AGRO-INTEL: AN INTEGRATED AI SOLUTION FOR AGRICULTURAL NEEDS

A PROJECT REPORT

Submitted by

UJJWAL KUMAR [Reg No: RA1811003010669]
KUNAL BHATNAGAR [Reg No: RA1811003010682]

Under the guidance of

Dr. D. Malathi

(Professor, Department of Computing Technologies)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

S.R.M. Nagar, Kattankulathur, Kancheepuram District

MAY 2022

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that 18CSP109L project report titled **“AGRO-INTEL: AN INTEGRATED AI SOLUTION FOR AGRICULTURAL NEEDS”** is the bonafide work of “ **UJJWAL KUMAR [Reg No: RA1811003010669]** and **KUNAL BHATNAGAR [Reg No: RA1811003010682]”**, who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

GUIDE
Dr. D. Malathi
Professor
Dept. of Computing Technologies

PANEL HEAD
Dr. K. Senthil Kumar
Professor
Dept. of Computing Technologies

HEAD OF THE DEPARTMENT
Dr. M. Pushpalatha
Professor
Dept. of Computing Technologies

INTERNAL EXAMINER

EXTERNAL EXAMINER

SRM Institute of Science & Technology
School of Computing
Department of Computing Technologies

Own Work Declaration Form

This sheet must be filled in and signed with date along with the student registration number, work will not be marked unless this is done.

To be completed by the student:

Degree/ Course :

Student Name :

Registration Number :

Title of Work :

I / We hereby certify that this work compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this project is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly references / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:
I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.
If you are working in a group, please write your registration numbers and sign with the date for every student in your group.

Signature of the Student(s)

ACKNOWLEDGEMENTS

We express our humble gratitude to **Dr. C. Muthamizhchelvan**, Vice Chancellor, SRM Institute of Science and Technology and **Dr. T. V. Gopal**, Dean, College of Engineering and Technology, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support. We extend our sincere thanks to **Dr. Revathi Venkataraman**, Professor & Chairperson, School of Computing, SRM Institute of Science and Technology, for her invaluable support. We wish to thank **Dr. M. Pushpalatha**, Professor & Head, Department of Computing Technologies, SRM Institute of Science and Technology, for her valuable suggestions and encouragement throughout the period of the project work.

We are extremely grateful to our Panel Head, **Dr. K. Senthil Kumar**, Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for his inputs during the project reviews. We register our immeasurable thanks to our Faculty Advisor, **Ms. D. Vanusha**, Assistant Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to my guide, **Dr. D. Malathi**, Professor, Department of Computing Technologies, for providing me an opportunity to pursue my project under his/her mentorship. She provided me the freedom and support to explore the research topics of my interest.

We sincerely thank staff and students of the Computing Technologies Department, SRM Institute of Science and Technology, for their help during my project work. Finally, we would like to thank my parents, our family members and our friends for their unconditional love, constant support and encouragement.

Ujjwal Kumar

Kunal Bhatnagar

ABSTRACT

Agriculture is the majority source of income for many people not just in the Indian subcontinent but around the world and hence forms the backbone of the economy. Present-day difficulties like unpredictability in weather conditions, water scarcity, and volatility due to demand-supply fluctuations create the need for the farmer to be equipped with modern day techniques. More specifically, topics like less yield of crops due to unpredictable climate, faulty irrigation resources, soil fertility level depletion need to be communicated. Hence there is a requirement to modify the abundant agriculture data into modern day technologies and make them conveniently accessible to farmers. A technique that can be implemented in crop yield prediction is Machine learning. Numerous machine learning techniques like regression, clustering, classification and prediction can be employed in crop yield forecasting. The wide array of available algorithms poses a selection dilemma with reference to the selected crop. Through this research, we aim to provide a clear and concise study to see the impact of different machine learning algorithms on predicting crop yield and irrigation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ML	Machine Learning
ANN	Artificial Neural Networks
DL	Deep Learning
LR	Logistic Regression
RF	Random Forest
XGB	Extreme Gradient Boosting

CHAPTER 1

INTRODUCTION

1.1 Overview

Agriculture is one of the oldest crafts that is still flourishing Technology has been evolving through decades and its contribution in the field of agriculture can play a major role in the development of mankind. The project is destined to be a smart farming platform which is aimed to implement up-to-date methods that uses the data of soil data and weather data. This will then be used for the benefits of the farmers as it can enable them to automate the process for increasing their crop yield.

1.2 Problem Definition

Crop selection, fertilizer recommendation, irrigation planning and crop yield prediction are wide areas where technology can provide ease in agriculture.

The aim of this project is to design a web platform using Machine Learning (ML) techniques to predict suitable crops based on the given parameters and location, predict the fertilizer to be used based on the nutrient deficiency of the soil, implement proper irrigation cycle to avoid under as well as over irrigation and then finally, predict the crop yield in a particular growing season.

1.3 Purpose

Farming is the key sector which influences the nation's economic process. In India, the bulk of the population depends on agriculture for bread and butter. Several new technologies, like ML and Deep Learning (DL) measures are being enforced into agriculture so that it's easier for farmers to grow and maximize their yield. During this project, we have develop a website during which the subsequent module measures were implemented:

- Crop recommendations, Fertiliser recommendations and Irrigation scheduling automation: The user will offer the soil knowledge from their aspect and therefore the application can predict the crop ought the user grow.
- For the fertiliser recommendation application, the user will input the soil knowledge and therefore the kind of crop they're growing, and therefore the application can predict what the soil lacks or has AN way over and can suggest enhancements.
- For the last application, that is the irrigation schedule designing, the user inputs the wetness level of the soil, rain and humidness levels, and therefore the application can predict whether or not the crop wants irrigation or not.

1.4 Scope

- Crop Plantation Schedule Recommendation: Recommendation about the type of crops to be cultivated which is best suited for the respective conditions and create a Schedule Recommendation System that includes recommendations sowing of the seeds, irrigation cycles, providing the required nutrients at particular intervals and finally harvesting the crop.
- Fertilizer Recommendation: The farmer has to give in some inputs like the area of his field, soil type, nutrient values of soil, etc and he will have the right fertilizer.
- Field Irrigation Recommendation: A Machine learning model to predict soil moisture in terms of soil temperature, air temperature and relative humidity.

1.5 Hardware Requirements

Table 1.1: Hardware Requirements

Specifications	**Value**
Device Required	Laptop/Desktop
Operating System	Windows/Mac/Linux/Any
Memory (RAM)	4GB or more
Processor	Minimum 2GHz or more
Internet	Ethernet Connection (LAN), Wireless Adapter (Wi-Fi)

1.6 Software Requirements

Table 1.2: Software Requirements

Specifications	**Package**
Editor	Google Collab/Jupyter Notebook/Kaggle
Framework	Flask
Database	RDBMS

CHAPTER 2

LITERATURE REVIEW

A technique called the Crop Selection Method (CSM) was suggested to address the crop selection problem, optimise the net yield rate of crop over the season, and achieve maximum economic growth of the country. Plantation days and anticipated yield rate are used as inputs to discover a sequence of crops that will produce the most per day over the course of a whole season, according to the programme. [1]

To estimate crop output based on input circumstances, the proposed system uses machine learning and prediction algorithms such as Multiple Linear Regression. The suggested method is put to the test using data gathered from 1997 to 2014. Data is gathered from all 640 districts and 5924 sub districts across India's states. The overall accuracy achieved is 78 percent using polynomial regression. [2]

The method proposed in the paper is a unique way to determine the best nutrients for increasing yield output while retaining soil fertility using time-series soil nutrients enhanced the population initialization technique. The suggested technique aids in the reduction of search space as well as the elimination of missing local optimization parameters. The model employs a genetic algorithm to analyse the data and make recommendations for remote environment optimization. This study suggested an improved genetic algorithm (IGA) for recommending the best nutrient settings for various crops. [3]

This paper recommended the suitable classifier to classify African countries and their crops in the harvested area using algorithms such as Naive Bayes classification, Decision Table classification, PART classification, J48 classification and Lazy classifier IBK. The dataset used, contains 163 crops which are produced in 59 African countries. The outcome demonstrated that the J48 algorithm has performed the best. [4]

The spatial fluctuation of individual soil nutrients in Kerala's Wayanad district was simulated using geostatistical approaches in this research. The soil data included pH, Organic Carbon (OC), Phosphorus (P), Potassium (K), Zinc (Zn), Iron (Fe), Copper (Cu), Manganese (Mn), Boron (B), and Sulphur (S) values, as well as the WGS84 coordinates of sample sites. There was a total of 19275 soil sample sites. With nugget values greater than 0, pH, Boron, and Copper show a smooth slope. With a range of more than 10 kilometres, pH and Boron show modest autocorrelation. Copper and sulphur have a strong autocorrelation with a range of less than 1 kilometre. [5]

