

**TRIBHUVAN UNIVERSITY**

**Prime College**

**Nayabazar, Kathmandu, Nepal**

**A Project Report**

**On**

**SwoBali-Crop & Fertilizer Prediction system**

Submitted By

Trilochan Aryal

A Project Report Submitted in partial fulfilment of the requirement of **Bachelors in Computer Application**

**(BCA) 6th Semester** of Tribhuvan University, Nepal

April 2022

**(SwoBali)**

**[CSC - 404]**

A project report submitted for the partial fulfilment of the requirement for the degree of Bachelosr in Computer Application awarded by Tribhuvan University.

**Submitted By**

Trilochan Aryal

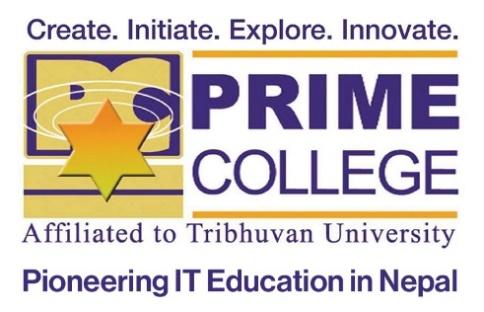
**Submitted To**

Prime College

Department of Computer science

Affiliated to Tribhuvan University

Khusibu, Nayabazar, Kathmandu



Ap­ril 2022

Date: ……………….

# SUPERVISOR’S RECOMMENDATION

It is my pleasure to recommend that a report on “SwoBali” has been prepared under my supervision by Trilochan Aryal in partial fulfilment of the requirement of the degree of Bachelors in Computer Application . Their report is satisfactory to process for the future evaluation.

…………………………………

Er. Rolisha Sthapit

IT Faculty

Prime College

­­

Date: …………….

# CERTIFICATE OF APPROVAL

The undersigned certify that he has read and recommended to the Department of Computer Science and Information Technology for acceptance of report entitled “**SwoBali**” submitted by **Trilochan Aryal** in partial fulfilment for the degree of Bachelors in Computer Application (BCA), Tribhuvan University.

|  |  |  |  |
| --- | --- | --- | --- |
| **…………………………** | |  | **……………………………** |
| **Er. Rolisha Sthapit** | |  | **Er. Rolisha Sthapit** |
| **Internal Supervisor** | |  | **Program Coordinator** |
|  | |  |  |
|  |
|  |
|  |

# ACKNOWLEDGEMENT

The completion of any project depends upon cooperation, coordination and combined efforts of several sources of knowledge. The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible. Their guidance and encouragement have crowned all our efforts with success.

If words are considered as a symbol of approval and token of appreciation, then let the words play the heralding role expressing our gratitude. We are grateful to the Institute of Computer Science and Information Technologies for including the provision for the Major Project. We know this project will prove to be very helpful for our career. We are also thankful to Prime College for providing us with the necessary resources and assistance for the completion of this project.

**We are extremely thankful and pay our gratitude to our Supervisor Er. Rolisha Sthapit,** for the invaluable guidance, support and inspiration for this project. We are also indebted to all of our faculty teachers and staff for their support. Finally, we would also like to thank our friends who helped, encouraged and shared with us their knowledge for completing this project.

Trilochan Aryal

# ABSTRACT

Nepal is among the oldest countries which is still practicing agriculture due to its economy which predominantly depends on it and therefore, it requires use of Precision Agriculture to increase the yield of the agricultural crops. 'SwoBali' is a crop and fertilizer prediction system that focuses on recommending the best suitable crop to the user according to the type of soil, its composition, pH; precipitation, and weather details which help in maximization of the crop. Moreover, SwoBali considers the soil and crop parameters to recommend the appropriate fertilizer and its amount to the user and help in the maximum yield of the crops without harming the environment and sustainability of the agricultural sector.

The main objective of the system is to predict suitable crop and fertilizer using Random Forest and Support Vector Machine algorithms using crop recommendation dataset and fertilizer prediction dataset that ensures great accuracy and reliability to the user. The tools used to develop this system includes Python programming language, Tkinter, Random Forest and Support Vector Machine algorithms which help in the accurate prediction of crop and fertilizer for maximum yield of the crop. The crop and fertilizer dataset are cleaned, wrangled and split into training (80%) and testing (20%) data. The inputs like Nitrogen, Phosphorous, Potassium, pH, rainfall, temperature and humidity are processed statistically and analytically to predict suitable crop and fertilizer for the land through two different algorithms: Random Forest and Support Vector Machine. Each of the algorithm's predicted results along with its accuracy, precision report and confusion matrix are displayed to the user.

*KEYWORDS: crop recommendation, fertilizer recommendation, random forest, support vector machine*

# TABLE OF CONTENT

[SUPERVISOR’S RECOMMENDATION iii](#_Toc69149896)

[CERTIFICATE OF APPROVAL iv](#_Toc69149897)

[ACKNOWLEDGEMENT v](#_Toc69149898)

[ABSTRACT vi](#_Toc69149899)

[TABLE OF CONTENT vii](#_Toc69149900)

[LIST OF FIGURES x](#_Toc69149901)

[LIST OF TABLES xi](#_Toc69149902)

[ABBREVIATION xii](#_Toc69149903)

[CHAPTER 1 INTRODUCTION 1](#_Toc69149904)

[1.1 Background 1](#_Toc69149905)

[1.2 Problem Definition 2](#_Toc69149906)

[1.3 Objectives 2](#_Toc69149907)

[1.4.1 Scope 3](#_Toc69149908)

[1.4.2 Limitation 3](#_Toc69149909)

[1.5 Report Organization 3](#_Toc69149910)

[CHAPTER 2 REQUIREMENT ANALYSIS AND FEASIBILITY ANALYSIS 4](#_Toc69149911)

[2.1 Literature Review 4](#_Toc69149912)

[2.1.1 Study of Previous Literature 4](#_Toc69149913)

[2.1.2 Study of existing system 5](#_Toc69149914)

[2.2 Requirement Collection Methods 6](#_Toc69149915)

[2.2.1 Sources of data: 6](#_Toc69149916)

[2.3 Requirement Specification 8](#_Toc69149917)

[2.3.1 Functional Requirement: 8](#_Toc69149918)

[2.3.2 Non-Functional Requirement: 9](#_Toc69149919)

[2.4 Feasibility Study 10](#_Toc69149920)

[2.4.1 Technical Feasibility: 10](#_Toc69149921)

[2.4.2 Operational Feasibility: 11](#_Toc69149922)

[2.4.3 Economic Feasibility: 11](#_Toc69149923)

[2.4.4 Schedule Feasibility: 11](#_Toc69149924)

[CHAPTER 3 SYSTEM DESIGN 13](#_Toc69149925)

[3.1 System Architecture and Overview 13](#_Toc69149926)

[3.2 Algorithm Used 14](#_Toc69149927)

[3.2.1 Support Vector Machine (SVM) 14](#_Toc69149928)

[3.2.2 Random Forest 15](#_Toc69149929)

[3.3 UML diagram 18](#_Toc69149930)

[3.3.1 Class Diagram 19](#_Toc69149931)

[3.3.2 Sequence Diagram 19](#_Toc69149932)

[3.3.3 Activity Diagram 21](#_Toc69149933)

[3.3.4 Component Diagram 22](#_Toc69149934)

[3.3.5 State Diagram 23](#_Toc69149935)

[3.3.6 Deployment Diagram 24](#_Toc69149936)

[CHAPTER 4 SYSTEM IMPLEMENTATION AND TESTING 25](#_Toc69149937)

[4.1 Implementation: 25](#_Toc69149938)

[4.2 Tools used: 28](#_Toc69149939)

[4.2.1 Front end tools: 28](#_Toc69149940)

[4.2.2 Back-end tools: 28](#_Toc69149941)

[4.3 Module description: 29](#_Toc69149942)

[4.4 Testing: 30](#_Toc69149943)

[4.4.1 Unit testing: 30](#_Toc69149944)

[4.4.2 Integration testing: 38](#_Toc69149945)

[4.4.3 System testing: 40](#_Toc69149946)

[CHAPTER 5 CONCLUSION AND FUTURE ENHANCEMENTS 41](#_Toc69149947)

[5.1 Conclusion: 41](#_Toc69149948)

[5.2 Future enhancements of the project: 41](#_Toc69149949)

[REFERENCES/ BIBLIOGRAPHY 43](#_Toc69149950)

# LIST OF FIGURES

[Figure 2.1 Use Case Diagram 8](#_heading=h.z337ya)

[Figure 2.2 Gantt Chart 1](#_heading=h.32hioqz)1

[Figure 3.1 System Architecture 1](#_heading=h.vx1227)2

[Figure 3.2 Original Dataset to Bootstrapped Dataset 1](#_heading=h.1v1yuxt)5

[Figure 3.3 Decision Tree from Bootstrapped Dataset 1](#_heading=h.4f1mdlm)6

[Figure 3.4 Multiple Decision Trees from Bootstrapped Dataset 1](#_heading=h.2u6wntf)6

[Figure 3.5 Different outputs from Trees 1](#_heading=h.19c6y18)7

[Figure 3.6 Class Diagram 1](#_heading=h.28h4qwu)8

[Figure 3.7 Sequence Diagram 1](#_heading=h.37m2jsg)9

[Figure 3.8 Activity Diagram](#_heading=h.46r0co2) 20

[Figure 3.9 Component Diagram 2](#_heading=h.2lwamvv)1

[Figure 3.10 State Diagram 2](#_heading=h.111kx3o)2

[Figure 3.11 Deployment Diagram 2](#_heading=h.3l18frh)3

[Figure 4.1 V-model 2](#_heading=h.2zbgiuw)5

[Figure 4.2 Confusion Matrix for Crop Recommendation Module from SVM 30](#_heading=h.4bvk7pj)

[Figure 4.3 Confusion Matrix for Crop Recommendation Module from RF 3](#_heading=h.2r0uhxc)2

[Figure 4.4 Confusion Matrix for Fertilizer Recommendation Module from SVM 3](#_heading=h.1664s55)4

[Figure 4.5 Confusion Matrix for Fertilizer Recommendation Module from RF 3](#_heading=h.3q5sasy)6

# LIST OF TABLES

[Table 4. 1:Table for Crop Prediction Using Support Vector Machine Algorithm 29](#_Toc69157833)

[Table 4. 2: Table for Support Vector Machine Crop Report 29](#_Toc69157834)

[Table 4. 3: Table for Crop Prediction Using Random Forest Algorithm 31](#_Toc69157835)

[Table 4. 4: Table for Random Forest Crop Report 31](#_Toc69157836)

[Table 4. 5: Table for Fertilizer Prediction Using Support Vector Machine Algorithm 33](#_Toc69157837)

[Table 4. 6: Table for Support Vector Machine Fertilizer Report 33](#_Toc69157838)

[Table 4. 7: Table for Fertilizer Prediction Using Random Forest Algorithm 35](#_Toc69157839)

[Table 4. 8: Table for Random Forest Fertilizer Report 35](#_Toc69157840)

[Table 4. 9: Table for Integration Testing of Crop Predictor 37](#_Toc69157841)

[Table 4. 10: Table for Integration Testing of Fertilizer Predictor 38](#_Toc69157842)

# ABBREVIATION

**API :** Application Programming Interface

**DFD :** Data Flow Diagram

**ERD :** Entity Relationship Diagram

**ICT :** Information and Communication Technologies

**IT :** Information Technology

**ML :** Machine Learning

**NPK :** Nitrogen Phosphorus Potassium

**pH :** Potential of Hydrogen

**THM :** Temperature Humidity Moisture

# CHAPTER 1

# INTRODUCTION

## **1.1 Background**

Nepal is among the oldest countries which is still practicing agriculture due to its economy which predominantly depends on it. The unpredictable changes in the environment, increase in population, and globalization all have led to the degradation of Nepal’s health in agriculture. The increase in population causes demands for more agro products and hence to remain relevant in the contemporary competitive environment, farmers to enhance the agricultural yield indiscriminately apply chemical fertilizers and such inappropriate doses of fertilizer application not only affects the crop but also creates a serious impact on environment and sustainability of agricultural sector when the maximum yield can just be easily achieved through the selection of suitable crop and the right amount of fertilizer. Many new technologies have tried to help regain the health of agriculture. One such technology is "precision agriculture", which helps farmers to achieve maximum yield by making use of the newest technology and farming methodologies.

Crop prediction systems are among one of the first and important methods in precision agriculture that gives farmers the advantage of efficient input, output, and better decisions regarding farming. We are well informed about how much a soil contributes in the betterment of a crop and it’s both quality and quantity that is why crop prediction system focuses on the type of soil, its composition and pH to predict the best suitable crop for that area which in term results to the maximum yield and thus lead the farmers of our country towards profit.

