SUSTAINABLE AGRICULTURE PRACTICE USING MACHINE LEARNING

A
Project Stage-II Report

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

Bachelor of Technology
in
Computer Science and Engineering

Submitted by

ARATLA ANANYA	**(19SS1A0503)**
POTHUGANTI SAIKRISHNA	**(19SS1A0538)**
PUSULURI VARNITHA	**(19SS1A0546)**
GANKIDI SAI VIVEK REDDY	**(20SS5A0504)**

Under the guidance of

Dr. P. Krupa sagar
Assistant. Professor(C)

Department of Computer Science and Engineering

JNTUH University College of Engineering Sultanpur

Sultanpur (V), Pulkal (M), Sangareddy (Dist), Telangana-502273

July 2023

JNTUH UNIVERSITY COLLEGE OF ENGINEERING SULTANPUR

Sultanpur(V), Pulkal(M), Sangareddy-502273, Telangana

Department of Computer Science and Engineering

Certificate

This is to certify that the Project Stage-II Report work entitled **"SUSTAINABLE AGRICULTURE PRACTICE USING MACHINE LEARNING"** is a bonafide work carried out by a team consisting of **ARATLA ANANYA** bearing Roll no.**19SS1A0503**, **POTHUGANTI SAIKRISHNA** bearing Roll no.**19SS1A0538**, **PUSULURI VARNITHA** bearing Roll no.**19SS1A0546**, **GANKIDI SAI VIVEK REDDY** bearing Roll no.**20SS5A0504**, in partial fulfillment of the requirements for the degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING discipline to Jawaharlal Nehru Technological University Hyderabad University College of Engineering Sultanpur during the academic year 2022-2023.

The results embodied in this report have not been submitted to any other University or Institution for the award of any degree or diploma.

Guide
Dr. P. KRUPA SAGAR
Assistant. Professor(C)

Head of the Department
Dr. B. V. RAM NARESH YADAV
Professor, & HOD of CSE

External Examiner

Declaration

We hereby declare that Project Stage-II entitled "**SUSTAINABLE AGRICULTURE PRACTICE USING MACHINE LEARNING**" is a bonafide work carried out by a team consisting of **ARATLA ANANYA** bearing Roll no.**19SS1A0503**, **POTHUGANTI SAIKRISHNA** bearing Roll no.**19SS1A0538**, **PUSULURI VARNITHA** bearing Roll no.**19SS1A0546**, **GANKIDI SAI VIVEK REDDY** bearing Roll no.**20SS5A0504**, in partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science and Engineering discipline to Jawaharlal Nehru Technological University Hyderabad College of Engineering Sultanpur during the academic year 2022-2023. The results embodied in this report have not been submitted to any other University or Institution for the award of any degree or diploma.

ARATLA ANANYA (19SS1A0503)
POTHUGANTI SAIKRISHNA (19SS1A0538)
PUSULURI VARNITHA (19SS1A0546)
GANKIDI SAI VIVEK REDDY (20SS5A0504)

Acknowledgment

We wish to take this opportunity to express our deep gratitude to all those who helped us in various ways during our Project Stage-II report work. It is our pleasure to acknowledge the help of all those individuals who were responsible for foreseeing the successful completion of our Project Stage-II report.

We express our sincere gratitude to **Dr. G. Narsimha, Professor and Principal**, JNTUHUCES for his support during the course period.

We express our sincere gratitude to **Dr. Y. Raghavender Rao, Professor, Vice Principal and Head of the Department(ECE)**, JNTUHUCES for his support during the course period.

We are thankful to **Dr. B. V. Ram Naresh Yadav, Professor, and Head of the Department**, Computer Science and Engineering, for his support and encouragement throughout the course period.

We are thankful to our Guide **Dr. P. Krupa sagar, Assistant. Professor(C)** for his effective suggestions during the course period.

Finally, we express our gratitude with great admiration and respect to our faculty for their moral support and encouragement throughout the course.

ARATLA ANANYA (19SS1A0503)
POTHUGANTI SAIKRISHNA (19SS1A0538)
PUSULURI VARNITHA (19SS1A0546)
GANKIDI SAI VIVEK REDDY (20SS5A0504)

Contents

Abstract

Agriculture is a major contributor to the Indian economy. The common problem existing among the Indian farmers are they don't choose the right crop based on their soil requirements. Due to this they face a serious setback in productivity. This problem of the farmers has been addressed through precision agriculture. Precision agriculture is a modern farming technique that uses research data of soil characteristics, soil types, crop yield data collection and suggests the farmers the right crop based on their site-specific parameters. This reduces the wrong choice on a crop and increases the productivity. In this project, we are building an intelligent system using machine learning, which intends to assist the Indian farmers in making an informed decision about which crop to grow depending on the yield factors such as his soil characteristics. Further the system will also provide the farmer, the yield prediction if he plants the recommended crop. This is done using various Machine learning and Deep Learning algorithms.

List of Figures

Chapter 1

INTRODUCTION

1.1 Project Overview

Precision farming is in trend nowadays. It's an approach where inputs are utilised in very precise amounts to get increased average yields for optimum profitability. Crop prediction is methodology is used to predict the suitable crop by sensing various parameter of soil (such as PH, Nitrogen, phosphate,potassium)and also parameter related to atmosphere (such as humidity,rainfall,temperature). ML can be used to obtain the precise inputs to make the yield more lucrative.

The project allows farmers to make better choice of crop to be cultivated with the help of several parameters. The result of the recommended crop leads to better yield. This project would be helpful for reducing of usage of pesticides and any other chemicals for growth of the crop.

1.2 Purpose

Responses by the farmers will be checked automatically and instantly.

- Crop recommendation will reduce the wrong selection of crop to be cultivate by

farmers.

- Being a Precision farming it will reduce over usage of pesticides.
- Can generate more amount of yield than previous.