The proposed effort was based on historical agricultural output, and crops can be recommended to farmers using a recommendation system. Farmers will be provided recommendations based on the crop output season. A total of 1,20,000 entries from the Tamil Nadu Agriculture Dataset were collected. The user was provided recommendations based on crop productivity and the season in which the crops were grown. [6]

The researchers created a fuzzy logic-based software that provides an appropriate quantity of fertilizers to soil depending on nutrient levels. This research uses the triangle membership function. The fuzzy system calculates the appropriate amount of fertiliser based on the NPK and season values. The outcome demonstrates that the fuzzy logic system was effectively designed and simulated to provide appropriate fertiliser recommendations. [7]

This model incorporates Artificial Neural Networks (ANN) because they can provide extremely precise results and can evaluate and handle a huge quantity of data on soil moisture, pH, and temperature at various locations quickly, which is critical for any systems that need to produce immediate answers. [8]

The methodologies used in the research were time-series analysis, ANN, Multi-Linear Regression, Moving Average. The application is used to control the irrigation system. Inputs are supplied dependent on the crop and its stage. If there was an intruder or if the temperature rose, the farmer would receive a notification. Romyan's technique is used to calculate the amount of water required for each crop. [9]

This study describes an advanced technology-based smart system that uses K-NN algorithm to anticipate a field's irrigation needs by monitoring ground parameters such as soil moisture, temperature-humidity, and water level. Engineering of the framework to effectively deal with the water system process, has been proposed to acquire, communicate, and process the physical parameters of the agricultural area. The proposed method (K-NN) is meant to address the overfitting problem, consuming less memory and making it compatible for all types of low memory devices and has given an accuracy of 93 percent. [10]

Based on the area and crop type Naive Bayesian, Linear Support, Vector Classification, and K Nearest Neighbour algorithms were used to determine the type of fertiliser to use. The numerous types of soil, their healthiness and features, and the link between various types of pesticides and manures and the types of soil they are to be identified with are all needed for this undertaking. The results revealed that KNN had the highest accuracy of 0.8145 in terms of the type of fertiliser to be used, while Naive Bayesian and Linear SVC had 0.759 and 0.777 accuracy, respectively. [11]

2.1 Research Gap

- The most quintessential problem for this present application is the unavailability of data present in real time, most farmers doesn't possess the necessary educational background.
- IoT based application to gather environmental data from agricultural land.
- The proper data analytical techniques to unravel real-time problems such as this this application which relies on the application of crop and weather data.

CHAPTER 3

SYSTEM ARCHITECTURE AND DESIGN

So now we come to System Architecture. It consists of 4 layers - IoT, Machine Learning, Cloud and Network. The IoT level consists of Interior and Exterior sensors. The sensors are used to gather the real-time data of the field and its surroundings. IoT level will basically give the information of the present field that will be used for farming. The output of the IoT level will be used for the Machine Learning layer where our project mainly surrounds. Now let's get to the Cloud Layer. Cloud Layer consists of database clusters about the soil, weather, humidity, temperature, precipitation, wind and other attributes of the past few decades. This data will be extensively used in the ML layer to predict the crop yield.

As you can see in figure 3.1, the data goes from the IoT layer as well as Cloud layer to the ML layer. Here, main crunching of data takes place. Data is pre-processed and visualized to gain insights into the data. ML models are applied to predict our different objectives. Finally, these layers are connected to the end-user through a gateway known as the Network layer. Where they will be able to input their farmland conditions into the model and the model will predict for them.

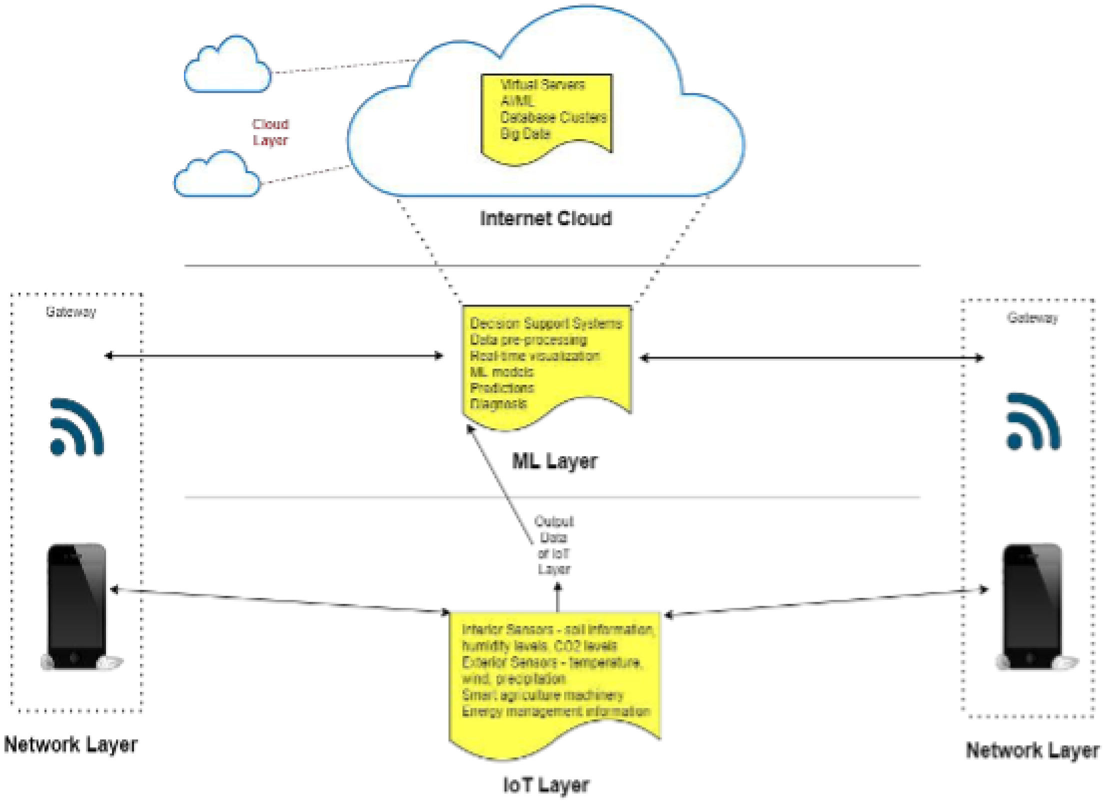

Figure 3.1: System Architecture

CHAPTER 4

METHODOLOGY

4.1 Proposed Idea

The basic idea behind this project is to solve the problem of less transparency and less awareness in the agriculture sector. This project aims to bring up a digital platform where people can come on the platform and from the platform can simply rent out a section of land from our farms and can start producing on their piece of land digitally. This platform will allow users to select the crops they want to sow and generate a request to do this on the application itself. All the requests the user generates are being fulfilled by our farmer working on their land. This project enables maximum transparency to the users by providing regular updates and providing them maximum power to exercise which includes visits to the farm, video and image updates to be received by the user of all the activities performed with the crop.

This project also includes the scope of an embedded recommendation system which assists users to make decisions on what to grow on their land to have a healthy yield. This recommendation system is to have advancements for predicting the crop to sow at a particular time of the year according to the seasonal factors and on top of that it is also enabled with the skills to recommend the crop on the basis of users' health/medical background.

To explain any system, first the problem must be solved in the present system, which is accomplished by stating the system's problem statement and then offering a solution to that research problem, after which subsequent discussion or inferences can be made based on the supplied answer.

Earlier the farmers predicted crop yields based on his previous experience with a particular land and crop. However, this isn't always the case. As a result, our suggested system requires specific input parameters. For these criteria, the farmer's state, district, season, land area, and crop type must be provided. Crop output is estimated for the harvesting year information that is provided by farmers. To estimate crop production based on input circumstances, the suggested system uses a machine learning and prediction algorithm. The projection will be more accurate if the previous year's production is taken into account. The portal must be able to receive and transmit accurate data. The method should assist farmers in cultivating the right crop for higher yields. Farmers will have access to a user-friendly interface.

4.2 Data Description

We collected an open-source dataset that was available on Kaggle. It consists of most of the variables that were required by our predictive model to make recommendations on the crop. The dataset used here focuses only on crop recommendation for the Indian subcontinent. The dataset was augmented by using fertilizer and climate data for India. Data fields:

- NPK ratio with temperature, humidity, pH value and rainfall.