'SwoBali' is a crop prediction system that focuses on recommending the best suitable crop to the user according to the type of soil, its composition, pH; precipitation, and weather details which help in maximization of the crop. Moreover, SwoBali considers the soil and crop parameters to recommend the appropriate fertilizer and its amount to the user and help in the maximum yield of the crops without harming the environment and sustainability of the agricultural sector. In return, the maximization of the crop does not only help the farmers generate more income but also helps the government to minimize the import and increase the export in agroindustry. The system collects the input parameters for crop and fertilizer from the user and organizes them to statistically interpret and help generate appropriate and accurate conclusions. It also provides the user with the necessary information on the crops so that the application does not only limit its use to farmers (people knowledgeable about the crop and fertilizer) but can also be used in cities to encourage urban vegetation even in a small amount of land.

## **1.2 Problem Definition**

Research shows that Nepal imports 13 times more than it exports even though Nepal has a clear advantage on the agricultural industry due to its geography and the soil type. About 36 percent of Nepali children under the age of 5 years suffer from chronic malnutrition, or stunting, which causes debilitating effects such as blindness, brain damage, and infectious diseases, which can result in lifetime damage. In such a condition, increasing the yield of crops can help a farmer increase his revenue which can deliberately help the situation of Nepal as agriculture has been the highest priority for economic growth of Nepal. Farmers tend to spend a lot of money on seeds and fertilizers and yet cannot earn as much to compensate for the input. [1]

The project represents a direct solution for all the problems stated above. The application does not only benefit the farmers of rural areas but might as well be helpful for urban agriculture and home vegetation which would fairly decrease the amount imported on agroindustry.

## **1.3 Objectives**

The core objectives of this project are:

* + - To predict crop and fertilizer using random forest classifiers and support vector machines.
    - To help farmers in their decision making through predicting the best suitable crop and fertilizer.
    - To predict the required Nitrogen, phosphorus and potassium value for fertilizer.

**1.4 Scope and Limitation**

### **1.4.1 Scope**

* The application helps users predict the right crop for the land.
* The application also helps increase productivity of the soil by recommending suitable fertilizer for the crop.
* It can help increase productivity and revenue from agro-based-industry.
* SwoBali helps governments & organizations to make appropriate planning like storing, selling, fixing minimum support price, importing/exporting etc.

### **1.4.2 Limitation**

* The accuracy may not be completely reliable.
* The application wouldn’t function without internet connection.

## **1.5 Report Organization**

The report starts with the introduction of the proposed application, crop predictor, “SwoBali” along with the problem statement and objectives of the project. Here, we have introduced why the system is built and the conditions of why the system is reliable. It also provides objectives and Scope of the project and lists the possible measures that can be applied to solve them. The chapter 2 analyses the similar existing models and reviews them on the basis of our project along with requirement and feasibility analysis of those systems. It also includes both functional and non-functional requirements. Chapter 3 provides a detailed overview of the system design along with the various algorithms used in the project. Chapter 4 explains the tools used on the project’s front and back end and purpose of it. Testing of the system is also explained in this chapter. Chapter 5 discusses the conclusion of how the project is accomplished, its findings etc. We further discuss the recommendation of the future enhancements in the project and how it can be improved.

# CHAPTER 2

# REQUIREMENT ANALYSIS AND FEASIBILITY ANALYSIS

## **2.1 Literature Review**

### **2.1.1 Study of Previous Literature**

Agriculture is a crucial part of income for an agricultural-based country like Nepal. The idea of precision agriculture for the increase of soil's productivity and better yield of crops has been in the picture for a long time. Making use of technologies like machine learning and artificial neural networks, people now have been able to achieve a system that predicts the suitable crop for cultivation. In countries like India and Bangladesh, many similar methods are used to help implement precision agriculture.

**According to S.Pudumalan and E.Ramanujam** [2]using data mining techniques to incorporate precision agriculture increases the productivity of the soil and increases the yield of the field. The system makes use of the Voting technique and the conclusion of the recommendation was drawn through bagging. The authors start from the basics of precision agriculture and move towards developing a model that would support it. They help to recommend crop to the user using two methods: dataset collection and crop prediction using ensemble tests with comparative analysis of the classification algorithms to be used like: Random Tree, K-Nearest Neighbor and Naïve Baye.

**Miftahul Jannat Mokarrama and Mohammad Shamsul Arefin** [3] describeda recommendation system for farmers in Bangladesh. It contains all the information about the methods and system architecture. The proposed system in the paper is a semantic web-based architecture where the recommendation of the crop is subdivided into three sub problems: location detection, information processing, and recommendation generation and tends to recommend crops according to the location and previous year’s crop results.

**E. Manjula and S. Djodiltachoumy** [4] states information on crop selection, weather forecasting, and smart irrigation systems in India. The paper touches the two phases of crop prediction: training and testing. The author’s explain data clustering, data conversion and mining techniques to get respected predictions for improving the crop quality.

**Rahul katarya** [5] describes the different machine learning methods used for accelerating crop yield. In this paper they went through different artificial intelligence techniques such as machine learning algorithms, big data analysis for precision agriculture. They explained crop recommender systems using KNN, Ensemble-based Models, Neural networks.

### **2.1.2 Study of existing system**

The detailed study of similar systems in context of Nepal and other countries has been properly done here and again. Available applications on the stores and constant articles about the need of precision agriculture for the maximum yield of crops points towards the need of applications that would help farmers with the proper choice of crop and fertilizers according to the different soil and weather inputs. Unfortunately, even though the Nepal Government is quite aware of precision agriculture and its need, the applications provided by the agriculture sector of government includes only informative application which informs users on the different crops and seeds. Moreover, the agriculture applications of Nepal found in stores are just information-based applications. Other countries like Bangladesh whereas had similar applications which would help farmers predict the crops to use in the fields but to overcome the challenge of using smart phones the project texted the farmers the answers this prevented the application from having real time data access to the farmers.

**G. Ravichandran and R.S** [6] have developed an agricultural crop prediction using ANN for smartphones. The system takes various inputs as parameters, processes them and predicts the suitable crop for that land. The authors have made the proposed system as an android application in order for it to be accessible to common people easily. The proposed system also helps predict the productivity status of the desired crop and suggests fertilizers to improve the productivity.

**P.S Vijayabaskar and R. Sreemathi** [7] describes a model which tests soil fertility and suggests the crop which has to be planted depending upon the value obtained from he sensor. The proposed system also provides regional wise information about the crops.

## **2.2 Requirement Collection Methods**

### **2.2.1 Sources of data:**

**Dataset:**

The system uses two different datasets for two different modules: crop and fertilizer prediction.

**Crop dataset:**

The dataset used for crop prediction is ‘Crop Recommendation Dataset’ which was extracted from Kaggle [8] and it consists of 8 labels and 17608 tuples. The labels in the dataset are:

* N: N represents the value of nitrogen in the soil.
* P: P represents the value of phosphorus in the soil.
* K: K represents the value of potassium in the soil.
* Temperature: It represents the value of temperature of the place and has float data type.
* Humidity: It represents the amount of water vapor in the air and too has float data type.
* Ph: The ‘ph’ label represents pH(potential of Hydrogen) of the soil.
* Rainfall: The rainfall label represents the value of rainfall required for the certain crop.
* Label: label consists of the name of the crops. The column has 22 unique values of the crop which includes rice, maize, chickpea, kidney beans, pigeon peas, moth beans, mung bean, black gram, lentil, pomegranate, banana, mango, grapes, watermelon, muskmelon, apple, orange, papaya, coconut, cotton and jute.

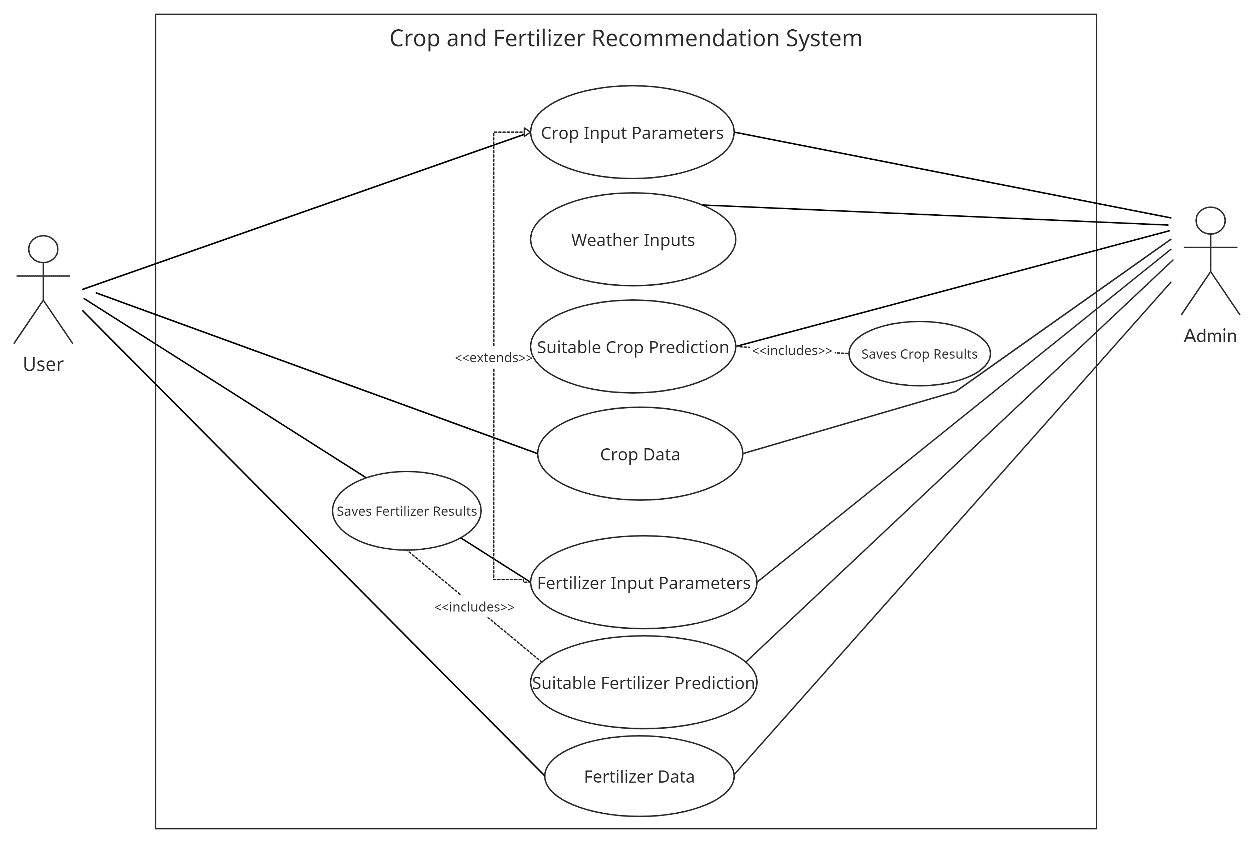
**Fertilizer dataset:**

The dataset used for fertilizer prediction is ‘Fertilizer Prediction dataset’ which was extracted from Kaggle [8] and it consists of 9 labels and 100 tuples. The labels used in the dataset are:

* Temperature: It represents the value of temperature of the place.
* Humidity: It represents the value of amount of water vapor in the air
* Moisture: Moisture represents the value of the water present in the soil.
* Crop Type: It represents the value of different crop types and contains 11 unique values: Maize, Sugarcane, Cotton, Tobacco, Paddy, Barley, Wheat, Millets, Oil seeds, Ground Nuts and Pulses.
* Sand Type: It represents the generic values of soil fields. It has 5 unique values including Sandy, Loamy, Black, Red and Clayey.
* Potassium: It represents the value of potassium in the soil.
* Phosphorous: It represents the value of phosphorus in the soil.
* Nitrogen: It represents the value of phosphorus in the soil.
* Name: Name consists of the name of the fertilizers that the system will predict. It has 6 unique values which include Urea, DAP, 14-35-14, 28-28, 20-20 and 17-17-17

## **2.3 Requirement Specification**

### **2.3.1 Functional Requirement:**



*Figure 2.1 Use Case Diagram*

1. **Crop input parameters:**

The crop input parameters help the user to input the given parameters to the system. Users can simply input the data of parameters like soil type and soil Ph by going through a test in a soil lab.

1. **Weather Inputs:**

The crop and fertilizer prediction system requires weather inputs like Temperature, Humidity, Moisture and Rainfall.

1. **Crop prediction:**

The system helps to recommend a crop with the help of the crop input parameters and the weather inputs.

1. **Crop data:**

The crop that gets recommended is now shown to the user along with its necessary information. The data fetched from the database are displayed on the user’s screen.