1.3 Existing System

Extensive work has been done, and many ML algorithms have been applied in the agriculture sector. The biggest challenge in agriculture is to increase farm production and offer it to the end-user with the best possible price and quality. It is also observed that at least 50 percent of the farm produce gets wasted, and it never reaches the end-user. The proposed model suggests the methods for minimizing farm produce wastage. Some of the recent works presented a model where the crop yield is predicted using KNN algorithms by making the clusters. It has been shown that KNN clustering proved much better than SVM or regression. And some models predicted the crop yield for the specific year with the help of advanced regression techniques like Enet, Lasso and Kernel Ridge algorithms. The Stacking regression helped to enhance the accuracy of the algorithms.

1.3.1 Drawbacks of Existing System

- The main challenge faced in agriculture sector is the lack of knowledge about the changing variations in climate. Each crop has its own suitable climatic features. This can be handled with the help of precise farming techniques.
- The existing system which recommends crop yield is either hardware-based being costly to maintain, or not easily accessible
- Despite many solutions that have been recently proposed, there are still open challenges in creating a user-friendly application with respect to crop recommendation

1.4 Proposed System

In this project, we have proposed a model that addresses the existing issues. The novelty of the proposed system is to guide the farmers to maximize the crop yield as well as suggest the most profitable crop for the specific region. The proposed model provides crop selection based on economic and environmental conditions, and benefit to maximize the crop yield that will subsequently help to meet the increasing demand for the country's food supplies. The proposed model predicts the crop yield by studying factors such as rainfall, temperature, area, season, soil type etc. The system also helps to determine the best time to use fertilizers. The user provides an area under cultivation and soil type as inputs. According to the requirement, the model predicts the crop yield for a specific crop. The model also recommends the most profitable crop and suggests the right time to use the fertilizers. The main objective is to obtain a better variety of crops that can be grown over the season. The proposed system would help to minimize the difficulties faced by farmers in choosing a crop and maximize the yield.

1.4.1 Advantages of Proposed System

- The proposed model predicts the crop yield for the data sets of the given region. Integrating agriculture and ML will contribute to more enhancements in the agriculture sector by increasing the yields and optimizing the resources involved. The data from previous years are the key elements in forecasting current performance.
- The proposed system uses recommender system to suggest the right time for using fertilizers.
- The methods in the proposed system includes increasing the yield of crops, real-time analysis of crops, selecting efficient parameters, making smarter decisions and getting better yield.

1.5 Scope

This project would be very useful to suggest the best crop to be cultivated for farmers. We could Recommend varitey of crops based on the testing parameters. Further this project would be helpful in increasing the crop yield.

1.6 Conclusion

Random Forest Regression Model gives the highest R-Squared value and least MSE among all the regressors. Hence the Random Forest Regressor has been selected in the project. User selects the location, puts the soil ph value, gets from ph meter and then puts the area which is in acres. The result shows the appropriate crop based on the climatic conditions as well as the production in tonnes. Web page also displays the data that user inputs and the weather data.

Chapter 2

LITERATURE SURVEY

2.1 Agriculture Decision Support System Using Data Mining

This paper was proposed by Prof. Rakesh Shirsath; Neha Khadke; Divya More, 2017 International Conference on I2C2. The purpose of the paper is to propose a system which helps the users to make decisions regarding the crop to be planted. The system used is a subscription based system which would have personalized information of every farmer registered. The system includes a module which maintains the information of the previous crops planted collected from various sources and shows a matching crop that can be planted. The whole process is done with the help of artificial neutral networks. At the end a feedback system is provided so that the developer can make changes required if the farmer finds some difficulty while using the system.

2.2 A Study on Various Data Mining Techniques for Crop Yield Prediction

This paper is a review for studying various algorithms and their accuracy in the agricultural field proposed by Yogesh Gandge and Sandhya,2017 International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques

(ICEECCOT). It was observed that Multiple Linear Regression gave an accuracy of 90-95% for rice yield. Decision tree using ID3 algorithm was considered for soybean crop and the recommendations were generated. The third algorithm was SVM which was used on all the crops and the accuracy was good with computationally less requirements. Neural network was used on corn data to achieve 95% of accuracy. Other algorithms were also used which are KNN, C4.5, K-means, J48, LAD Tree and Naïve Bayes. The conclusion was that still improvement is needed for the algorithms to achieve better accuracy.

2.3 Crop Recommendation System for Precision Agriculture

This paper was proposed by S.Pudumalar, E.Ramanujam, R.Harine Rajashreen, C.Kavyan, T.Kiruthikan, J.Nishan, 2016 IEEE Eighth International Conference on Advanced Computing (ICoAC). The pupose of this paper is to use an ensemble technique called Majority Voting Technique which combines the power of multiple models to achieve greater prediction accuracy. The methods used are Random Trees, KNN, CHAID and Naïve Bayes for ensemble so that even if one method predicts incorrectly, the other models are likely to make correct predictions and since the majority voting technique is used, the final prediction is correct one. If-then rules are the main components which are used in the prediction process. The accuracy obtained is 88% using the ensemble model.

2.4 Conclusion

In conclusion, the literature survey chapter has provided a comprehensive overview of the existing research and knowledge in the field of crop recommendation. Through a systematic analysis of various studies, publications, and methodologies, this survey has shed light on the challenges, approaches, and advancements in crop recommendation systems. The identification of key factors such as soil conditions, weather patterns, and crop characteristics has emerged as crucial components in developing accurate and context-specific recommendation models. While numerous studies have contributed to the development of crop recommendation systems, there is still a need for further research to address challenges related to data quality, scalability in agricultural settings.

Chapter 3

REQUIREMENT SPECIFICATION

3.1 Hardware and Software Requirements

Computer Hardware:

Hardware refers to the physical components of a computer. Computer Hardware is any part of the computer that we can touch these parts. These are the primary electronic devices used to build up the computer. Examples of hardware in a computer are the Processor, Memory Devices, Monitor, Printer, Keyboard, Mouse, and the Central Processing Unit.

Hardware Requirements:

- System: Intel i3 processor and above
- Input devices: Mouse, Keyboard
- RAM: 4GB and above
- Hard disk: 512 GB

Computer Software:

Software is a collection of instructions, procedures, and documentation that performs different tasks on a computer system. we can say also Computer Software is a programming code executed on a computer processor. The code can be machine-level code or the code written for an operating system. Examples of software are Ms Word, Excel, PowerPoint, Google Chrome, Photoshop, MySQL, etc.

