RANKING BASED DECISION SUPPORT SYSTEM FOR CROP RECOMMENDATION WITH NUTRIENT ANALYSIS USING IOT AND MACHINE LEARNING ALGORITHMS

A PROJECT REPORT

Submitted by

KARTHIKEYAN S

KISHORE P

KISHORE SHAW M R

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

EASWARI ENGINEERING COLLEGE, CHENNAI

(Autonomous Institution)

affiliated to

ANNA UNIVERSITY :: CHENNAI - 600025

JUNE 2022

EASWARI ENGINEERING COLLEGE, CHENNAI

(AUTONOMOUS INSTITUTION)

AFFILIATED TO ANNA UNIVERSITY, CHENNAI-600025

BONAFIDE CERTIFICATE

Certified that this project report **"RANKING BASED DECISION SUPPORT SYSTEM FOR CROP RECOMMENDATION WITH NUTRIENT ANALYSIS USING IOT AND MACHINE LEARNING ALGORITHMS"** is the bonafide work of **"KARTHIKEYAN S (310618104042), KISHORE P (310618104046), KISHORE SHAW M R (310618104048)"** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.G.S.ANANDHA MALA	**Mr.Y.JUSTIN DHAS**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
	Assistant Professor
Department of Computer	Department of Computer
Science and Engineering	Science and Engineering
Easwari Engineering College	Easwari Engineering College
Ramapuram, Chennai 600089	Ramapuram, Chennai 600089

CERTIFICATE OF EVALUATION

College Name : Easwari Engineering College

Branch & Semester : Computer Science and Engineering & VIII

S.No	Name of the Students	Title of the project	Name of the supervisor with designation
1.	KARTHIKEYAN S (310618104042)	RANKING BASED DECISION SUPPORT SYSTEM FOR CROP RECOMMENDATION WITH NUTRIENT ANALYSIS USING IOT AND MACHINE LEARNING ALGORITHMS	Mr.Y.JUSTIN DHAS ASSISTANT PROFESSOR
2.	KISHORE P (310618104046)		
3.	KISHORE SHAW M R (310618104048)		

The report of the project work submitted by the above students in partial fulfilment for the award of Bachelor of Engineering Degree in Computer Science and Engineering of Anna University were evaluated and confirmed to be a report of the work done by the above students.

The viva voce examination of the project was held on __________________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We hereby place our deep sense of gratitude to our beloved Founder Chairman of the institution, **Dr.T.R.Pachamuthu,B.Sc.,M.I.E.**, for providing us with the requisite infrastructure throughout the course. We would also like to express our gratitude towards our Chairman **Dr.R.Shivakumar,M.D.,Ph.D.,** for giving the necessary facilities.

We convey our sincere thanks to **Dr.R.S.Kumar,M.Tech.,Ph.D.,** Principal, Easwari Engineering College, for his encouragement and support. We extend our hearty thanks to **Dr.V.Elango,M.E.,Ph.D.**, Vice Principal (Academic) and **Dr.S.Nagarajan,M.E.,Ph.D.,** Vice Principal (Admin), Easwari Engineering College for their constant encouragement.

We take the privilege to extend our hearty thanks to **Dr.G.S.Anandha Mala,M.S.,M.E.,Ph.D.,** Head of the Department, Department of Computer Science and Engineering, Easwari Engineering College for her suggestions, support and encouragement towards the completion of the project with perfection.

We would like to express our gratitude to our Project Coordinator, **Mrs.D.Kavitha,M.Tech**.**,** Assistant Professor, Department of Computer Science and Engineering, Easwari Engineering College for her constant support and encouragement.

We would also like to express our gratitude to our guide **Mr.Y.Justin Dhas**,**M.Tech.,** Assistant Professor, Department of Computer Science and Engineering, Easwari Engineering College for his constant support and encouragement.

Finally, we wholeheartedly thank all the faculty members of the Department of Computer Science and Engineering for warm cooperation and encouragement.

ABSTRACT

Agriculture contributes significantly to our country's economic prosperity. Agriculture has grown in recent years as a result of globalization, but the price of agricultural products like vegetables, fruits, etc. are increasing due to low production by the farmers. The reason for low production is that the farmers are not choosing the right crop which is suitable for their soil. As a result of this wrong selection of crops, the farmers face a great amount of setback in their productivity. So, we've progressed to the next level of smart farming in order to boost yield and output for the high-demand goods. Because yield is influenced by a variety of factors, soil plays a crucial role. . Farmers also use the same fertilizers over and over again without knowing anything about them. This has an impact on agricultural productivity and soil fertility. Even if the land is fertile or not, variables such as the content of various nutrients have an impact on it. As a result, we use sensors to collect data on critical nutrients in the soil and send it to the cloud via IoT, where we compute the numbers using machine learning algorithms and make suitable crop recommendations to the farmers. This ML based crop recommendation system can result in reduction of wrong crop selection and increase in crop productivity.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVATIONS

KNN	K Nearest Neighbor
ML	Machine Learning
SVM	Support Vector Machine
NPK	Nitrogen Phosphorus Potassium
DAP	Di Ammonium Phosphate
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
OLED	Organic Light Emitting Diodes
LXDE	Lightweight X11 Desktop Environment
GUI	Graphics User Interface
IDE	Integrated Development Environment
OS	Operating System
LAN	Local Area Network
RAM	Random Access Memory
API	Application Program Interface
CPU	Central Processing Unit
SDK	Software Development Kit

CHAPTER 1

INTRODUCTION

1.1 GENERAL

Soil analysis is the process of determining the amount of minerals and nutrients in the soil. We are using major nutrients in soil to determine the best crop for that soil in our project. We have taken the three key nutrients, Nitrogen, Phosphorus, and Potassium, for analysis. We are also considering the pH value in the soil for crop recommendation. The prediction model should be established in such a way that it can measure crop production with a reduced risk of loss based on the soil nutrients and pH available at the desired site. Farmers nowadays generally utilize urea, DAP, and single phosphate as fertilizers, but they have no idea how much they are using. Farmers also use the same fertilizers over and over again without knowing anything about them. This has an impact on agricultural productivity and soil fertility.

To provide the most suitable crop suggestions, our crop uses IoT technology for data collection, and machine learning algorithms for crop suggestions. NPK and pH sensor are used to collect the soil's Nitrogen, Phosphorus, Potassium and alkaline level. For crop suggestion there exist many algorithms like K Nearest Neighbor, Random Forest, Support Vector Machine, Naïve Bayes, and Decision Tree Classification. We are using three algorithms which has the highest amount of accuracy in crop prediction.

1.2 PROBLEM DESCRIPTION

To avoid wrong crop plantation in particular land based soil. The system uses contents of soil to predict and analysis the suitable crop for the given soil sample using different machine learning algorithms and boxing the best nearest prediction.

1.3 OBJECTIVES

The objectives of the system are mentioned as follows:

- To detect the soil parameters such as nutrients, pH
- To design a highly accurate and prediction system for crop suggestion for the given soil.
- To display the values of NPK, pH and the crop suitable for the soil to the user.

1.4 SCOPE OF THE PROJECT

The Scope of the project is to predict the most suitable crop for the location of the soil sample based on, so this project would be a great help to the society and farmer's necessity. Prediction of most suitable crop thus results in more production and less fertilizers.

1.5 ORGANISATION OF THE THESIS

The report consists of 6 chapters, the contents of which are mentioned below:

Chapter 1 is the introduction that explains the basic information of the system. Chapter 2 is the literature survey that elaborates on the research works on the existing system and their issues. It also proposes a new system which makes an attempt on improving the existing system. Chapter 3 describes the system designand the roles played by various modules. Chapter 4 gives details regarding the system implementation along with the various techniques and algorithms used. Chapter 5 gives details about performance analysis of the proposed system. It involves different test case scenarios and performance factors that are tabulationsbased on the results obtained. Chapter 6 provides the conclusion and future workswhich summarizes the efforts undertaken in the proposed system and states the findings and various shortcomings in the proposed system.

CHAPTER 2

LITERATURE SURVEY

2.1 GENERAL

Literature survey is the documentation of a comprehensive and exhaustive review of the published, unpublished work and secondary sources data corresponding to a specific topic or research problem. It is an overall summary of current knowledge and techniques that allows you to identify relevant methods, theories and gaps in the existing research. This involves various steps such as collecting, evaluating, analyzing publications and user reviews that relate to the particular research topic or problem. Books, journal articles, and thesis are few of the publications that can be used for analyzing to obtain the above-mentioned information which can be used for the survey.

Many research works have been carried out in detecting or classifying Alzheimer Disease. Some of the works include the usage of various algorithms in machine learning and other deep learning techniques in detecting Alzheimer disease. The recently carried out works have been explained in detail below.

2.2 EXISTING SYSTEM

The contributions of various authors are analyzed and surveyed to figure out the merits and demerits of the existing systems in order to make the proposed system work efficiently.

Parameswari p, et al., [18] A two-phase technique is used to predict soil nutrients. Based on earlier time series, Phase 1 employs a fuzzy model to forecast the pH value of the soil. Stage 2 use a regression framework to determine the amount of nitrogen (N), phosphorous (P), and potassium (K) in the soil based on the projected pH value. Advantages: According to the findings analysis, experimental results surpass statistical parameters.

Jyoti Nanwal, et al., [9] "Identifying Best Suited Soil Based on its Physical and Chemical Properties Using Machine Learning". The proposed system uses physical components to collect values and to classify soil samples, they utilized support vector machine (SVM) and artificial neural network (ANN) values. Following the application of these two techniques to our soil samples, discovered that ANN outperforms SVM in terms of classifying the soil based on its physical and chemical properties.

Bhuwan Kashyap, et al., [4] "Sensing Methodologies in Agriculture for Soil Moisture and Nutrient Monitoring". The nutrient monitoring methods are reviewed beginning with laboratory-based methods, ion-selective membrane based sensors, bio-sensors, spectroscopy-based methods, and capillary electrophoresis-based systems for inorganic ion detection. Attention has been given to the core principle of detection while reporting recent sensors developed using the mentioned concepts. The latest works reported on the different sensing methodologies point towards the trend of developing low-cost, easy to use, field-deployable or portable sensing systems aimed towards improving technology adoption in crop production leading to efficient site-specific soil and crop management which in turn will bring us closer to reaching sustainability in the practice of agriculture.

Muhammad Shoaib Farooq, et al., [13] proposes many aspects of technologies involved in the domain of IoT in agriculture. It explains the major components of IoT based smart farming. A rigorous discussion on network technologies used in IoT based agriculture has been presented, that involves network architecture and layers, network topologies used, and protocols. Furthermore, the connection of IoT based agriculture systems with relevant technologies including cloud computing, big data storage and analytics has also been presented. In addition, security issues in IoT agriculture have been highlighted. A list of smart phone based and sensor based applications developed for different aspects of farm management has also been presented.