1. **Fertilizer input parameters:**

The user can either predict the crop first or the fertilizer. The fertilizer input parameters are similar to the crop input parameters but they also include the crop type.

1. **Fertilizer prediction:**

The algorithms help predict the most accurate fertilizer suitable for the crop according to the input parameters.

1. **Fertilizer data:**

After the prediction of the fertilizer, the amount of fertilizer and information about the fertilizer is accessible for the users.

### **2.3.2 Non-Functional Requirement:**

1. **User-friendly:**

The system is easy to access and work with. Even with the basic or no knowledge and skills in IT, one will be able to operate the system without any difficulty. SwoBali makes use of simple design as well as design to improve the user friendliness of the system.

1. **Easy Access:**

SwoBali can be easily accessed at anytime and anywhere with the help of an internet connection. It also provides the weather and temperature details easily and hence is easily accessible.

1. **Responsive:**

SwoBali is designed to be a responsive system. It takes the users input and analyses it to provide the user with the best suitable response.

1. **Reliability:**

The application is designed to help users with correct and suitable data. It will help the user to take the right decision through the reliable data provided by the application. Hence, SwoBali is reliable.

## **2.4 Feasibility Study**

Feasibility study is an analysis that takes all of a project's relevant factors into accounting including economic, technical, operation and scheduling to ascertain the likelihood of completing the project successfully. Feasibility studies provide the user with the crucial information on the system.

### **2.4.1 Technical Feasibility:**

Technical Feasibility helps validate if a system is technically possible. The proposed system is a web-based user interface that helps the users to get required data through operating a mobile phone. The algorithm used to extract the output provides the user technical guarantee of accuracy and the system itself is designed with the current equipment and existing software technology. Hence, making it technically feasible. The project used different tools such as Python, Tkinter, Sklearn, Numpy and Pandas to make it technically feasible. These tools are open source and helps to make the experience of user better. Furthermore, Random Forest and Support Vector Machine was used to develop system with high accuracy.

**Software** **requirements:** The system is compatible with desktop computers and laptops. The system itself is platform independent and can be ran on any operating system if the required libraries and the system is available.

**Hardware requirements:** The recommendation system requires support of powerful processor and RAM of at least 4gb. SwoBali was developed and ran on Intel Core i5 and i7: 7th and 8th generation. Although the system runs fine on i3 processors, for smooth implementation, better Intel Core and GPU enabled processor is suggested. The proposed hardware and software requirements are feasible for almost all processors and isn’t too expensive. Therefore, the project is technically feasible.

### **2.4.2 Operational Feasibility:**

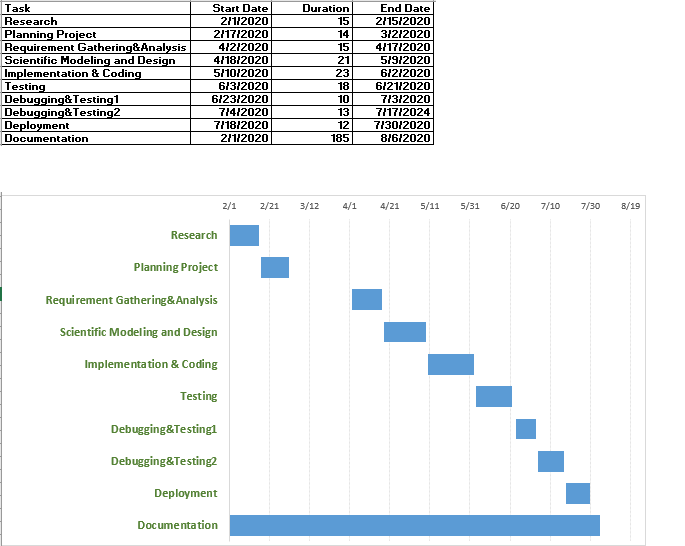
Operational feasibility is the study of how capable a system is while solving the problems and how well it satisfies the identified requirements. SwoBali takes all the necessary inputs from users and helps the system to come up with reliable and accurate prediction. The user is able to make smart decisions according to its output. The system’s user interface is also easy to use and can be easily accessed by people of all age groups making the application reliable and operationally feasible.

### **2.4.3 Economic Feasibility:**

Economic Feasibility is defined as the analysis of the cost-benefit of the project’s cost in order to determine whether the completion of a project is feasible or not. The proposed system is economically feasible as the user only requires a mobile phone and has no extra charges of any sorts.

### **2.4.4 Schedule Feasibility:**

Schedule feasibility is one of the most important one as it helps the organization to estimate the time required to complete the project and helps keep a track of the given schedule.



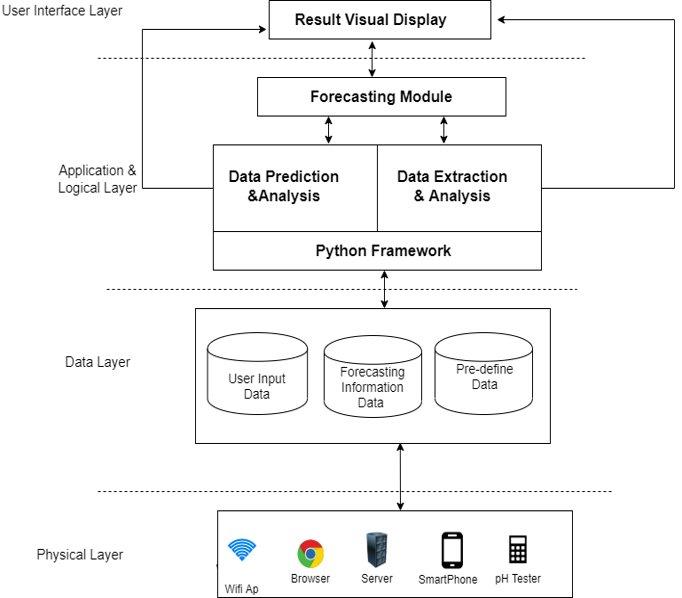
*Figure 2 .2 Gantt Chart*

# CHAPTER 3

# SYSTEM DESIGN

System Design is defined as the process of designing the system like the architecture, modules and interface. The system design serves the purpose of planning the way of how a system is to be implemented in order to solve the problem efficiently. The system design phase is further divided into smaller sub-phase which work together to achieve the goal of the system.

## **3.1 System Architecture and Overview**

****

*Figure 3.1 System Architecture*

To explain the process in above figure:

* The user then inputs data like pH of the soil taken by pH tester and soil type.
* The user also inputs weather data which is directly fed to the application.
* The inputs of the soil and weather are then processed.
* The model predicts suitable output for the processed data and displays the result.

The most important element of the application is that it takes real time input from the user to make sure that the output is data driven and gives the accurate real time result. Even though the application is real time, the inputs and the output it produces are also saved in the user and admin database. So, if a user has no access to internet connection can still make predictions from the databases.

The system model for ‘SwoBali’ is divided into four layers. The first layer is the physical layer where the user and the application interact. This layer helps the application take all the necessary input required for the system to make its predictions. The input from the physical layer is caught by the data layer and is passed to the application/logical layer where the data is fed to the system to make accurate results. Finally, the prediction from the logical layer gets displayed at the user interface layer.

The application ‘SwoBali’ uses a python framework to work with the data analysis, extraction and prediction of data. Users through the system get a real time output with its basic information as well which makes the application user friendly and informative for the users new to farming. Overall, the system helps the user to predict the suitable crop and fertilizer for the maximum yield and helps in crop rotations which also helps in preserving the nutrients of the soil and makes the yield maximum.

## **3.2 Algorithm Used**

### **3.2.1 Support Vector Machine (SVM)**

SVM constructs a hyperplane or set of hyperplanes in a high- or infinite dimensional space, which can be used for classification, regression, or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the 3 largest distance to the nearest training-data point of any class, since in general the larger the margin the lower the generalization error of the classifier**.**

The computational load should be sensible, the mappings are used by the SVM scheme to ensure the dot products will be computed in terms of the variable in the original scope, for that a kernel function k(x,y) selected to get the optimal computational time.

The higher-dimensional space in the hyper planes is distinct as the set of points whose dot product with a vector in that space is constant. These vectors in the hyper planes defining the hyper planes can be chosen to be linear combinations with parameters of images of feature vectors that occur in the database. With this choice of a hyperplane, the points in the feature space that are mapped into the hyperplane are defined by the relation: The equation of the output from a linear SVM is

Where w is the normal vector of the hyperplane, and x is the input vector.

The SVM predicts the output in following steps:

Step (1): The datasets are divided into classes of positives and negatives. A hyperplane is used to divide these classes whose equation is represented by

Step (2): A pair of parallel margin lines are created in the hyperplane which helps these classes to be easily separated. The distance between these margins is calculated through and the goal of the support vector machine is to maximize this distance.

### **3.2.2 Random Forest**

Random forest is an ensemble learning method used for classification, regression and other tasks that operate by constructing a multitude or decision trees at training time and outputting the class that is the mode of the classes or regression of the individual trees.

Given a training set with responses bagging repeatedly (*B* times) selects a random sample with replacement of the training set and fits trees to these samples:

For

1. Sample, with replacement, *n* training examples from call these
2. Train a classification or regression tree on

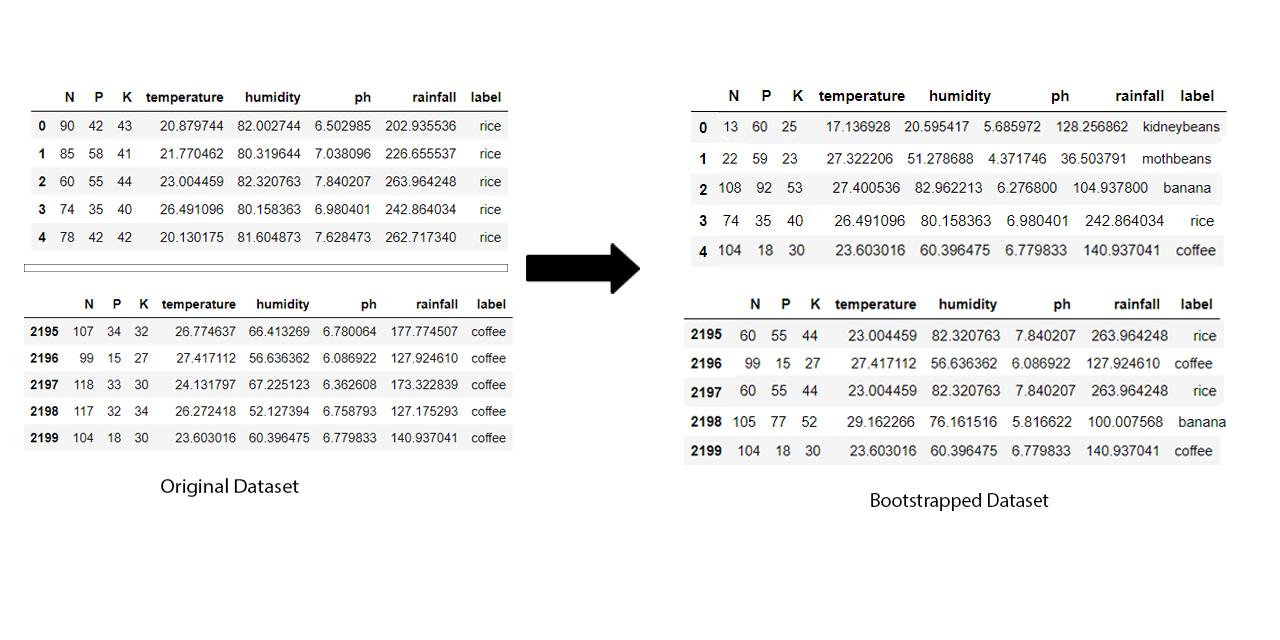
After training, predictions for unseen samples can be made by averaging the predictions from all the individual regression trees on

2

Or by taking the majority vote in the case of classification trees. Additionally, an estimate of the uncertainty of the prediction can be made as the standard deviation of the predictions from all the individual regression trees on