The dataset for the crop selection contains 3100 rows and the following attributes temperature, humidity, pH, label, and rainfall. The dataset contains a total of 31 crops as shown in figure 4.1 The dataset for fertilizer recommendation contains 1843 rows and the following attributes Nitrogen, phosphorus, potassium, and pH level. This dataset has a total of 93 crop labels as shown in figure 4.2 The dataset for irrigation contains 8 attributes – Humidity of Air, Temperature, the Air Pressure, the overall Wind Gust, Speed and Direction, Soil humidity and Irrigation. The dataset is spread over 23,995 entries giving detailed information about irrigation scheduling.

4.3 Methodologies

The first step involves collection of data, data is being collected from several websites like data.gov.in, Kaggle, GitHub etc. The extracted data would contain a lot of noise and missing data, so the next step would include the cleaning and pre-processing of the

```
1 df.head()
```

	N	P	K	temperature	humidity	ph	rainfall	label
0	90	42	43	20.879744	82.002744	6.502985	202.935536	rice
1	85	58	41	21.770462	80.319644	7.038096	226.655537	rice
2	60	55	44	23.004459	82.320763	7.840207	263.964248	rice
3	74	35	40	26.491096	80.158363	6.980401	242.864034	rice
4	78	42	42	20.130175	81.604873	7.628473	262.717340	rice

```
1 df.tail()
```

	N	P	K	temperature	humidity	ph	rainfall	label
2195	107	34	32	26.774637	66.413269	6.780064	177.774507	coffee
2196	99	15	27	27.417112	56.636362	6.086922	127.924610	coffee
2197	118	33	30	24.131797	67.225123	6.362608	173.322839	coffee
2198	117	32	34	26.272418	52.127394	6.758793	127.175293	coffee
2199	104	18	30	23.603016	60.396475	6.779833	140.937041	coffee

Figure 4.1: The first 5 and last 5 fields of the dataset of Crop Selection Dataset

collected data. This step will include dealing with missing and outlier data, normalization and standardisation. After the data is cleaned and pre-processed it is bifurcated between train and test sets. The following step involves the training of various predictive algorithms on the train data. Once the models are trained the next step involves the testing of the model which is to be tested on the test data. Once the models are tested and visualised, the model which outperforms is selected and used for the recommendation of crops, fertilizers and requirement of irrigation. On the web framework user inputs, the necessary details like soil type, pH level, humidity etc. This is visible as a flowchart in figure 4.3

The methodology involves the following steps:

- Dataset collection - The dataset was collected from various sources like data.gov.in, Kaggle and GitHub.
- Pre-processing - The pre-processing of data involves converting the raw data into understandable format. The pre-processing includes 4 major steps:
- Cleaning the Data, Maintaining Integrity of the Data, Reducing Data and Transformation.

app > Data > fertilizer.csv

```
,Crop,N,P,K,pH,soil_moisture
0,rice,80,40,40,5.5,30
3,maize,80,40,20,5.5,50
5,chickpea,40,60,80,5.5,60
12,kidneybeans,20,60,20,5.5,45
13,pigeonpeas,20,60,20,5.5,45
14,mothbeans,20,40,20,5.5,30
15,mungbean,20,40,20,5.5,80
18,blackgram,40,60,20,5,60
24,lentil,20,60,20,5.5,90
60,pomegranate,20,10,40,5.5,30
61,banana,100,75,50,6.5,40
62,mango,20,20,30,5,15
63,grapes,20,125,200,4,60
66,watermelon,100,10,50,5.5,70
67,muskmelon,100,10,50,5.5,30
69,apple,20,125,200,6.5,50
74,orange,20,10,10,4,60
75,papaya,50,50,50,6,20
88,coconut,20,10,30,5,45
93,cotton,120,40,20,5.5,70
94,jute,80,40,40,5.5,20
```

Figure 4.2: Excerpt from Fertilizer Data

Figure 4.3: Process Flowchart

- Classification - The pre-processed data was then split into train/test data and various classification models were applied on the train data. The performance of the models was assessed based on the accuracy and the model outperforming the dataset was selected to make further predictions on the given input data.
- Prediction - In this step the test data is passed from the selected ML/DL model. The model is further hyper tuned to reduce the error and get most efficient output.
- Result - The final results which is the output from the predictive model involves the recommendation of crop and fertilizer and also suggest the need of irrigation in the farm.

4.4 Machine Learning Algorithms Experiments

- Logistic Regression (LR) — This is one of the most used supervised machine learning algorithm that uses labelled dataset to make predictions using the sigmoid function.
- K-Nearest-Neighbour - K Nearest Neighbour algorithm is a versatile algorithm that falls under the Supervised Learning category. This algorithm is mainly used for classification. This flexible algorithm can also be used for imputing missing values and resampling datasets.

- Decision Tree Classifier — This algorithms is good for classification problems. Each node in the tree represents a variable and each connecting lines represents a decision the tree makes based on the impurity of the node.
- Gaussian-NB — Naive Bayes classifier is based on the bayes theorem of probability and uses the gaussian function. It is called a Naive algorithm because it assumes that all variables that are used here are independent of each other.
- Random Forest Classifier — It is like the Decision Tree classifier but is an ensemble method where a number of Decision Trees are used instead of one. Here the decision of majority of the trees are used to make an accurate prediction.
- Artificial Neural Networks — Neural networks are part of a much broader spectrum in Deep Learning. These networks usually have a lot of interconnected nodes which possesses weight and biases on them. They are optimized using the gradient decent algorithm to reach the global minima and minimizes the loss function accurately.

4.5 Performance/Evaluation Measures Used

- Confusion Matrix: It gives us a real view of how well the model is performing and therefore the styles of errors it's making. Figure 4.5 shows the generated confusion matrix for the module that recommends the crop to the farmer. This confusion matrix is of dimension 22*22 where N is the number of crops. This matrix shows the comparison of actual crops to the crops predicted by the model. Figure 4.6 shows the confusion matrix of dimension 2*2 for the irrigation recommendation model which demonstrates a comparison of actual need of irrigation to the predicted need of irrigation. Figure 4.4 gives us how the values in a confusion matrix are divied.

- Accuracy: Accuracy represents the share of images that were correctly classified by the model.
- Precision: The metric of precision gives a concise metric which is better than accuracy, and denotes the accuracy of all positive predictions made by the model.
- Recall: Recall or Sensitivity represents the true positive rate of the model or the ratio of positive images that are correctly detected by the model.
- Specificity: Specificity represents the true negative rate of the model and goes hand in hand with recall or Sensitivity.

