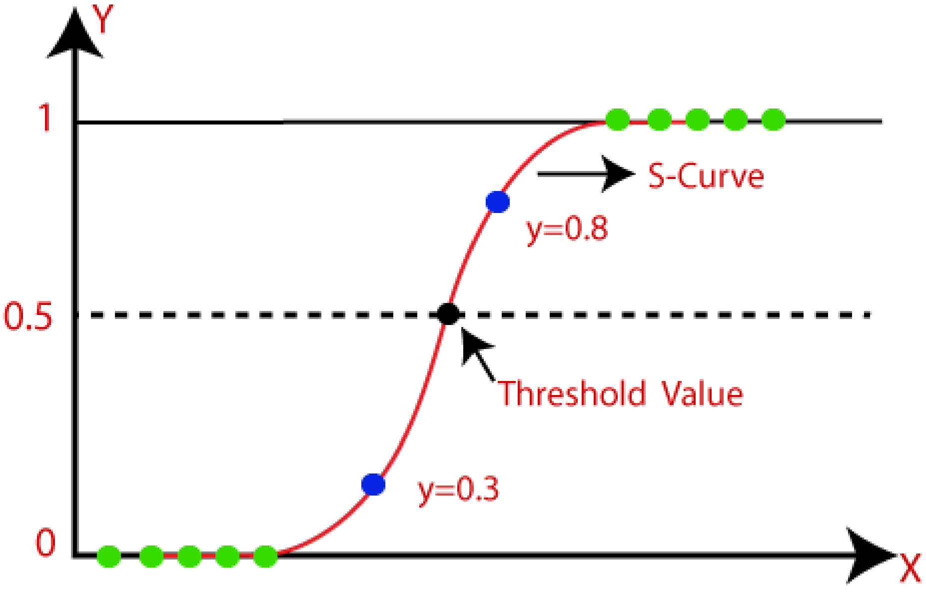

Fig 3.5.5 Logistic Regression Model Graph Representation

In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).

The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.

Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.

Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification.

3.6 CNN(Convolutional Neural Network):

A convolutional neural network, or CNN, is a deep learning neural network sketched for processing structured arrays of data such as portrayals. CNN are very satisfactory at picking up on design in the input image, such as lines, gradients, circles, or even eyes and faces.

This characteristic that makes convolutional neural network so robust for computer vision. CNN can run directly on a underdone image and do not need any preprocessing.

A convolutional neural network is a feed forward neural network, seldom with up to 20. The strength of a convolutional neural network comes from a particular kind of layer called the convolutional layer.

CNN contains many convolutional layers assembled on top of each other, each one competent of recognizing more sophisticated shapes. With three or four convolutional layers it is viable to recognize handwritten digits and with 25 layers it is possible to differentiate human faces.

The agenda for this sphere is to activate machines to view the world as humans do, perceive it in a alike fashion and even use the knowledge for a multitude of duty such as image and video recognition, image inspection and classification, media recreation, recommendation systems, natural language processing, etc.

The construction of a convolutional neural network is a multi-layered feed-forward neural network, made by assembling many unseen layers on top of each other in a particular order. It is the sequential design that give permission to CNN to learn hierarchical attributes. In CNN, some of them followed by grouping layers and hidden layers are typically convolutional layers followed by activation layers.

The pre-processing needed in a ConvNet is kindred to that of the related pattern of neurons in the human brain and was motivated by the organization of the Visual Cortex.

Chapter-4

SYSTEM ANALYSIS

4.1 Feasibility Study:

Analysis is the process of finding the best solution to the problem. System analysis is the process by which we learn about the existing problems, define objects and requirements and evaluates the solutions. It is the way of thinking about the organization and the problem it involves, a set of technologies that helps in solving these problems. Feasibility study plays an important role in system analysis which gives the target for design and development.

4.1.1 Technical Feasibility:

This study is carried out to check the technical feasibility, that is, the technical re-quirements of the system. Since machine learning algorithms is based on pure math there is very less requirement for any professional software. And also most of the tools are open source. The best part is that we can run this software in any system without any software requirements which makes them highly portable. Also most of the documentation and tutorials make easy to learn the technology.

4.1.2 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 main purpose of this project which is based on crop prediction is to prevent the farmer from incurring losses and improve productivity. This also ensures that there is no scarcity of food as lack of production may lead to severe consequences. Thus, this is a noble cause for the sake of the society, a small step taken to achieve a secure future.

4.2 Analysis:

4.2.1 Performance Analysis:

Most of the software we use is open source and free. The models which we use in this software, learn only once they are trained they need not be again fed in for the training phase. One can directly predict for values, hence time-complexity is very less. Therefore this model is temporally sound.

4.2.2 Technical Analysis:

As mentioned earlier, the tools used in building this software is open source. Each tool contains simple methods and the required methods are overridden to tackle the problem.

4.2.3 Economical Analysis:

The completion of this project can be considered free of cost in its entirety. As the software used in building the model is free of cost and all the data sets used are being downloaded from Govt. of India website.

Chapter-5

IMPLEMENTATION

5.1 Data Analysis:

One of the first steps we perform during implementation is an analysis of the data. This was done by us in an attempt to find the presence of any relationships between the various attributes present in the dataset.

5.1.1 Acquisition Of Training Dataset:

The accuracy of any machine learning algorithm depends on the number of parameters and the correctness of the training dataset. We In this project analyzed multiple datasets collected from Government website -https://data.gov.in/ carefully selected the parameters that would give the best results.

Many work done in this field have considered environmental parameters to predict crop sustainability some have used yield as major factor where as in some works only economic factors are taken into consideration.

We have tried to combine both environmental parameters like rainfall, temperature, pH, nutrients in soil, soil type, location and economic parameters like production, and yield to provide accurate and reliable recommendation to the farmer on which crop will be most suitable for his land.

Here are some images of different datasets that are collected from govt India website:

	A	B	C	D	E	F
1		Crop	N(nitrozen (kg/ha-1))	P(phosphorus (kg/ha-1))	K(potassium (kg/ha-1))	pH
2	0	Rice	80	40	40	5.5
3	1	Jowar(Sorghum)	80	40	40	5.5
4	2	Barley(JAV)	70	40	45	5.5
5	3	Maize	80	40	20	5.5
6	4	Ragi(naachnnii)	50	40	20	5.5
7	5	Chickpeas(Channa)	40	60	80	5.5
8	6	French Beans(Farasbi)	90	125	60	5
9	7	Fava beans (Papdi - Val)	90	125	60	5
10	8	Lima beans(Pavta)	40	60	20	5
11	9	Cluster Beans(Gavar)	25	50	25	5
12	10	Soyabean	20	60	20	5.5
13	11	Black eyed beans(chawli)	20	60	20	5.5
14	12	Kidney beans	20	60	20	5.5
15	13	pigeon peas(Toor Dal)	20	60	20	5.5
16	14	Moth bean(Matki)	20	40	20	5.5
17	15	Mung beans	20	40	20	5.5
18	16	Green Peas	40	35	55	6
19	17	Horse Gram(kulthi)	20	60	20	6
20	18	Black Gram	40	60	20	5
21	19	Rapeseed (Mohri)	50	40	20	5
22	20	Coriander seeds	90	20	20	6.5
23	21	Mustard seeds	100	30	15	6.5
24	22	sesame seed	30	15	30	6.5
25	23	Cumin seeds	90	60	20	6.5
26	24	Lentils(Masoor Dal)	20	60	20	5.5

Fig 5.1 (a) Fertilizer Dataset

	A	B	C	D	E
1	temperature(celcius)	humidity	ph	rainfall(mm)	label
2	20.87974371	82.00274423	6.502985292	202.9355362	rice
3	21.77046169	80.31964408	7.038096361	226.6555374	rice
4	23.00445915	82.3207629	7.840207144	263.9642476	rice
5	26.49109635	80.15836264	6.980400905	242.8640342	rice
6	20.13017482	81.60487287	7.628472891	262.7173405	rice
7	23.05804872	83.37011772	7.073453503	251.0549998	rice
8	22.70883798	82.63941394	5.70080568	271.3248604	rice
9	20.27774362	82.89408619	5.718627178	241.9741949	rice
10	24.51588066	83.5352163	6.685346424	230.4462359	rice
11	23.22397386	83.03322691	6.336253525	221.2091958	rice
12	26.52723513	81.41753846	5.386167788	264.6148697	rice
13	23.97898217	81.45061596	7.50283396	250.0832336	rice
14	26.80079604	80.88684822	5.108681786	284.4364567	rice
15	24.01497622	82.05687182	6.98435366	185.2773389	rice
16	25.66585205	80.66385045	6.94801983	209.5869708	rice
17	24.28209415	80.30025587	7.042299069	231.0863347	rice
18	21.58711777	82.7883708	6.249050656	276.6552459	rice
19	23.79391957	80.41817957	6.970859754	206.2611855	rice
20	21.8652524	80.1923008	5.953933276	224.5550169	rice
21	23.57943626	83.58760316	5.85393208	291.2986618	rice
22	21.32504158	80.47476396	6.442475375	185.4974732	rice
23	25.15745531	83.11713476	5.070175667	231.3843163	rice
24	21.94766735	80.97384195	6.012632591	213.3560921	rice