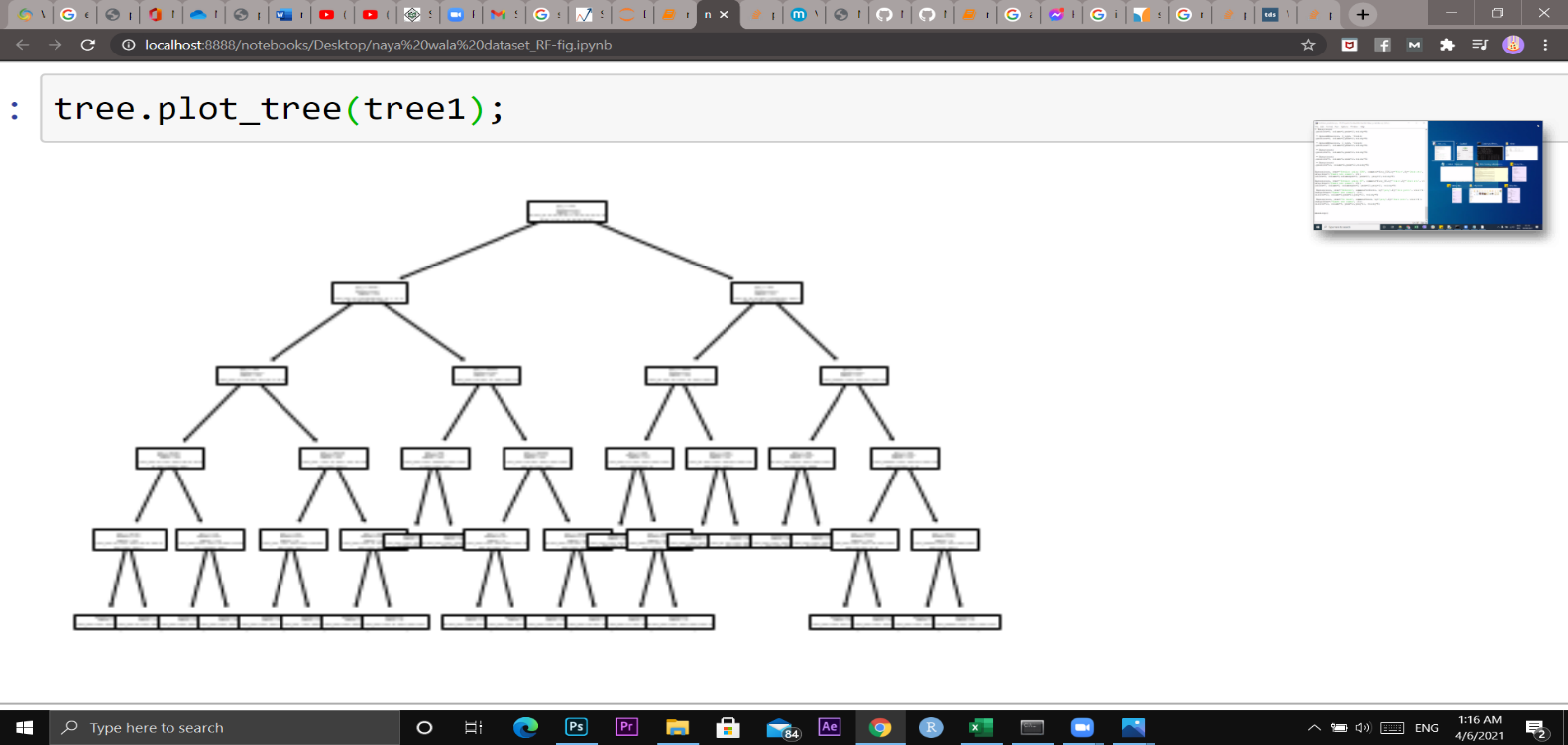
The Random Forest predicts the output in following ways:

Step (1): Bootstrap dataset is created from the original dataset whose size is same as the original dataset.



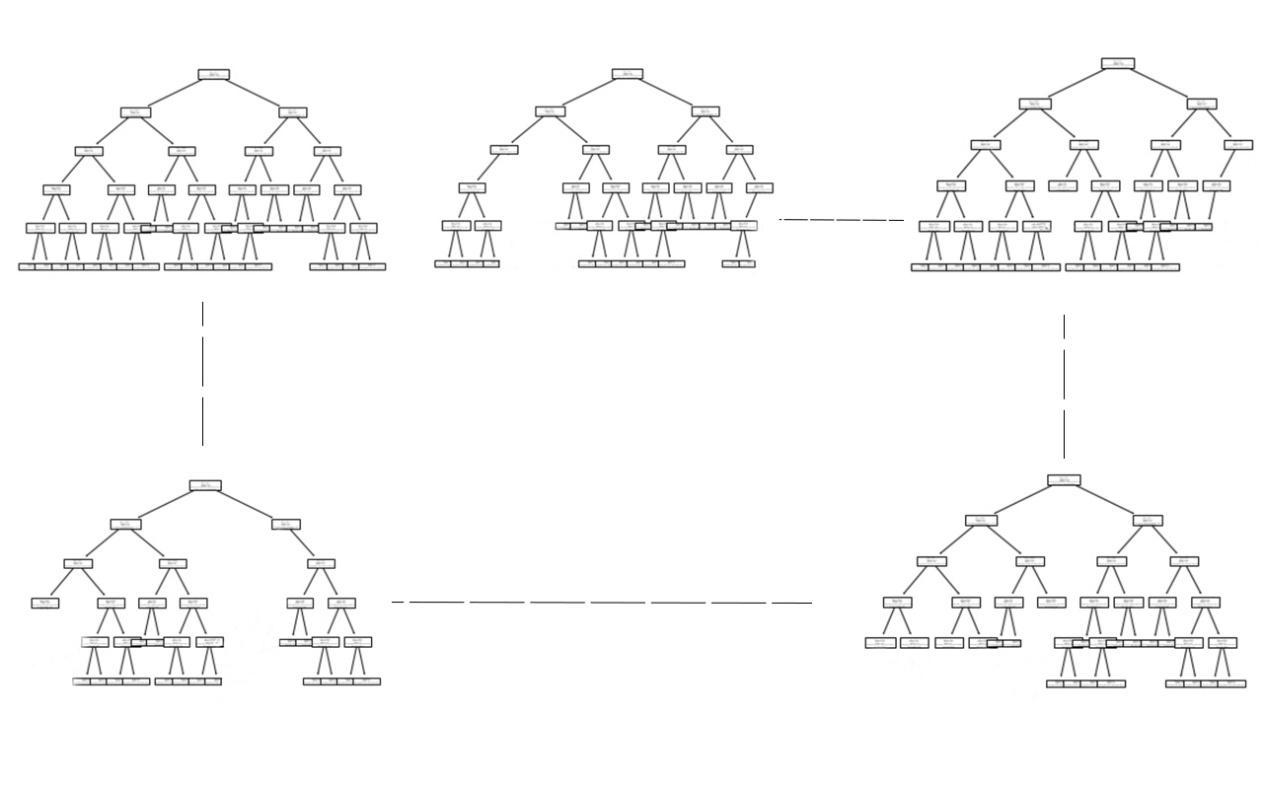
*Figure 3.2 Original Dataset to Bootstrapped Dataset*

Step (2): Decision Tree is created from the bootstrapped dataset by considering random subset or variable.



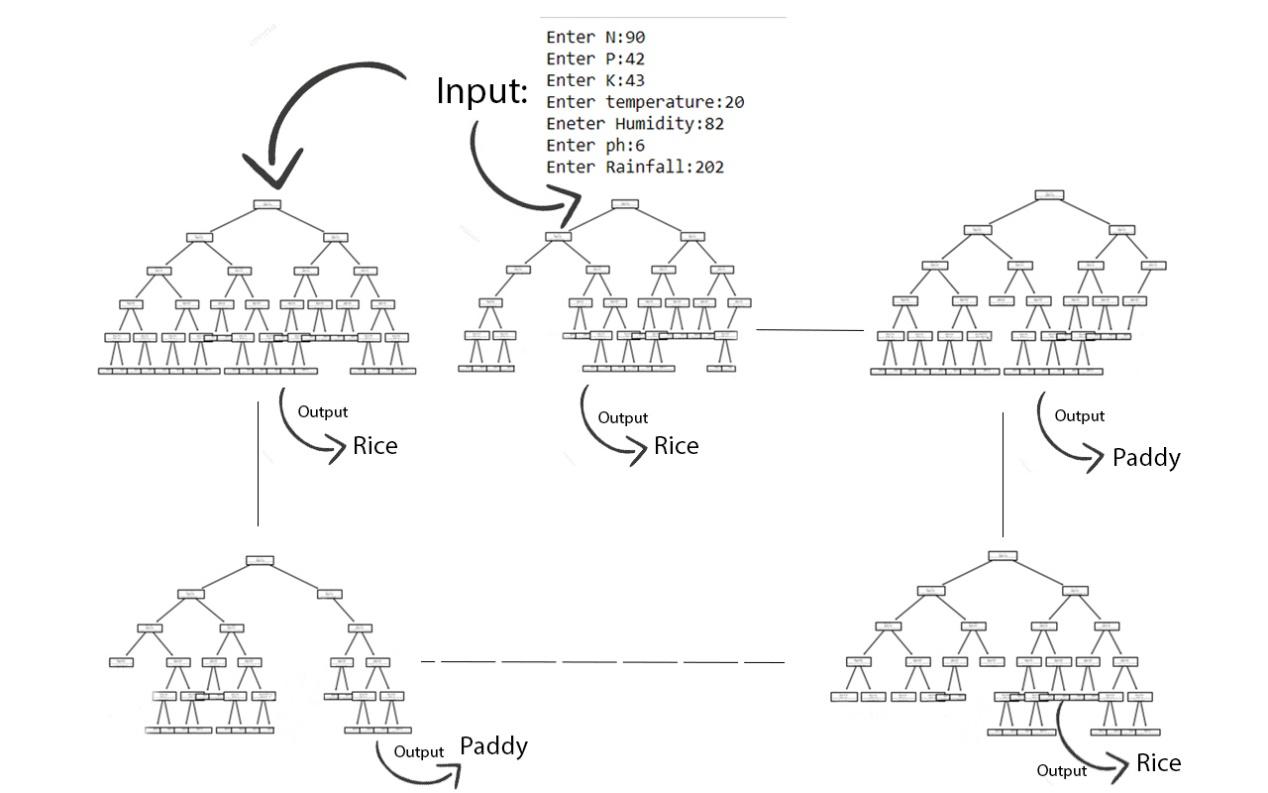
*Figure 3.3 Decision Tree from Bootstrapped Dataset*

Step (3): Step 1 and 2 are repeated continuously to give a number of decision trees. Each decision tree is made from a different bootstrapped dataset with different inputs.



*Figure 3.4 Multiple Decision Trees from Bootstrapped Dataset*

Step (4): The result for the classification goes through each of these trees and gives output of its own. The mode of the outputs is then displayed as the result.



*Figure 3.5 Different outputs from Trees*

Now, Taking the mode of the given outputs from 2199 bootstrapped dataset trees. The result we get in the above figure is ‘Rice.’

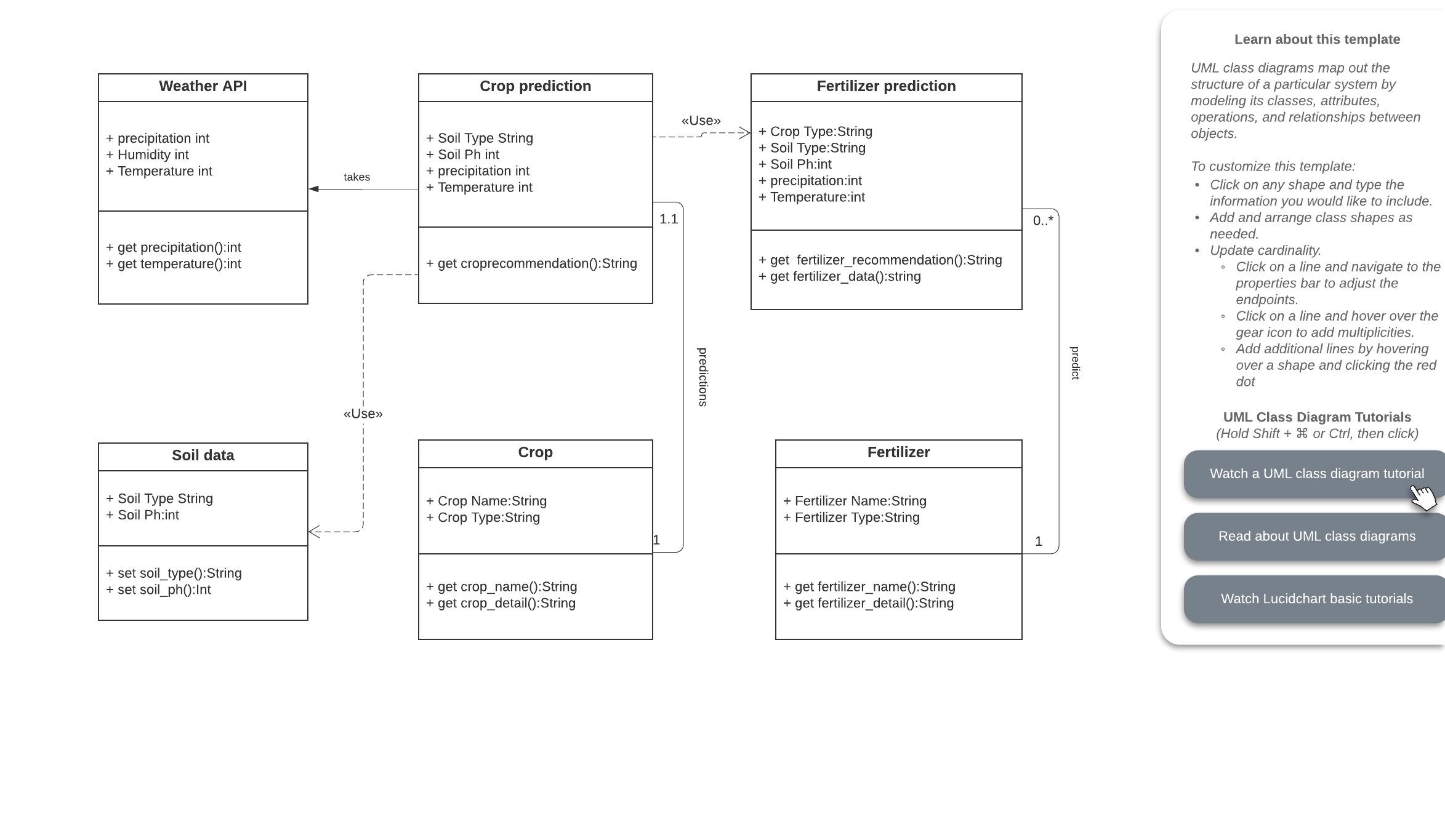
Step (5): The out-of-the-bag dataset or the dataset that gets skipped while making a bootstrapped dataset is used to predict the accuracy of the system.

