PRECISION AGRICULTURE – COIMBATORE BASED CROP SUGGESTION SYSTEM USING ML AND IOT

Bona fide record of work done by

Gayathri B (18z218)
Poovizhi Mangai K (18z236)
Praja S S (18z237)
Shivathmica C (18z249)
Sneha Sridharan (18z252)

Dissertation submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF ENGINEERING

Branch: COMPUTER SCIENCE AND ENGINEERING

of Anna University

April 2021

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
PSG COLLEGE OF TECHNOLOGY
(Autonomous Institution)
COIMBATORE – 641 004

PSG COLLEGE OF TECHNOLOGY

(Autonomous Institution)

COIMBATORE – 641 004

PRECISION AGRICULTURE – COIMBATORE BASED CROP SUGGESTION SYSTEM USING ML AND IOT

Bona fide record of work done by

Gayathri B (18z218)
Poovizhi Mangai K (18z236)
Praja S S (18z237)
Shivathmica C (18z249)
Sneha Sridharan (18z252)

Dissertation submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF ENGINEERING

Branch: COMPUTER SCIENCE AND ENGINEERING

of Anna University

April 2021

....

Dr. N. Gopika Rani **Dr.G.Sudha Sadasivam**

Faculty guide Head of the Department

Certified that the candidate was examined in the viva-voce examination held on **15-04-2021**

....

(Internal Examiner) (External Examiner)

CERTIFICATE

This is to certify that the project titled **"Precision Agriculture – Coimbatore based crop suggestion system using ML and IoT"**for **Niagara Automation, Coimbatore**was completedby the students studying 3rdyear B.EComputer Science & Engineeringat PSG College of Technology, Coimbatore,under the guidance of**Dr. N. Gopika Rani**and the duration of the project is from 28.12.2020 to 12.04.2021.

The students involved in the projects are as given below:

1. Gayathri B (18Z218)
2. Poovizhi Mangai S K (18Z236)
3. Praja S S (18Z237)
4. Shivathmica C (18Z249)
5. Sneha Sridharan (18Z252)

Date: 12 - 04 - 2021

For Niagara Automation Company

NIAGARA AUTOMATIONS

Authorized Signature

CONTENTS

AKNOWLEDGEMENT

We would like to express our special thanks to **Dr. K. Prakasan**, Principal In-charge for giving us this opportunity to develop and complete our project in our field of study.

We express our sincere thanks to **Dr. G.Sudha Sadasivam**, Professor & Head, Department of Computer Science and Engineering for encouraging and allowing us to present the project at our department for the partial fulfilment of the requirements leading to the award of BE degree.

We profoundly thank **Dr. G.R.Karpagam**, Professor & Program Coordinator for her constant support.

We take immense pleasure in conveying our thanks and a deep sense of gratitude to our project guide, **Dr.N.Gopika Rani**, Assistant Professor (SG), Department of Computer Science and Engineering, for her valuable inputs, guidance, encouragement, wholehearted cooperation, and constructive criticism throughout the duration of our project. It was a great privilege to work under her guidance.

It is our privilege to express our sincere regards to our Tutor, **Dr. C.Kavitha**, Assistant Professor (SG), Department of Computer Science and Engineering, for her valuable suggestions and motivation all through the course of the project.

We are obliged in taking the opportunity to sincerely thank **Mr.Uthamaraj**, Managing Director of Niagara Automations for providing us an internship and advising us during this project. We convey our sincere gratitude to the Soil Testing Lab and Tamil Nadu Agriculture University, Coimbatore for allowing us to research and collect the necessary data.

We are extremely grateful to all the faculty members of the department, friends and family who have always been there to guide us.

SYNOPSIS

This project is to help the farmers resolve various uncertainties about agriculture and learn new techniques to promote and make agriculture more profitable. The proposed system predicts crop for any farmland situated at Coimbatore, Tamil Nadu. This system operates based on a combination of machine learning models such as Random forest, K - Nearest Neighbours and Decision tree as an ensemble model. IoT sensors – temperature and soil pH sensor are also deployed for a precise outcome.

This automated crop suggestion system alleviates farmers from the pitfalls of traditional farming practices and paves way for better agricultural profit and productivity. Furthermore, it proves to give the best accuracy and present the most accurate result. Thus, the most precise crop to be grown in the farmer’s agricultural land in Coimbatore is suggested.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

A crop recommendation system has been developed, with an intention to assist the farmers in the Coimbatore district of Tamil Nadu in helping them select appropriate crops based on the soil in their farms. The goal is to mitigate the traditional farming methods, and to achieve this, a combination of machine learning models such as Random forest, K-Nearest Neighbour and Decision tree as an ensemble model is trained. The soil quality is governed by certain parameters like soil and environmental characteristics like temperature and rainfall. The aforementioned ensemble model is constructed and trained based on these input parameters. The trained model is further given real time inputs from the Internet of Things system. In addition, user inputs are also given to generate the output from the ensemble system. In the final step, the most appropriate crop to be grown in that specific soil is recommended to the farmer by considering all the pertinent soil based and environmental parameters by the ensemble model.

1.2 INTRODUCTION

India is the world's second largest producer of wheat and rice and world's first producer of several other crops. So, it's not surprising that agriculture rules more than half of India's population economically.

By taking into account of the undulating nature of the environment and soil parameters that Indian agriculture relies on, farmers don't find the most precise agriculture techniques for better yield. But they are unaware of such influencing parameters. They always prefer to trust their instincts and follow traditional farming methods. Such unsure practices may unfortunately cause the anticipated productivity and profitability become unsubstantial. Since soil is a non-renewable natural resource the soil health and land productivity may deteriorate overtime. This may lead to unpleasant implications on the farmer’s life as well as the status of food security in the specific region. In the thought of extenuating this situation, smart farming techniques can be imbibed.

A crop recommendation system developed for the Coimbatore region will be beneficial. In our system Machine learning models such as Random forest, KNN and Decision tree are combined to form an ensemble model. The ensemble model gives more prediction and efficiency than any of its models could achieve alone. This system is developed based on certain input parameters like soil pH, EC (Electrical Conductivity), type, N, P, K and environmental conditions like temperature and rainfall will be beneficial. Sensor readings from IoT along with user inputs like soil type, rainfall is fed into the trained model to give the output. This system alleviates the conventional method by providing precise suggestion on the most appropriate crop to be grown in the farmer’s agricultural land in Coimbatore.

1.3 MOTIVATION

Coimbatore is a district in the state of Tamil Nadu which is situated on the banks of the Noyyal River and surrounded by the Western Ghats. Coimbatore is the second largest city in Tamil Nadu. The cultivable soil of the district is classified into 16 different soil series having different soil properties. All sorts of traditional farming practices do exist here. Farmers of Coimbatore are unaware of the heterogeneous nature of soil. There are differences in soil properties occurring even within short distances. Growing the same crop as the neighbouring farmer will not be as profitable as it is for them. Also, choosing a crop for cultivation based only on market trends will cause adverse loss financially and worsen the quality of soil.

1.4 SCOPE

The proposed system addresses various issues of traditional farming techniques. It is hugely beneficial for farmers in Coimbatore. It provides precise suggestion on the crop to be grown for imbibing better farming techniques. Enhanced techniques, collection of data and improved efficiency are all inclusive in the scope of the implementation of this system.

CHAPTER 2

LITERATURE SURVEY

2.1 OVERVIEW

This chapter consists of the different methods and algorithms used for crop recommendation.

Zeel Doshi, et.al [2018] discussed about environmental parameters like temperature, rainfall, farm's latitude, longitude, altitude and distance from the sea and soil characteristics such as pH value, soil type and thickness of aquifer and topsoil. Crops used were bajra, jowar, maize, rice and wheat, barley, cotton, groundnut, gram, jute, other pulses, potato, ragi, tur, rapeseed and mustard, sesame, soybean, sugarcane, sunflower and tobacco crops. They compared various machine learning models that have in built support for Multi-label classification (MLC): Decision Tree, K Nearest Neighbor (K-NN), Random Forest and Neural Network and obtained 90.2%, 88.78%, 90.43% and 91% efficiency respectively. Here we learnt the various models that are suitable for MLC and thus our proposed solution combines the power of all 3 models to get a better efficiency.