M.Kalimuthu, et al., [11] the need for collecting the preceding and present is to perform consolidation process for checking accuracy between them. When there is a reasonable accuracy then there will be no occurrence of problem in predicting output. To perform consolidation process there need to perform the python programming in a Google Colab platform that has panda & numpy library in it to perform mathematical operation like average. Thus, both the data is fed into the program to obtain the average output value in graphical form. The consolidation of data and decision making is the major step involved in this mentioned work. Wherein. Naive Bayes classifier, a Supervised learning algorithm in this precision agriculture helps to construct the most accurate and efficient model to predict crop.

Vaibhavi Vanarase, et al., [24] proposes a system with a machine learning algorithm that recommends the best crop based on the nitrogen, phosphorus, and potassium content of the soil. The system results of the three algorithms like SVM, Naïve bays, and decision tree are compared, among which decision tree gives higher accuracy.

Dr. V.Geetha, et al., [23] uses crop growth datasets from various sources. These datasets are used for both training and testing process. Random Forest classifier was found huge ability to predict crop yield. From different outputs, it shows that Random Forest is an efficient learning algorithm to analyze crop at current climatic condition and has a huge exactness in data investigation.

Abhinav Sharma, et al., [3] present a systematic review of ML applications in the field of agriculture. The areas that are focused are prediction of soil parameters such as organic carbon and moisture content, crop yield prediction, disease and weed detection in crops and species detection. ML with computer vision are reviewed for the classification of a different set of crop images in order to monitor the crop.

Aditya Motwani [1] indicates experiments Convolutional Neural Networks with symmetrical architectures gave significantly better results for image classification for the selected soil classification dataset. The final CNN architecture had an amazing accuracy of 95.21%. The Random Forest Algorithm had an accuracy of 75%. Future experiments can adapt this system to a mixture of soil types, by accounting for the composition present in the minority of the mixture. Hence the different papers represent the different machine learning algorithm efficiency and involving various factors according to the process.

2.3 ISSUES IN THE EXISTING SYSTEM

Most The display of values is shown in the OLED display connected to the sensors which is difficult to retrieve data each time of the survey. Arduino boards are used which doesn't give the advantage of latest features and encoding facilities such as Wi-Fi module, data retrieval and operating system to program for ourselves. The values are compared with the basic NPK values through simple code whereas ML algorithm provides more accurate output. The models use mostly of a single algorithm for machine learning which comprises of low accuracy compared to each other. As the models need to improve from itself it needs a bagging necessity to involve different algorithm to maximum its accuracy and choosing the optimum value produced. The datasets has been limited to self-observing numbers whereas the newer datasets have been evolved and repositories are made in turn to achieve a better procrastination of result on the farming sector. Various repositories has been created in different domains for these projects to use.

2.4 PROPOSED SYSTEM

The proposed system aims to create an accurate and efficient way to predict the most suitable crop the given soil sample. The system initially reads the soil nutrient data which gathered using sensor and the input is passed to the prediction model which using KNN, Random Forest and SVM algorithms. The Final result is chosen based on the best of three from the results of the algorithms used in the prediction model. The final predicted result and soil nutrient values are displayed to the users via a mobile application.

2.5 SUMMARY

The literature survey has covered information regarding the existing technologies and algorithms used in the crop suggestion system. The issues have been identified from the existing system and also this chapter briefs about the proposed model. The design of the proposed model is covered in the next chapter.

CHAPTER 3

SYSTEM DESIGN

3.1 GENERAL

The system design phase plays an important role as its main objective is to propose solutions to the problems specified in the existing system. System design mainly deals with planning and designing of the proposed system. It explains the overall flow of operations in the proposed system. System designcould be seen as the application of systems theory to product development. In this chapter, the proposed system is explained using system architecture, functional architecture and the interaction of modules. A system architecture isthe conceptual model that defines the structure, behavior, and more views of a system.

A Functional Architecture is an architectural model that identifies system function and their interactions. It defines how the functions will operatetogether to perform the system missions. These designs help to clearly explain about the work flow of the system. Modular system explains the various modules involved in the functional architecture and gives a clear picture about its interaction.

3.2 SYSTEM ARCHITECTURE

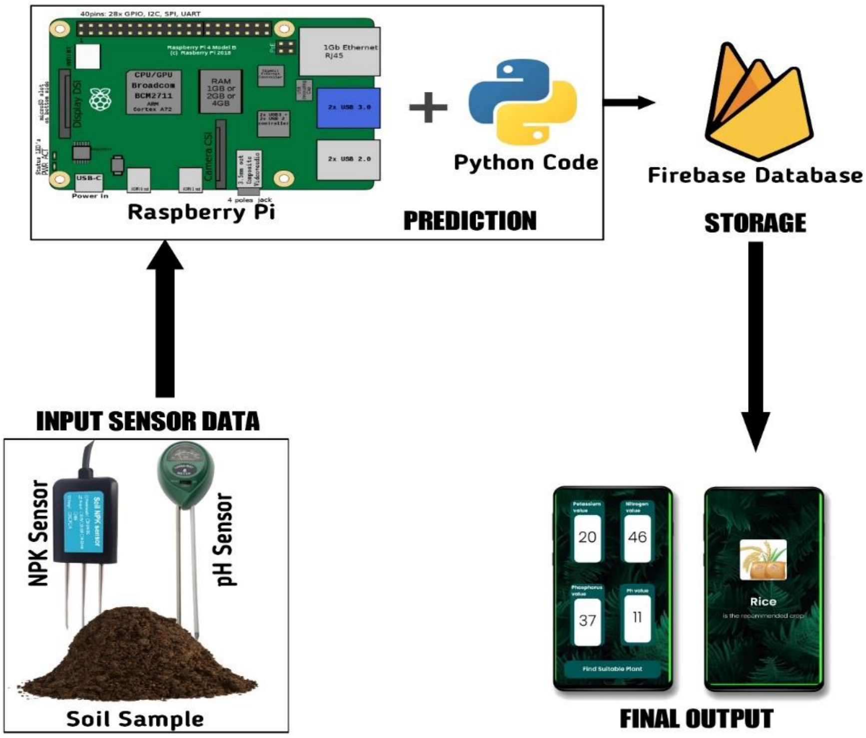


Fig. 3.2: System Architecture

The fig. 3.2 shows the system architecture of the proposed system. The soil sample is taken and analyzed using the NPK sensor and pH sensor and the collected data is sent to the IoT device , here the device is Raspberry pi which act as a Wi-Fi module and gateway for data and using the data for preprocessing. The pre-processed data is sent to the data analytics where the data is compared to existing datasets collected through Kaggle and trained and testes using the following machine learning algorithms such as KNN , SVM and random forest comparing the accuracies the ranking based algorithm finds the optimal algorithm to publish the result of which crop is to be suggested.

3.3 FUNCTIONAL ARCHITECTURE

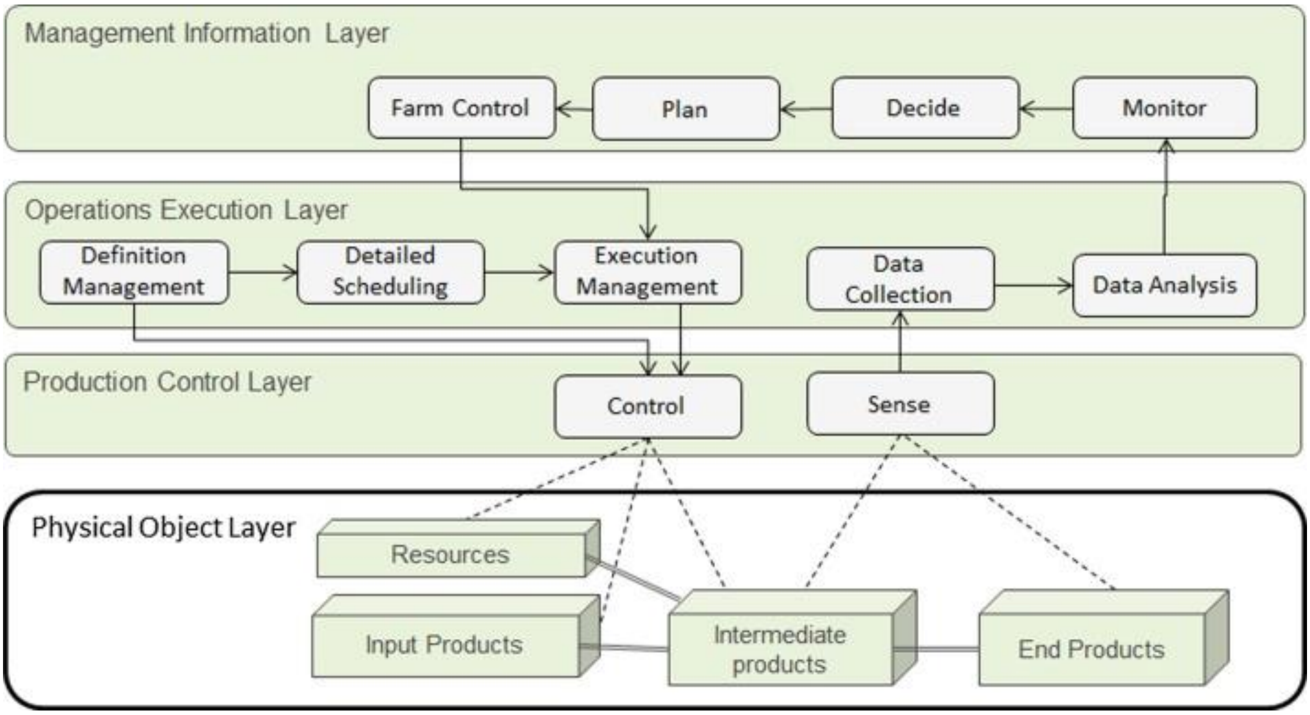


Fig. 3.3: Functional Architecture

The above fig. 3.3 is the functional architecture of the proposed system. The functional architecture of the system shows how all the functions work together to achieve a single objective. In the physical object layer, the soil sample is added with water and made into an intermediate product. This intermediate product is tested by sensing the nutrient level using sensor by the production control layer. The sensor data is passed to data collection module in the operations layer and then the data is analyzed. For the data analysis purpose we use python and machine learning algorithms to determine the most suitable crop that can be planted in the soil sample provided. The result of analysis consists of the soil's Nitrogen, Phosphorus, Potassium, pH level and suitable crop is displayed to the user via monitor in the management layer. The user then based on the analyzed data decides to take necessary action to increase production by managing the operations Layer and executing this changes into the soil. The soil then makes an intermediate product and when the user is satisfied with the analysis result of their soil, then they stop the execution from operation level and the soil at final level of execution is considered as end product.