## **3.3 UML diagram**

UML is an acronym that stands for Unified Modelling Language. It is a diagrammatic representation of software components that help in the documentation of several business processes or workflows. UML diagram helps to have a better understanding of the system as well as the errors in the system. Some of the UML diagrams are given below.

### **3.3.1 Class Diagram**

Class diagrams are the type of UML diagram that describes the structure of a system through the system's classes, their attributes, operations and the relationship between them. Class diagrams are also used for data modelling and each class in the diagram represents both the main element as well as its interaction with the environment.

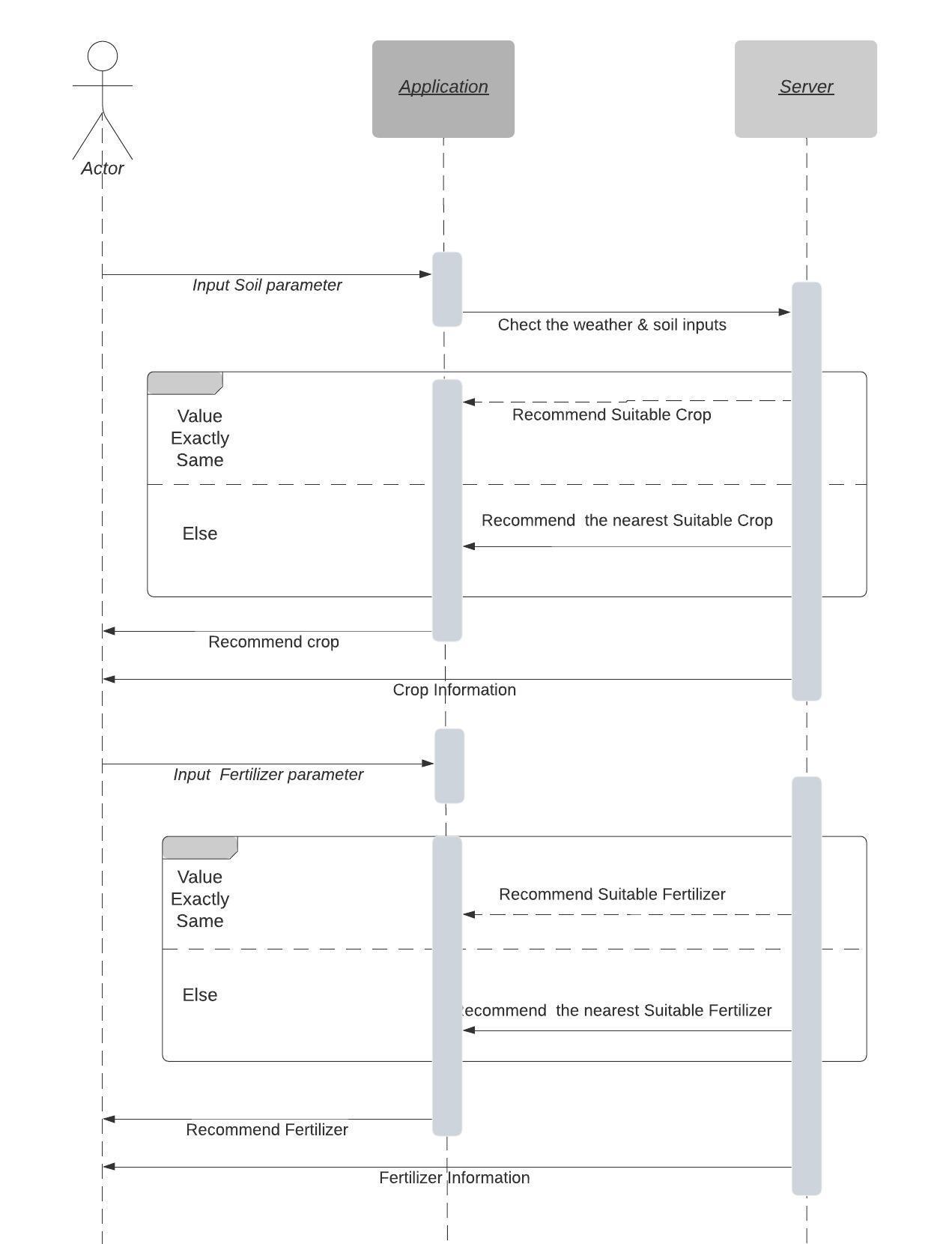


*Figure 3.6 Class Diagram*

The class diagram of SwoBali consists of 6 classes named weather API, Crop prediction, Fertilizer prediction, soil data, crop and fertilizer. Each class has their own attributes and since they are public it is represented by ‘+’. Each attribute is also followed by its type. The relationship between classes is shown by a solid line or dotted line.

### **3.3.2 Sequence Diagram**

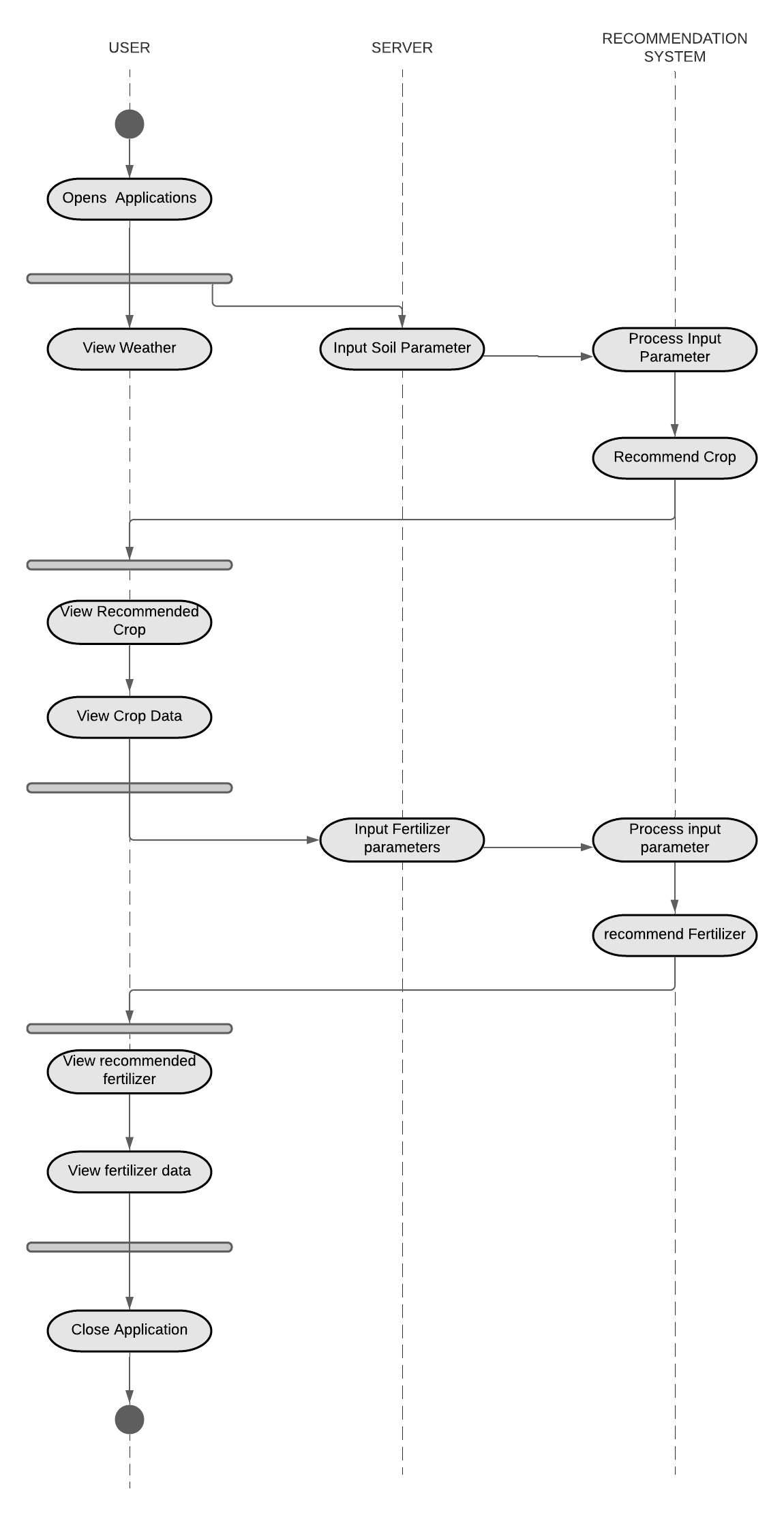
A Sequence diagram is the interaction between the objects in a sequential order. Sequence diagrams describe how and in what order the objects in a system will function. Sequence diagram helps in understanding the existing or new system better.



*Figure 3.7 Sequence Diagram*

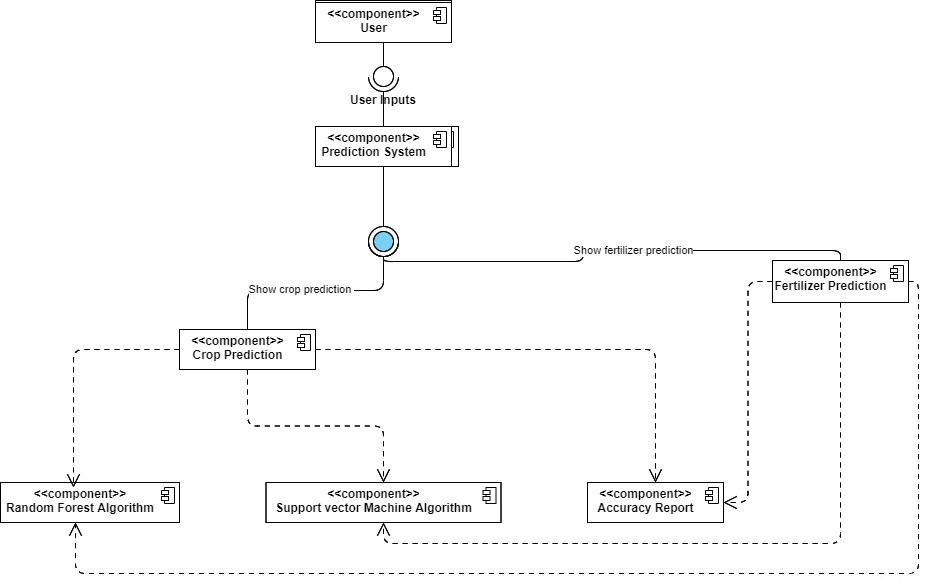
### **3.3.3 Activity Diagram**

An activity diagram is basically a flowchart used to represent the flow from one activity to another. It helps to represent the dynamic perspective of the system. The activity diagram below shows how the working of the activity of a project comes together to help recommend suitable crops and fertilizer to the user.