S.Pudumalar, et.al [2017] stated that a suggestion system based on site specific and soil parameters would be the most useful for farmers. The attributes considered in this paper: Depth, Texture, pH, Soil Color, Permeability, Drainage, Water holding and Erosion. The crops chosen by them include millet, groundnut, pulses, cotton, vegetables, banana, paddy, sorghum, sugarcane, coriander. A recommendation system using an ensemble model with majority voting technique was proposed. The base learners used were Random tree, CHAID, K-Nearest Neighbor and Naive Bayes and it was observed that the prediction accuracy of the model was 88%. Ensemble model is one of the most preferred for this system according to these set of papers and therefore we decided to opt for it.

Kevin Tom Thomas, et.al [2020] used environmental factors like temperature, humidity etc and soil parameters such as N, P, K and pH of the soil in their input dataset. The algorithms: kNN, kNN with cross validation, Decision Tree, Naïve Bayes and SVM were considered for evaluation and their respective efficiencies were 85%, 88%, 81%, 82% and 78%. It was inferred that kNN with cross validation was the ideal model and thus we decided to adopt kNN as one of the base learners in our ensemble model. The idea of adding IoT sensors to measure soil properties in their scope for improvement section inspired us to include the same in our project.

Harshitha L, et.al [2020] discussed about improving crop productivity through crop recommendation. Nutritional features in soil like pH values, organic Carbon, iron, zinc, nitrogen, phosphorus, sulphur have been used as it is assumed that chemical analysis of the soil helps in improving crop production. Other attributes include rainfall and temperature. Algorithm chosen by them is Naïve Bayes Algorithm.

K. R. Akshatha, et.al [2018] considered soil based attributes like depth, texture, pH, soil color, permeability, drainage, water holding capacity and erosion in their dataset. The crop recommendation system is built using an ensemble model – majority voting technique, using a combination of models like Random tree model, K-Nearest Neighbor model and Naïve Bayes model. The crops recommended by mapping with appropriate soil parameters are millet, groundnut, pulses, cotton, vegetables, banana, paddy, sorghum, sugarcane, coriander.

Srilakshmi A, et.al [2020] aims to get the best solution for crop prediction and recommendation system by comparing machine learning algorithms like random forest, decision tree and SVM. Accuracy obtained for random forest is 95.09%, higher than the rest of the models. The datasets are trained using SVM – grid search and random forest algorithms. The soil parameters taken into account for predicting crop based on environmental factors are temperature, humidity, pH and rainfall. The dataset consists of 3100 rows x 5 columns obtained from online that required no preprocessing. It was also found that combination of SVM and decision tree gave an accuracy of 91.8%, randomized search used on random forest gave 94.7% accuracy. Hence, it is concluded that random forest algorithm performs the best accuracy when compared to other models. Future work comprises of including more soil parameters to perform crop prediction.

Miss Snehal S.Dahikar, et.al [2014] focused on making the farmers attain maximum crop yield at minimum cost using ANN – feed forward back propagation technique. The yield for a crop is predicted based on the soil specific parameters like pH, depth, nutrient contents like nitrogen, phosphate, potassium, organic carbon, magnesium, sulphur, manganese, copper, iron and environment-based parameters like temperature, and rainfall. It is concluded that ANN can be a beneficial tool for prediction.

Varsha A, et.al [2020] proposed to predict a specific crop grown in a soil. The dataset collected IoT system are soil moisture, soil pH and gas content of soil. Firstly, data mining techniques are applied on the sensor values then machine learning steps are applied. A sophisticated front end web interface is also developed for the farmers to view the recommended crop for their soil. The idea of adding IoT sensors to measure soil properties inspired us.

Avinash Kumar, et.al [2019] proposed a recommendation system using soil type, average rainfall, average temperature, pH and its corresponding crop type as attributes for 15 different crops. They have used 3 different algorithms individually to figure out which one provides the best accuracy (SVM – 89.66%, Decision Tree – 86.80%, Logistic Regression – 86.04%). It also provides details of Pests and its management for the recommended crop using Information Retrieval.

Tanmay Banavlikar, et.al [2018] designed a system that deploys soil moisture sensor, temperature sensor, humidity sensor embedded on Raspberry Pi to detect the physical parameters of the soil. Artificial Neural Networks concept is used to train the datasets to suggest the appropriate crop. The attributes for training the datasets include soil moisture content, humidity, temperature and the apt crop.

Devdatta A. Bondre, et.al [2019] described how to predict crop yield based on soil nutrients and recommends fertilizers. The datasets are collected for attributes such as fertilizer, location and crop yield. Crops considered are rice, wheat, soybean, sunflower, cotton, sugarcane, tobacco and dry chilli. Random Forest and Support Vector Machine were used for Crop Yield Prediction and fertilizer recommendation. They arrived at an accuracy of about 86.35% for support vector machine and 99.47% for random forest which inspired us to use random forest as one of the model for our project.

Dr.A.K.Mariappan, et.al [2020] proposed a system to predict crop suitability for particular location and recommends the most suitable fertilizer. They collected data sets for soil and crops. The algorithm used here is Random forest and KNN algorithm.

Preethi G, et.al [2020] described about the method that takes the soil and pH sample as inputs to recommend the suitable crop and fertilizer as the solution. So, the soil information is collected through sensors and the data is transmitted from the Arduino through Zigbee and WSN (Wireless Sensor Network) to MATLAB. Analysis of the soil data and processing is done with help of ANN (Artificial Neural Network) and crop recommendations is done using SVM (Support Vector Machine). We gathered some IoT ideas from this paper.

Soumya Sri Attaluri, et.al [2020] provided a crop recommendation system with a prime motive of creating economic welfare of farmers. Factors such as cost of planting, cost of harvesting, rainfall, crop demand, cost of seed, cost of fertilizer and yield of crop are considered to generate a more accurate prediction of whether harvesting a crop will yield profits or not. Dataset for this research is sourced from Indian government websites published by different ministries related to agriculture.

2.2 COMPARING WITH EXISTING SYSTEM

Paper Name	Methodology used	Advantages	Disadvantages
AgroConsultant: Intelligent Crop Recommendation	Decision Tree (90.2%)	Compared various algorithms and	Absence of sensors for input making it essential

System Using Machine Learning Algorithms	K Nearest Neighbor (K-NN) (88.78%) Random Forest (90.43%) Neural Network (91%)	chose the best one. Accuracy of the model is very high.	for users to type all the inputs. Multiple models are not used together.
Crop Recommendation System for Precision Agriculture	Ensemble Model (Voting Based classifier) Random tree, CHAID, K-Nearest Neighbor and Naive Bayes (88%)	Combined the power of multiple models. Use of varying soil parameters.	Environmental parameters are not used in this model.
Crop Prediction Using Machine Learning	KNN (85%) KNN with cross validation (88%) Decision Tree (81%) Naïve Bayes (82%) Support Vector Machine (SVM) (78%)	Compared various algorithms and chose the best one. Tried two different methods for KNN.	Absence of sensors for input making it essential for users to type all the inputs (mentioned in their scope for improvement)
Improving Crop Productivity	Naïve Bayes	Nutrient content of the soil is given	Use of only one model in the entire solution.

Through Crop Recommendation		importance in this system.	
Implementation of Machine Learning Algorithms for Crop Recommendation Using Precision Agricuture	Ensemble model with majority voting technique, using a combination of models like Random tree model, K-Nearest Neighbor model and Naïve Bayes model.	-Sufficient soil specific attributes are considered. - Ensemble model gives more prediction and efficiency than any of its models could achieve alone.	- The combination of these models may not be complementing each other so hence might not produce high accuracy in recommendation.
Machine learning approach: Recommendation of suitable crop for land using meteorological factors	Random forest, decision tree and SVM	-Ample number of datasets are collected for training and testing for better accuracy. - Aimed at deciding which model has highest accuracy: Random forest - 95.09%.	-Less soil parameters are considered. -Only static data is provided to the trained model.
Agricultural Crop Yield Prediction Using Artificial Neural Network Approach	Artificial Neural Networks	-Good number of soil-based attributes like different soil nutrients used. -Use of ANN gives best results for pattern recognition.	-Using ANN alone does not give higher accuracy when compared to ensemble models for prediction systems.