3.4 MODULAR DESIGN

The modular design explains the detailed working of each module and its interactions. The modules in the proposed system are:

1. Data collection using sensors
2. Crop prediction using prediction model
3. Display soil data and crop suggestion

3.4.1 Data collection using sensors

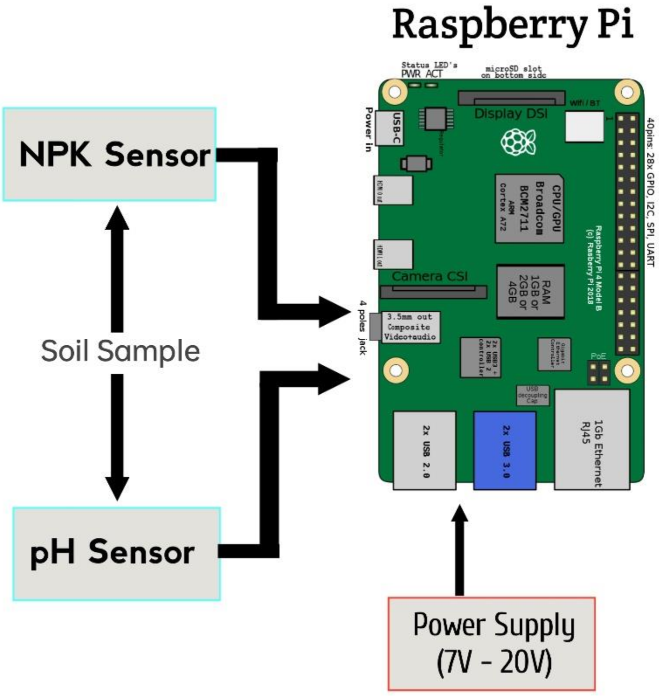


Fig. 3.4.1: Module diagram for data collection

The data collection module serves the purpose of collecting data from the provided soil sample using sensors. The sensors used for this data collection purpose are NPK sensor and pH sensor. The NPK sensor is used to gather the soil's Nitrogen, Phosphorus and Potassium level in the soil. The pH sensor is used to identify the soil's alkaline level. The data gathered from soil is collected from the sensors using raspberry pi. Fig. 3.4.1 displays the design of data collection module.

3.4.2 Crop prediction using prediction model

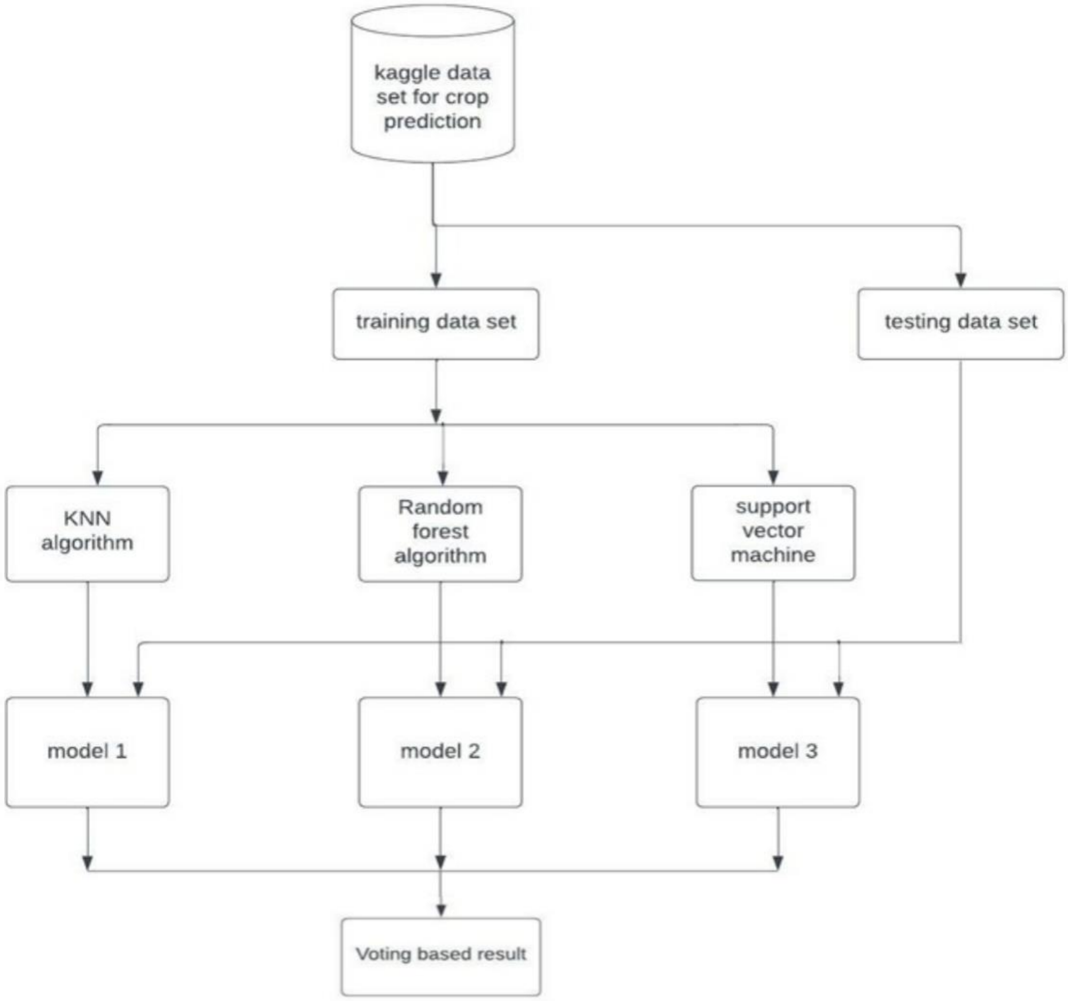


Fig. 3.4.2: Flow diagram of prediction model

The prediction model is displayed in fig. 3.4.2. The model uses crop recommendation dataset available in Kaggle. The model uses three sub models, these sub models are made using the KNN, Random Forest and SVM algorithms. The best of three among the results of these algorithms are provided as a final result of the prediction model in our proposed system. The KNN, Random Forest and SVM are the most used algorithms for crop suggestion, so these three algorithms are used in our proposed system.

3.4.3 Display soil data and crop suggestion

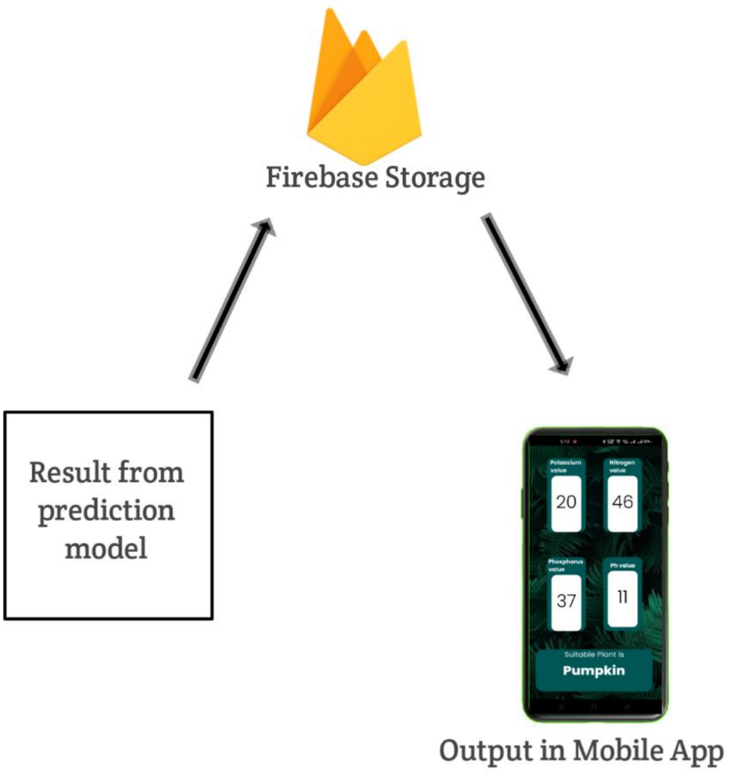


Fig. 3.4.3: Soil data and crop suggestion displayed in Mobile App

The fig. 3.4.3 shows the flow from prediction result to displaying result. The result of prediction model is uploaded in the Google's Firebase storage using APIs. The data in Firebase is displayed in the mobile App which was created using TypeScript and the firebase and mobile App is connected using API. When the value is updated in firebase, the value is mobile app also gets updated.

3.5 SYSTEM REQUIREMENTS

3.5.1 Hardware Specification

Processor : Pentium IV/III

Hard disk : minimum 80 GB

RAM : minimum 2 GB

LAN or Wi-Fi

3.5.2 Software Specification

OS: Windows/ Linux

Packages: NumPy, Pandas, Matplotlib, Sklearn, Seaborn

Language: Python

IDE: Anaconda with Jupyter Notebook

Mobile App Design: TypeScript, JavaScript, JSON

3.6 SUMMARY

This chapter gives an overview of the system architecture, functional architectures and modular design are explained. The system architecture explains the system design of the proposed system. The functional architecture explains the entire/complete functionality along with its modular structure, its interactions between the modules and the input - output of the proposed system. System requirements are also mentioned.

CHAPTER 4

SYSTEM IMPLEMENTATION

4.1 GENERAL

This chapters deal with how the implementation of the system is carried it. It describes about the various tools, frameworks, and platforms which are used. It also explains how these tools and platforms are incorporated within the systems and implemented. The following shows the implementation of the proposed system along with its detailed explanation and snapshots. The following presents the input and output used in the implementation of the system.

4.2 OVERVIEW OF THE PLATFORM

4.2.1 Python

Python is a general purpose and high-level programming language. We can use Python for developing desktop Graphical User Interface (GUI) applications, websites and web applications. Also, Python, as a high-level programming language, allows us to focus on core functionality of the application by taking care of common programming tasks. The simple syntax rules of the programming language further make it easier for us to keep the code base readable and application maintainable. Like other modern programming languages, Python also supports several programming paradigms. It supports object oriented and structured programming fully. Also, its language features support various concepts in functional and aspect-oriented programming. Scripts written in Python (.PY files) can be parsed and run immediately. Similar to this there are various reasons why we should prefer python over other programming languages

4.2.2 Jupyter Notebook

The Jupyter Notebook application allows you to create and edit documents display the input and output of a Python or R language script. Once saved, you can share these files with others. Python and R language are included by default,but with customization, Notebook can run several other kernel environments. The easiest way to install the Jupyter Notebook App is installing a scientific python distribution which also includes scientific python packages. The most common distribution is called Anaconda.