Software Requirements:

- IDE: Jupyter (or) Python Ide (or) Visual Studio code (or) Google Colab
- OS: Windows7 and above
- Coding languages: Python

3.2 Functional Requirements

- Data Collection
- Data Preprocessing
- Training and Testing
- Modeling
- Predicting

3.3 Conclusion

Requirements specification is a must when it comes to developing software. Some good practices lead to good documentation. Since RS is useful for both customers and software development team, it is essential to develop a complete and clear requirements document. RS helps the customers to define their needs with accuracy, while it helps the development team understand what the customers need in terms of development.

Chapter 4

SYSTEM ANALYSIS

4.1 Architecture of the System

The Architecture of the system depicts how the actual process of the system is working. Initially, data is taken as input and then it is pre-processed. The system is trained with a given dataset further it uses the NLP to test the new data and produces output as the polarity of the review.

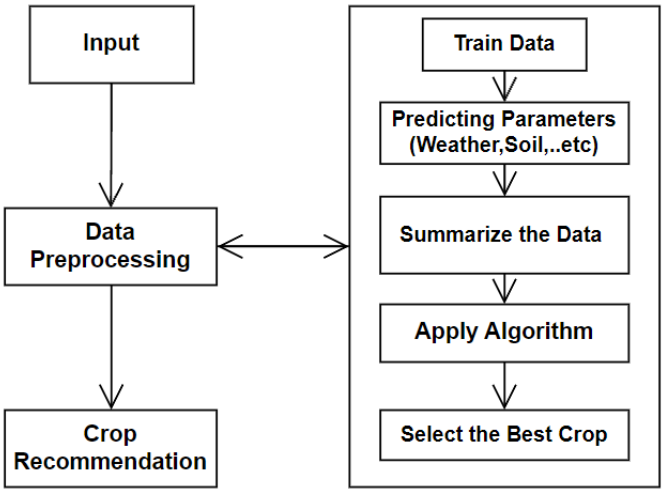

Figure 4.1: Architecture of the System

4.2 Feasibility Study

The next step in analysis is to verify the feasibility of the proposed system. ”All projects are feasible given unlimited resources and infinite time”. But in reality both resources and time are scarce. Project should confirm to time bounce and should be optimal in there consumption of resources. This places a constant are approval of any project. These feasibility study are 3 types:

- Technical Feasibility
- Operational Feasibility
- Economical Feasibility

4.2.1 Technical Feasibility

To determine whether the proposed system is technically feasible, we should take into consideration the technical issues involved behind the system.

- Technical Feasibility is the process of figuring out how you’re going to produce your product or service to determine whether it’s possible for your company.
- Before launching your offerings, you must plan every part of your operations, from first sourcing your production materials all the way to tracking your sales.
- By looking at all the logistics of this process, you can determine potential challenges and figure out ways to overcome them.
- Technical feasibility also involves the evaluation of the hardware, software, and other technical requirements of the proposed system.

4.2.2 Operational Feasibility

To determine the operational feasibility of the system we should take into consideration the awareness level of the users. This system is operational feasible since the users are

familiar with the technologies and hence there is no need to gear up the personnel to use system. Also the system is very friendly and to use.

4.2.3 Economic Feasibility

To decide whether a project is economically feasible, we have to consider various factors as:

- Cost-benefit analysis
- Long-term returns candidates appearing
- Maintenance costs

It requires average computing capabilities and access to internet, which are very basic requirements and can be afforded by any organization hence it doesn't incur additional economic overheads, which renders the system economically feasible. The examination system being an online system should be available anytime.

4.3 Algorithm

4.3.1 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.
- 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.
- The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

4.3.2 How does Random Forest Algorithm Works?

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase. The Working process can be explained in the below steps and diagram:

- **Step-1:** Select random K data points from the training set.
- **Step-2:** Build the decision trees associated with the selected data points (Subsets).
- **Step-3:** Choose the number N for decision trees that you want to build.
- **Step-4:** Repeat Step 1 & 2.
- **Step-5:** For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

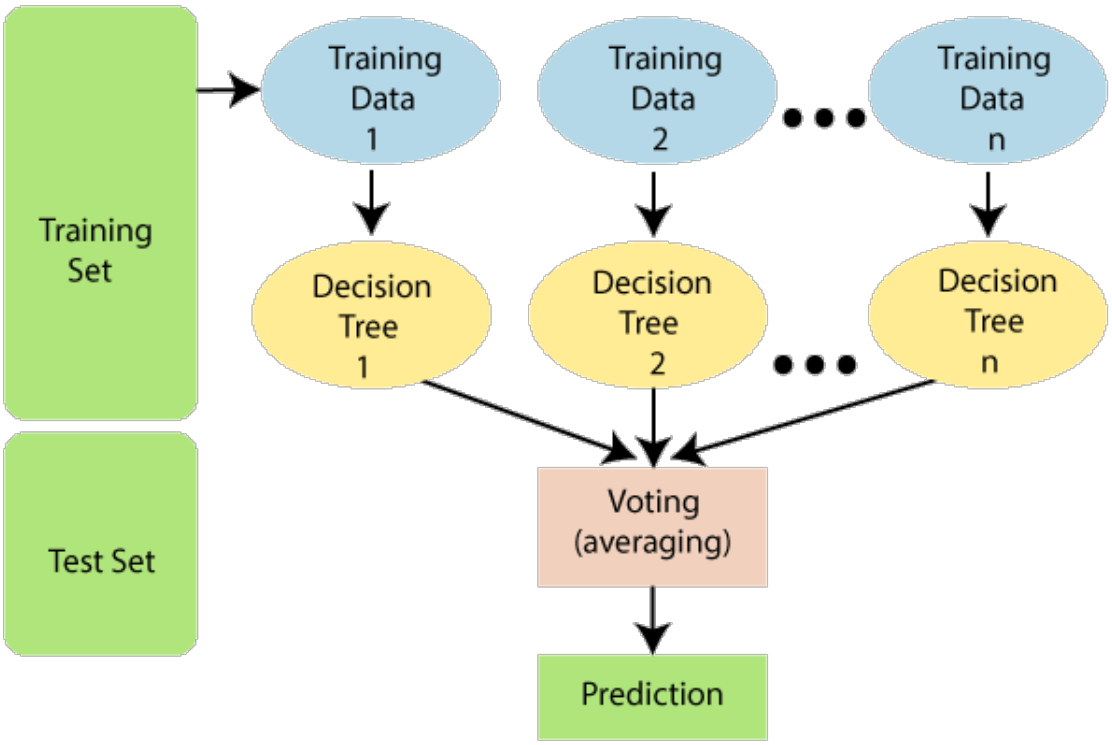

Figure 4.2: Random Forest

4.3.3 Python Flask

Flask is a web framework that provides libraries to build lightweight web applications in python. It is developed by Armin Ronacher who leads an international group of python enthusiasts (POCCO). It is based on WSGI toolkit and jinja2 template engine. Flask is considered as a micro framework.