Soil classification and crop recommendation using IoT and Machine Learning	NA	-Use of IoT for getting real time datasets -Well developed front end for farmer's usability.	- Preprocessing of inputs from IoT system (sensors) every time are time consuming and makes the system complex.
Recommendation System for Crop Identification and Pest Control Technique in Agriculture	SVM Classification (89.66%) Decision Tree (86.80%) Logistic Regression (86.04%)	All major parameters are included for recommendation system. Used different algorithms to choose the one with highest accuracy. Apart from identification, Pest control techniques is also suggested.	Since the user manually input the values, he may be unaware of the exact values which might mislead.
Recommendation System Using Neural Networks	Artificial Neural Networks	The input data are collected using sensors to get the accurate value which gives the correct recommendation.	Few more parameters could be taken into account for more effective recommendation.
Prediction of crop yield and fertilizer recommendation using machine learning algorithms	Support vector machine (86.35%) Random Forest (99.47%)	Compared various algorithms and chose the best one	

Crop Recommendation System through Soil Analysis Using Classification in Machine Learning	Random Forest and KNN algorithm	Compared various algorithms and chose the best one	Absence of sensors for input making it essential for users to type all the inputs.
Agro based crop and fertilizer recommendation system using machine learning	Artificial Neural Networks and Support Vector Machine	Gathered ideas For using soil sensors	Accurate results are not mentioned by using artificial neural networks and environmental parameters are not used properly.
Crop Plantation Recommendation using Feature Extraction and Machine Learning Techniques.	KNN, ANN (regression model)	Simulations are performed using to improve the model.	Market price of the crop and yield of the crop are major impacting factor

Table 2.1 – Overview of Literature Survey

CHAPTER 3

SYSTEM DESIGN

3.1 OVERVIEW

A detailed design of the proposed system is discussed in this chapter along with the workflow of the entire system and the IoT module.

3.2 WORKFLOW

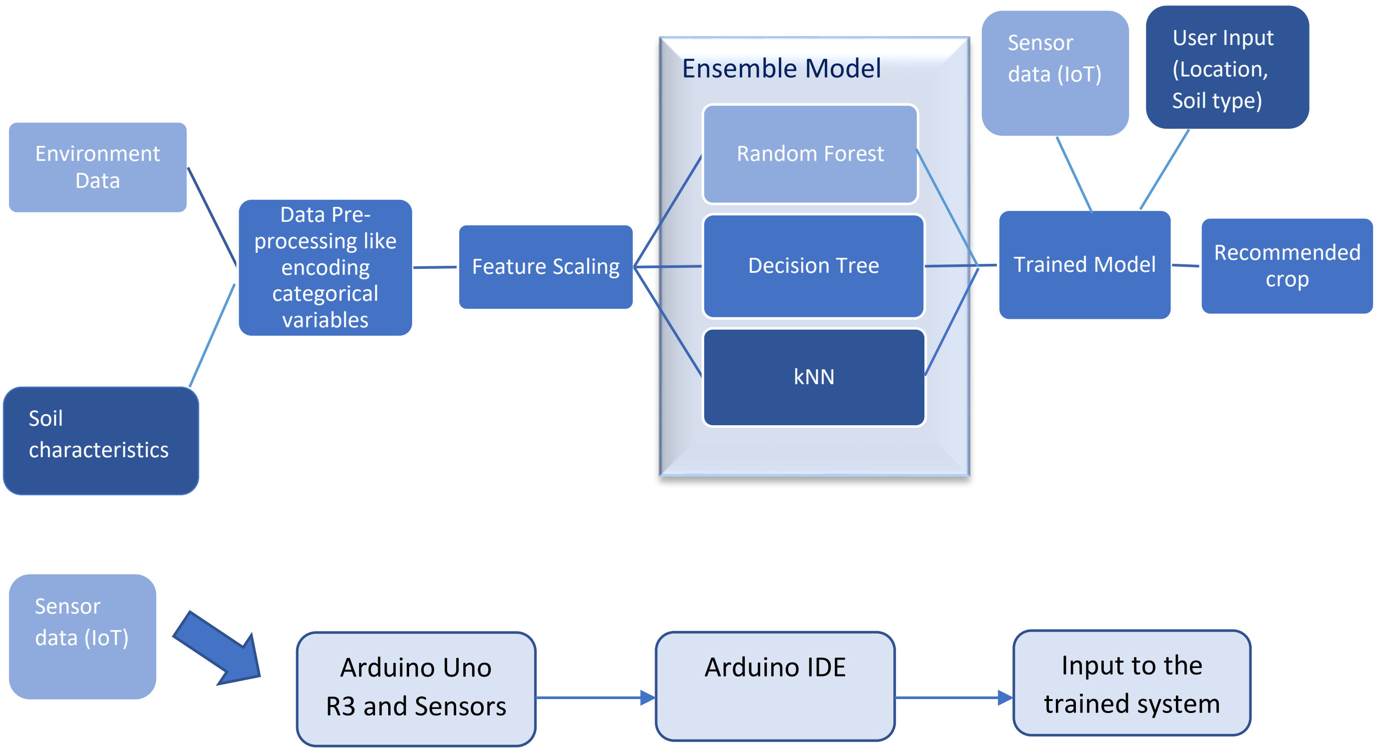

Fig 3.1 - Workflow of the Proposed Model

Fig 3.1 describes the workflow starting from the data pre-processing stage till the user input prediction stage. The IoT Module is separately shown below where the sensor reading is read and transmitted from the Arduino IDE to the input system.

3.3 ARCHITECTURE DEFINITION

- Firstly, data is collected from Soil Survey department and the Tamil Nadu Agriculture University.
- The data collected is then one hot encoded to convert categorical variables into numeric variables.
- After pre-processing, data is normalized to give equal priority to all the attributes.
- It is then provided as input to the ensemble model for training.
- The sensors are connected to the Arduino board and pH and temperature values are collected. This data is then sent to python serially so that it can serve as input to the trained model.
- User input and input from the IoT sensors is supplied to the trained model and the output is collected.
- The recommended crop is then presented as the output.

CHAPTER 4

SYSTEM REQUIREMENTS

4.1 OVERVIEW

This chapter describes the various hardware and software required for implementing the proposed algorithm.

4.2 HARDWARE REQUIREMENTS:

IoT:

- Arduino Uno R3
- Temperature and Humidity Sensor
- Soil pH Sensor

Operating Systems: Windows 8 or above

Memory: Minimum 4 GB RAM

4.3 SOFTWARE REQUIREMENTS:

- Arduino IDE 1.8.0
- Python
- Jupyter Notebook

4.4 SOFTWARE DESCRIPTION

4.4.1 JUPYTER NOTEBOOK

Jupyter Notebook version 6.3.0, an Open Source IDE provides an interactive environment by running a sequence of cells for development and presenting our project. It is a popular python IDE to perform modelling, machine learning, data transformation and visualization. Libraries like Scikit Learn, Numpy are available without additional installations.

4.4.2 ARDUINO IDE

Arduino IDE version 1.8.13 is another open source IDE that is mainly used for writing and compiling the code into the Arduino UNO module. It provides readily available libraries suitable for the sensors. The main code known as sketch, written on the IDE generates a Hex File which is then transferred and uploaded in the controller on the board. The IDE environment consists of editor and compiler. Editor is used for writing the required code and compiler is used for compiling and uploading the code into Arduino UNO.

CHAPTER 5

PROPOSED METHODOLOGY

5.1 OVERVIEW

The proposed methodology is explained elaborately. The machine learning models used along with mathematical formulas are discussed. Description of the sensors used in the IoT module is also specified.

5.2 MACHINE LEARNING MODULE

5.2.1 ENSEMBLE LEARNING TECHNIQUE

Ensemble learning technique in machine learning integrates multiple models to achieve increased performance. It combines the power of multiple ML algorithms so that one model can correct the errors of another and predict results with much higher accuracy. Two or more models can be used as the base learners in this technique. The base learners should be chosen such that each of them perform well individually so that the combined model gives better results.