Fig 5.1 (b) Rainfall Dataset

	A	B	C	D	E	F	G
1	State_Name	District_Name	Crop_Year	Season	Crop	Area(hactares)	Production
2	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Arecanut	1254	2000
3	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Other Kharif pulses	2	1
4	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Rice	102	321
5	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Banana	176	641
6	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Cashewnut	720	165
7	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Coconut	18168	65100000
8	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Dry ginger	36	100
9	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Sugarcane	1	2
10	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Sweet potato	5	15
11	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Tapioca	40	169
12	Andaman and Nicobar Islands	NICOBARS	2001	Kharif	Arecanut	1254	2061
13	Andaman and Nicobar Islands	NICOBARS	2001	Kharif	Other Kharif pulses	2	1
14	Andaman and Nicobar Islands	NICOBARS	2001	Kharif	Rice	83	300
15	Andaman and Nicobar Islands	NICOBARS	2001	Whole Year	Cashewnut	719	192
16	Andaman and Nicobar Islands	NICOBARS	2001	Whole Year	Coconut	18190	64430000
17	Andaman and Nicobar Islands	NICOBARS	2001	Whole Year	Dry ginger	46	100
18	Andaman and Nicobar Islands	NICOBARS	2001	Whole Year	Sugarcane	1	1
19	Andaman and Nicobar Islands	NICOBARS	2001	Whole Year	Sweet potato	11	33
20	Andaman and Nicobar Islands	NICOBARS	2002	Kharif	Rice	189.2	510.84
21	Andaman and Nicobar Islands	NICOBARS	2002	Whole Year	Arecanut	1258	2083
22	Andaman and Nicobar Islands	NICOBARS	2002	Whole Year	Banana	213	1278
23	Andaman and Nicobar Islands	NICOBARS	2002	Whole Year	Black pepper	63	13.5
24	Andaman and Nicobar Islands	NICOBARS	2002	Whole Year	Cashewnut	719	208
25	Andaman and Nicobar Islands	NICOBARS	2002	Whole Year	Coconut	18240	67490000
26	Andaman and Nicobar Islands	NICOBARS	2002	Whole Year	Dry chillies	413	28.8

Fig 5.1 (c) Raw District-wise Yield Dataset

5.2 Data Preprocessing:

After analyzing and visualizing the data, the next step is preprocessing. Data preprocessing is an important step as it helps in cleaning the data and making it suitable for use in machine learning algorithms. Most of the focus in preprocessing is to remove any outliers or erroneous data, as well as handling any missing values.

Missing data can be dealt with in two ways. The first method is to simply remove the entire row which contains the missing or erroneous value. While this an easy to execute method, it is better to use only on large datasets. Using this method on small datasets can reduce the dataset size too much, especially if there are a lot of missing values. This can severely affect the accuracy of the result. Since ours is a relatively small dataset, we will not be using this method.

The dataset that we used had values that were in string format so we had to transform and encode the into integer valued so as to pass as an input to the SVM. First we converted the data into pandas categorical data and then generated codes for crops and fertilizer respectively we than appended these and created separated datasets.

Step1: first we read the data from the csv file using below code and some functions

```
import pandas as pd
```

```
# Reading the data

crop_data_path = '../Data-raw/cpdata.csv'
fertilizer_data_path = '../Data-raw/Fertilizer.csv'

crop = pd.read_csv(crop_data_path)
fert = pd.read_csv(fertilizer_data_path)
```

Fig 5.2 (a) Data Preprocessing syntax 1

Step2: Then we make some changes in fertilizer data using below code and some functions for labelling the crops

```
# Function for lowering the cases
def change_case(i):
    i = i.replace(" ", "")
    i = i.lower()
    return i
```

```
fert['Crop'] = fert['Crop'].apply(change_case)
crop['label'] = crop['label'].apply(change_case)
```

```
#make some changes in ferttilizer dataset

fert['Crop'] = fert['Crop'].replace('mungbeans','mungbean')
fert['Crop'] = fert['Crop'].replace('lentils(masoordal)','lentil')
fert['Crop'] = fert['Crop'].replace('pigeonpeas(toordal)','pigeonpeas')
fert['Crop'] = fert['Crop'].replace('mothbean(matki)','mothbeans')
fert['Crop'] = fert['Crop'].replace('chickpeas(channa)','chickpea')
```

Fig 5.2 (b) Data Preprocessing syntax 2

Step3: Then we are using extract labels on crop to get all the data related to those labels

```
# using extract labesl on crop to get all the data related to those labels
new_crop = pd.DataFrame(columns = crop.columns)
new_fert = pd.DataFrame(columns = fert.columns)
```

```
for label in extract_labels:
    new_crop = new_crop.append(crop[crop['label'] == label])
```

```
for label in extract_labels:
    new_fert = new_fert.append(fert[fert['Crop'] == label].iloc[0])
```

Fig 5.2 (c) Data Preprocessing syntax 3

Step4: Output of data processing

```
new_crop
```

	temperature	humidity	ph	rainfall	label
0	20.879744	82.002744	6.502985	202.935536	rice
1	21.770462	80.319644	7.038096	226.655537	rice
2	23.004459	82.320763	7.840207	263.964248	rice
3	26.491096	80.158363	6.980401	242.864034	rice
4	20.130175	81.604873	7.628473	262.717340	rice
...	...	...	...	...	...
895	26.774637	66.413269	6.780064	177.774507	coffee
896	27.417112	56.636362	6.086922	127.924610	coffee
897	24.131797	67.225123	6.362608	173.322839	coffee
898	26.272418	52.127394	6.758793	127.175293	coffee
899	23.603016	60.396475	6.779833	140.937041	coffee

2200 rows × 5 columns

Fig 5.2 (d1) Data Preprocessing Output 1

```
new_fert
```

	Crop	N	P	K	pH
0	rice	80	40	40	5.5
3	maize	80	40	20	5.5
5	chickpea	40	60	80	5.5
12	kidneybeans	20	60	20	5.5
13	pigeonpeas	20	60	20	5.5
14	mothbeans	20	40	20	5.5
15	mungbean	20	40	20	5.5
18	blackgram	40	60	20	5.0
24	lentil	20	60	20	5.5
60	pomegranate	20	10	40	5.5
61	banana	100	75	50	6.5
62	mango	20	20	30	5.0
63	grapes	20	125	200	4.0
66	watermelon	100	10	50	5.5
67	muskmelon	100	10	50	5.5
69	apple	20	125	200	6.5
74	orange	20	10	10	4.0
75	papaya	50	50	50	6.0
88	coconut	20	10	30	5.0
93	cotton	120	40	20	5.5
94	jute	80	40	40	5.5

Fig 5.2 (d2) Data Preprocessing Output 1

Step5: Then save the files using below code

```
new_crop.to_csv('../Data-raw/MergeFileCrop.csv')
new_fert.to_csv('../Data-raw/FertilizerData.csv')
```

Fig 5.2 (e) Saving OutPut

5.3 Machine Learning Models:

These machine learning models used in crop and fertilizer recommendation.

5.3.1 Logistic Regression Model:

Logistic Regression is used to solve the classification problems in machine learning. They are similar to linear regression but used to predict the categorical variables. It can predict the output in either Yes or No, 0 or 1, True or False, etc. However, rather than giving the exact values, it provides the probabilistic values between 0 & 1.

The syntax we used in our project shown below:-

```
from sklearn.linear_model import LogisticRegression

LogReg = LogisticRegression(random_state=2)

LogReg.fit(Xtrain,Ytrain)

predicted_values = LogReg.predict(Xtest)

x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('Logistic Regression')
print("Logistic Regression's Accuracy is: ", x)

print(classification_report(Ytest,predicted_values))
```

Fig 5.3 (a) Logistic Regression Model Code Snippet

5.3.2 Support Vector Machine Model:

Support vector machine or SVM is the popular machine learning algorithm, which is widely used for classification and regression tasks. However, specifically, it is used to solve classification problems.

The main aim of SVM is to find the best decision boundaries in an N-dimensional space, which can segregate data points into classes, and the best decision boundary is known as Hyperplane. SVM selects the extreme vector to find the hyperplane, and these vectors are known as support vectors.

The syntax we used in our project is shown below:-

```
from sklearn.svm import SVC
# data normalization with sklearn
from sklearn.preprocessing import MinMaxScaler
# fit scaler on training data
norm = MinMaxScaler().fit(Xtrain)
X_train_norm = norm.transform(Xtrain)
# transform testing dataabs
X_test_norm = norm.transform(Xtest)
SVM = SVC(kernel='poly', degree=3, C=1)
SVM.fit(X_train_norm,Ytrain)
predicted_values = SVM.predict(X_test_norm)
x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('SVM')
print("SVM's Accuracy is: ", x)

print(classification_report(Ytest,predicted_values))
```

Fig 5.3 (b) SVM Model Code Snippet

5.3.3 Random Forest Model:

Random Forest is the ensemble learning method, which consists of a large number of decision trees. Each decision tree in a random forest predicts an outcome, and the prediction with the majority of votes is considered as the outcome.