Figure 4.3: Flask

4.4 Conclusion

We can say that system analysis is a problem solving strategy that includes glimpsing at the more extensive system, breaking the separated parts, and sorting out how it works to accomplish a specific objective. There are several definitions of system analysis like another definition is its examination of a specific system to observe the sectors of modifications and prepare any essential enhancements, if required.

Chapter 5

SYSTEM DESIGN

5.1 UML

The Unified Modeling Language (UML) is a standard language for writing software blue prints. The UML is a language which provides vocabulary and the rules for combining words in that vocabulary for the purpose of communication. A modeling language is a language whose vocabulary and the rules focus on the conceptual and physical representation of a system. Modeling yields an understanding of a system. A UML diagram is a diagram based on the UML (Unified Modeling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system.

UML Concepts:

The Unified Modeling Language (UML) is a standard language for writing software blue prints. The UML is a language for

- Visualizing
- Specifying
- Constructing
- Documenting the artifacts of a software intensive system

The UML is a language which provides vocabulary and the rules for combining words in that vocabulary for the purpose of communication. A modeling language is a language whose vocabulary and the rules focus on the conceptual and physical representation of a system. Modeling yields an understanding of a system.

Characteristics of UML:

The UML has the following features:

- It is a generalized modeling language.
- It is distinct from other programming languages like C++, Python, etc.
- It is interrelated to object-oriented analysis and design.
- It is used to visualize the workflow of the system.
- It is a pictorial language, used to generate powerful modeling artifacts.

A UML diagram is a diagram based on the UML (Unified Modeling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system.

Building Blocks of the UML:

The vocabulary of the UML encompasses three kinds of building blocks.

- **Things:** Things are the abstractions that are first-class citizens in a model
- **Relationships:** relationships tie these things together
- **Diagrams:** diagrams group interesting collections of things

5.2 Use Case Diagram

Use case diagrams are a set of use cases, actors, and their relationships. They represent the use case view of a system.

A use case represents a particular functionality of a system. Hence, use case diagram is used to describe the relationships among the functionalities and their internal/external controllers. These controllers are known as actors. In this project, faculty and student are the actors.

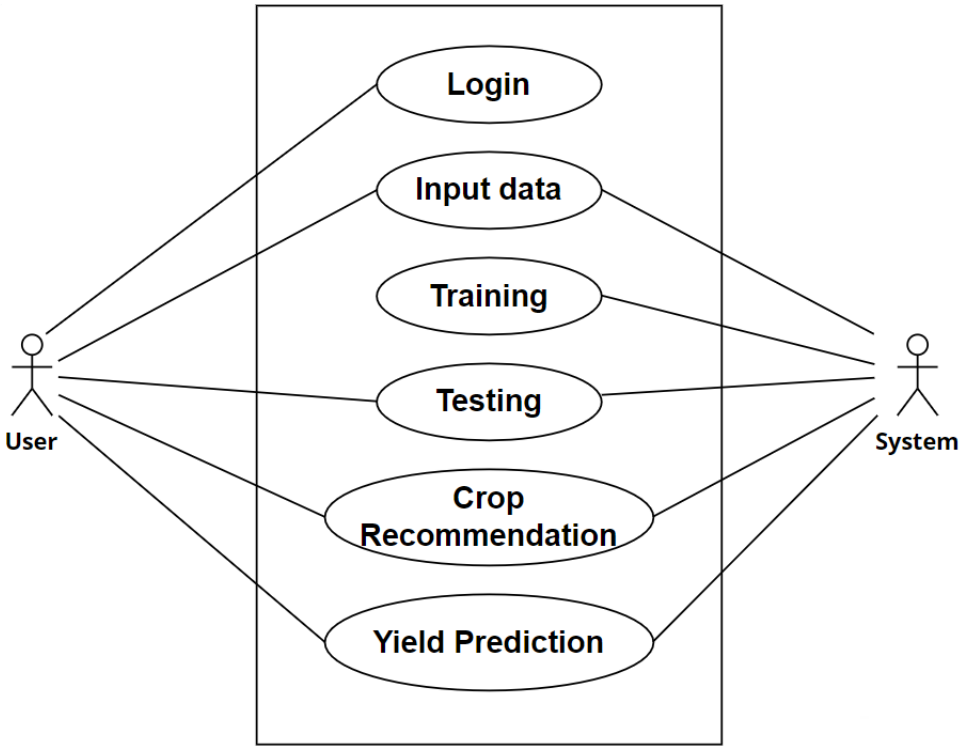

Figure 5.1: Use Case Diagram of System

5.3 Activity Diagram

Activity diagrams are used to document workflows in a system, from the business level down to the operational level. The general purpose of Activity diagrams is to focus on flows driven by internal processing vs. external events.

Activities are nothing but the functions of a system. Numbers of activity diagrams are prepared to capture the entire flow in a system.