In our project, we are planning to use k-Nearest Neighbours, Decision Tree and Random Forest as our base learners for the Majority Voting technique. Majority Voting is an ensemble technique that is one of the best models to solve classification problems. In this, the training set is supplied to all the models and each of them gets trained individually. The inputs for prediction is then fed to all the models separately. Each prediction result is counted as a vote and the output with the maximum number of votes is given as the final result.

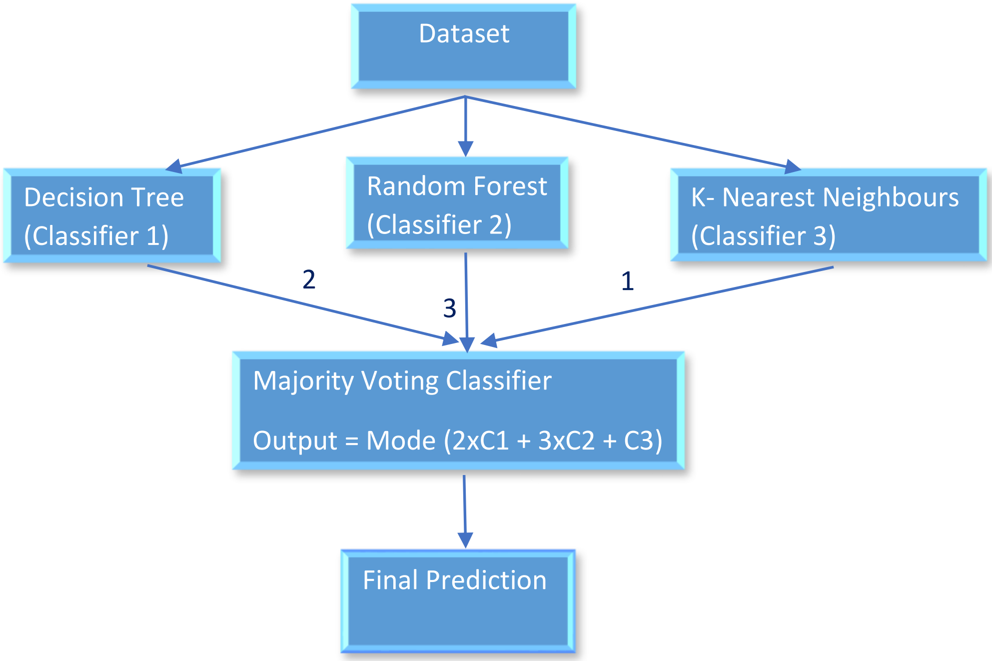

Fig 5.1 – Weighted Ensemble Learning Model Flowchart

Fig 5.1 shows the working of the proposed system. Weighted ensemble technique is used with Decision Tree having weight 2, Random Forest having weight 3 and K-Nearest Neighbour having weight 1.

5.2.2 LEARNERS USED IN THE MODEL

5.2.2.1 DECISION TREE

Decision Tree is a non-parametric supervised machine learning technique used for solving both classification and regression problems. It often mimics a human's thinking ability while making a decision. It is a tree-like structure where the internal nodes represent the features of a dataset; the branches act as the decision rules, and leaf nodes are the final result of the decisions. This uses different attribute selection measure to decide on where to split. They are namely Gini Index, Information Gain, Reduction in Variance and Chi Square. The tree algorithm depends on the selected attribute measure.

GENERAL WORKING:

1. Start the tree with root node S (complete dataset)
2. Find the best attribute using Attribute Selection Measure, Information Gain. The categorical feature that will maximize the information gain value using the impurity criterion entropy is chosen at each level.

$$\mathrm{Entropy(T)} = \sum_{i=0}^{n} - f_i \log(f_i)$$

n – number of unique values of the target variable

f_i – probability of i^{th} value of target variable

$$\mathrm{Gain(T,X)} = \mathrm{Entropy(T)} - \mathrm{Entropy(T,X)}$$

T = target variable

X = Feature to be split on

Entropy(T,X) = The entropy calculated after the data is split on feature X

3. Divide S into subsets
4. Generate decision tree node (which contains the best attributes)
5. Recursively continue from step 3

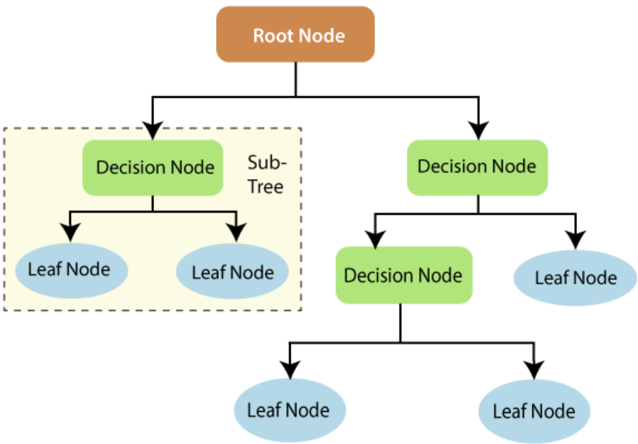

Fig 5.2 – Structure of Decision Tree

Fig 5.2 shows the branches and internal node structure of the decision tree. The branches are the unique categories of the attributes and the leaves indicate the attribute name.

5.2.2.2 RANDOM FOREST:

Random Forest is a popular algorithm that belonging to the supervised learning technique. It basically involves the process of combining multiple classifiers to solve a complex problem and to improve the performance of the model, which is generally called as ensemble learning. Random Forest combines the use of many decision trees. It takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

GENERAL WORKING:

1. Number of decision trees to be constructed is chosen as N.
2. Any random subset of the training set is considered to construct each decision tree.
3. The decision trees for the selected data points is constructed in a similar way using information gain as attribute selection measure.
4. Repeat steps 2 and 3 for N times.
5. The category of decision trees that wins the majority votes in the predictions will be assigned for the new data points.

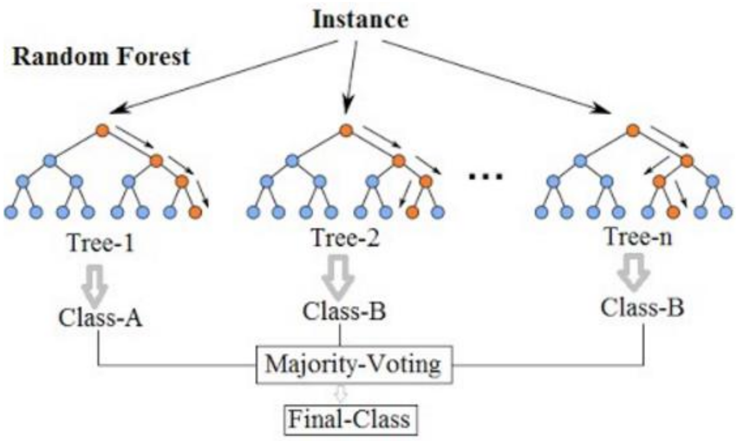

Fig 5.3 – Construction of Random Forest with N Decision Trees

Fig 5.3 illustrates the construction of random forest by considering the outputs of many (N) decision trees.

5.2.2.3 K-NEAREST NEIGHBOUR:

K-NN is a non-parametric algorithm, that stores the dataset and at the time of classification, it performs an action on the dataset. For this reason, it is also called a lazy learner algorithm. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

GENERAL WORKING:

1. Number K of the neighbours is selected.
2. The Euclidean distance (Minowski distance with p=2) is calculated to find the K nearest neighbours.

$$\text{Minowski Distance} = \sum_{i=1}^{n} \sqrt[p]{|x_i - y_i|^p}$$
$$\text{When } p = 2, \text{Euclidean Distance} = \sum_{i=1}^{n} \sqrt{|x_i - y_i|^2}$$

x and y are the two points

n – number of dimensions of the two points

3. The number of data points that belong to each category of target variable is counted.
4. The category that has the maximum number of neighbour will be assigned to the new data point.