A random forest model can be used for both regression and classification problems.

For the classification task, the outcome of the random forest is taken from the majority of votes. Whereas in the regression task, the outcome is taken from the mean or average of the predictions generated by each tree.

The syntax we used in our system is shown below:-

```
from sklearn.ensemble import RandomForestClassifier

RF = RandomForestClassifier(n_estimators=20, random_state=0)
RF.fit(Xtrain,Ytrain)

predicted_values = RF.predict(Xtest)

x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('RF')
print("RF's Accuracy is: ", x)

print(classification_report(Ytest,predicted_values))
```

Fig 5.3 (c) Random Forest Model Code Snippet

5.3.4 Gaussian Naive Bayes Model:

Naïve Bayes is a probabilistic machine learning algorithm used for many classification functions and is based on the Bayes theorem. Gaussian Naïve Bayes is the extension of naïve Bayes. While other functions are used to estimate data distribution, Gaussian or normal distribution is the simplest to implement as you will need to calculate the mean and standard deviation for the training data.

The syntax we used in our system is shown below:-

```python
from sklearn.naive_bayes import GaussianNB

NaiveBayes = GaussianNB()

NaiveBayes.fit(Xtrain,Ytrain)

predicted_values = NaiveBayes.predict(Xtest)
x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('Naive Bayes')
print("Naive Bayes's Accuracy is: ", x)

print(classification_report(Ytest,predicted_values))
```

Fig 5.3 (d) Gaussian Naïve Bayes Model Code Snippet

5.3.5 Decision Tree Model:

Decision trees are the popular machine learning models that can be used for both regression and classification problems. A decision tree uses a tree-like structure of decisions along with their possible consequences and outcomes. In this, each internal node is used to represent a test on an attribute; each branch is used to represent the outcome of the test. The more nodes a decision tree has, the more accurate the result will be.

The syntax we used in our project shown below:-

```python
from sklearn.tree import DecisionTreeClassifier

DecisionTree = DecisionTreeClassifier(criterion="entropy",random_state=2,max_depth=5)

DecisionTree.fit(Xtrain,Ytrain)

predicted_values = DecisionTree.predict(Xtest)
x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('Decision Tree')
print("DecisionTrees's Accuracy is: ", x*100)

print(classification_report(Ytest,predicted_values))
```
Python

Fig 5.3 (e) Decision Tree Model Code Snippet

5.4 Disease Prediction System:

In our system we use cnn (convolutional neural network) for in object detection and image classification. An ML and Deep Learning based Web app has been developed to query the results of machine learning analysis. The app is compatible with Android OS version 7. The pages were written in Java language. The app has a simple, easy to use interface requiring only few taps to retrieve desired results. Just only giving the location and area of the field the Android app gives the name of right crop to grown there. By accessing the user entered details, app will queries the machine learning analysis. Using the location, API will give out details of weather data. The retrieved weather data get acquired by machine learning classifier to predict the crop and calculate the yield. The output is then fetched by the server to portray the result in application. The main activities in the application were account creation, detail entry and results fetch. The account creation helps the user to actively interact with application interface. The user fill the field in home page to move onto the results activity. The retrieved data passed to machine learning model and crop name is predicted.

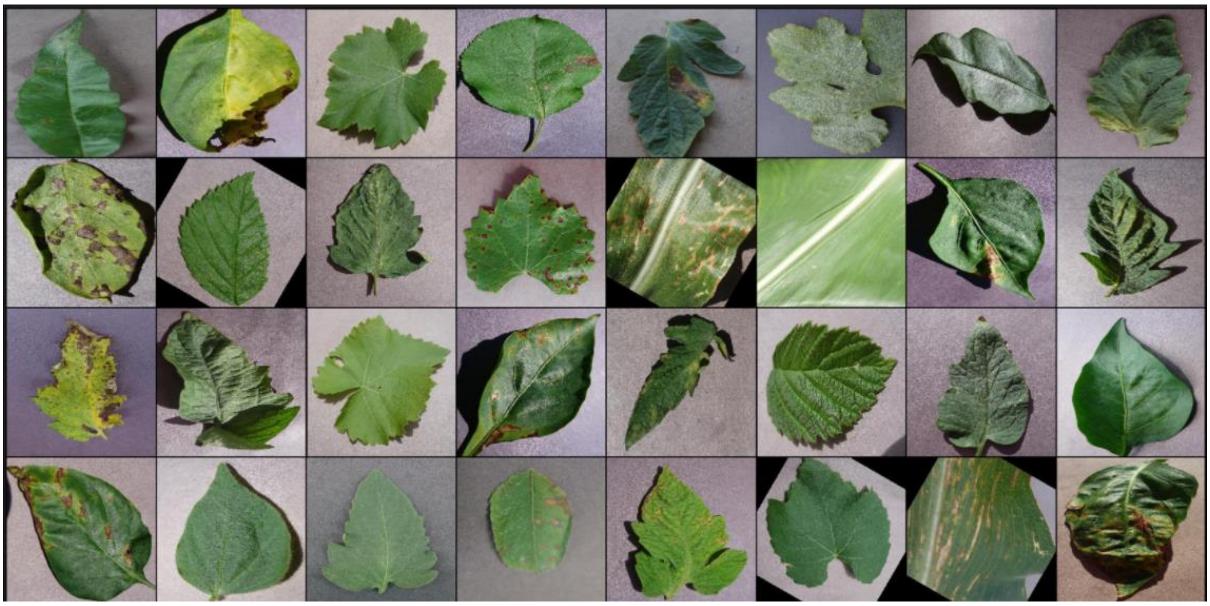

Fig 5.4 (a) A Sample Of Leaf Images From Dataset

Due to its extremely high speed, it is also suitable for consumer applications on smartphones. To develop an accurate image classifier for diagnosing plant diseases, a large, validated dataset containing images of healthy plants with the disease was needed. Until recently, such datasets did not exist and smaller datasets were not freely available. To address this issue, the Plant Village dataset has begun collecting images of tens of thousands of healthy and diseased crops. making them open and freely available. Here we report 26 disease classifications of 14 crop species from 54,306 images using a convolutional neural network approach. Model performance is measured by the ability to predict the correct pair of plant diseases from 38 possible classes. The best performing model achieves an average F1 value of (99.35% overall accuracy), demonstrating the technical realizability of our approach. Our results are the first step towards a smartphone-based plant disease diagnostic system.

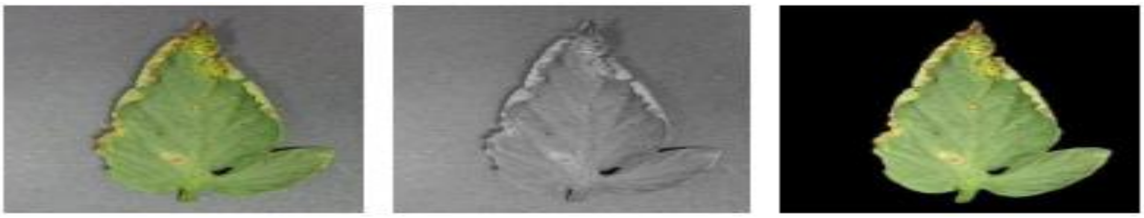

Fig 5.4 (b) Sample Images From The Three Different Versions Of The Plant Village Dataset

A cross all our experimental the overall accuracy obtained with the Plant Village dataset varied from 85.53% (AlexNet :: Training from Scratch :: GrayScale :: 8020). (For GoogLe Net :: Transfer Learning :: Color :: 8020), showing a strong prospect for a deep learning approach to similar prediction problems. Table 1 shows the average F1 score, average fit, average recall, and overall accuracy for all experimental configurations.

All experimental configurations are each run with a total of 30 epochs, and in most cases converge after the first step down of the learning rate. To address the problem of overfitting, changing the ratio of test sets to training sets, even in the extreme case where only 20% of the data is trained and the trained model is tested with the remaining 80% of the data. Observe the model. For GoogLeNet :: Transfer Learning :: Color :: 2080, we achieve an overall accuracy of 98.21% (average F1 score 0.9820). As expected, further increasing the ratio of test set to train set will reduce the overall performance of both AlexNet and GoogLeNet but the performance degradation is expected in the actual model.

Not as big as it gets. Over-adapted. also shows that there is no difference between validation loss and training loss, confirming that overfitting does not contribute to the results obtained in all experiments. Within the AlexNet and GoogLeNet architectures, GoogLeNet consistently outperforms AlexNet and based on training methods, Transfer Learning always provides better results. Both were expected. The three versions of the dataset (color, grayscale, and segmentation) show characteristic performance variability across all experiments, keeping the rest of the experimental composition constant. The model performs best with the color version of the dataset. When designing the experiment, I was worried that the neural network might only be learning to recognize the inherent bias associated with lighting conditions, methods, and equipment.

We therefore experimented with the grayscaled version of the same dataset to test the model`s adaptability in the absence of color information, and its ability to learn higher level structural patterns typical to particular crops and diseases.

As expected, the performance did decrease when compared to the experiments on the colored version of the dataset, but even in the case of the worst performance, the observed mean F1 score was 0.8524 (overall accuracy of 85.53%).