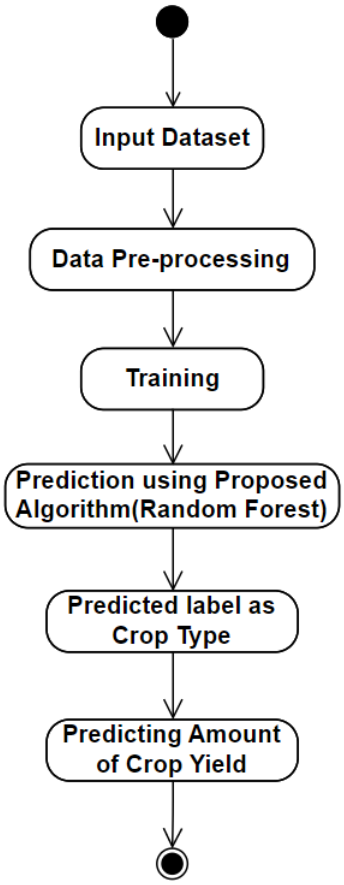

Figure 5.2: Activity Diagram of System

5.4 Sequence Diagram

A sequence diagram is a Unified Modeling Language (UML) diagram that illustrates the sequence of messages between objects in an interaction. A sequence diagram consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction.

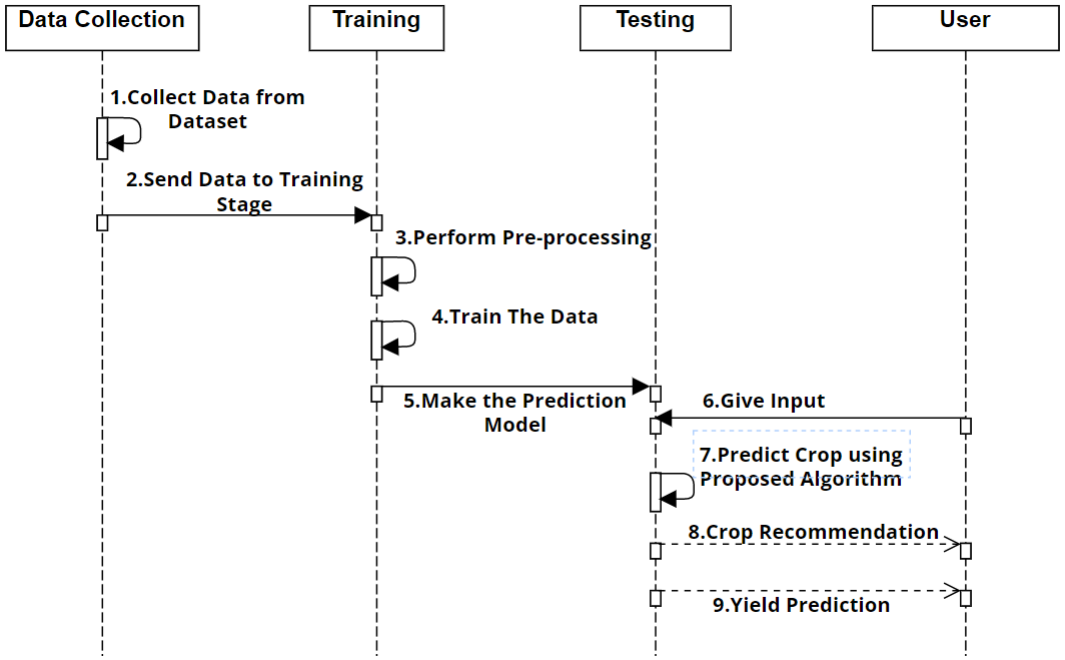

Figure 5.3: Sequence Diagram of System

5.5 Conclusion

A UML diagram is a diagram based on the UML (Unified Modeling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system.

Chapter 6

IMPLEMENTATION

6.1 Coding

6.1.1 major.ipynb

Import required Libraries:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
import seaborn as sns
import matplotlib.pyplot as plt
```

Load Crop_recommendation data and show top 5 rows:

```
crop = pd.read_csv('Crop_recommendation.csv')
crop.head()
```

Data Preprocessing:

```
crop.shape
crop.isnull().sum()
crop.duplicated().sum()
crop.info()
crop.describe()
corr = crop.corr()
corr
```

Apply LabelEncoding to make target feature into numerical:

```
crop['label'].value_counts()
crop_dict = {
    'rice': 1,
    'maize': 2,
    'jute': 3,
    'cotton': 4,
    'coconut': 5,
    'papaya': 6,
    'orange': 7,
    'apple': 8,
    'muskmelon': 9,
    'watermelon': 10,
    'grapes': 11,
    'mango': 12,
    'banana': 13,
    'pomegranate': 14,
    'lentil': 15,
    'blackgram': 16,
    'mungbean': 17,
    'mothbeans': 18,
    'pigeonpeas': 19,
    'kidneybeans': 20,
    'chickpea': 21,
    'coffee': 22
}
crop['label_num'] = crop['label'].map(crop_dict)

crop.drop('label',axis=1,inplace=True)
```

```
crop.head()
```

Train Test Split:

```
# Split the dataset into features and labels
X = crop.iloc[:, :-1]
y = crop.iloc[:, -1]

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=42)
```

Scale the Feature using MinMaxScaler:

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Standardization:

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Training Models:

```
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
```

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier

# create instances of all models
models = {
    'Linear Discriminant Analysis': LinearDiscriminantAnalysis(),
    'Logistic Regression': LogisticRegression(),
    'Naive Bayes': GaussianNB(),
    'Support Vector Machine': SVC(),
    'K-Nearest Neighbors': KNeighborsClassifier(),
    'Decision Tree': DecisionTreeClassifier(),
    'Random Forest': RandomForestClassifier(),
}

from sklearn.metrics import accuracy_score
for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    acc = accuracy_score(y_test, y_pred)
    print(f'{name}:\nAccuracy: {acc:.4f}')
```

Selecting Random Forest Model:

```
rdf = RandomForestClassifier()
rdf.fit(X_train,y_train)
y_pred = rdf.predict(X_test)
print(accuracy_score(y_test,y_pred))
```

Predictive System:

```
def predict_crop(N, P, K, temperature, humidity, pH, rainfall):
    input_values = np.array([[N, P, K, temperature,
    humidity, pH, rainfall]])
```

```
    # Use the model to make a prediction
    prediction = rdf.predict(input_values)

    # Return the predicted crop label
    return prediction[0]
```