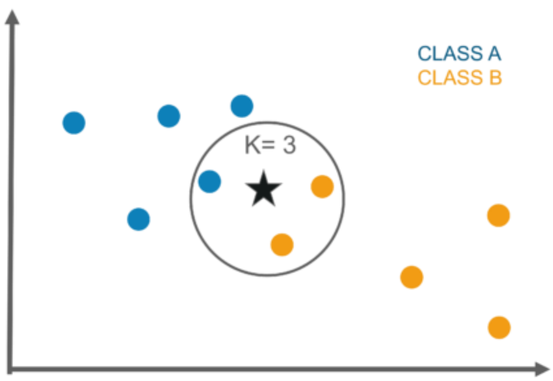

Fig 5.4 – Working of K-Nearest Neighbour Algorithm

Fig 5.4 demonstrates the selection of the nearest neighbours in the kNN algorithm.

5.3 INTERNET OF THINGS (IoT) MODULE

It refers to the different hardware components that are connected to the internet, collecting and sharing data. IoT devices comprises of sensors and mini-computer processors that works on the data collected by the sensors using machine learning.

5.3.1 DHT11 TEMPERTURE SENSOR:

The DHT11 is a simple, low-cost digital temperature and humidity sensor which can measure the temperature and humidity of the surroundings. It uses a thermistor to measure the input data and convert it into electronic data to record, monitor, or signal temperature changes.

5.3.2 pH SENSOR:

pH scale shows the acidity and basicity of a solution. It has values ranging from 0 (acidic) to 14 (basic). pH sensor consists of two main components – pH electrode and signal convertor. pH electrode measures the potential between the electrodes and converts it to pH units. Signal convertor is a voltage regulator chip which supports the voltage supply of 3.3-5.5V DC.

CHAPTER 6

IMPLEMENTATION

6.1 OVERVIEW

This chapters entails the entire process of implementation of the proposed system from dataset collection to user input prediction.

6.2 DATASET COLLECTION

The dataset was obtained from the Soil Testing Lab and Tamil Nadu Agricultural University, Coimbatore. This dataset consists of the soil and environmental parameters of all the areas in Coimbatore district. The input parameters considered are Soil N, P, K, EC, pH, Soil Type, Temperature and Rainfall. The data has 20 different crops that are widely grown here like Banana, Bengalgram, Bhendi, Brinjal, Blackgram, Onion, Paddy, Redgram, Sugarcane, Tapioca, Cotton, Cowpea, Cumbu, Greengram, Groundnut, Sorghum, Horsegram, Tomato, Turmeric, Coconut.

6.3 DATA PRE-PROCESSING

Data pre-processing is a technique used in data mining, machine learning which is used to transform the raw data into useful datasets that can be used to perform various tasks. It is generally the first step of any learning model.

Categorical attributes need to be encoded to have numerical values for correct working of the machine learning system. One Hot Encoding is a popular technique used for the same where a binary variable is created for each of the unique values of that categorical variable. This ensures that no ordinal relationship between the different values of the variable is assumed.

6.4 FEATURE SCALING

Feature scaling is a method used to normalize data in each attribute. This is mainly used to keep all attributes in the same value range to avoid giving higher priority to attributes with higher values.

The most appropriate method to do this is: Standardization (Z-score Normalization)

$$X' = \frac{X - \mu}{\sigma}$$

Where μ – mean of the sample

σ – standard deviation of the sample

6.5 MODEL BUILDING

6.5.1 MACHINE LEARNING MODULE

Random Forest, Decision Tree and kNN are combined to build the majority voting ensemble model to recommend the most suitable crop. Here, each model is first defined. Information gain is used as the criteria to decide the attributes at each level of the decision tree. In Random Forest, 10 decision trees are constructed and combined to predict the result. Euclidean distance (p=2) is considered to determine 26 nearest neighbours in kNN whose output value determine the predicted crop.

These results are combined with weights 2,3,1 for Decision Tree, Random Forest and kNN respectively to give the final predicted crop of the machine learning system. The accuracy score obtained is 0.8927.

6.5.2 IoT MODULE

Temperature sensor and pH sensor are attached to Arduino UNO board to collect the soil attributes like temperature and pH and give it as an input for prediction.

DHT11 Temperature Sensor:

This sensor has 3 pins – Vcc, Gnd and Data.

Pins in Sensor	Arduino Connection
Vcc	Vin
Gnd	Gnd
Data	A1

Table 6.1 - DHT11 Connections

DHT library <dht.h> is added to the Arduino. Temperature values are sensed using DHT.read11(dht_pin) function and the data is printed using the function Serial.print(DHT.temperature)

pH Sensor:

It consists of Signal conversion module and a pH electrode. It has 6 pins – V+, Gnd, Gnd, Po, Do,To

Pins in Sensor	Arduino Connection
V+	5V
Gnd	Gnd
Po	A0

Table 6.2: pH Sensor Connections

pH electrode should be calibrated in order to get accurate values. A solution whose pH value is known is taken (distilled water pH = 7) and the electrode is dipped onto it to find the value it prints. Finally the difference is added to the calibration value (calibration_value = 20.44).

pH values are sensed using analogRead(A0) function and the pH is calculated using nernst equation

$$\text{volt} = \text{avgval}*5.0/1024/6$$

$$\text{ph} = -5.70 * \text{volt} + \text{calibration\_value}$$

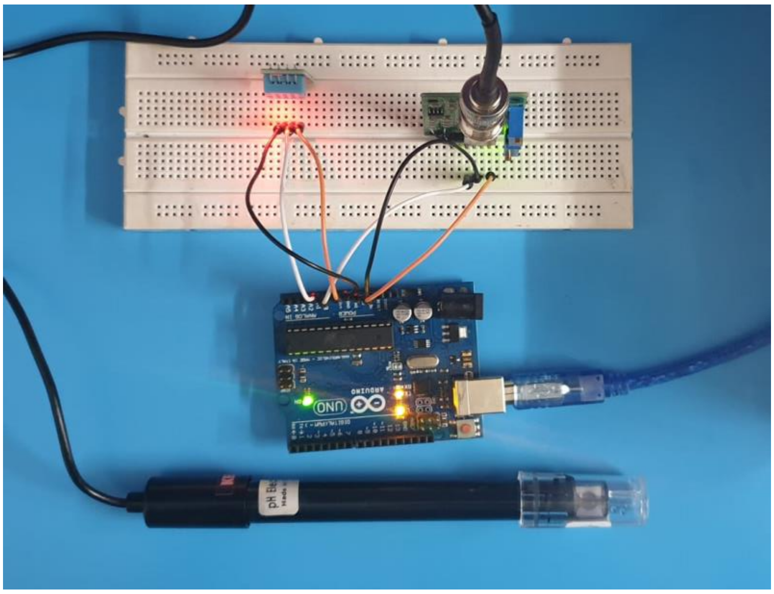

Fig 6.1 - Circuit Connections

Fig 6.1 displays the connections made between temperature sensor, pH sensor and the Arduino Board.

Connecting IoT module to Python:

The data collected from the sensors is collected and sent via serial port. PySerial package is used to access serial port. The average of the sensor values are computed and stored for prediction.

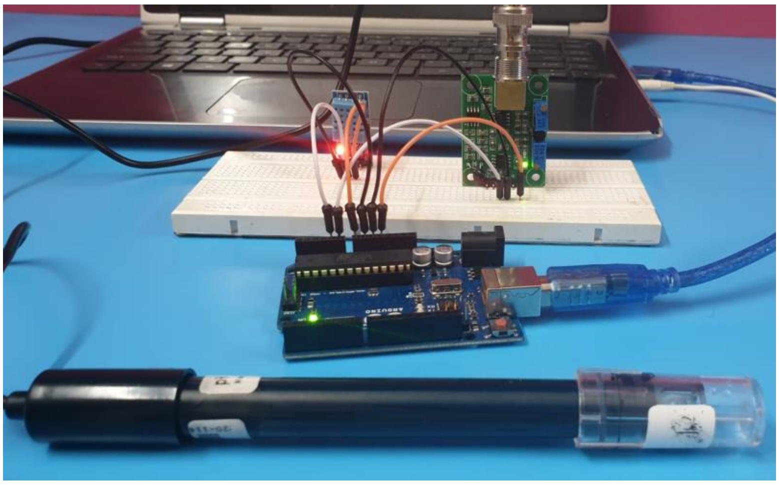

Fig 6.2 - Arduino Connected to Laptop

Fig 6.2 Lateral view of the connections that shows the Arduino connections with the laptop

6.6 USER INPUT PREDICTION

The most appropriate crop is recommended to users for a user defined input. The values entered by the users for attributes such as Soil Type, Rainfall, N, P, K, EC are combined with the readings received from the Temperature and pH sensors and the most appropriate crop is predicted.