The segmented versions of the whole dataset was also prepared to investigate the role of the background of the images in overall performance, and as shown in Figure 4(e), the performance of the model using segmented images is consistently better than that of the model using grayscaled images, but slightly lower than that of the model using the colored version of the images.

Finally, while these approaches yield excellent results on the Plant Village dataset which was collected in a controlled environment, we also assessed the model`s performance on images sampled from trusted online sources such as academic agriculture extension services. Large numbers of such images are not available.

We combined automatic downloads from Bing image search with a visual validation step by one of us (MS) to get a small validated dataset of 121 images (see supplement for more information). please). Process description). GoogLeNet: Segmented: TransferLearning: Using a model trained in 8020, 31.40% overall by successfully predicting the correct class name (ie, plant and disease information) from 38 possible class names. Achieved accuracy.

You can see that the average accuracy achieved by the random classifier is only 2.63%. Specifying the section to which each image belongs will improve the accuracy to 47.93%. In all images, the correct class was included in the top 5 predictions with a 52.89% chance.

The power of convolutional neural networks in object detection and image classification has made great strides in recent years. Previously, the traditional approach to image classification tasks was to manually build features such as SIFT, HoG, SURF and use some form of learning algorithm in these functional spaces. Was based on. This makes the performance of all these approaches highly dependent on the underlying predefined characteristics. Feature engineering itself is a complex and time-consuming process that has to be revisited each time the problem at hand or the associated dataset changes significantly. This problem has occurred in all conventional attempts to detect plant diseases using computer vision. This was due to the heavy reliance on handmade features, image enhancement techniques, and various other complex, labor-intensive methods. AlexNet was a few years ago when end-to-end supervised training using a deep convolutional neural network architecture was a practical option for image classification problems involving a large number of classes.

It was first shown to be superior to traditional approaches with handdesigned features. By a significant lead in the standard benchmark. The lack of a labor-intensive feature engineering phase and the generalizability of the solution make it a very promising candidate for a practical and scalable approach to computational plant disease inference.

Fig 5.4 (c) Example Image Of A Leaf Suffering From Apple Cedar Rust

Using a deep convolutional neural network architecture, we modeled on plant leaf images with the aim of classifying crop species and both the presence and identity of diseases on images that the model has never seen before. I trained. This goal was achieved as shown with the highest levels of accuracy within the Plant Village dataset of 54,306 images, including 38 classes of 14 plant species and 26 diseases.While this is a clear path to global plant disease diagnosis using smartphones, there are some limitations that need to be addressed in future efforts at this stage. First, when tested on a series of images taken under different conditions than the images used for training, the accuracy of the model drops significantly to 31.4%.

It is important to note that this accuracy is much higher than the accuracy based on 38 classes of random selection (2.6%), but to improve accuracy requires a more diverse set of training data. Current results show that more (and more variable) data alone is sufficient to significantly improve accuracy, and data collection efforts are underway. The second limitation is that we are currently limited to classifying individual leaves ostensibly with a uniform background.

These are simple terms, but in the real world, applications need to be able to classify images of the disease when the disease appears directly on the plant. In fact, many diseases appear on different parts of the plant, not just on the leaves (or at all). Therefore, new image acquisition efforts should try to capture images from many different perspectives, ideally from the most realistic environment possible. At the same time, tackled the challenge with 38 classes, including both plant species and diseases.

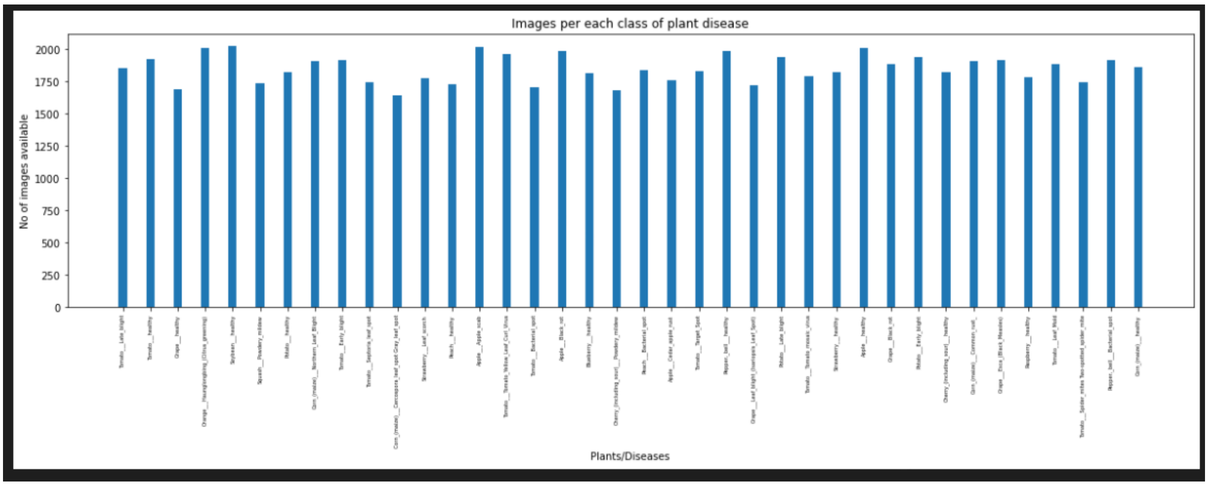

Fig 5.4 (d) Images Per Each Class Of Plant Disease

Chapter-6

TESTING

6.1 Testing Methodologies:

The program comprises of several algorithms which are tested individually for the accuracy. We check for the correctness of the program as a whole and how it performs.

6.1.1 Unit Testing:

The various modules can be individually run from a command line and tested for correctness. The tester can pass various values, to check the answer returned and verify it with the values given to him/her. The other work around is to write a script, and run all the tests using it and write the output to a log file and using that to verify the results. We tested each of the algorithms individually and made changes in preprocessing accordingly to increase the accuracy.

6.1.2 System Testing:

System Testing is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the systems compliance with the specified requirements. System Testing is the testing of a complete and fully integrated software product. And White Box Testing. System test falls under the black box testing category of software testing. Different Types of System Testing:

Usability Testing – Usability Testing mainly focuses on the users ease to use the application, flexibility in handling controls and ability of the system to meet its objectives.

Load Testing – Load Testing is necessary to know that a software solution will perform under real-life loads.

Regression Testing- - Regression Testing involves testing done to make sure none of the changes made over the course of the development process have caused new bugs.

Recovery Testing – Recovery testing is done to demonstrate a software solution is reliable, trustworthy and can successfully recoup from possible crashes.

Migration Testing – Migration testing is done to ensure that the software can be moved from older system infrastructures to current system infrastructures without any issues.

6.1.3 Quality Assurance:

Quality Assurance is popularly known as QA Testing, is defined as an activity to ensure that an organization is providing the best possible product or service to customers. QA focuses on improving the processes to deliver Quality Products to the customer. An organization has to ensure, that processes are efficient and effective as per the quality standards defined for software products.

6.1.4 Functional Test:

Functional Testing is also known as functional completeness testing, Functional Test- ing involves trying to think of any possible missing functions. As chat-bot evolves into new application areas, functional testing of essential chatbot components. Functional testing evaluates use-case scenarios and related business processes, such as the behavior of smart contracts.

Chapter-7

RESULT AND CONCLUSION

7.1 Result:

For the purposes of crop and fertilizer recommendation in this project we used SVM, LOGISTIC REGRESSION, GAUSSIAN NAÏVE BAYES, RANDOM FOREST, DECISION TREE algorithms.

Our overall system is divided into two modules:

- Crop and Fertilizer recommendation
- Disease Prediction

7.1.1 Crop And Fertilizer Recommendation:

7.1.1.1 SVM(Support Vector Machine) Result:

	precision	recall	f1-score	support
apple	1.00	1.00	1.00	13
banana	1.00	1.00	1.00	17
blackgram	1.00	1.00	1.00	16
chickpea	1.00	1.00	1.00	21
coconut	1.00	1.00	1.00	21
coffee	1.00	0.95	0.98	22
cotton	0.95	1.00	0.98	20
grapes	1.00	1.00	1.00	18
jute	0.83	0.89	0.86	28
kidneybeans	1.00	1.00	1.00	14
lentil	1.00	1.00	1.00	23
maize	1.00	0.95	0.98	21
mango	1.00	1.00	1.00	26
mothbeans	1.00	1.00	1.00	19
mungbean	1.00	1.00	1.00	24
muskmelon	1.00	1.00	1.00	23
orange	1.00	1.00	1.00	29
papaya	1.00	1.00	1.00	19
pigeonpeas	1.00	1.00	1.00	18
pomegranate	1.00	1.00	1.00	17
rice	0.80	0.75	0.77	16
watermelon	1.00	1.00	1.00	15
...				
accuracy			0.98	440
macro avg	0.98	0.98	0.98	440
weighted avg	0.98	0.98	0.98	440