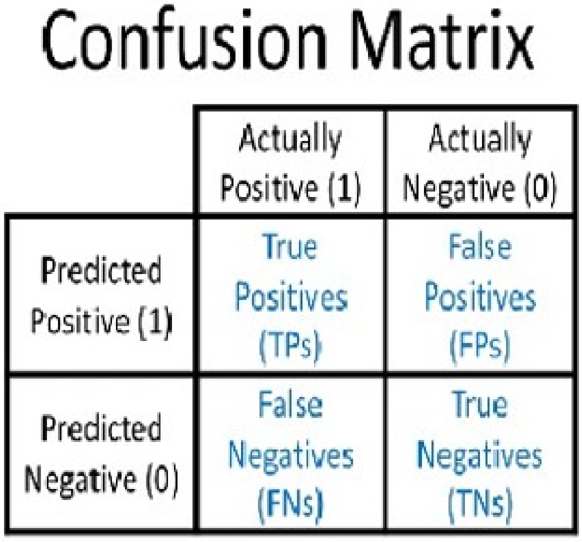

Figure 4.4: Confusion Matrix Schematics

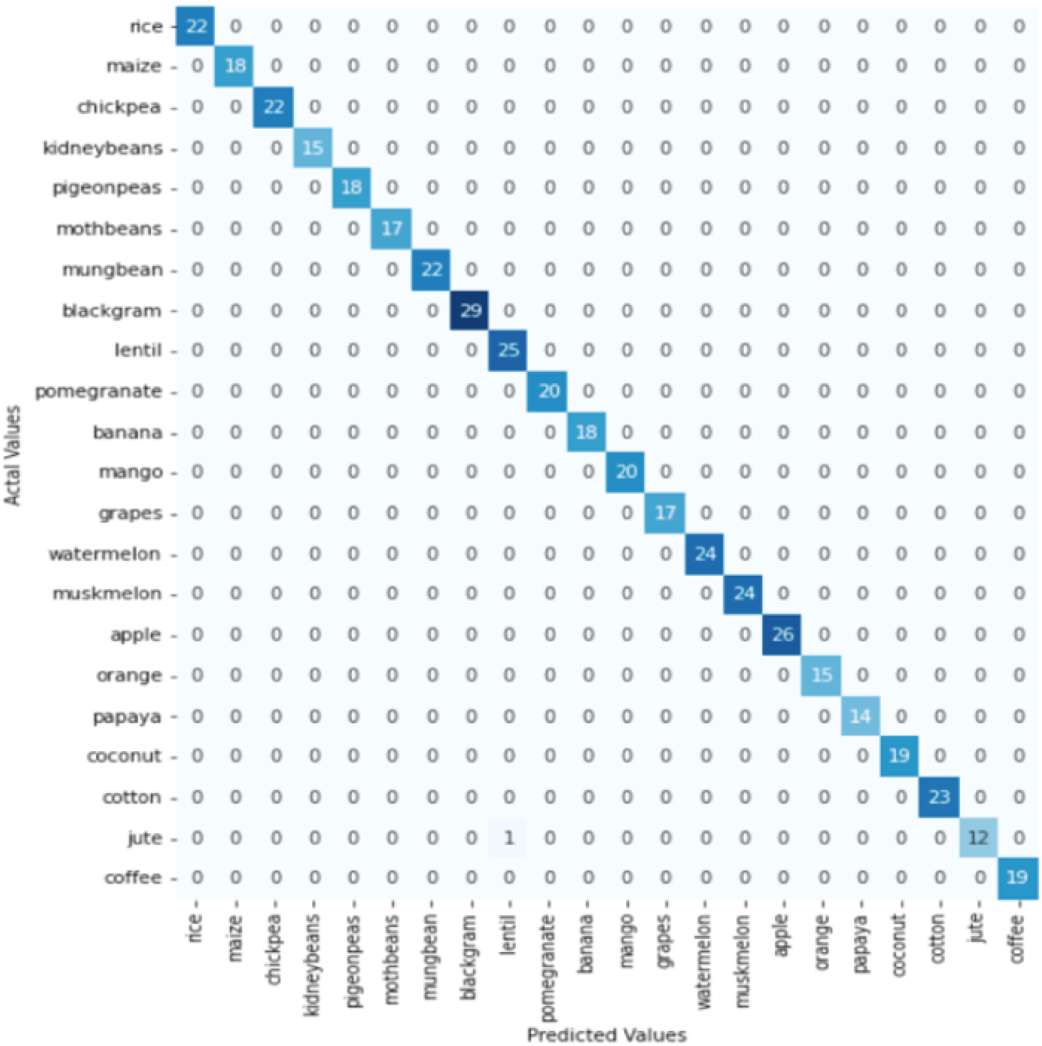

Figure 4.5: Confusion Matrix for Crop Prediction

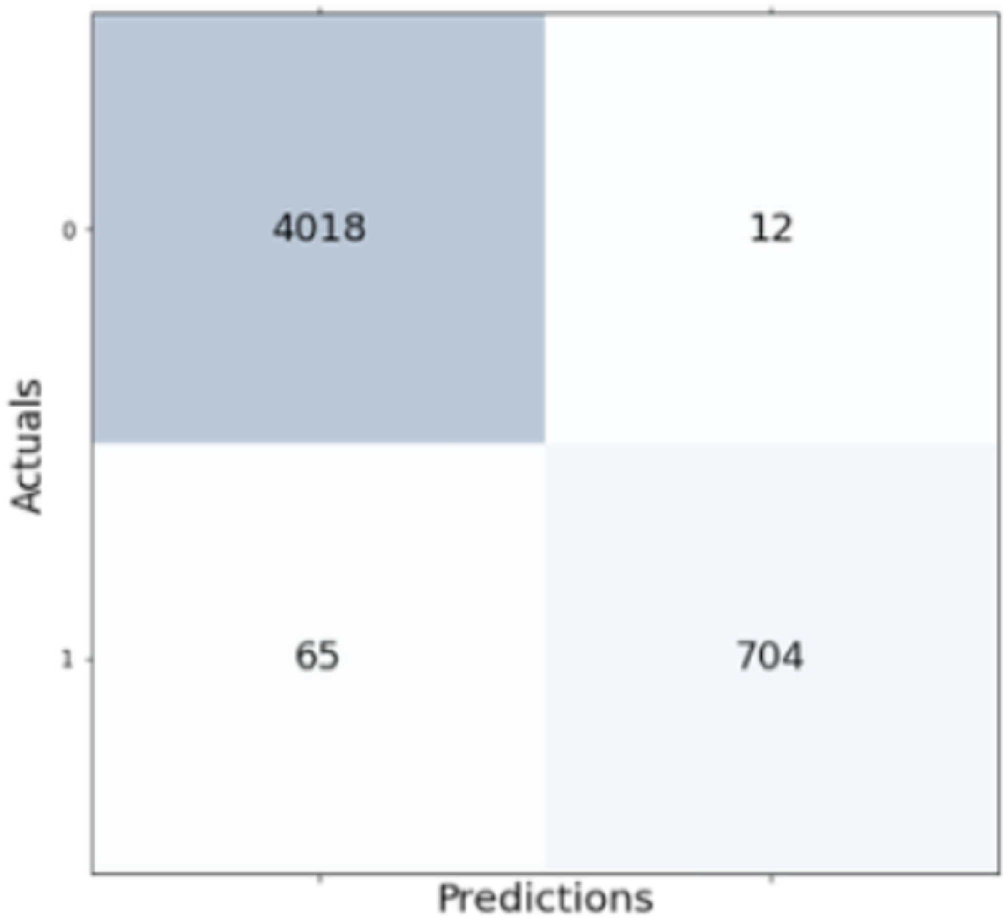

Figure 4.6: Confusion Matrix for Irrigation Recommendation

CHAPTER 5

CODING AND TESTING

```
import torch
import torch.nn as nn
import torch.nn.functional as F


def ConvBlock(in_channels, out_channels, pool=False):
    layers = [nn.Conv2d(in_channels, out_channels, kernel_size=3, padding=1),
              nn.BatchNorm2d(out_channels),
              nn.ReLU(inplace=True)]
    if pool:
        layers.append(nn.MaxPool2d(4))
    return nn.Sequential(*layers)


# Model Architecture
class ResNet9(nn.Module):
    def __init__(self, in_channels, num_diseases):
        super().__init__()

        self.conv1 = ConvBlock(in_channels, 64)
        self.conv2 = ConvBlock(64, 128, pool=True) # out_dim : 128 x 64 x 64
        self.res1 = nn.Sequential(ConvBlock(128, 128), ConvBlock(128, 128))

        self.conv3 = ConvBlock(128, 256, pool=True) # out_dim : 256 x 16 x 16
        self.conv4 = ConvBlock(256, 512, pool=True) # out_dim : 512 x 4 x 44
        self.res2 = nn.Sequential(ConvBlock(512, 512), ConvBlock(512, 512))

        self.classifier = nn.Sequential(nn.MaxPool2d(4),
                                        nn.Flatten(),
                                        nn.Linear(512, num_diseases))

    def forward(self, xb): # xb is the loaded batch
        out = self.conv1(xb)
        out = self.conv2(out)
        out = self.res1(out) + out
        out = self.conv3(out)
        out = self.conv4(out)
        out = self.res2(out) + out
        out = self.classifier(out)
        return out
```

```python
# Importing essential libraries and modules

from flask import Flask, render_template, request, Markup
import numpy as np
import pandas as pd
from utils.fertilizer import fertilizer_dic
import requests
import config
import pickle
import io
# ===============================================================================================
```

```python
# render fertilizer recommendation form page


@ app.route('/fertilizer')
def fertilizer_recommendation():
    title = 'AgroIntel - Fertilizer Recommendation'

    return render_template('fertilizer.html', title=title)


@ app.route('/irrigation')
def irr_recommendation():
    title = 'AgroIntel - Irrigation Recommendation'

    return render_template('irrigation.html', title=title)

# ===============================================================================================
```

```python
# ----------------------LOADING THE TRAINED MODELS -----------------------------------------

# Loading crop recommendation model

import pickle

crop_recommendation_model_path = 'models/crop_model.pkl'
crop_recommendation_model = pickle.load(
    open(crop_recommendation_model_path, 'rb'))


irr_recommendation_model_path = 'models/irr_model.pkl'
irr_recommendation_model = pickle.load(
    open(irr_recommendation_model_path, 'rb'))