CHAPTER 7

RESULT ANALYSIS

The dataset obtained from Soil Testing Lab and Tamil Nadu Agricultural University, Coimbatore consists of 1162 records of crop data for different areas in Coimbatore. This was split into train and test set with 930 records in the train set and 232 record in test set.

```
[[23  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  5  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  3  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  1  0  3  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0 33  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 1  0  0  0  0  0  5  0  0  0  0  0  0  0  0  2  0  0  0  0]
 [ 0  0  0  0  0  0  0 13  0  0  0  0  0  0  0  3  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  4  0  0  0  0  0  0  1  0  0  0  0]
 [ 1  0  0  0  0  0  2  0  0  7  0  1  0  0  1  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  3  0  0 11  1  0  0  0  3  0  0  0  0]
 [ 0  0  0  0  0  0  0  1  0  0  0  8  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  1  0  0  0  0  9  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  4  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  5  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 45  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  5  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  7  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 14  0]
 [ 0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  3]]
Out[40]: 0.8927038626609443
```

Fig 7.1 – Confusion Matrix and Accuracy Score

Fig 7.1 represents the confusion matrix and accuracy score which comes to 89.27%.

This is the confusion matrix that we obtained for our model. It is a 20 x 20 matrix that shows the accuracy of crop predictions. The diagonal of the matrix gives the number of correctly predicted crops and the rest shows the number of incorrect predictions. The accuracy score obtained is 0.8927 which indicates that the system is 89.27% accurate.

```
Temperature readings from Arduino: 26.0
pH readings from Arduino: 7.25
Temperature readings from Arduino: 26.0
pH readings from Arduino: 7.25
Temperature readings from Arduino: 26.0
pH readings from Arduino: 7.27
Temperature readings from Arduino: 26.0
pH readings from Arduino: 7.44
Temperature readings from Arduino: 26.0
pH readings from Arduino: 7.26
Average Temperature: 26
Average pH: 7.29
```

Fig 7.2 – Readings from Sensor

Fig 7.2 displays the temperature and pH readings recorded by the sensors.

The temperature and pH from the sensor is received from the Arduino board. The average of five values is taken for better accuracy.

```
Area: Pogalur
Soil type: Irugur Series
N value: 218.41
P value: 12.63
K value: 470.6
EC value: 0.27
pH: 7.29
Temperature: 26
Rainfall: 1500
Predicted Crop: Coconut
```

Fig 7.3 – User Input Prediction

Fig 7.3 shows the prediction of the proposed system for a user specified input along with sensor input.

The required attributes are collected as input from the user. This along with the sensor values are used to recommend the most suitable crop.

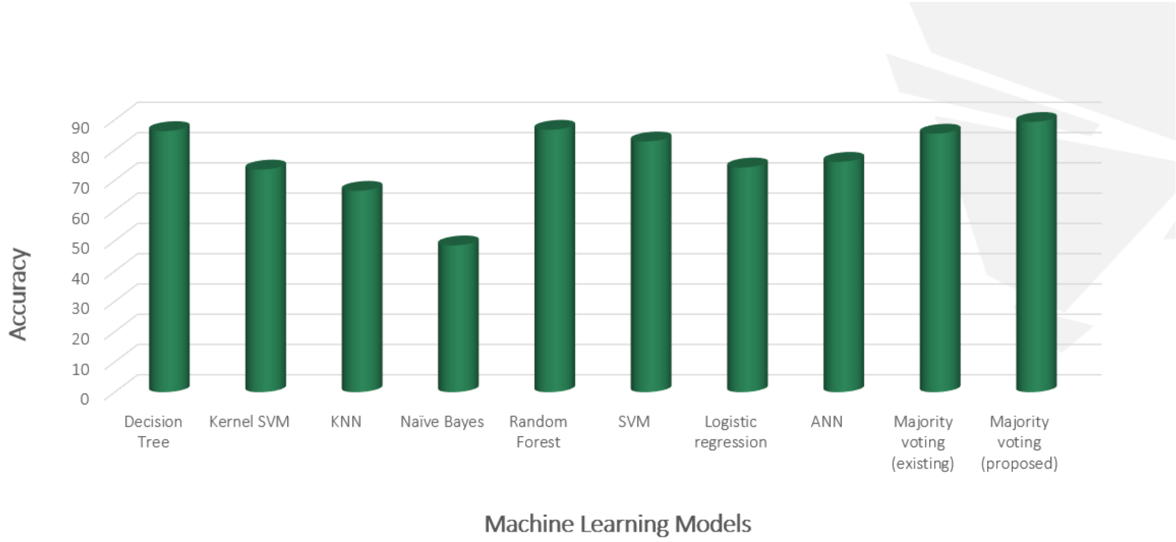

Fig 7.4 – Comparison with Existing Models

Fig 7.4 gives the rundown of the accuracy of the various existing models along with that of the proposed model.

The proposed system was compared with other existing models and the proposed system showed maximum accuracy. The existing majority voting technique with learners such as Random forest, Decision tree and SVM gave an accuracy of 84.54%. Another existing system that used ANN showed an accuracy of 76.1%.

CHAPTER 8

CONCLUSION

The proposed system plays a major role in assisting the farmers to make a decision on the best crop to be grown on their farm land located in Coimbatore. This system considers various soil and environmental factors, and IoT sensor values to predict the accurate crop. The ensemble learning model used provides an advantage by combining the accuracies of KNN, random forest and decision tree algorithms and gives a resulting accuracy of 89.27% which is higher than the accuracy of the existing majority voting techniques and ANN which are 84.54% and 76.1% respectively.

8.1 FUTURE ENHANCEMENTS

As an enhancement of the proposed system, additional IoT sensors for other parameters can be used. Also, crop rotation can be included as a new feature to predict the crop based on the previously grown crop.

BIBLIOGRAPHY

1. Z. Doshi, S. Nadkarni, R. Agrawal and N. Shah, "AgroConsultant: Intelligent Crop Recommendation System Using Machine Learning Algorithms", 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 2018, pp. 1-6. doi-10.1109/ICCUBEA.2018.8697349
2. S. Pudumalar, E. Ramanujam, R. H. Rajashree, C. Kavya, T. Kiruthika and J. Nisha, "Crop recommendation system for precision agriculture", 2016 Eighth International Conference on Advanced Computing (ICoAC), Chennai, 2017, pp. 32-36. doi: 10.1109/ICoAC.2017.7951740.
3. Kevin Tom Thomas, Varsha S, Merin Mary Saji, Lisha Varghese, Er. Jinu Thomas "Crop Prediction Using Machine Learning", International Journal of Future Generation Communication and Networking Vol. 13, No. 3, (2020), pp. 1896–1901
4. Harshitha L, Rashmi M N, Vaishnavi I Dodamani, Yamini S, Nandini M S, "Improving Crop Productivity Through Crop Recommendation", International Research Journal of Engineering and Technology (IRJET) Volume: 07 Issue: 08 | Aug 2020
5. K. R. Akshatha, K. S. Shreedhara, "Implementation of Machine Learning Algorithms for Crop Recommendation Using Precision Agriculture", International Journal of Research in Engineering, Science and Management Volume-1, Issue-6, June 2018
6. Srilakshmi A, Madhumitha K, Geetha K, "Machine learning approach: Recommendation of suitable crop for land using meteorological factors", Proceedings of the 2nd International Conference on IoT, Social, Mobile, Analytics & Cloud in Computational Vision & Bio-Engineering (ISMAC-CVB 2020)
7. Miss Snehal S.Dahikar, Dr.Sandeep V.Rode, Agricultural Crop Yield Prediction Using Artificial Neural Network Approach", INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING, Vol. 2, Issue 1, January 2014
8. Varsha A, Midhuna V M,Rilty, Assistant Prof. Divya R, "Soil Classification and Crop recommendation using IoT and Machine Learning", International Journal of Scientific Research & Engineering Trends Volume 6, Issue 3, May-June-2020, ISSN (Online): 2395-566X