*Figure 3.8 Activity Diagram*

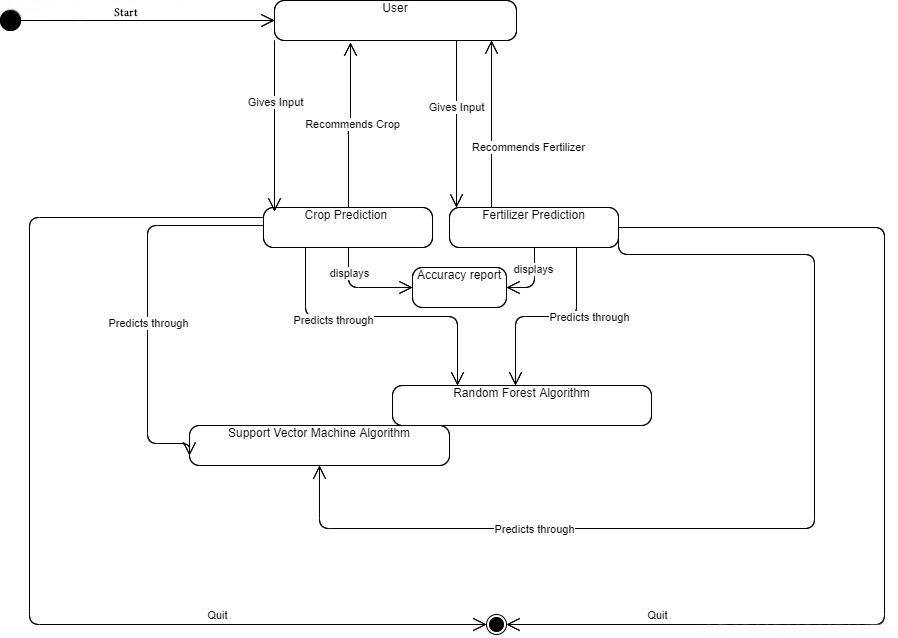
### **3.3.4 Component Diagram**



*Figure 3.9 Component Diagram*

Component diagrams are static implementation views of a system that does not describe the functionality of the system but it describes the components used to make those functionalities. The components used in the system are prediction systems that depend on the value of user input. The prediction system has two different subdivisions which predict crops and fertilizers respectively with the help of random forest and support vector machine algorithms.

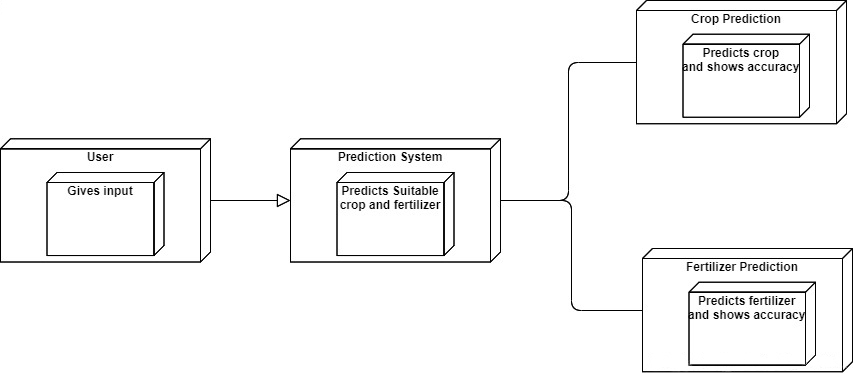
### **3.3.5 State Diagram**



*Figure 3.10 State Diagram*

State Diagrams describe the behaviour of the system through sequence of finite state transitions. The above diagram represents the state diagram of SwoBali which starts through user inputs and ends either when the crop or fertilizer is predicted.

### **3.3.6 Deployment Diagram**



*Figure 3.11 Deployment Diagram*

A deployment diagram shows the execution architecture of a system. The diagram represented above shows the deployment diagram of the system. It contains nodes like a user and prediction system. The user gives input parameters through which the prediction system provides the predicted output.

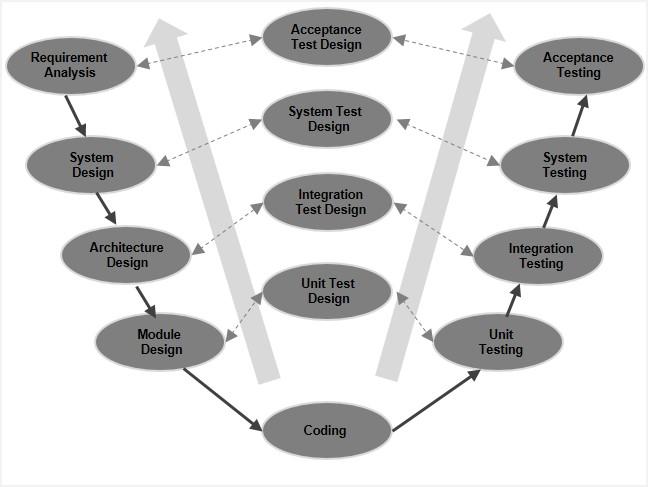
# CHAPTER 4

# SYSTEM IMPLEMENTATION AND TESTING

## **4.1 Implementation:**

The implementation phase of the project is the process of putting all the decisions and planning done in the earlier stage of the project into effect. All the models designed earlier are put into effect through implementation and hence this part is very crucial for a programmer as it can introduce a range of both logical and conceptual issues to the programmer.

So, choosing an appropriate development model is very crucial for the project to get the right direction. There are several development models and each development model has its own advantages and disadvantages. In the case of ‘SwoBali’, we’ve decided to use the V-model (Validation and Verification model). The V-model is a linear model with each stage having a corresponding testing activity. V-model is the most appropriate development model in the case of ‘SwoBali’ because our project is a data-driven application and needs to be accurate for the users to get the benefit from it. V-model assures the quality control of our model and helps it be as accurate as possible. The model helps us to recognize the requirement specification, coding and architecture errors beforehand which ensures overall quality and assures the accuracy for our model.



*Figure 4.1 V-model*

The process that takes place in the V-model are:

1. **Requirement analysis:** Requirement analysis is the phase where the requirements of the system are understood. In this phase, gathering of required information was done. With the help of various research papers, the input parameters and algorithms that would benefit the accuracy of the system were selected. The steps that the system would follow were developed in the requirement analysis phase.
2. **System design:** Once the requirements like parameters and algorithms were clear, the next phase was system design where a rough design of the system and the complete hardware and communication were set up. In this phase, the programming language to be used for the system was decided. Since the system uses machine learning, it required higher processors hence these hardware factors were also considered in system design.
3. **Architecture design:** The system’s architectural specification was designed in this phase. Different algorithms and architectures were proposed and the most feasible one among them were chosen. Considering technical, operational, economic and schedule feasibility, the system was designed and made sure it was user-friendly and cost-effective.
4. **Module design:** The system design is broken down into modules and each module is given different functionalities. The system was divided into training, pre-processing, data loading and testing modules. In this phase, each module was clearly defined with its functions and were allocated to respective people.
5. **Coding Phase:** The module designs that were created and assigned in the module design phase were actually coded in this phase. According to the architectural and Module designs, python was selected as a suitable language in this phase. The modules were then separately executed by each individual to make the system functional.
6. **Testing:** The code is tested along the way and is verified and validated. The main reason the system works on V-model is because the model provides testing phase simultaneously to the coding phase. SwoBali required accuracy and precision in order to meet its objective due to which the results and system were tested in each step.
7. Unit Testing: The bugs on the early stage were detected and fixed in this testing. Tests regarding the front-end implementation were done during this phase.
8. System Testing: The machine learning part of the system was tested in this stage. The accuracy bias was checked during this test to make sure the system doesn’t overfit the data and predicts incorrect value.
9. Implementation Testing: After the testing and validation of the machine learning part of the system, the code was implemented to back end such that it displayed results with accuracy and precision reports. The testing involved checking whether the implementation was correct.
10. Acceptance Testing: In this testing phase, the UI of the application was checked. The non-functional requirements were also checked during this testing.

## **4.2 Tools used:**

### **4.2.1 Front end tools:**

**Python:**

The front-end programming is programmed using Python, Tkinter. Tkinter is a standard GUI library for Python and helps create faster and easy GUI applications. The system’s UI was designed through Tkinter making sure it is user friendly.

### **4.2.2 Back-end tools:**

**Jupyter Notebook:**

Jupyter notebook is used to train and test the ML model which helps predict the output. The datasets fetched are divided into training (80%) and testing (20%) and are fitted into the Random Forest and support vector machine algorithms. The algorithm is chosen by comparing the accuracy level of all the algorithms. The list of libraries used in jupyter are:

* Pandas: The Pandas library was used to load the data and clean it.
* Numpy: The Numpy library was used to convert the training and testing datasets into multi-dimensional arrays.
* MatplotLib: MatplotLib was used to display confusion matrix, tree figures and heat maps.
* Sklearn: Sklearn was used to preprocess the data, fit it and display the result.
* Pickles: Pickles was used to save and load the system.

**Tkinter:**

Tkinter is a library written in Python that is widely used to create GUI applications. It is very easy to build GUI using Tkinter and the process is even faster. Tkinter has several widgets that can be used while developing GUI. These include buttons, radio buttons, checkboxes, etc

## **4.3 Module description:**

**Crop Prediction module:**

The crop dataset that has 8 labels and 17608 targets is wrangled and split into training (80%) and testing (20%) data. The training part of the data is used to fit and train the model to predict suitable crop output. The other 20% of testing data is used to find the accuracy of the system. The crop prediction module consists of the user taking input parameters like Nitrogen, Phosphorous, Potassium, pH, rainfall, temperature and humidity. These input data are then processed and a suitable crop for the land is predicted through two different algorithms: Random Forest and Support Vector Machine. Each of the algorithm's predicted results along with its accuracy, precision report and confusion matrix are displayed to the user.

**Fertilizer Prediction module:**

The fertilizer dataset has 9 labels and 100 targets. The dataset is cleaned, wrangled and split into training (80%) and testing (20%) data. The fertilizer training data is fit to a model and suitable fertilizer for the crop is predicted for the user. The other 20% of fertilizer testing data is used to find the accuracy of the system. The fertilizer Prediction module consists of a fertilizer prediction system where the user needs to input parameters like Nitrogen, Phosphorous, Potassium, Temperature, Humidity, Moisture, crop Type and Soil Type. These data are statistically processed through two different algorithms: Random Forest and Support Vector Machine. Both the outputs from the algorithms are displayed to the user along with the accuracy, precision report and confusion matrix. The predicted fertilizer helps the user know the required Nitrogen-Phosphorus-Potassium value for their soil for the better yield of the crops.

**Report Generation Module:**

The report generation module deals with the report generation part of the system. After the prediction is done, the system also displays accuracy score and report for each of the algorithm. The report consists of recall, precision, f1-score and support. Each of these elements help the users to identify how accurate is the predicted result where,

## **4.4 Testing:**

Testing is a method used to check whether the actual software product matches the expected requirements and to ensure that the software is defect free.

### **4.4.1 Unit testing:**

Unit testing is the level of software testing where individual components of software are tested. They are usually done to check whether the individual modules of code are working properly or not.

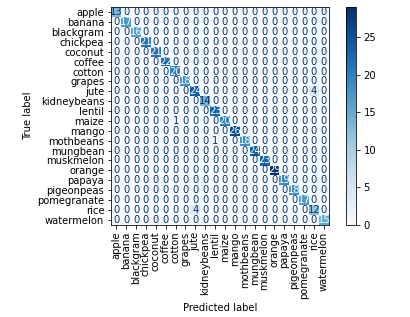
**Crop Prediction Module:**

Table 4. 1:Table for Crop Prediction Using Support Vector Machine Algorithm

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Support Vector Machine Algorithm | | | | | | | | | |
| Steps | Input | | | | | | | Output | Accuracy |
| Nitrogen | Phosphorous | Potassium | Temperature | Humidity | pH value | Rainfall |
| Input Value for the parameters and check the accuracy of the output | 30 | 25 | 12 | 36 | 85 | 6 | 56 | mungbean | 97.73 |
| 83 | 52 | 52 | 30 | 85 | 7 | 156 | jute |
| 23 | 35 | 43 | 23 | 91 | 7 | 106 | pomegranate |
| 39 | 28 | 30 | 29 | 47 | 7 | 100 | mango |
| 99 | 32 | 30 | 26 | 71 | 7 | 130 | coffee |
| 67 | 57 | 36 | 25 | 87 | 8 | 280 | rice |
| 83 | 52 | 52 | 30 | 91 | 7 | 156 | papaya |
| 57 | 17 | 16 | 25 | 87 | 8 | 180 | coconut |
| 89 | 52 | 30 | 59 | 47 | 7 | 80 | maize |
| 39 | 72 | 63 | 59 | 97 | 6 | 50 | papaya |

Table 4. 2: Table for Support Vector Machine Crop Report

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SVM Crop Report | | | | |
|  | 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 | 0.95 | 1.00 | 0.98 | 20 |
| Grapes | 1.00 | 1.00 | 1.00 | 18 |
| Jute | 0.86 | 0.86 | 0.86 | 28 |
| Kidneybeans | 1.00 | 1.00 | 1.00 | 14 |
| Lentil | 0.96 | 1.00 | 0.98 | 23 |
| Maize | 1.00 | 0.95 | 0.98 | 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 |
| Pigeon Peas | 1.00 | 1.00 | 1.00 | 18 |
| Pomegranate | 1.00 | 1.00 | 1.00 | 17 |
| Rice | 0.75 | 0.75 | 0.75 | 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 |