```

```python
# Custom functions for calculations


def weather_fetch(city_name):
    """
    Fetch and returns the temperature and humidity of a city
    :params: city_name
    :return: temperature, humidity
    """
    api_key = config.weather_api_key
    base_url = "http://api.openweathermap.org/data/2.5/weather?"

    complete_url = base_url + "appid=" + api_key + "&q=" + city_name
    response = requests.get(complete_url)
    x = response.json()

    if x["cod"] != "404":
        y = x["main"]

        temperature = round((y["temp"] - 273.15), 2)
        humidity = y["humidity"]
        return temperature, humidity
    else:
        return None
```

```python
# ---------------------------------- FLASK APP ----------------------------------------------


app = Flask(__name__)

# render home page


@ app.route('/')
def home():
    title = 'AgroIntel - Home'
    return render_template('index.html', title=title)

# render crop recommendation form page


@ app.route('/crop-recommend')
def crop_recommend():
    title = 'AgroIntel - Crop Recommendation'
    return render_template('crop.html', title=title)

# render fertilizer recommendation form page


@ app.route('/fertilizer')
def fertilizer_recommendation():
    title = 'AgroIntel - Fertilizer Recommendation'

    return render_template('fertilizer.html', title=title)


@ app.route('/irrigation')
def irr_recommendation():
    title = 'AgroIntel - Irrigation Recommendation'

    return render_template('irrigation.html', title=title)

```

```python
# render crop recommendation result page


@ app.route('/crop-predict', methods=['POST'])
def crop_prediction():
    title = 'AgroIntel - Crop Recommendation'

    if request.method == 'POST':
        N = float(request.form['nitrogen'])
        P = float(request.form['phosphorous'])
        K = float(request.form['pottasium'])
        ph = float(request.form['ph'])
        rainfall = float(request.form['rainfall'])

        # state = request.form.get("stt")
        city = request.form.get("city")

        if weather_fetch(city) != None:
            temperature, humidity = weather_fetch(city)
            data = np.array([[N, P, K, temperature, humidity, ph, rainfall]])
            my_prediction = irr_recommendation_model.predict(data)
            final_prediction = my_prediction[0]


            return render_template('crop-result.html', prediction=final_prediction, title=title)

        else:

            return render_template('try_again.html', title=title)

# render irrigation recommendation result page
```

```
# render irrigation recommendation result page


@ app.route('/irr-predict', methods=['POST'])
def irr_prediction():
    title = 'AgroIntel - Irrigation Recommendation'

    if request.method == 'POST':
        air_temp = float(request.form['air_temp'])
        humidity = float(request.form['humidity'])
        pressure = float(request.form['pressure'])
        wind_speed = float(request.form['wind_speed'])
        wind_gust = float(request.form['wind_gust'])
        wind_dir = float(request.form['wind_dir'])
        soil_humidity = float(request.form['soil_humidity'])


        data = np.array([[air_temp, humidity, pressure, wind_speed, wind_gust, wind_dir, soil_humidity]])
        my_prediction_irr = irr_recommendation_model.predict(data)
        final_prediction = my_prediction_irr[0]
        if final_prediction==0:
            my_final_prediction = "You do not need to irrigate the field"
        else:
            my_final_prediction = "You need to irrigate the field"

        return render_template('irrigation-result.html', prediction=my_final_prediction, title=title)

```

```python
@ app.route('/fertilizer-predict', methods=['POST'])
def fert_recommend():
    title = 'AgroIntel - Fertilizer Recommendation'

    crop_name = str(request.form['cropname'])
    N = float(request.form['nitrogen'])
    P = float(request.form['phosphorous'])
    K = float(request.form['pottasium'])
    ph = float(request.form['ph'])

    df = pd.read_csv('Data/fertilizer.csv')

    nr = df[df['Crop'] == crop_name]['N'].iloc[0]
    pr = df[df['Crop'] == crop_name]['P'].iloc[0]
    kr = df[df['Crop'] == crop_name]['K'].iloc[0]

    n = nr - N
    p = pr - P
    k = kr - K
    temp = {abs(n): "N", abs(p): "P", abs(k): "K"}
    max_value = temp[max(temp.keys())]
    if max_value == "N":
        if n < 0:
            key = 'NHigh'
        else:
            key = "Nlow"
    elif max_value == "P":
        if p < 0:
            key = 'PHigh'
        else:
            key = "Plow"
    else:
        if k < 0:
            key = 'KHigh'
        else:
            key = "Klow"

    response = Markup(str(fertilizer_dic[key]))

    return render_template('fertilizer-result.html', recommendation=response, title=title)
```

```
<!DOCTYPE html>
<html lang="en">

<head>
    <title>{{ title }}</title>
    <link rel="shortcut icon" href="{{ url_for('static', filename='images/favicon.ico') }}"/>

    <!-- for-mobile-apps -->
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta charset="utf-8">
    <meta name="keywords" content="Agro Harvest Responsive web template, Bootstrap Web Templates, Flat Web Templates,
Smartphone Compatible web template, free webdesigns for Nokia, Samsung, LG, SonyEricsson, Motorola web design" />

    <style>
        html {
            font-size: 1rem;
        }

        @media (min-width: 576px) {
            html {
                font-size: 1.25rem;
            }
        }

        @media (min-width: 768px) {
            html {
                font-size: 1.5rem;
            }
        }

        @media (min-width: 992px) {
            html {
                font-size: 1.75rem;
            }
        }

        @media (min-width: 1200px) {
            html {
                font-size: 2rem;
            }

```

```python
import torch
import torch.nn as nn
import torch.nn.functional as F


def ConvBlock(in_channels, out_channels, pool=False):
    layers = [nn.Conv2d(in_channels, out_channels, kernel_size=3, padding=1),
              nn.BatchNorm2d(out_channels),
              nn.ReLU(inplace=True)]
    if pool:
        layers.append(nn.MaxPool2d(4))
    return nn.Sequential(*layers)


# Model Architecture
class ResNet9(nn.Module):
    def __init__(self, in_channels, num_diseases):
        super().__init__()

        self.conv1 = ConvBlock(in_channels, 64)
        self.conv2 = ConvBlock(64, 128, pool=True) # out_dim : 128 x 64 x 64
        self.res1 = nn.Sequential(ConvBlock(128, 128), ConvBlock(128, 128))

        self.conv3 = ConvBlock(128, 256, pool=True) # out_dim : 256 x 16 x 16
        self.conv4 = ConvBlock(256, 512, pool=True) # out_dim : 512 x 4 x 44
        self.res2 = nn.Sequential(ConvBlock(512, 512), ConvBlock(512, 512))

        self.classifier = nn.Sequential(nn.MaxPool2d(4),
                                        nn.Flatten(),
                                        nn.Linear(512, num_diseases))

    def forward(self, xb): # xb is the loaded batch
        out = self.conv1(xb)
        out = self.conv2(out)
        out = self.res1(out) + out
        out = self.conv3(out)
        out = self.conv4(out)
        out = self.res2(out) + out
        out = self.classifier(out)
        return out
```

```css
            html {
                font-size: 1rem;
            }

            h1 {
                font-size: 1.2rem;
            }

            h2 {
                font-size: 1.1rem;
            }

            @media (min-width: 768px) {
                html {
                    font-size: 1.1rem;
                }

                h1 {
                    font-size: 1.3rem;
                }

                h2 {
                    font-size: 1.2rem;
                }
            }

            @media (min-width: 991px) {
                html {
                    font-size: 1.2rem;
                }

                h1 {
                    font-size: 1.5rem;
                }

                h2 {
                    font-size: 1.4rem;
                }
            }