4.2.3 PyCharm

PyCharm is the most popular Integrated Development Environment (IDE) for python. PyCharm is a cross-platform IDE for Python. In addition to supporting versions 2.x and 3.x of Python, PyCharm is also compatible with Windows, Linux, and macOS. At the same time, the tools and features provided by PyCharm help programmers to write a variety of software applications in Python quickly and efficiently. The developers can even customize the PyCharm UI according to their specific needs and preferences. Also, they can extend the IDE by choosing from over 50 plug-ins to meet complex project requirements.

4.2.4 NumPy

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much 33 more. NumPy fully supports an object-oriented approach, starting, once again, with ndarray. For example, ndarray is a class, possessing numerous methods and attributes. Many of its methods are mirrored by functions in the outer-most NumPy namespace, allowing the programmer to code in whichever paradigm they prefer. This flexibility has allowed the NumPy array dialect and NumPy ndarray class to become the de-facto language of multi-dimensional data interchange used in Python.

4.2.5 Matplotlib

Matplotlib is a low-level graph plotting library in python that serves as a visualization utility. It is open source and we can use it freely. Matplotlib is mostly written in python, a few segments are written in C, Objective-C and JavaScript for Platform compatibility. It is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consistsof several plots like line, bar, scatter, histogram etc. Matplotlib was created by John D. Hunter. Matplotlib 2.0.x supports Python versions 2.7 through 3.10. Python 3 support started with Matplotlib 1.2. Matplotlib 1.4 is the last version tosupport Python 2.6.

4.2.6 Seaborn

Seaborn is an open source, BSD-licensed Python library providing high level API for visualizing the data using Python programming language. It provides a high-level interface for drawing attractive and informative statistical graphics. Seaborn is more comfortable in handling Pandas data frames. It uses basic sets of methods to provide beautiful graphics in python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on the top of Matplotlib library and also closely integrated to the data structures from pandas. Seaborn tries to make a well-defined set of hard things easy too. It aims to make visualization the central part of exploring and understanding data. It provides dataset-oriented APIs, so that we can switch between different visual representations for same variables for better understanding of dataset.

4.2.7 NPK Sensor

The soil npk sensor is suitable for detecting the content of nitrogen, phosphorus, and potassium in the soil, and judging the fertility of the soil. Thereby facilitating the systematic evaluation of the soil condition. Can be buried in the soil for a long time, resistant to long-term electrolysis, corrosion resistance, vacuum potting, and completely waterproof. Soil npk sensors are widely used in soil nitrogen, phosphorus and potassium detection, precision agriculture, forestry, soil research, geological prospecting, plant cultivation and other fields.

4.2.8 pH Sensor

A pH meter is a scientific instrument that measures the hydrogen-ion activity in water-based solutions, indicating its acidity or alkalinity expressed as pH. The pH meter measures the difference in electrical potential between a pH electrode and a reference electrode, and so the pH meter is sometimes referred to as a "potentiometric pH meter". The difference in electrical potential relates to the acidity or pH of the solution. The pH meter is used in many applications ranging from laboratory experimentation to quality control.

4.2.9 Raspberry Pi

Raspberry Pi OS has a desktop environment, PIXEL, based on LXDE, which looks similar to many common desktops, such as macOS and Microsoft Windows. The desktop has a background image. A menu bar is positioned at the top and contains an application menu and shortcuts to a web browser (Chromium), file manager, and terminal. The other end of the menu bar shows a Bluetooth menu, Wi-Fi menu, volume control, and clock. The desktop can also be changed from its default appearance, such as repositioning the menu bar.

4.2.10 FIREBASE

The firebase real time database is a cloud-hosted database. Data is stored as JSON and synchronized in real-time to every connected client. When you build cross-platform apps with our iOS, Android, and JavaScript SDKs, all of your clients share one real time Database instance and automatically receive updates with the newest data. Store and sync data with our NoSQL cloud database. Data is synced across all clients in real-time, and remains available when your app goes offline.

4.2.11 TypeScript

TypeScript is a programming language developed and maintained by Microsoft. It is a strict syntactical superset of JavaScript and adds optional static typing to the language. It is designed for the development of large applications and transpiles to JavaScript. As it is a superset of JavaScript, existing JavaScript programs are also valid TypeScript programs. TypeScript supports definition files that can contain type information of existing JavaScript libraries, much like C++ header files can describe the structure of existing object files. This enables other programs to use the values defined in the files as if they were statically typed TypeScript entities. There are third-party header files for popular libraries such as jQuery, MongoDB, and D3.js. TypeScript headers for the Node.js basic modules are also available, allowing development of Node.js programs within TypeScript.

4.2.12 Dataset Description

The dataset is taken from Kaggle website. It consists 2200 records and 22 distinct crops and 100 records for each crop. Each crop has a distinct values for columns N(Nitrogen), P(Phosphorus), K(Potassium) and pH (Alkaline level) The dataset is splitted into a train dataset(70% of entire dataset) and test dataset (30% of entire dataset). Using the train set and test set we can identify the accuracy of our prediction model.

4.3 MODULE IMPLEMENTATION

The hardware components like sensors and raspberry pi are connected using jump wires. The raspberry is connected with power supply and this supply provides power to the sensors also. The entire module implementation connection is displayed in Fig 4.3.

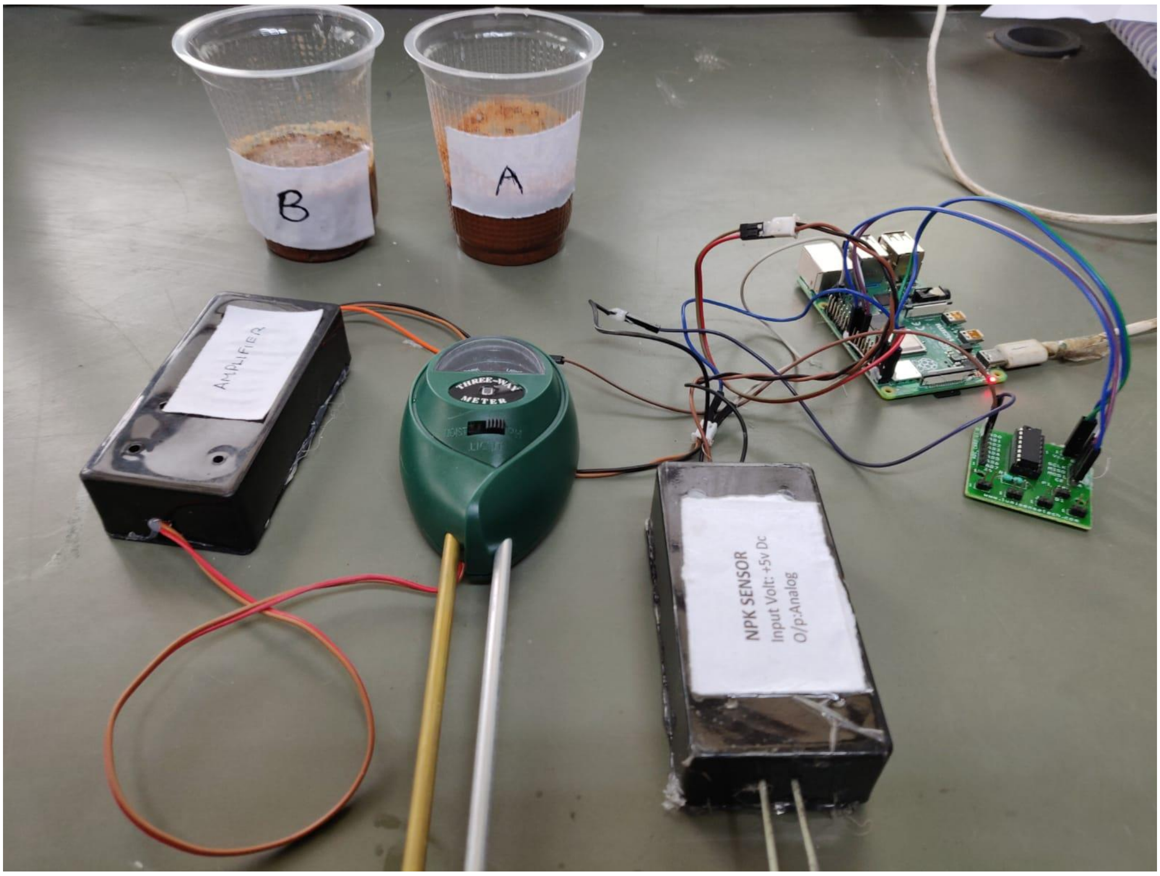


Fig 4.3 Implementation of System

4.3.1 Implementation of data collection module

This module collects the data from soil using Sensor and passes the data to raspberry pi. The probes of NPK and pH sensor are dipped into the soil sample. The power for sensor is provided through raspberry pi. Fig. 4.3.1.1 displays the collection of soil data using sensors.