Testing with new data:

```
new_data=[[21,26,27,27.003155,47.675254,5.699587,   95.851183]]
predicted_crop=rdf.predict(new_data)
print(f"Predicted Crop:{predicted_crop[0]}")

new_data=[[2,129,201,22.78234161,94.36803516,5.682343744,122.1449949]]
predicted_crop=rdf.predict(new_data)
print(f"Predicted Crop:{predicted_crop[0]}")
```

6.1.2 app.py

```
from flask import Flask, render_template, request
import pandas as pd
import numpy as np
import sklearn
import os
import pickle
import warnings

app = Flask(__name__)

loaded_model = pickle.load(open("model.pkl", 'rb'))
```

```
@app.route('/')
def home():
    return render_template('home.html')



@app.route('/predict', methods=['POST'])
def predict():
    N = int(request.form['Nitrogen'])
    P = int(request.form['Phosporus'])
    K = int(request.form['Potassium'])
    temp = float(request.form['Temperature'])
    humidity = float(request.form['Humidity'])
    ph = float(request.form['pH'])
    rainfall = float(request.form['Rainfall'])

    feature_list = [N, P, K, temp, humidity, ph, rainfall]
    single_pred = np.array(feature_list).reshape(1, -1)

    prediction = loaded_model.predict(single_pred)

    crop_dict = {1: "Rice", 2: "Maize", 3: "Jute", 4: "Cotton",
            5: "Coconut", 6: "Papaya", 7: "Orange",
            8: "Apple", 9: "Muskmelon", 10: "Watermelon",
            11: "Grapes", 12: "Mango", 13: "Banana",
            14: "Pomegranate", 15: "Lentil", 16: "Blackgram",
            17: "Mungbean", 18: "Mothbeans", 19: "Pigeonpeas",
            20: "Kidneybeans", 21: "Chickpea", 22: "Coffee"}

    if prediction[0] in crop_dict:
        crop = crop_dict[prediction[0]]
        result = "{} is the best crop to be cultivated right
        there".format(crop)
    else:
        result = "Sorry, we could not determine the best crop
        to be cultivated with the provided data."
```

```
    return render_template('home.html', prediction=result)


if __name__ == '__main__':
    app.run(debug=True)
```

6.1.3 Home.html

```
<!DOCTYPE html>
<html>
<head>
<title>Crop Recommender</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstrap/
4.5.2/css/bootstrap.min.css">
<scriptsrc="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/
jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/
1.16.0/umd/popper.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/
bootstrap.min.js"></script>
<style>
h1 {
color: mediumseagreen;
text-align: left;
}

.warning {
color: red;
font-weight: bold;
text-align: center;
}
.card{
```

```
margin-left:410px;
margin-top: 20px;
color: white;
}
.container{
font-weight: bold;
padding-bottom:10px;
border-radius: 15px;
}
</style>
</head>
<body style="background:#BCBBB8">

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
<div class="container-fluid">
<a class="navbar-brand" href="/">Crop Recommendation</a>
<button class="navbar-toggler" type="button" data-bs-toggle=
    "collapse" data-bs-target="#navbarSupportedContent"
    aria-controls="navbarSupportedContent"
    aria-expanded="false" aria-label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav me-auto mb-2 mb-lg-0">
    <li class="nav-item">
        <a class="nav-link active" aria-current="page" href="#">
        home</a>
    </li>
    <li class="nav-item">
        <a class="nav-link" href="#">Contact</a>
    </li>
    <li class="nav-item">
        <a class="nav-link disabled">About</a>
    </li>
    </ul>
    <form class="d-flex" role="search">
    <input class="form-control me-2" type="search" placeholder="Search"
    aria-label="Search">
```

```
    <button class="btn btn-outline-success" type="submit">Search
    </button>
    </form>
</div>
</div>
</nav>


<div class="container my-5 mt-5 bg-light">
<h1>Crop Recommendation <span class="text-success"></span></h1>

<form method="POST" action="{{ url_for('predict') }}" class="mt-5">
<div class="row">
<div class="col-md-4">
    <label for="Nitrogen">Nitrogen</label>
    <input type="number" id="Nitrogen" name="Nitrogen" placeholder=
    "Enter Nitrogen" class="form-control" required>
</div>
<div class="col-md-4">
    <label for="Phosporus">Phosphorus</label>
    <input type="number" id="Phosporus" name="Phosporus" placeholder=
    "Enter Phosphorus" class="form-control" required>
</div>
<div class="col-md-4">
    <label for="Potassium">Potassium</label>
    <input type="number" id="Potassium" name="Potassium" placeholder=
    "Enter Potassium" class="form-control" required>
</div>
</div>

<div class="row mt-4">
<div class="col-md-4">
    <label for="Temperature">Temperature</label>
    <input type="number" step="0.01" id="Temperature" name=
    "Temperature"placeholder="Enter Temperature in °C" class=
    "form-control" required>
</div>
<div class="col-md-4">
```

```
    <label for="Humidity">Humidity</label>
    <input type="number" step="0.01" id="Humidity" name="Humidity"
    placeholder="Enter Humidity in %" class="form-control" required>
</div>
<div class="col-md-4">
    <label for="pH">pH</label>
    <input type="number" step="0.01" id="pH" name="pH"
    placeholder="Enter pH value" class="form-control" required>
</div>
</div>

<div class="row mt-4">
<div class="col-md-4">
    <label for="Rainfall">Rainfall</label>
    <input type="number" step="0.01" id="Rainfall" name="Rainfall"
    placeholder="Enter Rainfall in mm" class="form-control" required>
</div>
</div>

<div class="row mt-4">

<div class="col-md-12 text-center">
    <button type="submit" class="btn btn-primary btn-lg">
    Get Recommendation</button>
</div>
</div>
</form>

{% if prediction %}
<div class="card bg-dark" style="width: 18rem;">
<img src="{{url_for('static', filename='img.jpg')}}" class=
"card-img-top" alt="...">
<div class="card-body">
<h5 class="card-title">Recommend Crop</h5>
<p class="card-text">{{prediction }}</p>
</div>
</div>
{% endif %}
```

```
</div>
</body>
</html>
```

6.2 Conclusion

In conclusion, the implementation chapter of the crop recommendation project documentation showcases the successful execution of the project's development and deployment stages. Through meticulous planning and effective collaboration, the project team achieved the translation of the crop recommendation system from concept to reality. The chapter highlights the selection of appropriate technologies, adherence to best practices, and the implementation of robust testing procedures. The successful implementation of the project demonstrates the team's technical expertise and dedication to delivering a reliable and efficient solution that can revolutionize agricultural practices.