9. Avinash Kumar, Sobhangi Sarkar and Chittaranjan Pradhan, "Recommendation System for Crop Identification and Pest Control Technique in Agriculture", International Conference on Communication and Signal Processing, April 4-6, 2019, India.
10. Tanmay Banavlikar, Aqsa Mahir, Mayuresh Budukh, Soham Dhodapkar, "Recommendation System Using Neural Networks", International Research Journal of Engineering and Technology(IRJET) Volume: 05 Issue: 05
11. Devdatta A. Bondre, Santosh Mahagaonkar, "Prediction of crop yield and fertilizer recommendation using machine learning algorithms", International Journal of Engineering Applied Sciences and Technology, 2019, Vol. 4, Issue 5, ISSN No. 2455-2143
12. Dr.A.K.Mariappan, Ms C. Madhumitha, Ms P. Nishitha, Ms S. Nivedhitha, "Crop Recommendation System through Soil Analysis Using Classification in Machine Learning", International Journal of Advanced Science and Technology Vol.29, No.03,(2020), pp.12738-12747
13. Preethi G, Rathi Priya V, Sanjula S M, Lalitha S D, Vijaya Bindhu B. "Agro based crop and fertilizer recommendation system using machine learning", European Journal of Molecular & Clinical Medicine Volume 7, Issue 4, 2020
14. Soumya Sri Attaluri, Nowshath K Batcha, Raheem Mafas "Crop Plantation Recommendation using Feature Extraction and Machine Learning Techniques", Journal of Applied Technology and Innovation (e -ISSN: 2600-7304) vol. 4, no. 4, (2020)

APPENDICES

DATASET

Area	N	P	K	EC	pH	Soil type	Temperatu	Rainfall	Crop
Anamalai	148.41	12.63	470.6	0.27	6.97	Anamalai Series	24	750	Banana
Periyapothu	159.66	13.96	513.38	0.18	6.93	Anamalai Series	27	705	Banana
A.Mettupalayam	187.52	11.35	672.23	0.45	7.56	Peelamedu Series	24	670	Banana
A.Sengappalli	178.74	12.58	537.35	0.33	7.53	Peelamedu Series	22	678	Banana
Allapalayam	183.08	13.71	568.3	0.77	7.53	Peelamedu Series	27	689	Banana
Ambothi	171.08	12.94	711.08	0.27	7.23	Peelamedu Series	24	689	Banana
Annur	183.59	13.46	519.02	0.4	7.4	Irugur Series	15	687	Banana
Kanjapalli	162.36	12.79	578.86	0.18	7.01	Peelamedu Series	19	724	Banana
Kanuvakkarai	174	13.78	487.16	0.29	7.59	Peelamedu Series	17	726	Banana
Karegoundenplm	169.83	13.66	720.14	0.36	7.25	Peelamedu Series	20	665	Banana
Kariampalayam	169.71	12.1	511.32	0.21	7.56	Peelamedu Series	25	650	Banana
Kunnathur	183.22	13.8	625.83	0.44	7.57	Peelamedu Series	18	668	Banana
Kuppanur	192.83	13.45	507.8	0.44	7.69	Irugur Series	16	675	Banana
Kuppepalayam	196.57	13.69	518.38	0.44	7.15	Peelamedu Series	22	690	Banana
Kuppepalayam	196.57	13.69	518.38	0.44	7.15	Peelamedu Series	35	675	Banana
M.G.C.Palayam	179.46	12.74	586.48	0.25	7.15	Peelamedu Series	32	735	Banana
Naranapuram	182.12	12.49	472.92	0.32	7.28	Peelamedu Series	33	748	Banana
Odderpalayam	174.32	11.45	572.72	0.15	7.29	Peelamedu Series	22	678	Banana
Pachapalayam	182.27	12.97	508.6	0.3	7.06	Peelamedu Series	18	715	Banana
Pasur	189.85	13.36	473.25	0.36	7.13	Peelamedu Series	27	710	Banana
Pogalur	169.82	13.79	468.29	0.33	7.17	Peelamedu Series	21	658	Banana
Vadakkalur	160.01	11.5	467.11	0.17	7.35	Peelamedu Series	25	695	Banana
Vadavalli	192.7	11.46	539.08	0.28	7.11	Peelamedu Series	17	693	Banana
Bellathi	188.5	13.57	564.45	0.41	7.4	Irugur Series	32	745	Banana
Bellapalayam	185.51	13.89	677.46	0.42	7.29	Irugur Series	35	688	Banana
Chikkadasampalaya	188.98	13.02	587.83	0.21	7.34	Irugur Series	17	674	Banana
Chikkarampalayam	181.36	12.96	635.27	0.41	7.4	Irugur Series	21	710	Banana
Chinnakallipatti	175.47	11.78	536.21	0.37	7.29	Irugur Series	18	682	Banana
Illuppananatham	168.61	12.68	549.64	0.17	7.23	Irugur Series	16	695	Banana

SOURCE CODE

Majority Voting Technique

Importing the libraries

In [22]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Importing the dataset

In [23]:
```
dataset = pd.read_csv('finalf.csv')
dataset = dataset.drop('Area',1)
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
```

In [24]:
```
X
```

Out[24]:
```
array([[148.41, 12.63, 470.6, ..., 'Anamalai Series', 24, 750],
       [159.66, 13.96, 513.38, ..., 'Anamalai Series', 27, 705],
       [187.52, 11.35, 672.23, ..., 'Peelamedu Series', 24, 670],
       ...,
       [169.27, 12.9, 546.33, ..., 'Irugur Series', 27, 1555],
       [209.93, 15.88, 607.06, ..., 'Palathurai Series', 26, 1646],
       [186.87, 13.09, 594.4, ..., 'Irugur Series', 29, 998]],
      dtype=object)
```

In [25]:
```
y
```
Out[25]:
```
array(['Banana', 'Banana', 'Banana', ..., 'Coconut', 'Coconut', 'Coconut'],
      dtype=object)
```

Encoding categorical data

Encoding the Independent Variable

In [26]:
```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [5])], remainder='passthrough')
temp=X[:,5]
X = np.array(ct.fit_transform(X))
temp1=X[:,:11]
soiltype={temp[i]:temp1[i] for i in range(len(temp))}
```

In [27]:
```
X[0]
```
Out[27]:
```
array([1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 148.41,
       12.63, 470.6, 0.27, 6.97, 24, 750], dtype=object)
```

Encoding the Dependent Variable

In [28]:
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y = le.fit_transform(y)
```

In [29]:
```
y
```
Out[29]:
```
array([0, 0, 0, ..., 5, 5, 5])
```

Splitting the dataset into the Training set and Test set

In [30]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 1)
```

In [31]:
```
print(X_train)
```
```
[[0.0 0.0 1.0 ... 7.12 27 1100]
 [0.0 0.0 1.0 ... 7.41 35 513]
 [0.0 0.0 1.0 ... 7.73 27 608]
 ...
 [0.0 0.0 0.0 ... 7.35 28 1285]
 [0.0 0.0 0.0 ... 7.76 32 2828]
 [0.0 1.0 0.0 ... 7.37 26 1682]]
```

In [32]:
```
X_test
```
Out[32]:
```
array([[0.0, 0.0, 0.0, ..., 7.43, 32, 462],
       [0.0, 0.0, 0.0, ..., 7.69, 21, 654],
       [0.0, 0.0, 1.0, ..., 7.53, 32, 401],
       ...,
       [0.0, 0.0, 1.0, ..., 7.15, 28, 632],
       [1.0, 0.0, 0.0, ..., 7.34, 29, 670],
       [0.0, 0.0, 1.0, ..., 7.31, 27, 539]], dtype=object)
```