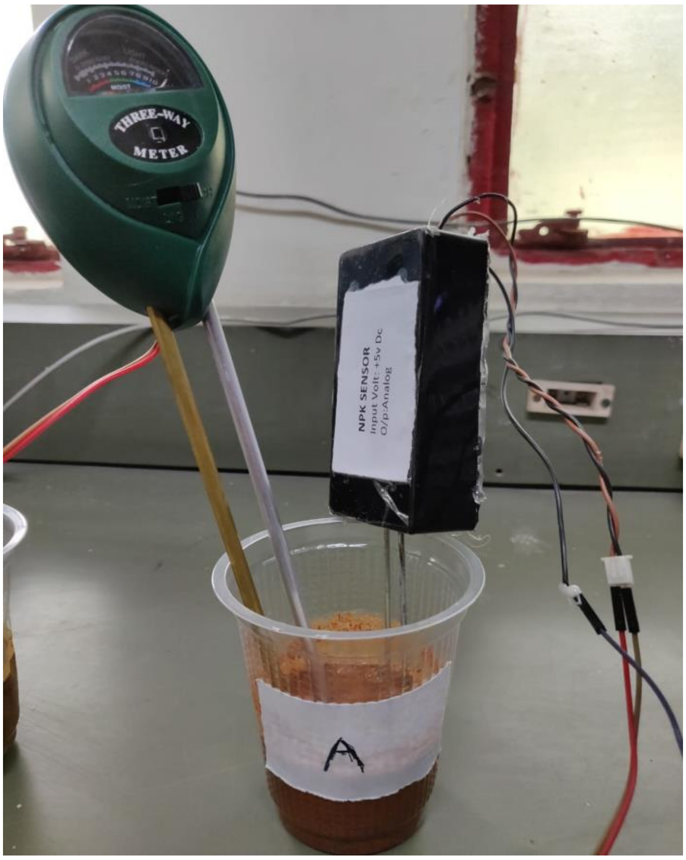
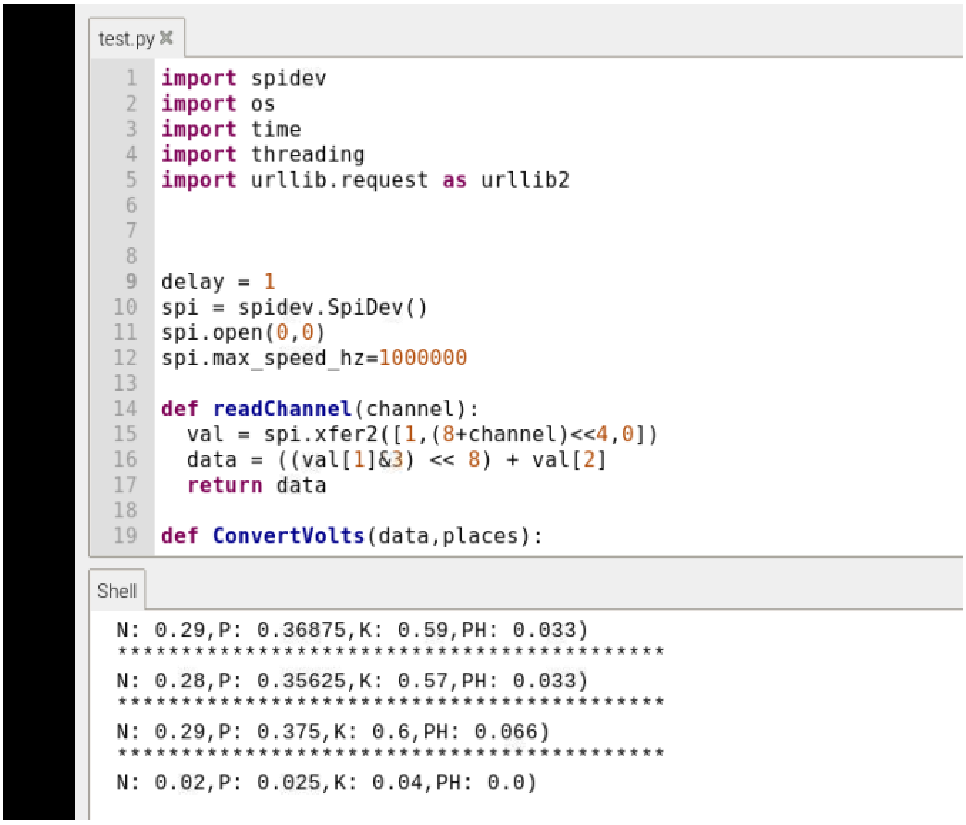


Fig. 4.3.1.1: Soil data collection using Sensors

The soil data is collected from sensor and available in raspberry pi. We can view the sensor data using python code in the raspberry pi desktop. Fig 4.3.1.2 displays the sensor data displayed in raspberry pi desktop IDE.



```
test.py

import spidev
import os
import time
import threading
import urllib.request as urllib2



delay = 1
spi = spidev.SpiDev()
spi.open(0,0)
spi.max_speed_hz=1000000

def readChannel(channel):
  val = spi.xfer2([1,(8+channel)<<4,0])
  data = ((val[1]&3) << 8) + val[2]
  return data

def ConvertVolts(data,places):
```

```
Shell

N: 0.29,P: 0.36875,K: 0.59,PH: 0.033)
*******************************************
N: 0.28,P: 0.35625,K: 0.57,PH: 0.033)
*******************************************
N: 0.29,P: 0.375,K: 0.6,PH: 0.066)
*******************************************
N: 0.02,P: 0.025,K: 0.04,PH: 0.0)
```

Fig. 4.3.1.2: Sensor data displayed in raspberry pi

4.3.2 Implementation of prediction model

In this module the data is to be analyzed using machine learning algorithms using the dataset taken from Kaggle, then training and testing the dataset, the algorithms used are KNN, Random forest and SVM and finally the most frequent out of the three results is used. The dataset sample is displayed in table 4.3.2.

Table 4.3.2: Dataset Sample

N	P	K	ph	label
90	42	43	6.502985	rice
85	58	41	7.038096	rice
60	55	44	7.840207	rice
74	35	40	6.980401	rice
78	42	42	7.628473	rice
69	37	42	7.073454	rice
69	55	38	5.700806	rice
94	53	40	5.718627	rice

KNN:

One of the most commonly used supervised and nonparametric machine learning techniques is the KNN, used in classification and regression problems. Supervised algorithms are the ones with labelled data. Labeled data refers to input data that is already tagged with the correct output in supervised learning. Supervised learning algorithms take the data and attempt to make models that predict the output data, given relevant inputs. There is, however, no actual learning in the k-NN algorithm, which follows the ‘‘lazy learning’’ principle, where all the work happens at the time a prediction is required. The algorithm depends on distances between points, which can be ascertained using one of a few methods. A key aspect for consideration is that the distance is always required to be either zero or positive. This is done by squaring the distance or raising it to a certain power or taking the absolute values.

SVM:

The Support Vector Machine, or SVM, is a popular Supervised Learning technique that may be used to solve both classification and regression issues. However, it is mostly utilized in Machine Learning for Classification difficulties.

The SVM algorithm's purpose is to find the optimum line or decision boundary for categorizing n-dimensional space into classes so that additional data points can be readily placed in the correct category in the future. A hyperplane is the name for the optimal choice boundary. The extreme points/vectors that assist create the hyperplane are chosen via SVM. Support vectors are the extreme instances, and the algorithm is called a Support Vector Machine. Take a look at the diagram below, which shows two different groups that are categorized using a decision boundary or hyperplane.

RANDAM FOREST:

Random Forest is a well-known machine learning algorithm that uses the supervised learning method. In machine learning, it can be utilized for both classification and regression issues. It is based on ensemble learning, which is a method of integrating numerous classifiers to solve a complex problem and increase the model's performance. Random Forest is a classifier that contains a number of decision trees on various subsets of a given dataset and takes the average to enhance the predicted accuracy of that dataset, according to the name. Instead than relying on a single decision tree, the random forest collects the forecasts from each tree and predicts the final output based on the majority votes of predictions.

```
#Build Model
models = {
    KNeighborsClassifier(n_neighbors=5):'KNN',
    RandomForestClassifier():'Random Forest',
    SVC():'Support Vector Machine'
}
for m in models.keys():
    m.fit(X_train,y_train)
for model,name in models.items():
     print(f"Accuracy Score for {name} is : ",model.score(X_test,y_test)*100,"%"
```

```
Accuracy Score for KNN is :  72.95454545454545 %
Accuracy Score for Random Forest is :  80.22727272727272 %
Accuracy Score for Support Vector Machine is :  75.9090909090909 %
```

Fig. 4.3.2: Accuracy of algorithms

As per the prediction the most frequent out of the three algorithms result will be displayed as final output and if all the three algorithms have different results then the result of Random forest will be displayed as it has more accuracy then KNN and SVM.

4.3.3 Implementation of storing and displaying result

The sensor data and prediction data are uploaded into firebase storage using the python library “firebase_admin”. In the firebase storage five fields are stored on each reading which are K, N, P, Ph and Plant. The value gets overwritten when the sensor reads new data. The overwriting of values is enabled to reduce the storage and the current system doesn’t require any history data. The firebase console is displayed in Fig. 4.3.3.1.

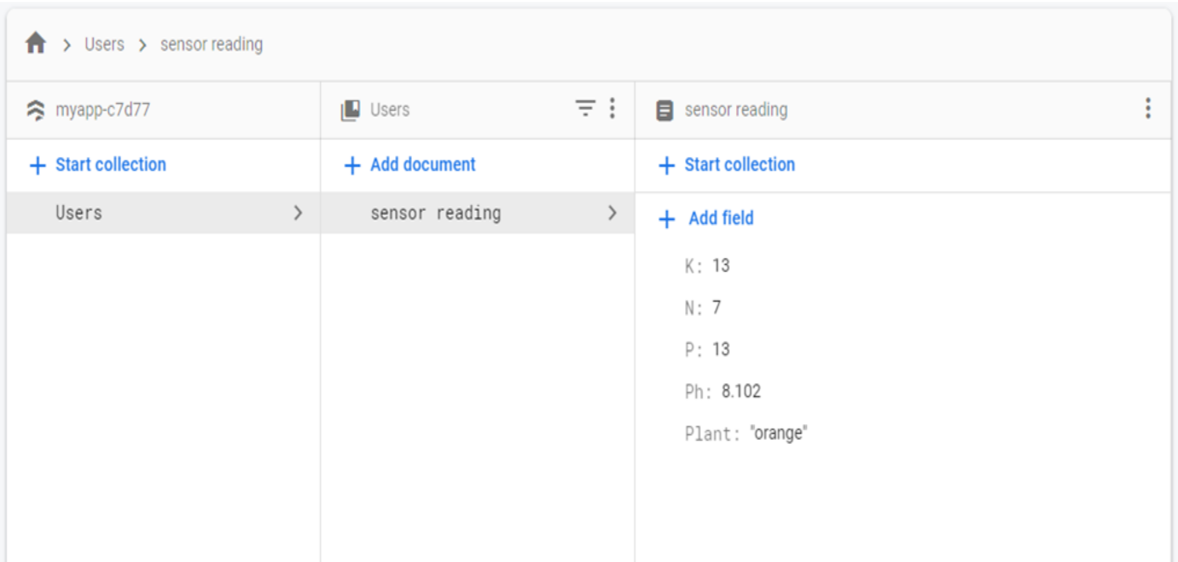


Fig. 4.3.3.1: Firebase Console

After data is stored in console, it gets immediately reflected in the mobile app too with a minimal delay of 0.3 seconds. The app includes a login page and once the user has logged in, they can view 2 options as give manual input and the other one as read from sensor. The user can go with option 1 of giving manual input, if they know the soil’s NPK values and pH value or else they can go with option 2 which includes usage of sensors to identify the soil data. The app displays all the five fields available in the firebase storage to the user for any given option. The sample screenshot of the app is displayed as Fig. 4.3.3.2 and Fig. 4.3.3.3.