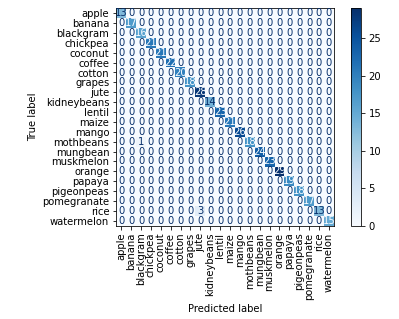
*Figure 4.2 Confusion Matrix for Crop Recommendation Module from SVM*

Table 4. 3: Table for Crop Prediction Using Random Forest Algorithm

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Random Forest Algorithm | | | | | | | | | |
| Steps | Input | | | | | | | Output | Accuracy |
| Nitrogen | Phosphorous | Potassium | Temperature | Humidity | pH value | Rainfall |
| Input Value for the parameters and check the accuracy of the output | 30 | 25 | 12 | 36 | 85 | 6 | 56 | mungbean | 99.09 |
| 83 | 52 | 52 | 30 | 85 | 7 | 156 | jute |
| 23 | 35 | 43 | 23 | 91 | 7 | 106 | pomegranate |
| 39 | 28 | 30 | 29 | 47 | 7 | 100 | mango |
| 99 | 32 | 30 | 26 | 71 | 7 | 130 | coffee |
| 67 | 57 | 36 | 25 | 87 | 8 | 280 | rice |
| 83 | 52 | 52 | 30 | 91 | 7 | 156 | papaya |
| 57 | 17 | 16 | 25 | 87 | 8 | 180 | coconut |
| 89 | 52 | 30 | 59 | 47 | 7 | 80 | coffee |
| 39 | 72 | 63 | 59 | 97 | 6 | 50 | papaya |

Table 4. 4: Table for Random Forest Crop Report

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| RF Crop Report | | | | |
|  | 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 |
| Pigeon Peas | 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 |



*Figure 4.3 Confusion Matrix for Crop Recommendation Module from RF*

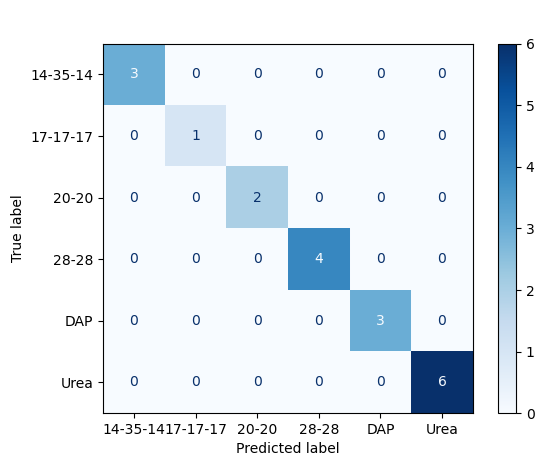
**Fertilizer Prediction Module:**

Table 4. 5: Table for Fertilizer Prediction Using Support Vector Machine Algorithm

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Support Vector Machine Algorithm | | | | | | | | | | |
| Steps | Input | | | | | | | | Output | Accuracy |
| Temperature | Humidity | Moisture | Soil Type | Crop Type | Nitrogen | Phosphorous | Potassium |
| Input Value for the parameters and check the accuracy of the output | 38 | 72 | 51 | Loamy | Wheat | 39 | 0 | 0 | Urea | 99% |
| 29 | 58 | 57 | Black | Sugarcane | 12 | 0 | 10 | 20-20 | 98.09% |
| 36 | 68 | 41 | Red | Ground Nut | 41 | 0 | 0 | Urea | 100% |
| 30 | 60 | 26 | Black | Oil Seeds | 8 | 9 | 30 | 14-35-14 | 100% |
| 26 | 52 | 59 | Loamy | Sugarcane | 11 | 0 | 9 | 20-20 | 99.09% |
| 26 | 52 | 39 | Clayey | Pulses | 21 | 0 | 23 | 28-28 | 98.90% |
| 30 | 60 | 49 | Sandy | Barley | 12 | 0 | 42 | DAP | 100% |
| 31 | 62 | 63 | Red | Cotton | 11 | 12 | 15 | 17-17-17 | 100% |
| 30 | 60 | 58 | Loamy | Cotton | 10 | 7 | 32 | 14-35-14 | 97.89% |
| 32 | 62 | 34 | Red | Tobacco | 22 | 0 | 20 | 28-28 | 99% |

Table 4. 6: Table for Support Vector Machine Fertilizer Report

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SVM Fertilizer Report | | | | |
|  | Precision | Recall | F1-score | support |
| 14-35-14 | 1.00 | 1.00 | 1.00 | 3 |
| 17-17-17 | 1.00 | 1.00 | 1.00 | 1 |
| 20-20 | 1.00 | 1.00 | 1.00 | 2 |
| 28-28 | 1.00 | 1.00 | 1.00 | 4 |
| DAP | 1.00 | 1.00 | 1.00 | 3 |
| Urea | 1.00 | 1.00 | 1.00 | 6 |
|  | | | | |
| Accuracy |  |  | 1.00 | 19 |
| Macro avg | 1.00 | 1.00 | 1.00 | 19 |
| Weighted avg | 1.00 | 1.00 | 1.00 | 19 |



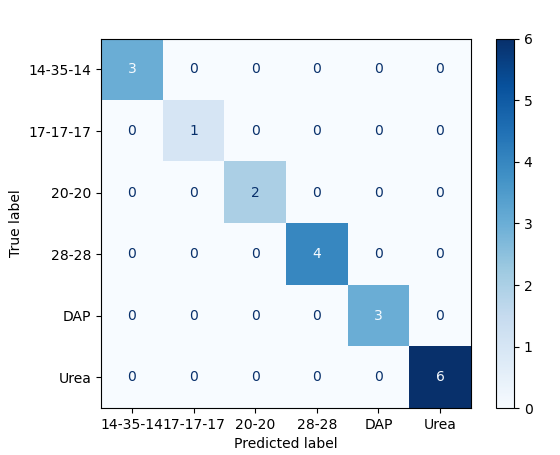
*Figure 4.4 Confusion Matrix for Fertilizer Recommendation Module from SVM*

Table 4. 7: Table for Fertilizer Prediction Using Random Forest Algorithm

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Random Forest Algorithm | | | | | | | | | | |
| Steps | Input | | | | | | | | Output | Accuracy |
| Temperature | Humidity | Moisture | Soil Type | Crop Type | Nitrogen | Phosphorous | Potassium |
| Input Value for the parameters and check the accuracy of the output | 26 | 52 | 35 | Sandy | Barley | 12 | 13 | 10 | 17-17-17 | 100% |
| 32 | 62 | 41 | clayey | paddy | 24 | 22 | 0 | 28-28 | 100% |
| 27 | 54 | 46 | Clayey | Paddy | 35 | 0 | 0 | Urea | 99.0909% |
| 37 | 70 | 32 | Black | Oil Seeds | 12 | 30 | 0 | DAP | 100% |
| 30 | 60 | 63 | Red | Cotton | 9 | 29 | 9 | 13-35-14 | 98.96% |
| 28 | 54 | 35 | Black | Millets | 41 | 0 | 0 | Urea | 98.99% |
| 25 | 50 | 56 | Loamy | Sugarcane | 11 | 15 | 13 | 17-17-17 | 100% |
| 32 | 62 | 34 | Red | Ground Nut | 15 | 37 | 0 | DAP | 100% |
| 27 | 53 | 59 | Loamy | Sugarcane | 10 | 15 | 0 | 20-20 | 100% |
| 26 | 52 | 44 | Sandy | Maize | 23 | 20 | 0 | 28-28 | 100% |

Table 4. 8: Table for Random Forest Fertilizer Report

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| RF Fertilizer Report | | | | |
|  | Precision | Recall | F1-score | support |
| 14-35-14 | 1.00 | 1.00 | 1.00 | 3 |
| 17-17-17 | 1.00 | 1.00 | 1.00 | 1 |
| 20-20 | 1.00 | 1.00 | 1.00 | 2 |
| 28-28 | 1.00 | 1.00 | 1.00 | 4 |
| DAP | 1.00 | 1.00 | 1.00 | 3 |
| Urea | 1.00 | 1.00 | 1.00 | 6 |
|  | | | | |
| Accuracy |  |  | 1.00 | 19 |
| Macro avg | 1.00 | 1.00 | 1.00 | 19 |
| Weighted avg | 1.00 | 1.00 | 1.00 | 19 |



*Figure 4.5 Confusion Matrix for Fertilizer Recommendation Module from RF*

### **4.4.2 Integration testing:**

After unit testing, integration testing is performed to validate the collection and interface modules.

We use the ‘Big Bang Integration Testing’ approach. In the approach almost all of the major units are combined together to perform integration testing.

Table 4. 9: Table for Integration Testing of Crop Predictor

|  |  |  |  |
| --- | --- | --- | --- |
| Crop Predictor | | | |
|  | Steps | Status | Comment |
| Validation | Random string value typed | Pass | Error message displayed |
| Random larger digits number was tried | Pass | Error message displayed. Value cannot be greater than 300 |
| All the values were entered for the prediction | Pass | Algorithms are integrated with GUI. It takes input from the user successfully for the prediction |
| SVM | Value entered and ‘Predict using SVM’ was entered | Pass | Predicted output and accuracy is displayed in GUI |
| Report and Confusion matrix button was entered | Pass | Report and confusion matrix displayed in GUI |
| Random Forest | Value entered and ‘Predict using RF’ was entered | Pass | Predicted output and accuracy is displayed in GUI |
| Report and Confusion matrix button was entered | Pass | Report and confusion matrix displayed in GUI |

Table 4. 10: Table for Integration Testing of Fertilizer Predictor

|  |  |  |  |
| --- | --- | --- | --- |
| Fertilizer Predictor | | | |
|  | Steps | Status | Comment |
| Validation | Random string value typed | Pass | Error message displayed |
| Random larger digits number was tried | Pass | Error message displayed. Value cannot be greater than 100 |
| All the values were entered for the prediction | Pass | Algorithms are integrated with GUI. It takes input from the user successfully for the prediction |
| SVM | Value entered and ‘Predict using SVM’ was entered | Pass | Predicted output and accuracy is displayed in GUI |
| Report and Confusion matrix button was entered | Pass | Report and confusion matrix displayed in GUI |
| Random Forest | Value entered and ‘Predict using RF’ was entered | Pass | Predicted output and accuracy is displayed in GUI |
| Report and Confusion matrix button was entered | Pass | Report and confusion matrix displayed in GUI |

### **4.4.3 System testing:**

System testing is the testing done on finished products. After integration testing, system testing helps to emphasize on the behaviour of the module as a whole.

# CHAPTER 5

# CONCLUSION AND FUTURE ENHANCEMENTS

## **5.1 Conclusion:**

‘SwoBali’ is a crop and fertilizer predictor that helps predict the suitable output and recommend it to the users. It is an ML-based application whose sole objective is to help farmers and non-farmers predict the best suitable crops based on different factors such as climate, soil type, composition, etc, and help them in their decision-making process. The application encourages precision agriculture and helps both the farmers and the government for the better yield of the crops. Using the Random Forest classifier and Support Vector Machine it helps recommend the best suitable crop and fertilizer.

The main objective of the system is to predict the suitable crop and fertilizer using Random Forest and Support Vector Machine with ‘Crop Recommendation dataset’ as its crop and ‘fertilizer prediction dataset’ as its fertilizer dataset. The system contains three modules: Crop Prediction module, Fertilizer Prediction module and Report generation module. Each of these modules predicts crop, fertilizer and generates report respectively with respect to given inputs and algorithms. The system successfully collects the input parameters for crop and fertilizer from the user and organizes them to statistically interpret and help generate appropriate and accurate conclusions. The application also provides the user with the necessary information on the crops so that the application does not only limit its use to farmers (people knowledgeable about the crop and fertilizer) but can also be used in cities to encourage urban vegetation even in a small amount of land.

The system limits the user to input unnecessary inputs and doesn’t predict the crop or fertilizer if the inputs are too high. Using Python as its main programming language, the application was designed and it has successfully met all the criteria and the objective of the project by predicting the suitable crop and recommending the suitable value of fertilizer with an accuracy of 99% from Random Forest and 97% from Support Vector Machine for crop prediction whereas 100% from both Random Forest and Support Vector Machine for Fertilizer Prediction.