In [33]:
```
y_train
```
Out[33]:
```
array([ 5, 15,  9, 18,  7, 17,  5, 11, 14,  0,  5,  2, 14, 15,  9, 15,  3,
       11, 18, 13, 10, 11,  3, 12, 15,  7,  8, 15,  8,  5, 14,  0, 16, 15,
       15,  5, 12, 15, 16, 15, 10,  8,  9,  3, 11,  7,  5,  7, 15,  9,  5,
       17, 15, 17,  5,  0,  0, 15, 18,  5, 11, 10,  5,  0, 15, 15,  7,  0,
       19, 15, 18, 18, 15, 13,  5,  0, 15,  7,  0,  5, 14,  5, 18, 15, 15,
        7,  5,  0,  1,  5,  9,  5,  0,  5,  5,  6, 15,  0, 10,  5, 14, 15,
        0,  7,  5,  0,  8,  7,  0,  5,  7,  1,  5,  7, 15, 15,  4, 15, 10,
        2, 15,  3, 11,  7, 11, 13, 18,  3, 19, 15, 10,  7, 15, 15, 16, 15,
```

Feature Scaling

```
In [35]:
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train[:, 11:] = sc.fit_transform(X_train[:, 11:])
X_test[:, 11:] = sc.transform(X_test[:, 11:])
```

```
In [36]:
X_train
```

```
Out[36]: array([[0.0, 0.0, 1.0, ..., -1.277762403572537, -0.2114336004211533,
         0.3461954464933725],
        [0.0, 0.0, 1.0, ..., -0.44051663757759596, 1.723756693088716,
         -0.616769625851324],
        [0.0, 0.0, 1.0, ..., 0.48334075938233967, -0.2114336004211533,
         -0.4609234897819097],
        ...,
        [0.0, 0.0, 0.0, ..., -0.6137398995075851, 0.030465186267580352,
         0.6496852904180214],
        [0.0, 0.0, 0.0, ..., 0.5699523903473317, 0.9980603330225151,
         3.180954637313876],
        [0.0, 1.0, 0.0, ..., -0.5559988121975878, -0.453332387109887,
         1.3009580906238893]], dtype=object)
```

```
In [37]:
X_test
```

```
Out[37]: array([[0.0, 0.0, 0.0, ..., -0.38277555026760124, 0.9980603330225151,
         -0.7004343936359568],
        [0.0, 0.0, 0.0, ..., 0.36785858476234773, -1.6628263205535554,
         -0.3854611502114565],
        [0.0, 0.0, 1.0, ..., -0.09407011371762009, 0.9980603330225151,
         -0.8005040178489492],
        ...,
        [0.0, 0.0, 1.0, ..., -1.1911507726075423, 0.030465186267580352,
         -0.42155183435384713],
        [1.0, 0.0, 0.0, ..., -0.6426104431625825, 0.27236397295631404,
```

Machine Learning Model

```
In [38]:
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import VotingClassifier
from sklearn.neighbors import KNeighborsClassifier
classifier1 = DecisionTreeClassifier(criterion = 'entropy', random_state = 0)
classifier2 = RandomForestClassifier(n_estimators = 18, criterion = 'entropy', random_state = 0)
classifier3 = KNeighborsClassifier(n_neighbors = 26, metric = 'minkowski', p = 2)
classifier = VotingClassifier(estimators=[('dt',classifier1),('rf',classifier2),('knn',classifier3)],voting='hard',weights=[
classifier.fit(X_train, y_train)
```

```
Out[38]: VotingClassifier(estimators=[('dt',
                              DecisionTreeClassifier(criterion='entropy',
                                                     random_state=0)),
                             ('rf',
                              RandomForestClassifier(criterion='entropy',
                                                     n_estimators=18,
                                                     random_state=0)),
                             ('knn', KNeighborsClassifier(n_neighbors=26))],
                 weights=[2, 3, 1])
```

```
In [39]:
y_pred = classifier.predict(X_test)
print(np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_test),1)),1))
```

```
[[15 15]
 [12 12]
 [15 15]
 [15 15]
 [ 7  7]
 [ 5  5]
 [15 10]
 [ 0  0]
 [13 13]
 [15 15]
```

```
[14 14]
[11 11]
[15 15]
[ 5  5]
```

In [40]:

```
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
print(cm)
accuracy_score(y_test, y_pred)
```

```
[[23  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  5  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  3  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  1  0  3  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0 33  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 1  0  0  0  0  0  5  0  0  0  0  0  0  0  0  2  0  0  0  0]
 [ 0  0  0  0  0  0  0 13  0  0  0  0  0  0  0  3  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  4  0  0  0  0  0  0  1  0  0  0  0]
 [ 1  0  0  0  0  0  2  0  0  7  0  1  0  0  1  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  3  0  0 11  1  0  0  0  3  0  0  0  0]
 [ 0  0  0  0  0  0  0  1  0  0  0  8  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  1  0  0  0  0  9  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  4  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  5  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 45  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  5  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  7  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 14  0]
 [ 0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  3]]
```

Out[40]: 0.8927038626609443

Input from sensor

In [46]:

```
import serial

def main_func():
    list_values=[]
    list_in_floats=[]
    arduino = serial.Serial('com4', 9600)
    arduino_data = arduino.readline()
    decoded_values = str(arduino_data[0:len(arduino_data)].decode("utf-8"))
    list_values = decoded_values.split(' ')
    for item in list_values:
        list_in_floats.append(float(item))
    arduino_data = 0
    arduino.close()
    return list_in_floats

i=0
sum_ph=0
sum_temp=0
while i<5:
    sensor_value=main_func()
    print(f'Temperature readings from Arduino:',sensor_value[0])
    sum_temp+=sensor_value[0]
    print(f'pH readings from Arduino:',sensor_value[1])
    sum_ph+=sensor_value[1]
    i+=1
avg_temp=int(sum_temp/5)
print(f'Average Temperature:',avg_temp)
avg_ph=round(sum_ph/5,2)
print(f'Average pH:',avg_ph)
```

```
Temperature readings from Arduino: 26.0
pH readings from Arduino: 7.25
Temperature readings from Arduino: 26.0
pH readings from Arduino: 7.25
Temperature readings from Arduino: 26.0
```

Final_Sensordata | Arduino 1.8.12

File Edit Sketch Tools Help

Final_Sensordata

```
#include <dht.h>
#define dht_apin A1 // Analog Pin sensor is connected to

dht DHT;

float calibration_value = 20.44;
int phval = 0;
unsigned long int avgval;
int buffer_arr[10],temp;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  delay(1000);//Wait before accessing Sensor
}

void loop() {
  DHT.read11(dht_apin);
  Serial.print(DHT.temperature);
  Serial.print(" ");
 for(int i=0;i<10;i++)
 {
 buffer_arr[i]=analogRead(A0);
 delay(30);
 }
 for(int i=0;i<9;i++)
 {
 for(int j=i+1;j<10;j++)
 {
 if(buffer_arr[i]>buffer_arr[j])

 {
 temp=buffer_arr[i];
 buffer_arr[i]=buffer_arr[j];
 buffer_arr[j]=temp;
 }
 }
 }
 avgval=0;
 for(int i=2;i<8;i++)
 avgval+=buffer_arr[i];
 float volt=(float)avgval*5.0/1024/6;
 float ph_act = -5.70 * volt + calibration_value;
 Serial.println(ph_act);
  delay(5000);
}
```

User Input Prediction

In [48]:

```
l=[]
l1=[]
area=input("Area: ")
soilt=input("Soil type: ")
vals=soiltype[soilt].tolist()
l.extend(vals)
l1.extend(vals)
nval=float(input("N value: "))
l.append(nval)
pval=float(input("P value: "))
l.append(pval)
kval=float(input("K value: "))
l.append(kval)
ec=float(input("EC value: "))
l.append(ec)
print("pH:",avg_ph)
print("Temperature:",avg_temp)
l.append(avg_ph)
l.append(avg_temp)
rainfall=int(input("Rainfall: "))
l.append(rainfall)
temp1=sc.transform([l[11:]]).tolist()
l1.extend(temp1[0])
predicted_value=classifier.predict([l1])
print("Predicted Crop:",le.inverse_transform(predicted_value)[0])
```

```
Area: Pogalur
Soil type: Irugur Series
N value: 218.41
P value: 12.63
K value: 470.6
EC value: 0.27
pH: 7.29
Temperature: 26
Rainfall: 1500
Predicted Crop: Coconut
```