Fig. 4.3.3.2: Mobile app front page UI

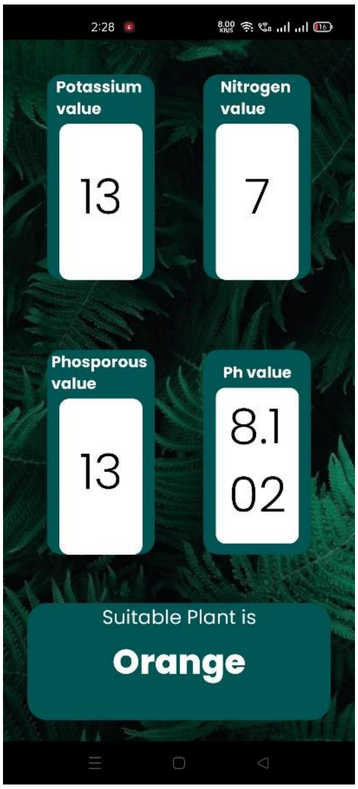


Fig. 4.3.3.3: Mobile app sensor data

4.4 SUMMARY

This chapter emphasizes the analysis of each technology incorporated in this project. This chapter also portrays the basic system requirements such as the platforms, languages, and tools used in the implementation. It also includes the snapshots of implementation of different modules.

CHAPTER 5

SYSTEM TESTING AND PERFORMANCE ANALYSIS

5.1 GENERAL

Testing phase plays a vital role as it is a process which helps to find out the possible fault or inaccuracy in the work project. This phase provides a functionality check of the components used, its interaction within the modules and the finished product. The performance, accuracy and robustness of the system is tested in this phase. There are several types of testing during the test phase, including quality assurance testing (QA), system integration testing (SIT), and user acceptance testing (UAT). It also undergoes thorough execution to ensure if it meets the system requirements and user expectations.

5.2 TEST CASE:

The system was tested with 5% data from the test data with all 22 crops present and the test case has 110 records. In the test cases, for a crop we have taken the result of all the three algorithms we used in our prediction model and final output of our prediction model.

The test case table consist of column N, P, K, pH, crop (actual crop from dataset), KNN (result of KNN algorithm), SVM (result of SVM algorithm), Random Forest (result of Random Forest algorithm) and our prediction (final result of our prediction model). These test cases helps us in identifying the accuracy of our prediction model. Sample test cases are available in the table 5.2.

Table 5.2: Test Case

N	P	K	pH	crop	KNN	SVM	Random Forest	Our Prediction
88	46	42	6.60	rice	rice	rice	rice	rice
93	47	37	6.50	rice	rice	rice	rice	rice
60	55	45	5.94	rice	maize	rice	maize	maize
78	35	44	7.07	rice	rice	rice	rice	rice
65	37	40	5.33	rice	rice	rice	rice	rice
90	57	24	6.16	maize	maize	maize	maize	maize
67	35	22	6.39	maize	maize	maize	maize	maize
60	54	19	6.42	maize	rice	maize	rice	rice
83	58	23	6.38	maize	maize	maize	maize	maize
83	57	19	6.88	maize	maize	maize	maize	maize
57	56	78	8.86	chickpea	chickpea	chickpea	chickpea	chickpea
48	65	78	7.86	chickpea	chickpea	chickpea	chickpea	chickpea
36	56	83	7.45	chickpea	chickpea	chickpea	chickpea	chickpea
40	58	75	7.17	chickpea	kidneybeans	chickpea	kidneybeans	kidneybeans
49	69	82	7.26	chickpea	chickpea	chickpea	chickpea	chickpea
27	65	18	5.60	kidneybeans	chickpea	kidneybeans	chickpea	chickpea
30	63	16	5.53	kidneybeans	kidneybeans	kidneybeans	Kidneybeans	kidneybeans
37	70	25	5.82	kidneybeans	kidneybeans	kidneybeans	kidneybeans	kidneybeans
27	63	19	5.56	kidneybeans	kidneybeans	kidneybeans	kidneybeans	kidneybeans
22	60	24	5.63	kidneybeans	kidneybeans	kidneybeans	kidneybeans	kidneybeans
35	71	17	6.93	pigeonpeas	pigeonpeas	pigeonpeas	pigeonpeas	pigeonpeas
11	72	22	6.84	pigeonpeas	pigeonpeas	pigeonpeas	pigeonpeas	pigeonpeas
20	60	22	6.88	pigeonpeas	pigeonpeas	pigeonpeas	pigeonpeas	pigeonpeas
10	71	18	6.15	pigeonpeas	pigeonpeas	pigeonpeas	pigeonpeas	pigeonpeas
33	61	24	4.57	pigeonpeas	pigeonpeas	pigeonpeas	pigeonpeas	pigeonpeas
4	59	22	8.91	mothbeans	mothbeans	mothbeans	mothbeans	mothbeans
22	51	16	8.64	mothbeans	mothbeans	mothbeans	mothbeans	mothbeans
33	47	17	8.62	mothbeans	mothbeans	mothbeans	mothbeans	mothbeans
2	51	17	5.84	mothbeans	mothbeans	mothbeans	lentil	mothbeans
16	51	21	3.53	mothbeans	mothbeans	mothbeans	mothbeans	mothbeans
10	37	22	7.07	mungbean	mungbean	mungbean	mungbean	mungbean

4	44	19	6.92	mungbean	mungbean	mungbean	mungbean	mothbeans
20	45	17	6.77	mungbean	mungbean	mungbean	mungbean	mungbean
23	45	23	6.98	mungbean	lentil	mungbean	mungbean	mungbean
25	48	21	6.27	mungbean	mungbean	mungbean	mungbean	mungbean
58	73	16	6.87	blackgram	mungbean	blackgram	blackgram	blackgram
55	77	22	7.76	blackgram	blackgram	blackgram	blackgram	blackgram
42	79	23	6.78	blackgram	blackgram	blackgram	blackgram	blackgram
44	77	21	7.33	blackgram	blackgram	blackgram	blackgram	blackgram
38	62	25	7.45	blackgram	blackgram	blackgram	blackgram	blackgram
26	56	22	7.01	lentil	lentil	lentil	lentil	lentil
9	77	17	6.28	lentil	lentil	lentil	lentil	lentil
4	59	19	7.62	lentil	lentil	lentil	lentil	lentil
34	73	15	7.63	lentil	lentil	lentil	lentil	lentil
33	77	15	7.80	lentil	lentil	lentil	lentil	lentil
33	23	45	7.12	pomegranate	pomegranate	pomegranate	pomegranate	pomegranate
4	14	41	6.43	pomegranate	pomegranate	pomegranate	pomegranate	pomegranate
13	17	45	7.16	pomegranate	banana	banana	pomegranate	banana
39	24	39	6.43	pomegranate	pomegranate	banana	pomegranate	pomegranate
8	28	37	6.08	pomegranate	pomegranate	pomegranate	pomegranate	pomegranate
108	94	47	6.39	banana	banana	banana	grapes	banana
92	81	52	5.89	banana	banana	banana	banana	banana
110	71	54	5.73	banana	banana	banana	banana	banana
82	75	55	6.28	banana	banana	banana	banana	banana
117	81	53	5.51	banana	grapes	grapes	banana	grapes
19	38	26	4.53	mango	mango	mango	mango	mango
21	21	30	5.40	mango	mango	mango	mango	mango
22	18	33	6.62	mango	mango	mango	mango	mango
23	138	200	5.97	grapes	pomegranate	grapes	pomegranate	pomegranate
40	143	201	6.48	grapes	grapes	grapes	grapes	grapes
6	142	202	6.22	grapes	grapes	grapes	grapes	grapes
96	18	50	6.90	watermelon	muskmelon	watermelon	muskmelon	muskmelon
82	26	47	6.57	muskmelon	muskmelon	muskmelon	muskmelon	muskmelon
106	21	52	6.29	muskmelon	watermelon	watermelon	muskmelon	watermelon
90	15	52	6.45	muskmelon	muskmelon	muskmelon	muskmelon	muskmelon
106	16	54	6.59	muskmelon	muskmelon	muskmelon	muskmelon	muskmelon

40	120	197	5.89	apple	apple	apple	apple	apple
25	132	198	5.73	apple	apple	apple	apple	apple
31	137	196	6.40	apple	apple	apple	apple	apple
36	144	196	6.50	apple	orange	apple	orange	orange
10	140	197	6.23	apple	apple	apple	apple	apple
27	30	5	7.66	orange	orange	orange	orange	orange
13	8	12	7.11	orange	orange	orange	orange	orange
6	7	7	7.20	orange	orange	orange	orange	orange
40	17	15	7.87	orange	orange	orange	orange	orange
31	26	9	7.57	orange	orange	orange	orange	orange
42	59	55	6.98	papaya	papaya	papaya	papaya	papaya
43	64	47	6.83	papaya	papaya	papaya	papaya	papaya
35	67	49	6.62	papaya	banana	banana	papaya	banana
56	59	55	6.55	papaya	papaya	papaya	papaya	papaya
39	64	53	6.60	papaya	papaya	papaya	papaya	papaya
22	8	33	5.67	coconut	coconut	coconut	coconut	coconut
28	27	32	5.76	coconut	coconut	coconut	coconut	coconut
19	26	29	5.67	coconut	coconut	coconut	coconut	coconut
113	38	20	6.36	cotton	cotton	cotton	cotton	cotton
102	53	21	6.91	cotton	cotton	cotton	cotton	cotton
110	39	18	7.77	cotton	coconut	coconut	cotton	coconut

5.3 PERFORMANCE MEASURES

5.3.1 Accuracy

The sum of true values divided by the total number of samples. This is only accurate if the model is balanced. It will give inaccurate results if there is an imbalance in the model.

Accuracy = No. of true values/ Total no. of samples

5.3.2 Time Taken

The amount of time taken by a particular test case is the total time elapsed from when the input is provided by the user to the time when the system returns the translated sentence. The average time taken is calculated by summing up the timetaken by each test case and dividing by the number of test cases

$$Average\ Time\ Taken = \frac{Total\ time\ taken\ to\ complete\ each\ test\ case}{Number\ of\ test\ cases}$$

5.4 PERFORMANCE ANALYSIS

The performance of the proposed system can be measured by the accuracy of the algorithms and compared

We have used 5% of the dataset, which is 110 records for testing our prediction model. From the test cases we are deriving our prediction model's accuracy. The 110 records includes 5 test records for all available in dataset. 22 crops x 5 records each gives 110 records. These 110 records are handpicked and the result for each record is calculated separately.

We have calculated the accuracy based on the number of correctly predicted records over total number of records.

Accuracy = No. of true values/ Total no. of samples

Our prediction model has predicted 934 records correctly out of 110 records. Accuracy of proposed prediction model = (94/110)*100 which gives the percentage of accuracy as 85.5%.

Our proposed system has achieved 85.5% of accuracy which is higher than the accuracy of the three algorithms used in our prediction model.