```

```
            @media (min-width: 1200px) {
                html {
                    font-size: 1.2rem;
                }

                h1 {
                    font-size: 1.7rem;
                }

                h2 {
                    font-size: 1.6rem;
                }
            }

        }
    </style>
    <script>
        addEventListener("load", function () {
            setTimeout(hideURLbar, 0);
        }, false);

        function hideURLbar() {
            window.scrollTo(0, 1);
        }

    </script>

    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
        integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
        crossorigin="anonymous"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
        integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
        crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
        integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
        crossorigin="anonymous"></script>
```

```html
        crossorigin="anonymous"></script>
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
        integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
        crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
        integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
        crossorigin="anonymous"></script>
    </body>
    <!-- css files -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
        integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
    <link href="{{ url_for('static', filename='css/bootstrap.css') }}" rel='stylesheet' type='text/css' />
    <!-- bootstrap css -->
    <link href="{{ url_for('static', filename='css/style.css') }}" rel='stylesheet' type='text/css' />
    <!-- custom css -->
    <link href="{{ url_for('static', filename='css/font-awesome.min.css') }}" rel="stylesheet"><!-- fontawesome css -
    <!-- //css files -->
    <!-- <link rel="icon" type="image/png" href="{{ url_for('static', filename='images/favicon.png?') }}"> -->


    <script type="text/JavaScript" src="{{ url_for('static', filename='scripts/cities.js') }}"></script>


    <!-- google fonts -->
    <link href="//fonts.googleapis.com/css?family=Thasadith:400,400i,700,700i&amp;subset=latin-ext,thai,vietnamese"
        rel="stylesheet">
    <!-- //google fonts -->

    <style>
        header {
            background-color: rgba(30, 30, 30, 1);
            margin-top: 0rem;
            display: block;

        }
    </style>
</head>

<body>
```

```
<!-- Navigation -->
<nav class="navbar navbar-expand-lg navbar-dark bg-dark static-top" style="background-color: #1C00ff00;">
    <div class="container">
        <a class="navbar-brand" href="{{ url_for('home') }}">
            <img src="{{ url_for('static', filename='images/logo.png') }}" style="width: 90px; height: 55px" alt='
        </a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarResponsive"
            aria-controls="navbarResponsive" aria-expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarResponsive">
            <ul class="navbar-nav ml-auto">
                <li class="nav-item">
                    <a class="nav-link" href="{{ url_for('home') }}">Home
                        <span class="sr-only">(current)</span>
                    </a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="{{ url_for('crop_recommend') }}">Crop</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="{{ url_for('fertilizer_recommendation') }}">Fertilizer</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="{{ url_for('irr_recommendation') }}">Irrigation</a>
                </li>
            </ul>
        </div>
    </div>
</nav>
```

```
    {% block body %} {% endblock %}

    <!-- footer -->
    <footer class="text-center py-5">
        <div class="container py-md-3">
            <!-- logo -->
            <h2 class="logo2 text-center">
                <a href="{{ url_for('home') }}">
                    AgroIntel
                </a>
            </h2>
            <!-- //logo -->
            <!-- address -->
            <div class="contact-left-footer mt-4">





                <!-- <a href="community.html">Community</a> -->
                </p>
            </div>
            <div class="w3l-copy text-center">
                <p class="text-da">NEXT GEN FARMING<br> </p>
            </div>
            <p class="homelogo">

        </div>
    </footer>
    <!-- //footer -->

    <!-- move top icon -->
    <a href="#home" class="move-top text-center"></a>
    <!-- //move top icon -->
</body>

</html>
```

CHAPTER 6

RESULTS AND OBSERVATIONS

Various machine learning algorithms were implemented and their performance was checked and evaluated on the custom dataset. After comparing each of the models performance, it was found out that the Random Forest (RF) Classifier performed the best in terms of accuracy (99.31 percent for crop recommendation and 98.73 percent for irrigation recommendation) and it was chosen to be used as the final model that was incorporated in the web application. The accuracy of various algorithms on both crop and irrigation dataset are visually represented in figures 6.1 and 6.2 respectively.

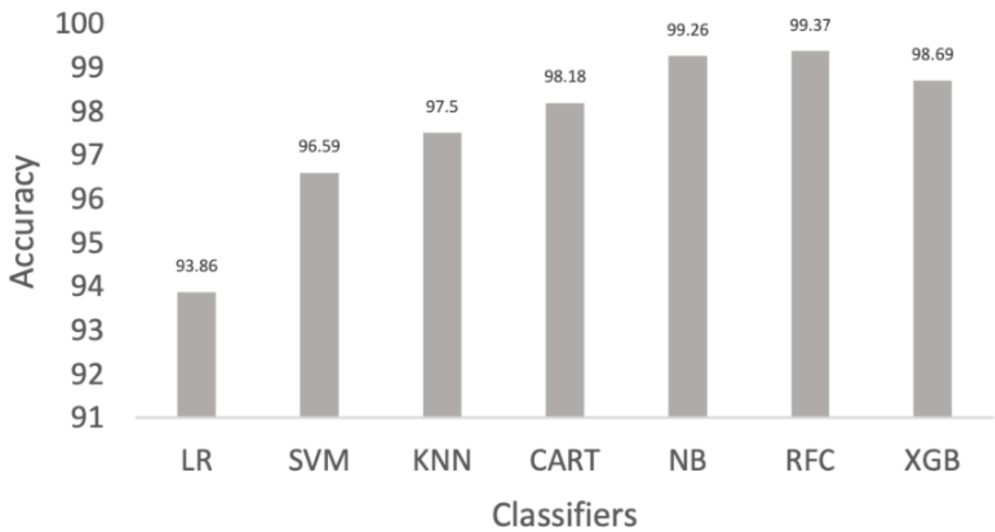

Figure 6.1: Accuracy of Classifiers on Crop Dataset

Figure 6.3 show the actual excerpt of the running website that is developed for recommending crops to the farmers. The end-user is required to input the soil data directly in the application and the AI model running in the backend outputs the best suitable crop that the farmer should sow to get the maximum yield in the season.

Based on the trained model that is running in the backend, the dynamic web-application developed by us can also accurately predict if the soil that is used by the farmer needs irrigation or not. This feature can be used by the end-user by simply