Fig 7.1.1 (a) SVM Result

7.1.1.2 Logistic Regression result:

	precision	recall	f1-score	support
apple	1.00	1.00	1.00	13
banana	1.00	1.00	1.00	17
blackgram	0.86	0.75	0.80	16
chickpea	1.00	1.00	1.00	21
coconut	1.00	1.00	1.00	21
coffee	1.00	1.00	1.00	22
cotton	0.86	0.90	0.88	20
grapes	1.00	1.00	1.00	18
jute	0.84	0.93	0.88	28
kidneybeans	1.00	1.00	1.00	14
lentil	0.88	1.00	0.94	23
maize	0.90	0.86	0.88	21
mango	0.96	1.00	0.98	26
mothbeans	0.84	0.84	0.84	19
mungbean	1.00	0.96	0.98	24
muskmelon	1.00	1.00	1.00	23
orange	1.00	1.00	1.00	29
papaya	1.00	0.95	0.97	19
pigeonpeas	1.00	1.00	1.00	18
pomegranate	1.00	1.00	1.00	17
rice	0.85	0.69	0.76	16
watermelon	1.00	1.00	1.00	15
...				
accuracy			0.95	440
macro avg	0.95	0.95	0.95	440
weighted avg	0.95	0.95	0.95	440

Fig 7.1.1 (b) Logistic Regression Result

7.1.1.3 Decision Tree Result:

	precision	recall	f1-score	support
apple	1.00	1.00	1.00	13
banana	1.00	1.00	1.00	17
blackgram	0.59	1.00	0.74	16
chickpea	1.00	1.00	1.00	21
coconut	0.91	1.00	0.95	21
coffee	1.00	1.00	1.00	22
cotton	1.00	1.00	1.00	20
grapes	1.00	1.00	1.00	18
jute	0.74	0.93	0.83	28
kidneybeans	0.00	0.00	0.00	14
lentil	0.68	1.00	0.81	23
maize	1.00	1.00	1.00	21
mango	1.00	1.00	1.00	26
mothbeans	0.00	0.00	0.00	19
mungbean	1.00	1.00	1.00	24
muskmelon	1.00	1.00	1.00	23
orange	1.00	1.00	1.00	29
papaya	1.00	0.84	0.91	19
pigeonpeas	0.62	1.00	0.77	18
pomegranate	1.00	1.00	1.00	17
rice	1.00	0.62	0.77	16
watermelon	1.00	1.00	1.00	15
...				
accuracy			0.90	440
macro avg	0.84	0.88	0.85	440
weighted avg	0.86	0.90	0.87	440

Fig 7.1.1 (c) Decision Tree Result

7.1.1.4 Gaussian Naïve Bayes Result:

	precision	recall	f1-score	support
apple	1.00	1.00	1.00	13
banana	1.00	1.00	1.00	17
blackgram	1.00	1.00	1.00	16
chickpea	1.00	1.00	1.00	21
coconut	1.00	1.00	1.00	21
coffee	1.00	1.00	1.00	22
cotton	1.00	1.00	1.00	20
grapes	1.00	1.00	1.00	18
jute	0.88	1.00	0.93	28
kidneybeans	1.00	1.00	1.00	14
lentil	1.00	1.00	1.00	23
maize	1.00	1.00	1.00	21
mango	1.00	1.00	1.00	26
mothbeans	1.00	1.00	1.00	19
mungbean	1.00	1.00	1.00	24
muskmelon	1.00	1.00	1.00	23
orange	1.00	1.00	1.00	29
papaya	1.00	1.00	1.00	19
pigeonpeas	1.00	1.00	1.00	18
pomegranate	1.00	1.00	1.00	17
rice	1.00	0.75	0.86	16
watermelon	1.00	1.00	1.00	15
...				
accuracy			0.99	440
macro avg	0.99	0.99	0.99	440
weighted avg	0.99	0.99	0.99	440

Fig 7.1.1 (d) Gaussian Naïve Bayes Result

7.1.1.5 Random Forest Result:

	precision	recall	f1-score	support
apple	1.00	1.00	1.00	13
banana	1.00	1.00	1.00	17
blackgram	0.94	1.00	0.97	16
chickpea	1.00	1.00	1.00	21
coconut	1.00	1.00	1.00	21
coffee	1.00	1.00	1.00	22
cotton	1.00	1.00	1.00	20
grapes	1.00	1.00	1.00	18
jute	0.90	1.00	0.95	28
kidneybeans	1.00	1.00	1.00	14
lentil	1.00	1.00	1.00	23
maize	1.00	1.00	1.00	21
mango	1.00	1.00	1.00	26
mothbeans	1.00	0.95	0.97	19
mungbean	1.00	1.00	1.00	24
muskmelon	1.00	1.00	1.00	23
orange	1.00	1.00	1.00	29
papaya	1.00	1.00	1.00	19
pigeonpeas	1.00	1.00	1.00	18
pomegranate	1.00	1.00	1.00	17
rice	1.00	0.81	0.90	16
watermelon	1.00	1.00	1.00	15
...				
accuracy			0.99	440
macro avg	0.99	0.99	0.99	440
weighted avg	0.99	0.99	0.99	440

Fig 7.1.1 (e) Random Forest Result

7.1.2 Disease prediction Result:

```
Output exceeds the size limit. Open the full output data in a text editor
['AppleCedarRust1.JPG',
 'AppleCedarRust2.JPG',
 'AppleCedarRust3.JPG',
 'AppleCedarRust4.JPG',
 'AppleScab1.JPG',
 'AppleScab2.JPG',
 'AppleScab3.JPG',
 'CornCommonRust1.JPG',
 'CornCommonRust2.JPG',
 'CornCommonRust3.JPG',
 'PotatoEarlyBlight1.JPG',
 'PotatoEarlyBlight2.JPG',
 'PotatoEarlyBlight3.JPG',
 'PotatoEarlyBlight4.JPG',
 'PotatoEarlyBlight5.JPG',
 'PotatoHealthy1.JPG',
 'PotatoHealthy2.JPG',
 'TomatoEarlyBlight1.JPG',
 'TomatoEarlyBlight2.JPG',
 'TomatoEarlyBlight3.JPG',
 'TomatoEarlyBlight4.JPG',
 'TomatoEarlyBlight5.JPG',
 'TomatoEarlyBlight6.JPG',
 'TomatoHealthy1.JPG',
 'TomatoHealthy2.JPG',
...
 'TomatoYellowCurlVirus2.JPG',
 'TomatoYellowCurlVirus3.JPG',
 'TomatoYellowCurlVirus4.JPG',
```

FIG 7.1.2 (a) Disease Prediction Output1

```
Output exceeds the size limit. Open the full output data in a text editor
Label: AppleCedarRust1.JPG , Predicted: Apple___Cedar_apple_rust
Label: AppleCedarRust2.JPG , Predicted: Apple___Cedar_apple_rust
Label: AppleCedarRust3.JPG , Predicted: Apple___Cedar_apple_rust
Label: AppleCedarRust4.JPG , Predicted: Apple___Cedar_apple_rust
Label: AppleScab1.JPG , Predicted: Apple___Apple_scab
Label: AppleScab2.JPG , Predicted: Apple___Apple_scab
Label: AppleScab3.JPG , Predicted: Apple___Apple_scab
Label: CornCommonRust1.JPG , Predicted: Corn_(maize)___Common_rust_
Label: CornCommonRust2.JPG , Predicted: Corn_(maize)___Common_rust_
Label: CornCommonRust3.JPG , Predicted: Corn_(maize)___Common_rust_
Label: PotatoEarlyBlight1.JPG , Predicted: Potato___Early_blight
Label: PotatoEarlyBlight2.JPG , Predicted: Potato___Early_blight
Label: PotatoEarlyBlight3.JPG , Predicted: Potato___Early_blight
Label: PotatoEarlyBlight4.JPG , Predicted: Potato___Early_blight
Label: PotatoEarlyBlight5.JPG , Predicted: Potato___Early_blight
Label: PotatoHealthy1.JPG , Predicted: Potato___healthy
Label: PotatoHealthy2.JPG , Predicted: Potato___healthy
Label: TomatoEarlyBlight1.JPG , Predicted: Tomato___Early_blight
Label: TomatoEarlyBlight2.JPG , Predicted: Tomato___Early_blight
Label: TomatoEarlyBlight3.JPG , Predicted: Tomato___Early_blight
Label: TomatoEarlyBlight4.JPG , Predicted: Tomato___Early_blight
Label: TomatoEarlyBlight5.JPG , Predicted: Tomato___Early_blight
Label: TomatoEarlyBlight6.JPG , Predicted: Tomato___Early_blight
Label: TomatoHealthy1.JPG , Predicted: Tomato___healthy
Label: TomatoHealthy2.JPG , Predicted: Tomato___healthy
...
Label: TomatoYellowCurlVirus3.JPG , Predicted: Tomato___Tomato_Yellow_Leaf_Curl_Virus
Label: TomatoYellowCurlVirus4.JPG , Predicted: Tomato___Tomato_Yellow_Leaf_Curl_Virus
Label: TomatoYellowCurlVirus5.JPG , Predicted: Tomato___Tomato_Yellow_Leaf_Curl_Virus
```

FIG 7.1.2 (b) Disease Prediction Outpu 2

7.2 CONCLUSION:

This system helps the farmer to choose the right crop by providing insights that ordinary farmers don’t keep track of thereby decreasing the chances of crop failure and increasing productivity. The system can be extended to the web and can be accessed by millions of farmers across the country.We could achieve an accuracy of 89.88 percent from the neural network and an accuracy of 88.26 percent from the linear regression model Further development is to integrate the crop recommendation system with another subsystem, yield predictor that would also provide the farmer an estimate of production if he plants the recommended crop.