KNN

Out of 110 records, 80 records are correctly predicted by KNN algorithm. The confusion matrix for KNN is available as Fig. 5.4.1.

Accuracy = (80/110)*100 which gives the percentage of accuracy as 73%

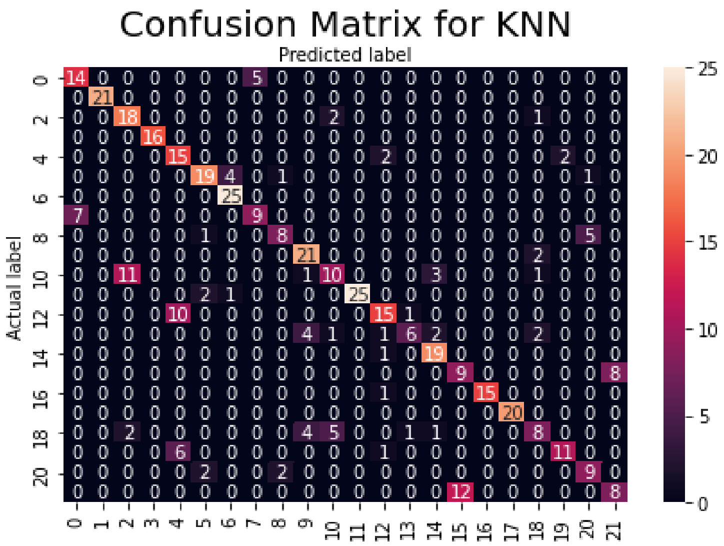


Fig. 5.4.1: Confusion matrix for KNN

SVM

Out of 110 records, 83 records are correctly predicted by SVM algorithm. The confusion matrix for SVM is available as Fig. 5.4.2.

Accuracy = (83/110)*100 which gives the percentage of accuracy as 75.5%

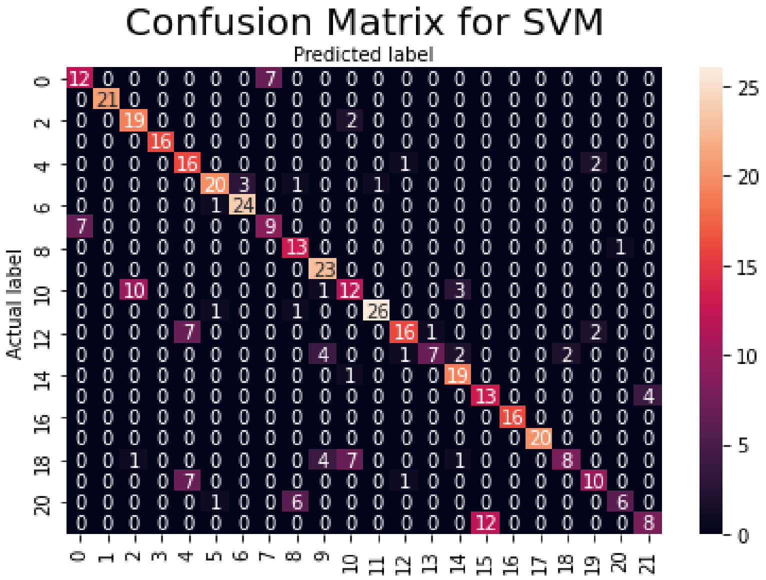


Fig. 5.4.2: Confusion matrix for SVM

RANDOM FOREST

Out of 110 records, 88 records are correctly predicted by random forest algorithm. The confusion matrix for random forest is available as Fig. 5.4.2.

Accuracy = (88/110)*100 which gives the percentage of accuracy as 80%

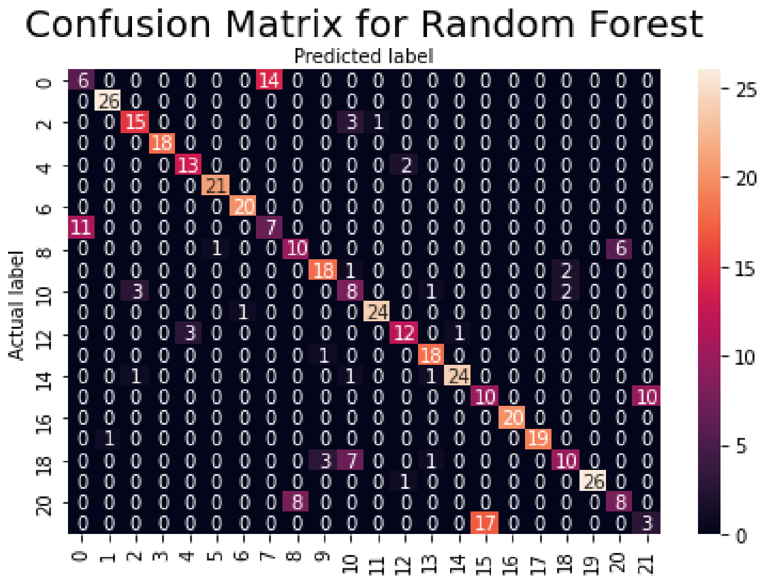


Fig. 5.4.3: Confusion matrix for random forest

5.5 SUMMARY

This chapter explains the mainstream illustration of the different test cases used and the testing processes applied in the entire development of the project. It also mentions about the performance metrics and its analysis conducted on the system.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 CONCLUSION

This system wraps up with a complete overview of all the ideas that has been discussed in the proposed system so far. The proposed system has created a more accurate prediction model than the existing single ML algorithm models like KNN, SVM, random forest etc. The current proposed system has a prediction accuracy of 85.5%. With this proposed system, we can identify the most suitable crop for a soil and increase productivity and revenue of farmers also reducing the money and investment on fertilizers. The current proposed system can also be remote monitored as we are displaying the final result on mobile app.

6.2 FUTURE WORK

The future work of this project is implemented in better and advanced mode of prediction and in order to provide a baseline for future research, we will set up and evaluate the performance of existing models and resources for crop and soil analysis as survey in the future and include more factors for a better accuracy model such as humidity, rainfall, temperature in the future.

APPENDICES

SAMPLE CODING

Code for mobile app

input.tsx

```
import React, { FC } from "react";
import { View, Text, StyleSheet, TextInput, Dimensions } from "react-
native";
interface Props {
  ph: string;
  oct: (text: string) => void;
  secureTextEntry: boolean;
  textcolor: string;
}
const width = Dimensions.get("window").width;
const height = Dimensions.get("window").height;

const Input: FC<Props> = (props) => {
  return (
    <View style={styles.container}>

      <TextInput placeholder={props.ph} onChangeText={props.oct}
secureTextEntry={props.secureTextEntry || false}
placeholderTextColor={props.textcolor} color={props.textcolor} />
    </View>
  )
```

```
}
export default Input;
const styles = StyleSheet.create({
  container: {
    marginVertical: 10,
    borderRadius: 10,
    height: 50,
    width: width - 50,
    backgroundColor: "lightgrey",
    alignItems: "center",
    justifyContent: "center",
  }
})
```

results.tsx

```
import React from "react";
import { FC, useState } from "react";
import { ImageBackground, Text, TextInput, View, Dimensions,
TouchableOpacity, Image } from "react-native";

import storage from '@react-native-firebase/storage';

const App: FC = ({ route }) => {
  const ItemId = route.params
  const name = ItemId.plant
  const reference = { uri: 'https://firebasestorage.googleapis.com/v0/b/myapp-
c7d77.appspot.com/o/' + name + '.jpg?alt=media&token=' + ItemId.token }
  const image = { uri: "https://images.pexels.com/photos/1687341/pexels-photo-
```

```
1687341.jpeg?auto=compress&cs=tinysrgb&w=1260&h=750&dpr=1" }

  return (
    console.log(name),
    <ImageBackground source={image} resizeMode='cover' style={{ flex: 1 }}>
      <View style={{ alignItems: 'center', justifyContent: 'center', marginTop:
'50%' }}>
        <Image source={reference} style={{ height: 150, width: 150,
borderRadius: 10 }} />
        <Text style={{ color: 'white', fontSize: 40, fontFamily: 'Poppins-
SemiBold', marginTop: 50 }}>
          {name}
        </Text>
        <Text style={{ fontSize: 20, fontFamily: 'Poppins-Light', color: 'white' }}>
          is the recommended crop
        </Text>
      </View>
    </ImageBackground>

  )
}
export default App
```

LiveValue.tsx

```
import React from "react";
import { FC, useState } from "react";
import { ImageBackground, Text, TextInput, View, Dimensions,
TouchableOpacity } from "react-native";
import firestore from '@react-native-firebase/firestore';
```

```
const width = Dimensions.get('window').width;
const height = Dimensions.get('window').height;
const App: FC = () => {
  const image = { uri: "https://images.pexels.com/photos/1687341/pexels-photo-
1687341.jpeg?auto=compress&cs=tinysrgb&w=1260&h=750&dpr=1" }
  const [K, setK] = useState<any>(null)
  const [N, setN] = useState(null)
  const [P, setP] = useState(null)
  const [ph, setph] = useState(null)
  const [plant, setplant] = useState(null)
  function onResult(QuerySnapshot) {
    console.log('Got Users collection result.');
    const data = QuerySnapshot["_data"]
    setK(data["K"])
    setP(data["P"])
    setN(data["N"])
    setph(data["Ph"])
    setplant(data["Plant"])

  }
  function onError(error) {
    console.error(error);
  }
  firestore().collection('Users').doc('sensor reading').onSnapshot(onResult,
onError);
  return (
    <ImageBackground source={image} resizeMode='cover' style={{ flex: 1 }}>
      <View style={{ alignItems: 'center' }}>
```

```
<View style={{ flexDirection: 'row', }} >
  <View style={{ marginTop: '10%', height: 210, width: width - 250,
backgroundColor: '#005555', alignItems: 'center', borderRadius: 15,
justifyContent: 'center' }}>
    <Text style={{ fontSize: 16, fontFamily: 'Poppins-Bold', color: 'white'
}}>Potassium value</Text>
    <View style={{ height: 160, width: width - 275, backgroundColor:
'white', borderRadius: 10, alignItems: 'center', justifyContent: 'center' }}>
      <Text style={{ fontSize: 50, fontFamily: 'Poppins-Light',
borderRadius: 15, color: 'black' }} >{K}</Text>
    </View>

  </View>

  <View style={{ marginTop: '10%', marginLeft: 50, height: 210, width:
width - 250, backgroundColor: '#005555', alignItems: 'center', borderRadius: 15,
justifyContent: 'center' }}>
    <Text style={{ fontSize: 16, fontFamily: 'Poppins-Bold', color: 'white'
}}>Nitrogen value</Text>
    <View style={{ height: 160, width: width - 275, backgroundColor:
'white', borderRadius: 10, alignItems: 'center', justifyContent: 'center' }}>
      <Text style={{ fontSize: 50, fontFamily: 'Poppins-Light',
borderRadius: 15, color: 'black' }} >{N}</Text>
    </View>

  </View>



</View>
```

```
    <View style={{ flexDirection: 'row' }} >
     <View style={{ marginTop: '20%', height: 210, width: width - 250,
backgroundColor: '#005555', alignItems: 'center', borderRadius: 15,
justifyContent: 'center' }}>
      <Text style={{ fontSize: 16, fontFamily: 'Poppins-Bold', color: 'white'
}}>Phosporous value</Text>
      <View style={{ height: 160, width: width - 275, backgroundColor:
'white', borderRadius: 10, alignItems: 'center', justifyContent: 'center' }}>
       <Text style={{ fontSize: 50, fontFamily: 'Poppins-Light',
borderRadius: 15, color: 'black' }} >{P}</Text>
      </View>

     </View>
     <View style={{ marginTop: '20%', marginLeft: 50, height: 210, width:
width - 250, backgroundColor: '#005555', alignItems: 'center', borderRadius: 15,
justifyContent: 'center' }}>
      <Text style={{ fontSize: 16, fontFamily: 'Poppins-Bold', color: 'white'
}}>Ph value</Text>
      <View style={{ height: 160, width: width - 275, backgroundColor:
'white', borderRadius: 10, alignItems: 'center', justifyContent: 'center' }}>
       <Text style={{ fontSize: 50, fontFamily: 'Poppins-Light',
borderRadius: 15, color: 'black' }} >{ph}</Text>
      </View>

     </View>



    </View>
   </View>
```

```
      <View style={{ alignItems: 'center', marginTop: 50 }}>
        <View style={{ width: width - 50, height: 120, backgroundColor:
'#005555', borderRadius: 20, alignItems: 'center', }}>
          <Text style={{ color: 'white', fontFamily: 'Poppins-Regular', fontSize: 20
}}>
            Suitable Plant is
          </Text>
          <Text style={{ fontSize: 35, fontFamily: 'Poppins-ExtraBold', color:
'white', textTransform: 'capitalize' }}>
            {plant}
          </Text>

        </View>
      </View>



    </ImageBackground>

  )

}
export default App;
```