## **5.2 Future enhancements of the project:**

* The application can have additional parameters and location-based inputs to increase the accuracy of the result.
* The application can only run with internet connection. Later on, the system could be developed to predict crops and fertilizers offline.
* The application could also be enhanced to web applications.

# REFERENCES/ BIBLIOGRAPHY

[1] S. Mishra, P. Paygude, S. Chaudhary, and S. Idate, “Use of data mining in crop yield prediction,” *Proc. 2nd Int. Conf. Inven. Syst. Control. ICISC 2018*, no. Icisc, pp. 796–802, 2018, doi: 10.1109/ICISC.2018.8398908.

[2] M. J. Mokarrama and M. S. Arefin, “RSF: A recommendation system for farmers,” *5th IEEE Reg. 10 Humanit. Technol. Conf. 2017, R10-HTC 2017*, vol. 2018-Janua, pp. 843–850, 2018, doi: 10.1109/R10-HTC.2017.8289086.

[3] B. K. R, B. K, B. C. A, S. M, and S. R, “Crop Recommendation System for Precision Agriculture,” *Int. J. Comput. Sci. Eng.*, vol. 7, no. 5, pp. 1277–1282, 2019, doi: 10.26438/ijcse/v7i5.12771282.

[4] Manjula, E., and S. Djodiltachoumy. "A model for prediction of crop yield." *International Journal of Computational Intelligence and Informatics* 6.4 (2017): 298-305.

[5] Katarya, Rahul & Raturi, Ashutosh & Mehndiratta, Abhinav & Thapper, Abhinav. (2020). Impact of Machine Learning Techniques in Precision Agriculture. 1-6. 10.1109/ICETCE48199.2020.9091741.

[6] Ravichandran, G., & Koteeshwari, R. S. (2016). *Agricultural crop predictor and advisor using ANN for smartphones. 2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS).* doi:10.1109/icetets.2016.7603053

[7] Institute of Electrical and Electronics Engineers. Madras Section and Institute of Electrical and Electronics Engineers, “2016 Eighth International Conference on Advanced Computing (ICoAC) : 19-21 Jan. 2017.,” pp. 32–36, 2016.

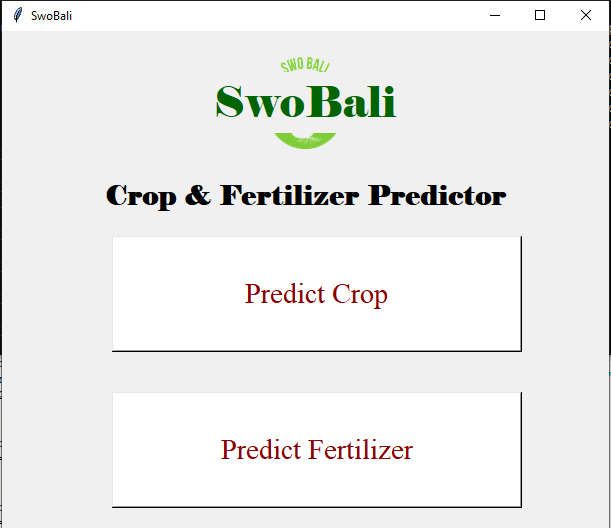
[8] <https://www.kaggle.com/atharvaingle/crop-recommendation-dataset>

[9] https://www.kaggle.com/gdabhishek/fertilizer-prediction

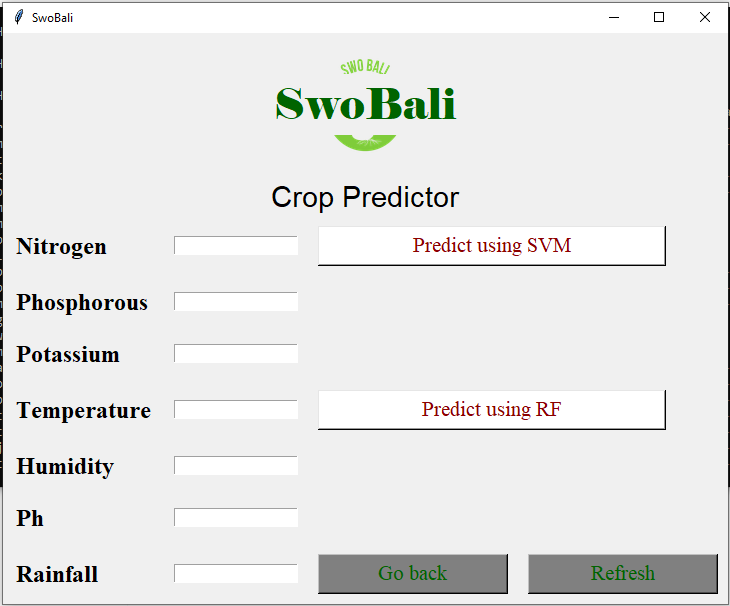
**Appendix**

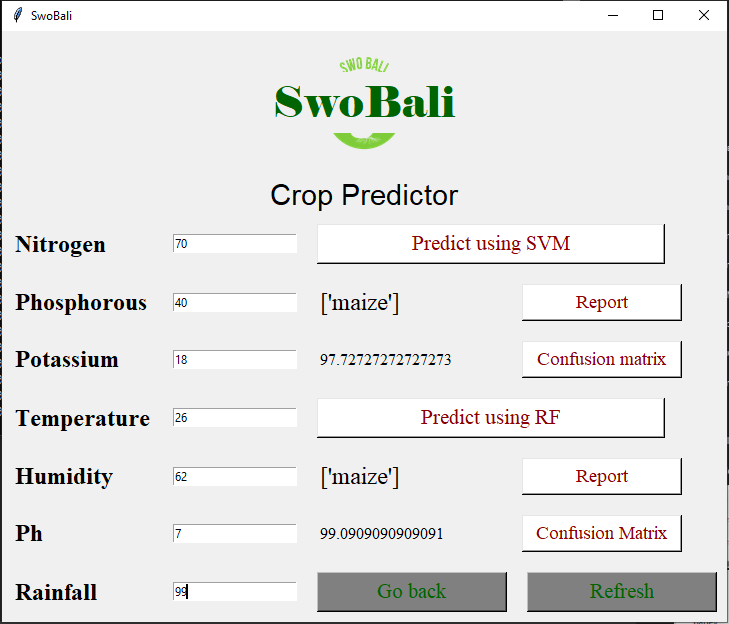
**Screenshots**

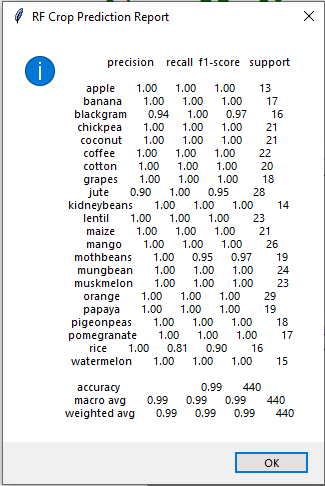
**Front page:**

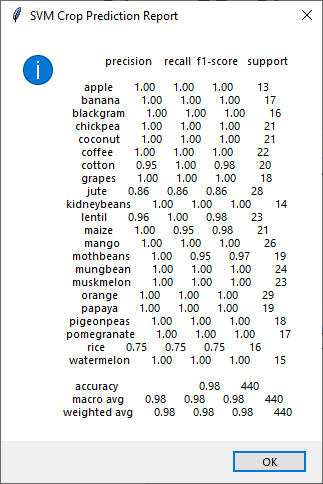


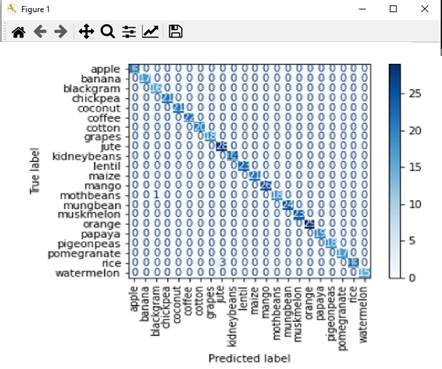
**Crop Prediction Module:**



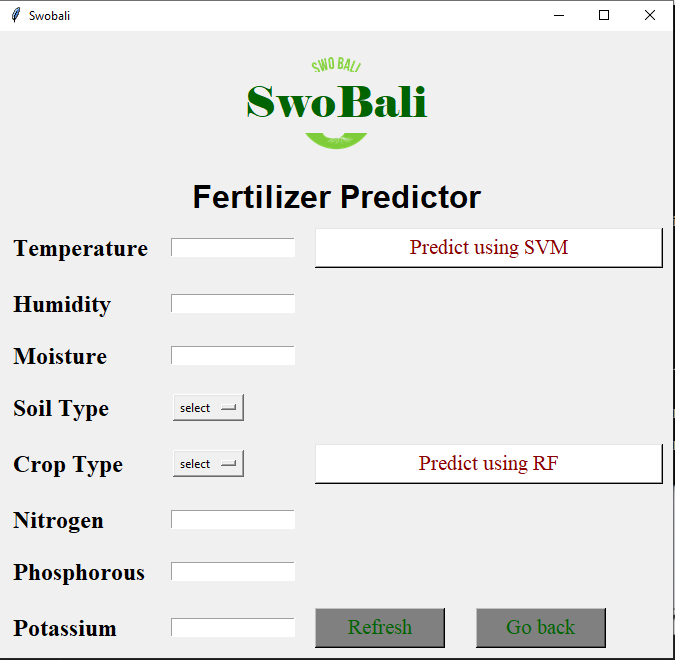
h

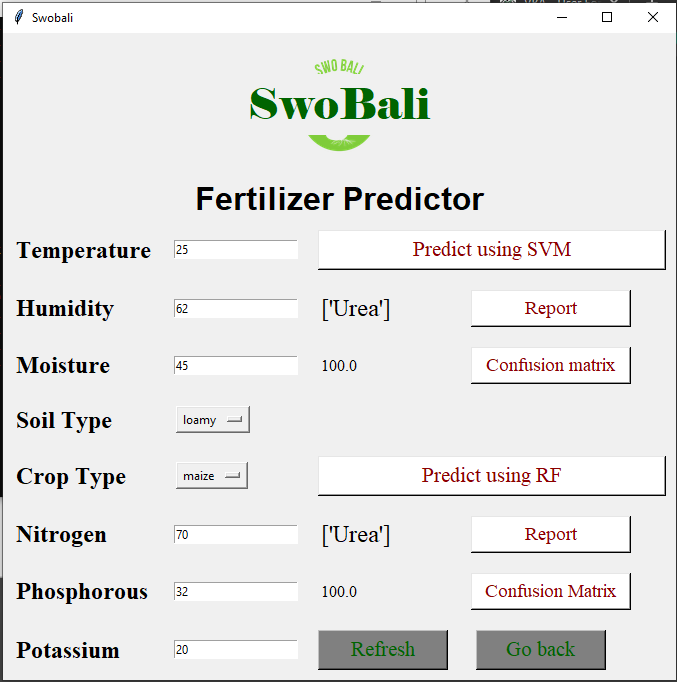


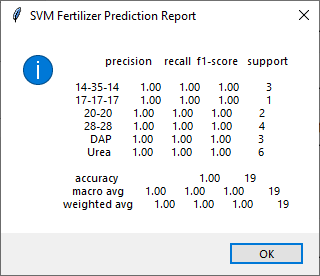


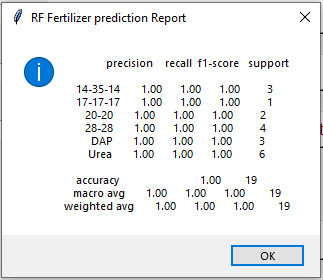


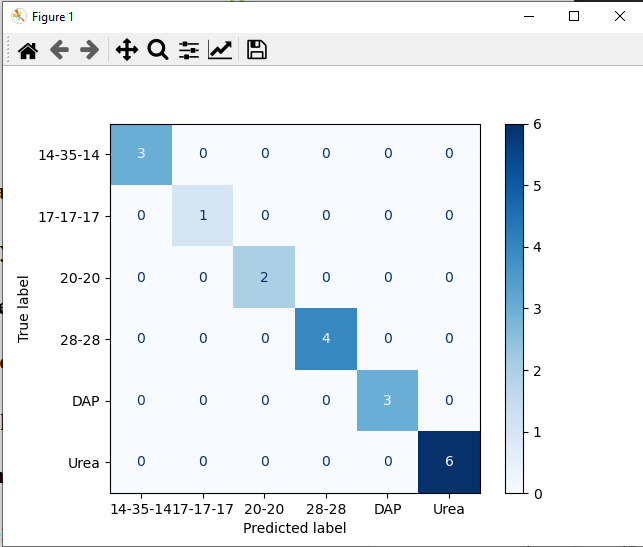
**Fertilizer Prediction Module:**











**Validation:**