Chapter-8

GUI PREVIEW

8.1 User interface:

User Interface is the front-end application view to which user interacts in order to use the software. User can manipulate and control the software as well as hardware by means of user interface.

Today, user interface is found at almost every place where digital technology exists, right from computers, mobile phones, cars, music players, airplanes, ships etc. User interface is part of software and is designed such a way that it is expected to provide the user insight of the software.

UI provides fundamental platform for human- computer interaction. UI can be graphical, text-based, audio-video based, depending upon the underlying hardware and software combination. UI can be hardware or software or a combination of both.

There are below some screenshots of the web application.

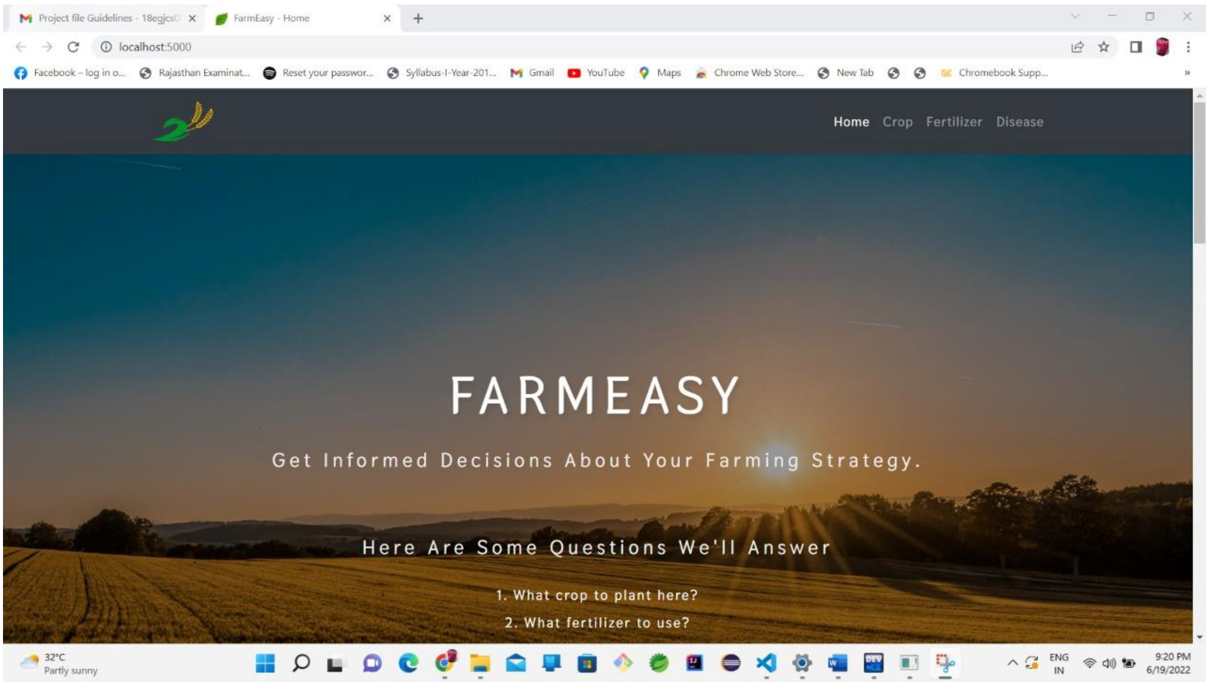

Fig 8.1 (a) Home Page

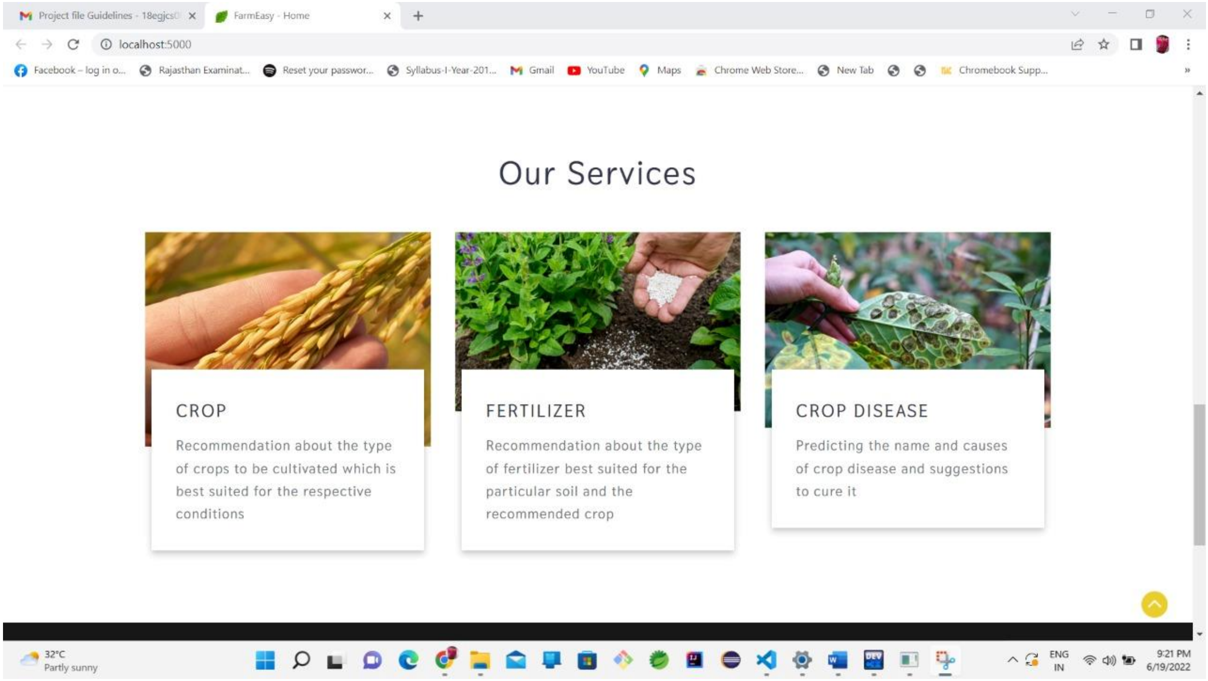

Fig 8.1 (b) Middle Home Page