Code for python Prediction

predict.py

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import io
from sklearn.metrics import confusion_matrix,classification_report
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
crop['no_label'] = pd.Categorical(crop.label).codes
plt.figure(figsize=(8,7))
sns.histplot(x='N',data=crop,color='b');
plt.title("Nitrogen for crops",{'fontsize':20});
plt.figure(figsize=(8,7))
sns.histplot(x='K',data=crop,color='b');
plt.title("Potassium for crops",{'fontsize':20});
plt.figure(figsize=(8,7))
sns.histplot(x='P',data=crop,color='b');
plt.title("Phosphorus for crops",{'fontsize':20});
X = crop.drop(['label','no_label'],axis=1)
y = crop.no_label
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2)
scalar = StandardScaler()
```

```
X_train = scalar.fit_transform(X_train)
X_test = scalar.transform(X_test)

rf = RandomForestClassifier()
rf.fit(X_train,y_train)
y_pred = rf.predict(X_test)
class_names = np.arange(0,21)
fig,ax = plt.subplots()
tick_marks = np.arange(len(class_names))
plt.xticks(tick_marks,class_names)
plt.yticks(tick_marks,class_names)
cnf_matrix = confusion_matrix(y_test,y_pred)
sns.heatmap(pd.DataFrame(cnf_matrix), annot = True,fmt = 'd')
ax.xaxis.set_label_position('top')
plt.tight_layout()
plt.title(f'Confusion Matrix for SVM', {'fontsize':20})
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
plt.show()
```

REFERNCES

1. A. Motwani, P. Patil, V. Nagaria, S. Verma and S. Ghane, "Soil Analysis and Crop Recommendation using Machine Learning," International Conference for Advancement in Technology (ICONAT), 2022, pp. 1-7, doi: 10.1109/ICONAT53423.2022.9725901 (2022)

2. A. S. H, S. Prathibha, P. K. P, S. F. M and Y. K. B, "Design of a Comprehensive Sensor based Soil and Crop Analysis System Model using Machine Learning Algorithms," 4th International Conference on Recent Developments in Control, Automation & Power Engineering (RDCAPE), 2021, pp. 327-332, doi: 10.1109/RDCAPE52977.2021.9633648. (2021)

3. A. Sharma, A. Jain, P. Gupta and V. Chowdary, "Machine Learning Applications for Precision Agriculture: A Comprehensive Review," in IEEE Access, vol. 9, pp. 4843-4873, doi: 10.1109/ACCESS.2020.3048415. (2021)

4. B. Kashyap and R. Kumar, "Sensing Methodologies in Agriculture for Soil Moisture and Nutrient Monitoring," in IEEE Access, vol. 9, pp. 14095-14121, doi: 10.1109/ACCESS.2021.3052478. (2021)

5. C. P. Wickramasinghe, P. L. N. Lakshitha, H. P. H. S. Hemapriya, A. Jayakody and P. G. N. S. Ranasinghe, "Smart Crop and Fertilizer Prediction System," 2019 International Conference on Advancements in Computing (ICAC), pp. 487-492, doi: 10.1109/ICAC49085.2019.9103422. (2019)

6. E. E. Vigneswaran and M. Selvaganesh, "Decision Support System for Crop Rotation Using Machine Learning," Fourth International Conference on Inventive Systems and Control (ICISC), 2020, pp. 925-930, doi: 10.1109/ICISC47916.2020.9171120. (2020)

7. G. S. Jayadeva, S. Shanmugasundaram, N. Sharath, C. Sanju, and S. M. Sidhalinga, “Smart Agricultural Crop Prediction Using Machine Learning”, IJRAMT, vol. 2, no. 7, pp. 144–147, Jul. (2021)

8. G. S. Pravallika, L. Kundana, K. S. Thanvi, G. Sirisha and C. Rupa, "Proficient Smart Soil based IoT System for Crop Prediction," 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), pp. 752-757, doi: 10.1109/ICIRCA48905.2020.9183054. (2020)

9. J. Nanwal and P. Sethi, "Identifying Best Suited Soil Based on its Physical and Chemical Properties Using Machine Learning," Fourth International Conference on Computational Intelligence and Communication Technologies (CCICT), 2021, pp. 35-38, doi: 10.1109/CCICT53244.2021.00018. (2021)

10. M. Jadhav, N. Kolambe, S. Jain and S. Chaudhari, "Farming Made Easy using Machine Learning," 2nd International Conference for Emerging Technology (INCET), 2021, pp. 1-5, doi: 10.1109/INCET51464.2021.9456351.C. (2021)

11. M. Kalimuthu, P. Vaishnavi and M. Kishore, "Crop Prediction using Machine Learning," 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), pp. 926-932, doi: 10.1109/ICSSIT48917.2020.9214190. (2020)

12. M. R. M. Kassim, "IoT Applications in Smart Agriculture: Issues and Challenges," 2020 IEEE Conference on Open Systems (ICOS), pp. 19-24, doi: 10.1109/ICOS50156.2020.9293672. (2020)

13. M. S. Farooq, S. Riaz, A. Abid, K. Abid and M. A. Naeem, "A Survey on the Role of IoT in Agriculture for the Implementation of Smart Farming," in IEEE Access, vol. 7, pp. 156237-156271, doi: 10.1109/ACCESS.2019.2949703. (2019)

14. M. U. Gul, K. John Pratheep, M. Junaid and A. Paul, "Spiking Neural Network (SNN) for Crop Yield Prediction," 2021 9th International Conference on Orange Technology (ICOT), pp. 1-4, doi: 10.1109/ICOT54518.2021.9680618 (2021)

15. Madhuri Shripathi Rao, Arushi Singh, N.V. Subba Reddy and Dinesh U Acharya “Crop prediction using machine learning” AICECS 2021 Journal of Physics: Conference Series 2161 012033 IOP Publishing doi:10.1088/1742-6596/2161/1/012033. (2022)

16. O. Friha, M. A. Ferrag, L. Shu, L. Maglaras and X. Wang, "Internet of Things for the Future of Smart Agriculture: A Comprehensive Survey of Emerging Technologies," in IEEE/CAA Journal of Automatica Sinica, vol. 8, no. 4, pp. 718-752, April, doi: 10.1109/JAS.2021.1003925. (2021)

17. P. A, S. Chakraborty, A. Kumar and O. R. Pooniwala, "Intelligent Crop Recommendation System using Machine Learning," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), pp. 843-848, doi: 10.1109/ICCMC51019.2021.9418375. (2021)

18. P. Parameswari, N. Rajathi and K. J. Harshanaa, "Machine Learning Approaches for Crop Recommendation," 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), pp. 1-5, doi: 10.1109/ICAECA52838.2021.9675480. (2021)

19. Renuka, Sujata Terdal, "Evaluation of Machine Learning Algorithms for Crop Yield Prediction", in IJEAT, vol. 8, pp. 4082-4086, doi: 10.35940/ijeat.F8640.088619. (2019)

20. R. Madhumathi, T. Arumuganathan and R. Shruthi, "Soil NPK and Moisture analysis using Wireless Sensor Networks," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1-6, doi: 10.1109/ICCCNT49239.2020.9225547. (2020)

21. S. M. PANDE, P. K. RAMESH, A. ANMOL, B. R. AISHWARYA, K. ROHILLA and K. SHAURYA, "Crop Recommender System Using Machine Learning Approach," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), pp. 1066-1071, doi: 10.1109/ICCMC51019.2021.9418351. (2021)

22. S. P. Raja, B. Sawicka, Z. Stamenkovic and G. Mariammal, "Crop Prediction Based on Characteristics of the Agricultural Environment Using Various Feature Selection Techniques and Classifiers," in IEEE Access, vol. 10, pp. 23625-23641, doi: 10.1109/ACCESS.2022.3154350. (2022)

23. V. Geetha, A. Punitha, M. Abarna, M. Akshaya, S. Illakiya and A. P. Janani, "An Effective Crop Prediction Using Random Forest Algorithm," 2020 International Conference on System, Computation, Automation and Networking (ICSCAN), pp. 1-5, doi: 10.1109/ICSCAN49426.2020.9262311. (2020)

24. V. Vanarase, V. Mane, H. Bhute, A. Tate and S. Dhar, "Crop Prediction Using Data Mining and Machine Learning Techniques," 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), pp. 1764-1771, doi: 10.1109/ICIRCA51532.2021.9544724. (2021)