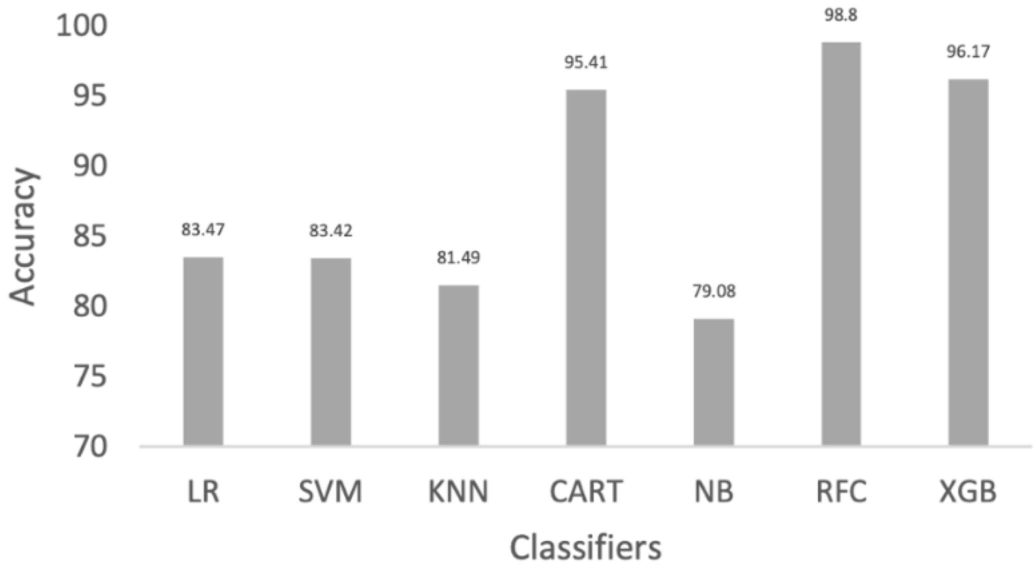

Figure 6.2: Accuracy of Classifiers on Irrigation Dataset

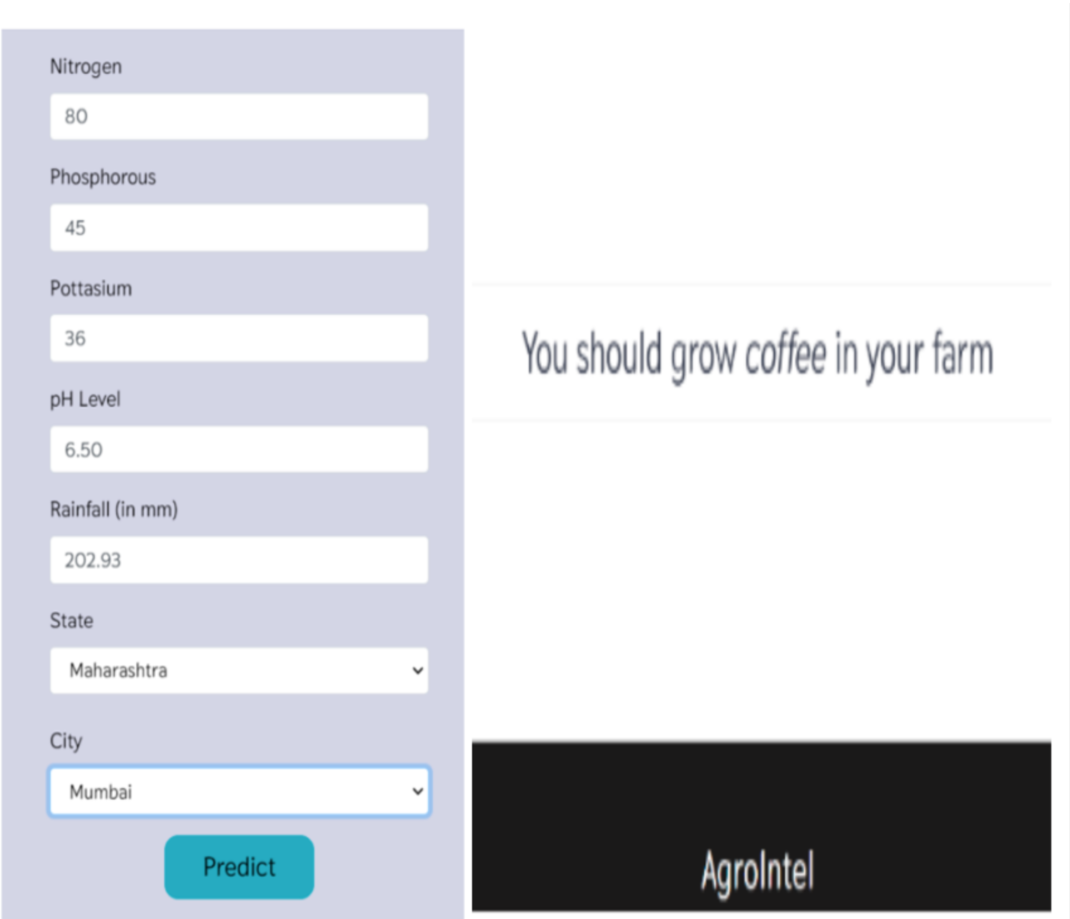

Figure 6.3: Screenshot of Crop Recommendation Module

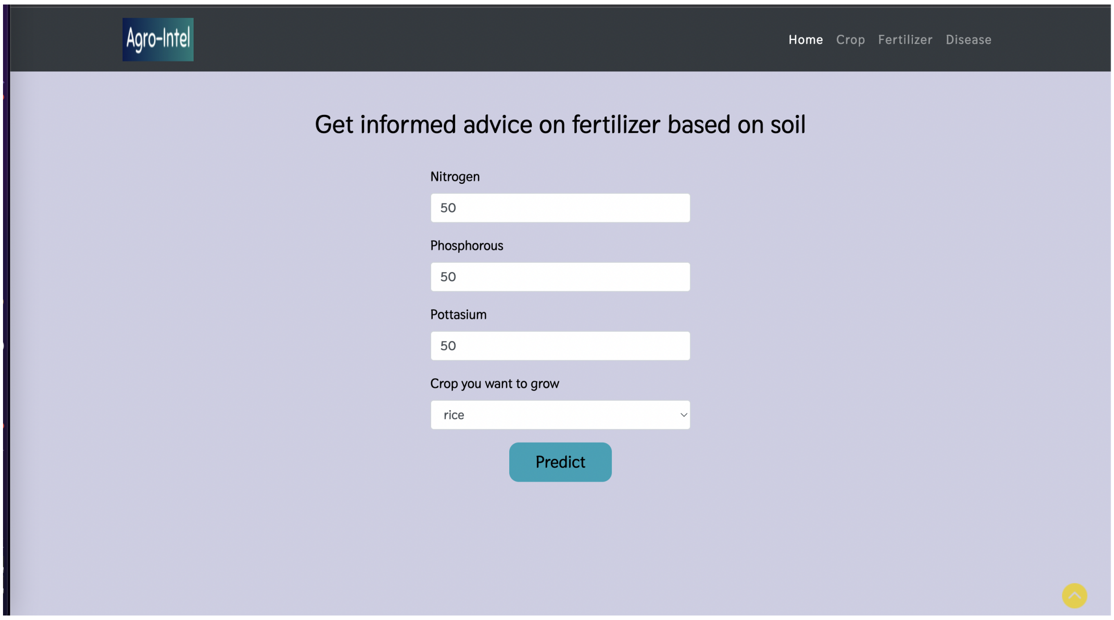

Figure 6.4: Screenshot of Fertilizer Recommendation Module (A)

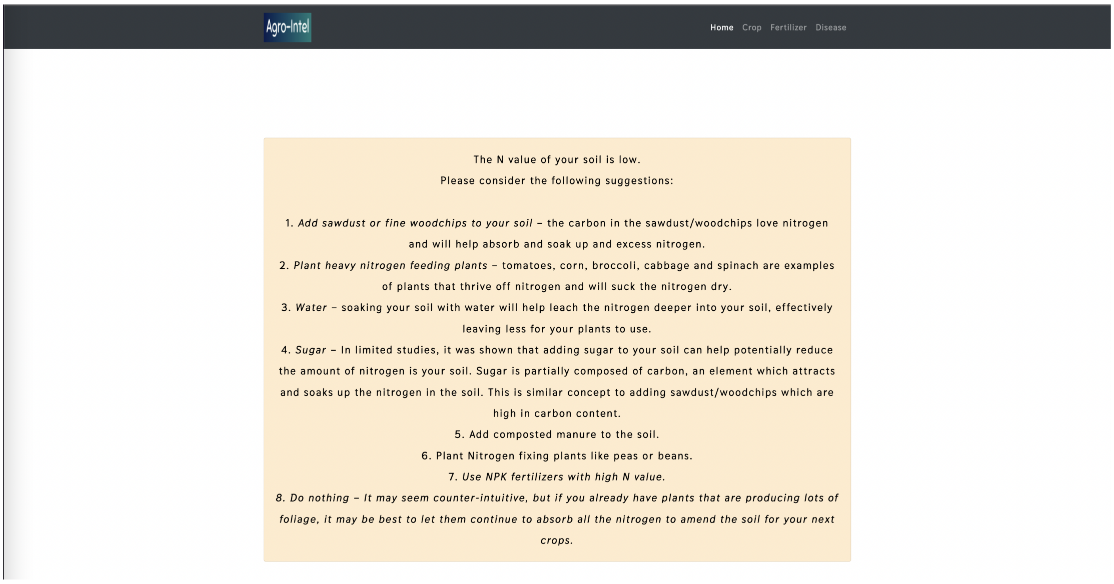

Figure 6.5: Screenshot of Fertilizer Recommendation Module (B)

entering the climatic and soil conditions around the crop. Figure 6.6 shows the user interface of this module and the actual output generated by the model.

The team also worked on developing the fertilizer recommendation module whose excerpt is shown in figures 6.4 and 6.5.

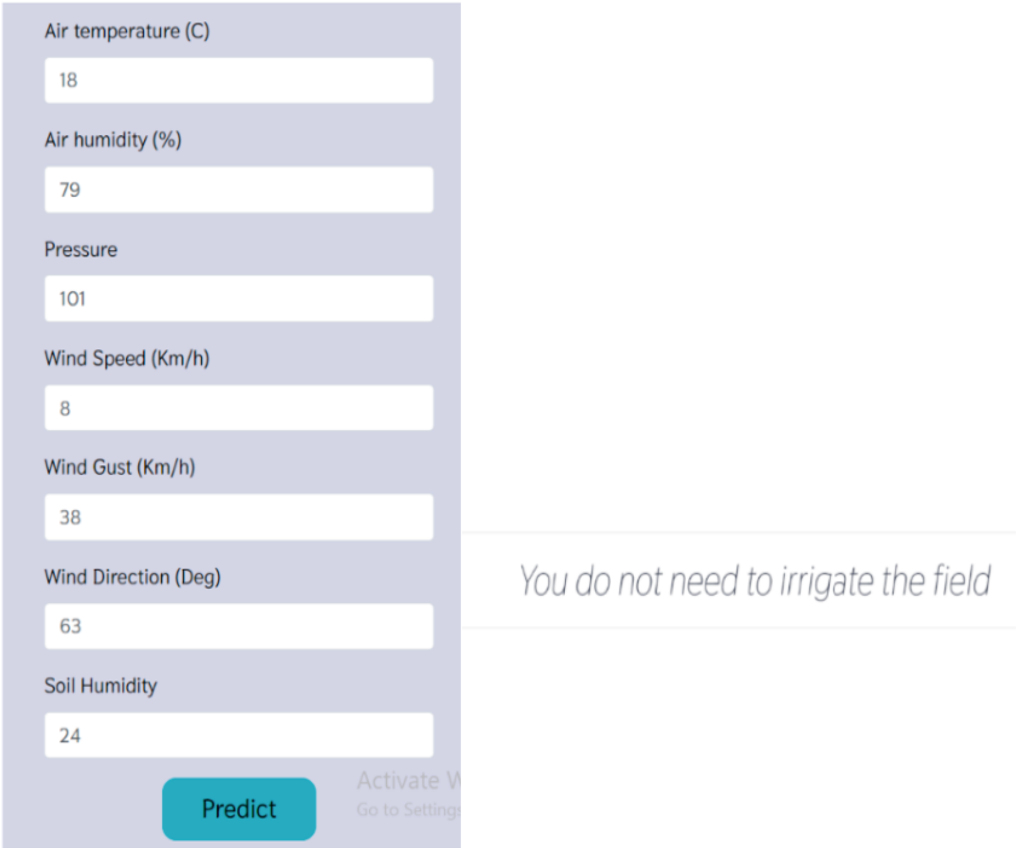

Figure 6.6: Screenshot of Irrigation Recommendation Module

CHAPTER 7

CONCLUSION

Because our farmers are not properly employing technology and analysis at the moment, there is a risk of incorrect crop selection, which will lower their revenue. To avoid such losses, we built a farmer-friendly system with a graphical user interface that would forecast which crop would be the best fit for a given plot of land, as well as information on necessary nutrients, required seeds for cultivation, projected yield, and market price. As a result, farmers are more likely to make the proper selection when choosing a crop to cultivate, resulting in the agricultural industry being improved through creative ideas.