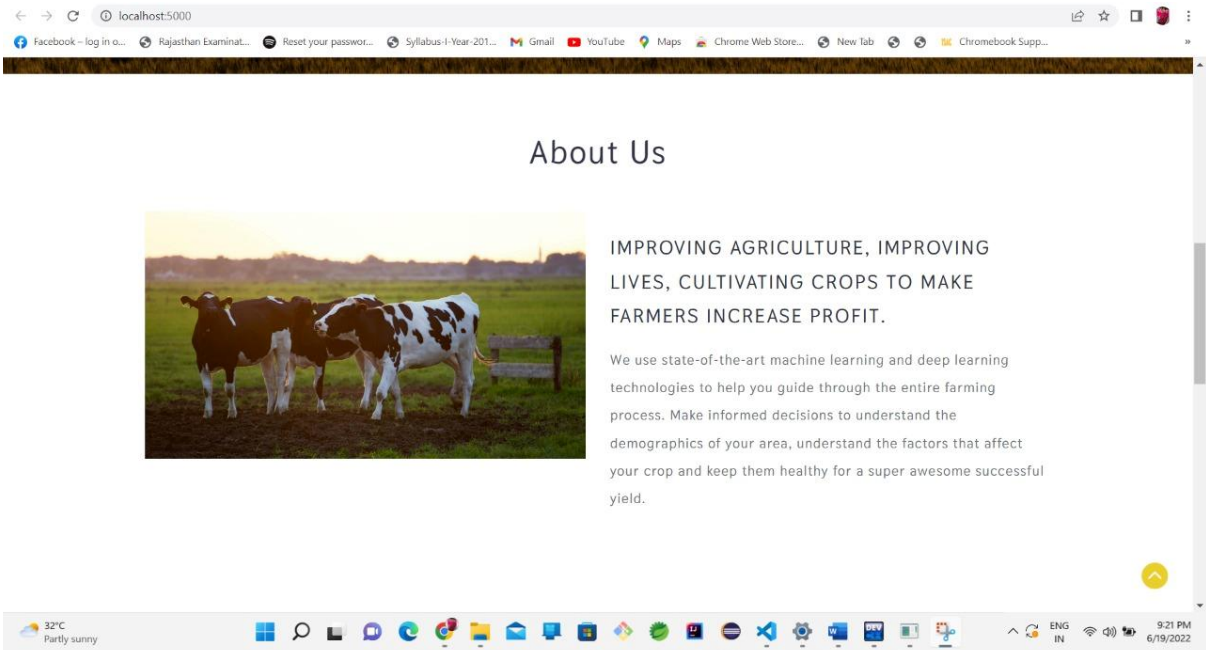

Fig 8.1 (c) About Us Page

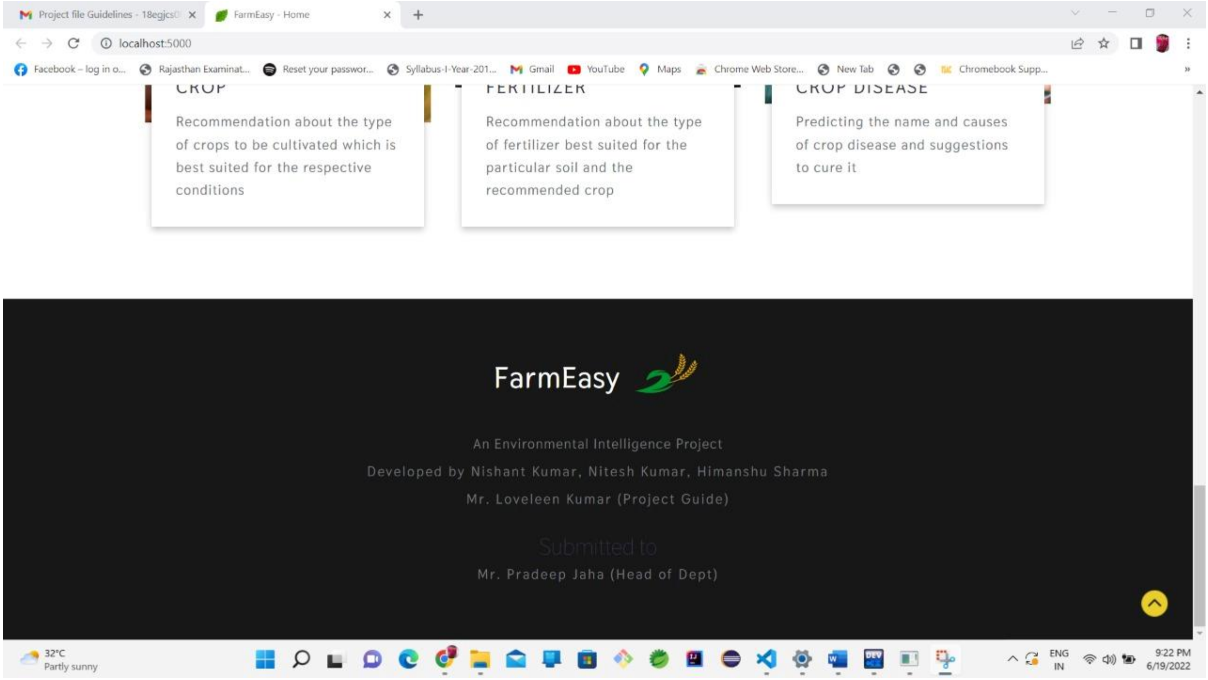

Fig 8.1 (d) Footer Section

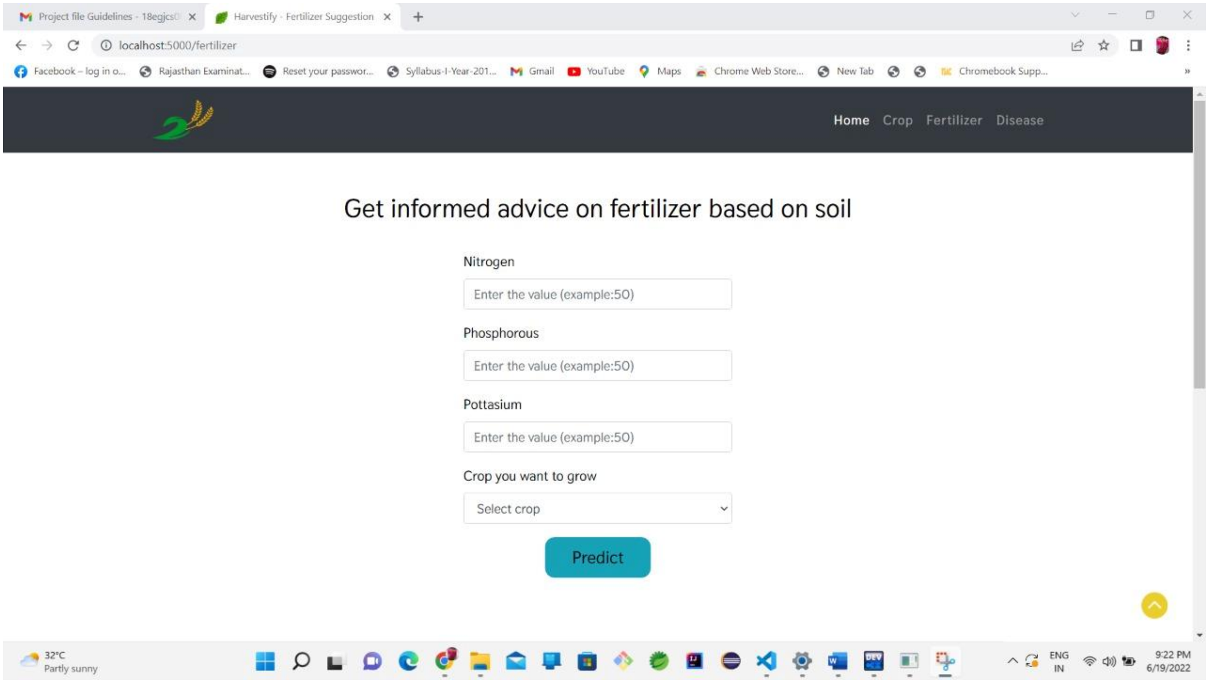

Fig 8.1 (e) Crop And Fertilizer Data Fill Up Service Page

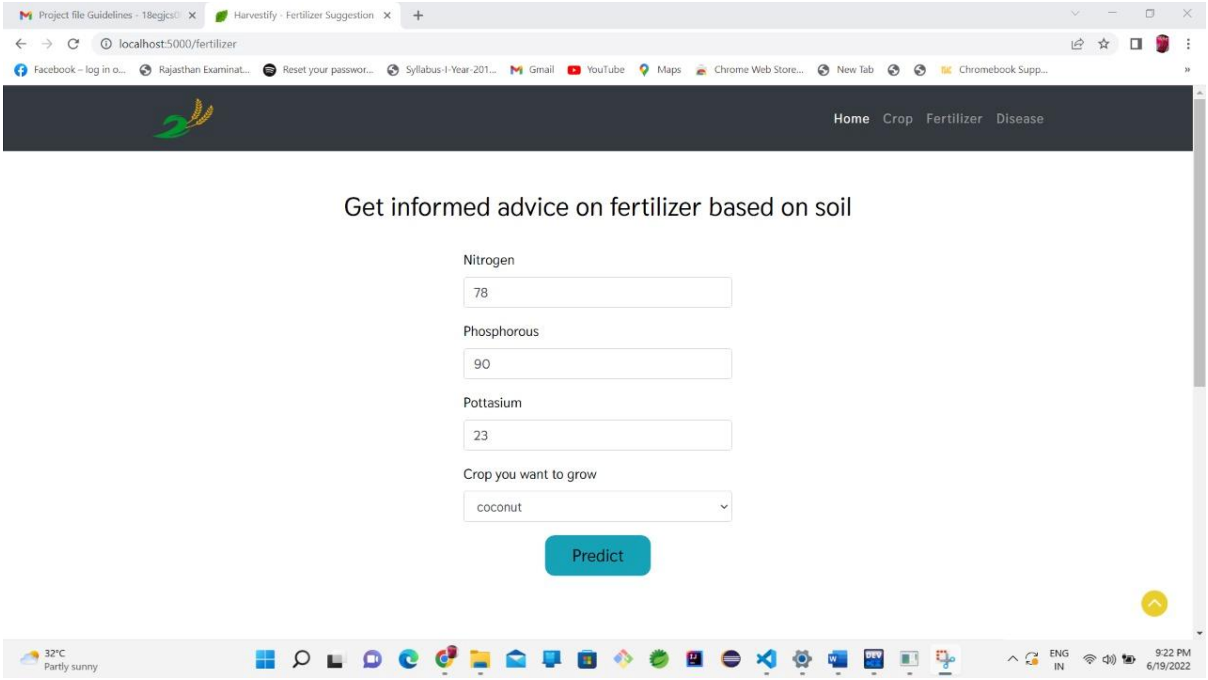

Fig 8.1 (f) Crop And Fertilizer Service Data Fill Page

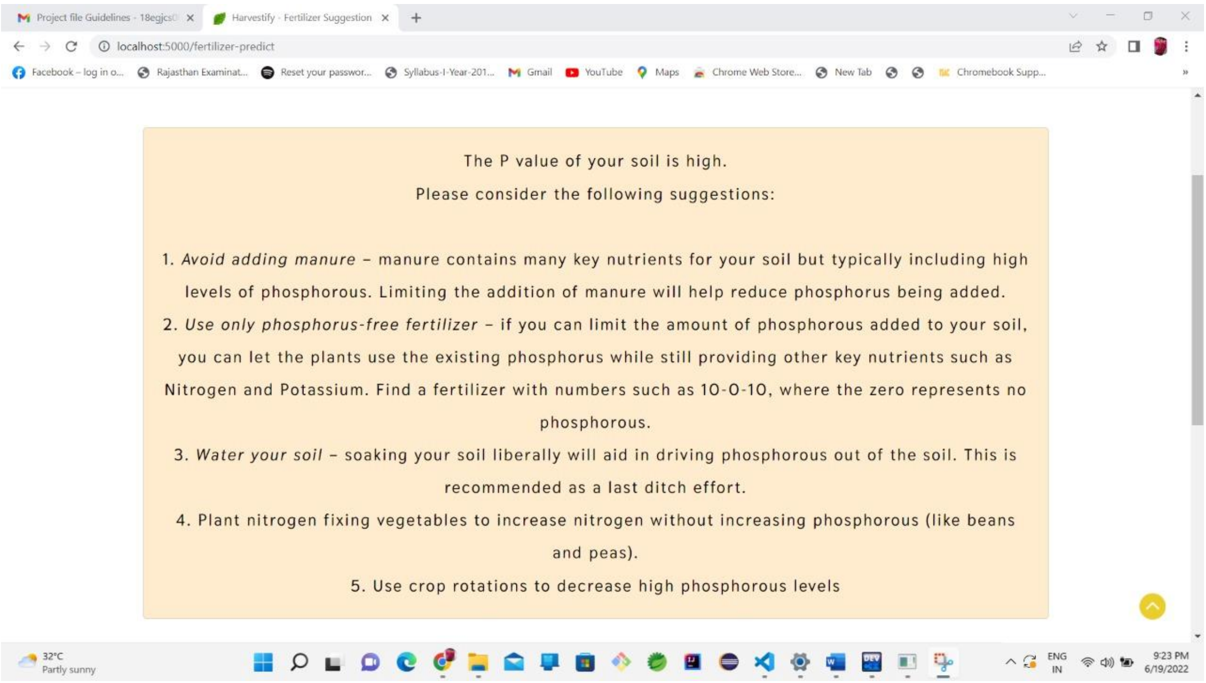

Fig 8.1 (g) Crop And Fertilizer Fill Up Data Result Page

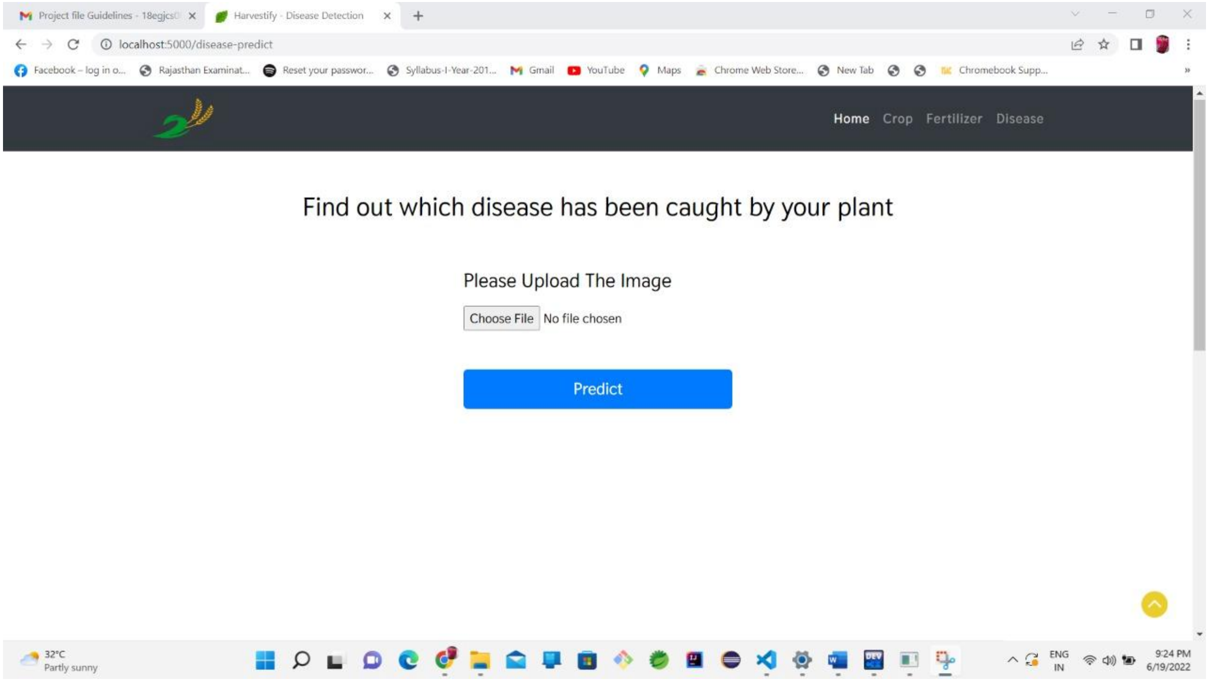

Fig 8.1 (h) Disease Prediction Service Page

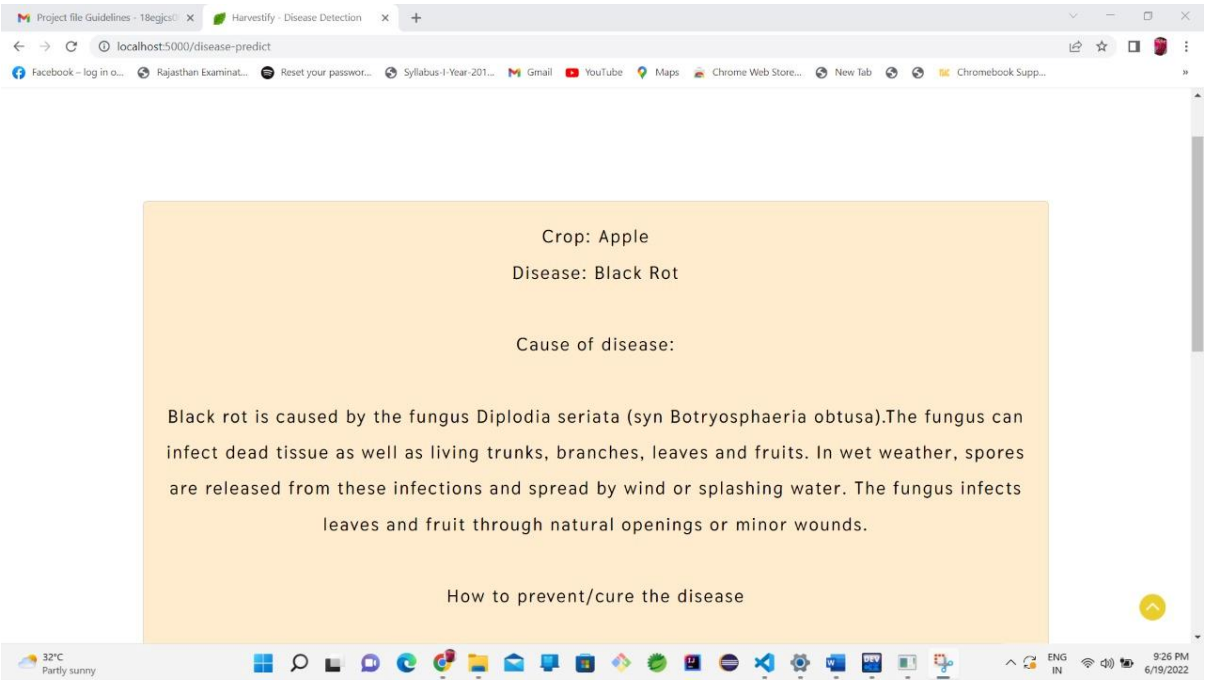

Fig 8.1 (i) Disease Prediction Result Page

REFERENCES:

1) Haedong Lee and Aekyung Moon, “Development of Yield Prediction System Based on Real-time Agricultural Meteorological Information”, 16th International Conference on Advanced Communication Technology, 2014
2) T.R. Lekhaa, “Efficient Crop Yield and Pesticide Prediction for Improving Agricultural Economy using Data Mining Techniques”, International Journal of Modern Trends in Engineering and Science (IJMTES), 2016, Volume 03, Issue 10
3) Jay Gholap, Anurag Ingole, Jayesh Gohil, Shailesh Gargade and Vahida Attar, “Soil Data Analysis Using Classification Techniques and Soil Attribute Prediction”, International Journal of Computer Science Issues, Volume 9, Issue 3
4) AgroConsultant: Intelligent Crop Recommendation System Using Machine Learning Algorithms. Zeel Doshi , Subhash Nadkarni , Rashi Agrawal, Prof. Neepa Shah.
5) Crop Recommendation System for Precision Agriculture S.Pudumalar*, E.Ramanujam*, R.Harine Rajashree, C.Kavya, T.Kiruthika, J.Nisha.
6) Tom M. Mitchell, Machine Learning, India Edition 2013, McGraw Hill Educa- tion.
7) https://data.gov.in/g
8) Indianwaterportal.org
9) Power.iarc.nasa.in
10) https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm
11) https://www.javatpoint.com/logistic-regression-in-machine-learning
12) https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm
13) https://towardsdatascience.com/gaussian-naive-bayes