Chapter 7

TESTING

7.1 White-Box Testing

White box testing is a testing case design method that uses the control structure of the procedure design to derive test cases. All independents path in a module are exercised at least once, all logical decisions are exercised at once, execute all loops at boundaries and within their operational bounds exercise internal data structure to ensure their validity. Here the customer is given three chances to enter a valid choice out of the given menu. After which the control exits the current menu.

7.2 Black-Box Testing

Black Box Testing attempts to find errors in following areas or categories, incorrect or missing functions, interface error, errors in data structures, performance error and initialization and termination error. Here all the input data must match the data type to become a valid entry.

The following are the different tests at various levels:

7.3 Unit Testing

Unit testing is essentially for the verification of the code produced during the coding phase and the goal is test the internal logic of the[5] module/program. In the Generic code project, the unit testing is done during coding phase of data entry forms whether the functions are working properly or not. In this phase all the drivers are tested they are rightly connected or not.

7.4 Integration Testing

All the tested modules are combined into sub systems, which are then tested. The goal is to see if the modules are properly integrated, and the emphasis being on the testing interfaces between the modules. In the generic code integration testing is done mainly on table creation module and insertion module.

7.5 Validation Testing

This testing concentrates on confirming that the software is error-free in all respects. All the specified validations are verified and the software is subjected to hard-core testing. It also aims at determining the degree of deviation that exists in the software designed from the specification; they are listed out and are corrected.

7.6 System Testing

This testing is a series of different tests whose primary is to fully exercise the computerbased system. This involves:

- Implementing the system in a simulated production environment and testing it.
- Introducing errors and testing for error handling.

7.7 Conclusion

Software testing is an important part of the software development process. It is not a single activity that takes place after code implementation, but is part of each stage of the lifecycle. A Successful test strategy will begin with consideration during requirements specification.