The limitations of current methods and their practical use in yield prediction were emphasised in this research. Then leads the farmers through a feasible yield prediction system, a suggested system that connects farmers via a web application. The app has a number of tools that users may utilise to choose a crop. The built-in predictor technology assists farmers in predicting crop yields. The built-in recommender system helps the user to explore the various crops and their yields in order to make better informed selections.

On the supplied datasets several machine learning algorithms such as LR,RF, ANN, LDA, Extreme Gradient Boosting (XGB) and KNN were applied and assessed for yield to determine accuracy. The proposed algorithm also looked at fertiliser application time and made recommendations for the best duration.

REFERENCES

[1] Kumar, R., Singh, M. P., Kumar, P., and Singh, J. P., 2015, "Crop Selection Method to maximize crop yield rate using machine learning technique," 2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), pp. 138–145.

[2] Reddy, D. and Kumar, M., 2021, "Crop Yield Prediction using Machine Learning Algorithm," 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 1466–1470.

[3] Ahmed, U., Lin, C.-W., Srivastava, G., and Djenouri, Y., 2021, "A nutrient recommendation system for soil fertilization based on evolutionary computation," Computers and Electronics in Agriculture, **189**, p. 106407.

[4] Bisoyi, S. and Panda, A., 2017, "Designing of an Improved Agricultural Classification System," IOSR Journal of Electrical and Electronics Engineering, **12**, pp. 83–86.

[5] Iyer, R. T., Krishnan, M. T., Pillai, S. K. K., and Lalitha, L. P. P., 2019, "Geostatistical Modeling of Soil nutrients and development of a mobile application for site-specific fertilizer recommendation," 2019 IEEE Recent Advances in Geoscience and Remote Sensing : Technologies, Standards and Applications (TENGARSS), pp. 148–150.

[6] Vaishnavi, S., Shobana, M., Sabitha, R., and Karthik, S., 2021, "Agricultural Crop Recommendations based on Productivity and Season," 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), **1**, pp. 883–886.

[7] Haban, J. J. I., Puno, J. C. V., Bandala, A. A., Kerwin Billones, R., Dadios, E. P., and Sybingco, E., 2020, "Soil Fertilizer Recommendation System using Fuzzy Logic," 2020 IEEE REGION 10 CONFERENCE (TENCON), pp. 1171–1175.

[8] Aggarwal, S. and Kumar, A., 2019, "A Smart Irrigation System to Automate Irrigation Process Using IOT and Artificial Neural Network," 2019 2nd International Conference on Signal Processing and Communication (ICSPC), pp. 310–314.

[9] Kondaveti, R., Reddy, A., and Palabtla, S., 2019, "Smart Irrigation System Using Machine Learning and IOT," , pp. 1–11.

[10] Akshay, S. and Ramesh, T. K., 2020, "Efficient Machine Learning Algorithm for Smart Irrigation," 2020 International Conference on Communication and Signal Processing (ICCSP), pp. 867–870.

[11] Kanuru, L., Tyagi, A. K., U, A. S., Fernandez, T. F., Sreenath, N., and Mishra,

S., 2021, “Prediction of Pesticides and Fertilizers using Machine Learning and Internet of Things,” 2021 International Conference on Computer Communication and Informatics (ICCCI), pp. 1–6.

Project Report VER2

ORIGINALITY REPORT

6%	1%	4%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Laasya Kanuru, Amit Kumar Tyagi, Aswathy S U, Terrance Frederick Fernandez, N Sreenath, Shashvi Mishra. "Prediction of Pesticides and Fertilizers using Machine Learning and Internet of Things", 2021 International Conference on Computer Communication and Informatics (ICCCI), 2021 Publication	2%
2	Radhakrishnan Thanu Iyer, Manojkumar Thananthu Krishnan, Sudhis Kumar Krishna Pillai, Lal Prakash Prabhakaran Lalitha. "Geostatistical Modeling of Soil nutrients and development of a mobile application for site-specific fertilizer recommendation", 2019 IEEE Recent Advances in Geoscience and Remote Sensing : Technologies, Standards and Applications (TENGARSS), 2019 Publication	1%
3	iosrjen.org Internet Source	1%

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

Office of Controller of Examinations

REPORT FOR PLAGIARISM CHECK ON THE SYNOPSIS/THESIS/DISSERTATION/PROJECT REPORTS

1	Name of the Candidates **(IN BLOCK LETTERS)**	UJJWAL KUMAR KUNAL BHATNAGAR
2	Address of the Candidate	VA 303, HRC Professionals Hub, Vaibhav Khand, Indirapuram, Ghaziabad, Uttar Pradesh – Ujjwal Kumar 5 Ashirwad Nagar, University Road, Udaipur, Rajasthan – Kunal Bhatnagar **Mobile Number:** 9818300702 (Ujjwal), 9587501959 (Kunal)
3	Registration Number	RA1811003010669 (Ujjwal), RA1811003010682 (Kunal)
4	Date of Birth	26/07/2000 (Ujjwal), 17/02/2000 (Kunal)
5	Department	Computer Science and Engineering
6	Faculty	Engineering and Technology
7	Title of the Synopsis/ Thesis/ Dissertation/Project	Agro-Intel: An Integrated Ai solution for Agricultural Needs
8	Name and address of the Supervisor / Guide	Dr. D. Malathi **Mail ID:** malathid@srmist.edu.in **Mobile Number:** 94425 54055
9	Name and address of the Co-Supervisor / Co- Guide (if any)	NA
10	Software Used	Python, Flask, Django, HTML, CSS, JavaScript
11	Date of Verification	8/05/2022

Date :

12	**Plagiarism Details: (to attach the final report)**			
Chapter	**Title of the Chapter**	**Percentage of similarity index (including self citation)**	**Percentage of similarity index (Excluding self citation)**	**% of plagarism after excluding Quotes, Bibliography, etc.,**
1	INTRODUCTION	1	1	1
2	LITERATURE REVIEW	3	3	3
3	SYSTEM ARCHITECURE AND DESIGN	1	1	1
4	METHODOLOGY	1	1	1
5	CODING AND TESTING	0	0	0
6	RESULTS AND OBSERVATION	0	0	0
7	CONCLUSION	0	0	0
8				
9				
10				
Thesis abstract		0	0	0
Appendices		NA	NA	NA
We declare that the above information have been verified and found true to the best of our knowledge.				
Signature of the Candidate		**Signature of the Supervisor / Guide**		
Signature of the Co-Supervisor/Co-Guide		**Signature of the HOD**		

Date :

Acceptance Letter

To: **Ujjwal Kumar, Kunal Bhatnagar, D Malathi**

Paper id: **ICICCT102**

Title: **Integrated Solution for Crop Yield Prediction, Fertilizer Recommendation and Irrigation Scheduling using Machine Learning Methodologies**

Dear Author,

With the heartiest congratulations, we are happy to inform you that based on double blind review process and the recommendations of the conference review committee, your paper mentioned above has been accepted for publication and oral presentation at the 6th International Conference on Intelligent Computing and Communication Technologies [ICICCT 2022].

ICICCT 2022 is a Springer approved conference and all the registered papers will be recommended for publication in springer "Lecture Notes in Networks and Systems". ICICCT 2022 gives due recognition to great achievements of students, researchers and industrialists in the promotion and effective utilization of their research works. Herewith, the conference committee sincerely invites you for oral presentation at ICICCT 2022 to be held in Namakkal, India, 12-13 May, 2022. For more information on the conference kindly visit ICICCT 2022.

Yours' Sincerely

Dr.G.Ranganathan
Conference Chair ICICCT 2022