Chapter 8

SCREENSHOTS

8.1 Home Page

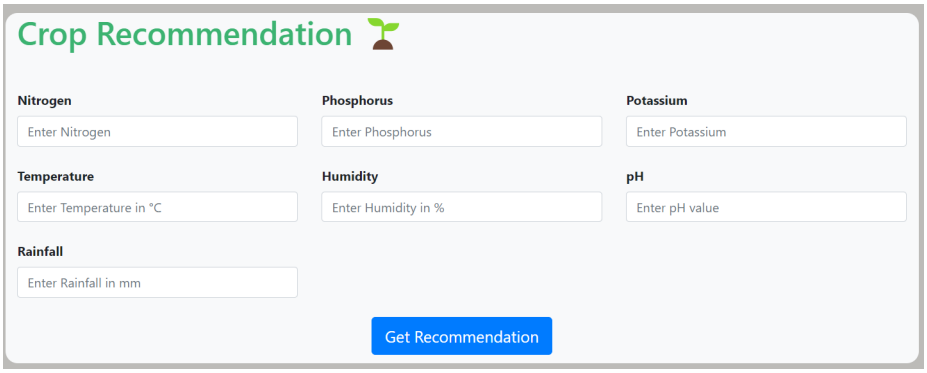

Figure 8.1: Home page of the Crop Recommendation System

8.2 User Inputs-1

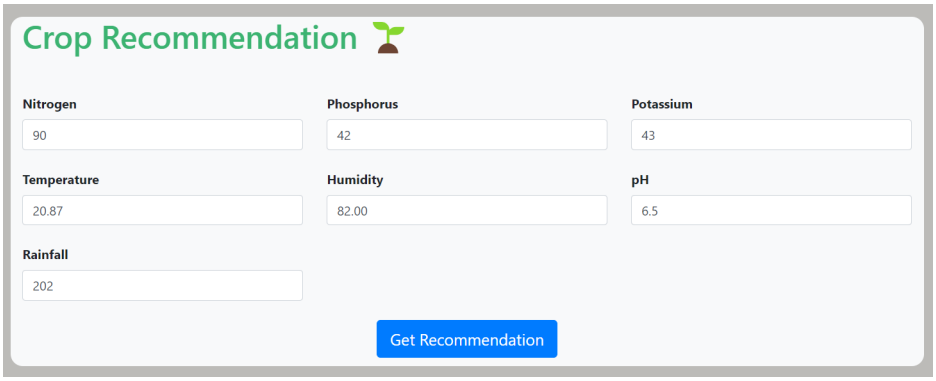

Figure 8.2: User-1 Inputs to Get Recommendation

8.3 Result-1

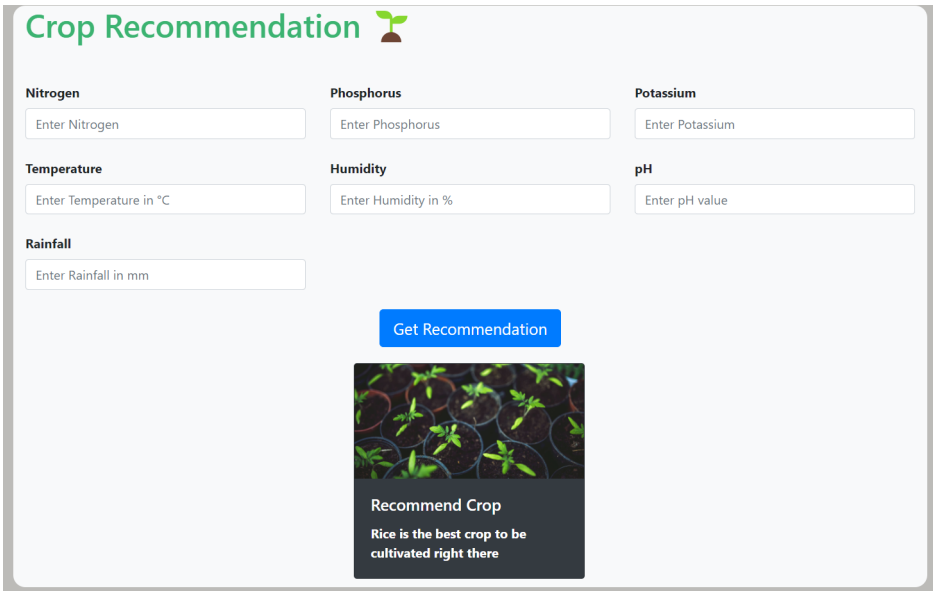

Figure 8.3: Crop to be Cultivated for the Specific Inputs

8.4 User Inputs-2

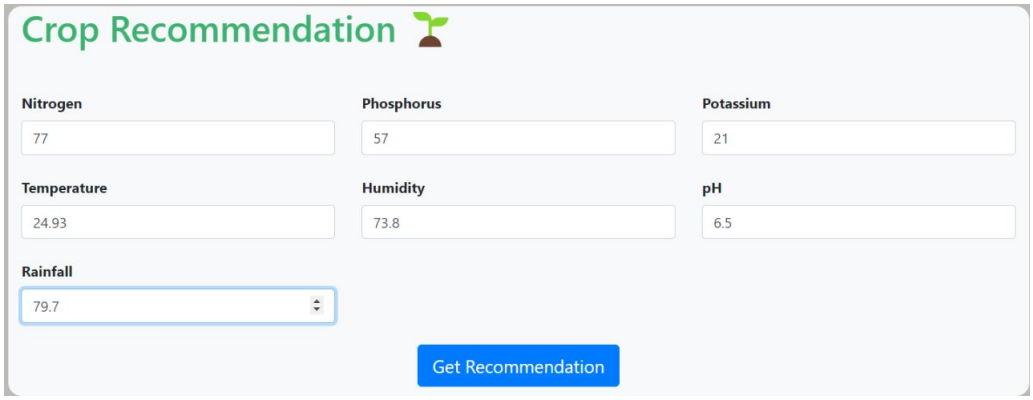

Figure 8.4: User-2 Inputs to Get Recommendation

8.5 Result-2

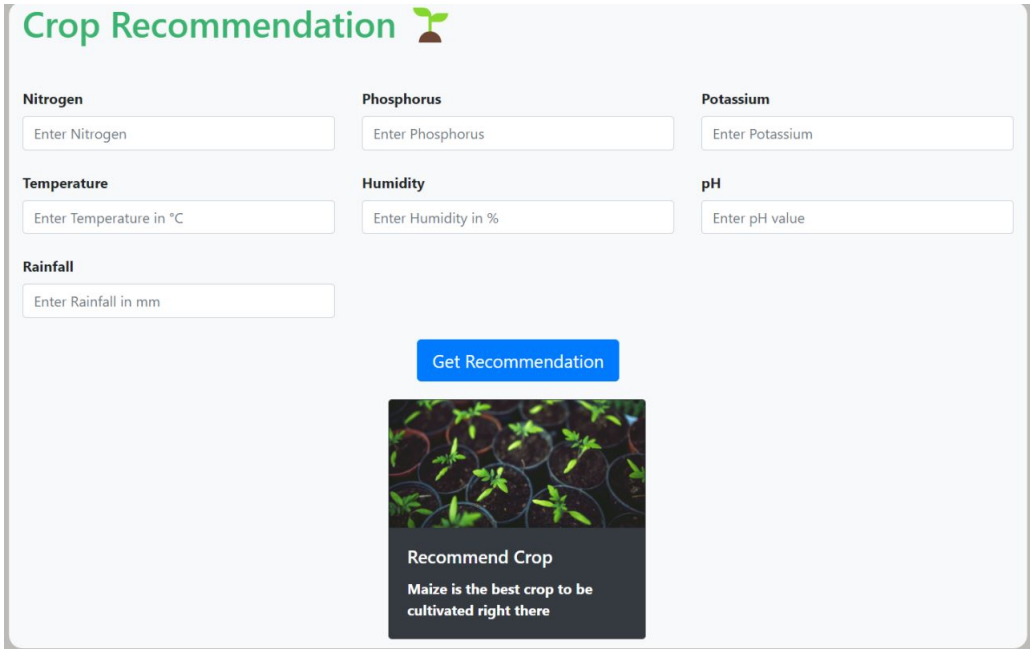

Figure 8.5: Crop to be Cultivated for the Specific Inputs

8.6 Conclusion

The inclusion of screenshots in the crop recommendation project documentation serves as a valuable visual representation of the system's user interface and functionality. These screenshots provide a clear and concise overview of the different screens, menus, and features of the application. The screenshots also serve as a reference point for future

development or enhancements, aiding in the documentation and communication of the project's design and implementation. Overall, the screenshots contribute to the comprehensive documentation of the crop recommendation project, enhancing its clarity and usability.

CONCLUSION

In conclusion, the crop recommendation project utilizing machine learning holds immense potential for enhancing agricultural practices. By analyzing various data inputs such as soil quality, weather patterns, and historical crop yields, machine learning algorithms can generate accurate recommendations for farmers. This technology enables farmers to optimize their crop selection, maximize productivity, and minimize resource wastage. With its ability to adapt to changing conditions and provide tailored suggestions, machine learning empowers farmers to make informed decisions, leading to sustainable agriculture and increased profitability. The integration of machine learning in crop recommendation systems marks a significant step towards achieving efficient and effective farming practices in the modern era.

REFERENCES

[1] *R. Shirsath, N. Khadke, D. More, P. Patil and H. Patil, "Agriculture decision support system using data mining," 2017 International Conference on Intelligent Computing and Control (I2C2), 2017, pp. 1-5, doi: 10.1109/I2C2.2017.8321888.*

[2] *Y. Gandge and Sandhya, "A study on various data mining techniques for crop yield prediction," 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), 2017, pp. 420-423, doi: 10.1109/ICEECCOT.2017.8284541.*

[3] *https://www.geeksforgeeks.org/crop-monitoring-smart-farming-using-iot/*

[4] *https://www.javatpoint.com/machine-learning-random-forest-algorithm*