University of Mumbai

Smart Agriculture System Using IOT & ML

Submitted at the end of semester VI in partial fulfillment of requirements

For the degree of

Bachelors in Technology

by

Harsh Panchal
Roll No: 1813033

Kaushal Patil
Roll No: 1813038

Shubh Dedhia
Roll No: 1813040

Rohit Shah
Roll No: 1813042

Guide

Dr. Anudeepa Kholapure

Department of Electronics and Telecommunication Engineering
K. J. Somaiya College of Engineering, Mumbai-77
(Autonomous College Affiliated to University of Mumbai)

Batch 2018 -2022

K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Certificate

This is to certify that the dissertation report entitled **Smart Agriculture System Using IOT & ML** submitted by Harsh Panchal, Kaushal Patil, Shubh Dedhia & Rohit Shah at the end of semester VI of TY B. Tech is a bonafide record for partial fulfillment of requirements for the degree of Bachelors in Technology in Electronics and Telecommunication Engineering of University of Mumbai

Guide/Examiner1 | Expert/Examiner2

Date:

Place: Mumbai-77

K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Certificate of Approval of Examiners

We certify that this dissertation report entitled **Smart Agriculture System Using IOT & ML** is bonafide record of project work done by Harsh Panchal, Kaushal Patil, Shubh Dedhia and Rohit Shah during semester VI. This project work is submitted at the end of semester VI in partial fulfillment of requirements for the degree of Bachelors in Technology in Electronics and Telecommunication Engineering of University of Mumbai.

____

Guide/Examiner1

____

Expert/Examiner2

Date:

Place: Mumbai-77

K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

DECLARATION

We declare that this written report submission represents the work done based on our and / or others' ideas with adequately cited and referenced the original source. We also declare that we have adhered to all principles of intellectual property, academic honesty and integrity as we have not misinterpreted or fabricated or falsified any idea/data/fact/source/original work/ matter in my submission.

We understand that any violation of the above will be cause for disciplinary action by the college and may evoke the penal action from the sources which have not been properly cited or from whom proper permission is not sought.

Signature of the Student **1813033** **Roll No.**	**Signature of the Student** **1813038** **Roll No.**
Signature of the Student **1813040** **Roll No.**	**Signature of the Student** **1813042** **Roll No.**

Date:

Place: Mumbai-77

Abstract

In the current age of high competition and risk in markets, technological advancements are a must for better growth and sustainability and same apply for agriculture industry. The purpose of this project is to improve the efficiency of the agriculture sector. In India, agriculture plays a vital role for development in food production and in global economy. Internet of Things (IoT) is a milestone in evolution of technology. IoT helps us in many fields among which agriculture is one of the primary ones.

With the help of IoT along with Machine Learning in the field of agriculture, we can increase the efficiency of crop production. Different weather parameters are taken into consideration with which the best suitable crops to be grown are predicted with the help of supervised learning like Decision Tree Regression. With the continuing expansion of the human population understanding worldwide crop yield is central to addressing food security challenges and reducing the impacts of climate change. The Agricultural yield primarily depends on weather conditions (rain, temperature, etc), pesticides. Accurate information about history of crop yield is important for making decisions related to agricultural risk management and future predictions. This project looks into developing an **Smart Agriculture System using IOT & ML.**

The aim of this project is to develop a microcontroller (NodeMCU) based IoT system which can automatically turn On or turn Off the motor pump based on the soil moisture level readings from the sensor. The data from the setup will be sent to the API (Think Speak Dashboard), where the user can check the soil moisture, humidity and temperature. Using machine learning methods we will be able to predict the crop yield using publicly available data from FAO and World Data Bank.

***Key words*:** Agriculture, Decision Tree Regressor (DTR), Farmers, Internet of Things, Machine Learning, NodeMCU, Think Speak.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter presents the introduction of the Smart Agriculture System using IOT & ML. It contains the background of the technology utilized in the project topic, the motivation behind its ideation, also the scope and overview of the developed project.

1.1 Introduction

The Internet of Things (IoT) focuses on automating processes by reducing human-human interaction. In the automation process, the IoT uses electronic sensors to collect data, the controller processes the data, and the actuators complete the automation process. IoT is regarded as very complex platform facilitating the connection of things based on objects being tagged for their identification, but also includes sensors, actuating elements, and other technologies. As per the recent predictions by Gartner by 2025, 40 billion IoT objects will be connected and installed everywhere. Generally, IoT refers to the networked interconnection of everyday objects, which are often equipped with ubiquitous intelligence. IoT will increase the ubiquity of Internet by integrating every object for interaction via embedded systems, which will lead to the development of highly distributed network of devices communicating and coordinating with human beings as well as other devices. The focus of the IoT in agriculture is to automate all aspects of agriculture and agricultural methods to make the process quite efficient.

Many researchers have focused on intelligent systems that monitor and control agricultural parameters by increasing productivity and efficiency. Intelligent systems collect data for measurements and get accurate results which can take the appropriate action. Current uses of smart farming systems include collecting data on environmental parameters such as soil moisture level, humidity, temperature, and pH levels. ML is a new way of computing intelligence using machines. Its observed that there are various researches have been done for smart agriculture in IoT and ML individually. Machine learning is an important decision support tool for crop yield prediction, including supporting decisions on what crops to grow and what to do during the growing season of the crops. Several machine learning algorithms have been applied to support crop yield prediction research.

1.2 Motivation

Farmers today face a huge challenge — feeding a growing global population with less available land. The world's population is expected to grow to nearly 10 billion by 2050, increasing the global food demand by 50%. As this demand for food grows, land, water, and other resources will come under even more pressure. Agriculture plays a vital role in the Indian economy. Over70% of the rural households depend on agriculture. Agriculture is an important sector of Indian economy as it contributes about 17% to the total GDP and provides employment to over 60% of the population. Indian agriculture has registered impressive growth over last few decades.

However, farmers' suicides in India is worrying topic. The expressed reasons in order of importance behind farmer suicides were – debt, environment, low produce prices, poor irrigation, increased cost of cultivation, use of chemical fertilizers and crop failure. The need of the hour is to design a system that could provide predictive insights to the Indian farmers, thereby helping them make an informed decision about which crop to grow. This calls for the need of smart farming, which requires use of IoT. Application of IoT in agriculture could be a life changing event for humanity and the whole planet. Sensor data analytics drives transparency into agricultural processes, as farmers get precious insights on the performance of their fields, greenhouses, etc. Farming powered by Machine Learning with its high-precision algorithms is a new concept emerging today. Aiming to increase the quantity and quality of products, this cutting-edge movement makes sustainable productivity growth for everyone working in the agriculture realm. With this in mind, we propose system for Smart Management for Crop Cultivation using IoT and Machine Learning – a smart system that can assist farmers in crop management by considering sensed parameters (Soil Moisture, Temperature, Humidity) and other parameters (Different Countries, location, rainfall, Pesticides in tonnes) that predicts the most suitable crop to grow in that environment.

1.3 Scope of the Project

IoT based data-driven farm management techniques can help increase agricultural yields by planning input costs, reducing losses, and using resources more efficiently. The IoT generates big amount data with different characteristics based on location and time. To improve productivity of agriculture through intelligent farm management, the data analyzing must be well analyzed and processed.

In our Project Smart Agriculture System improves the whole Agricultural process by monitoring the field in real-time. It keeps various factors like humidity, temperature, soil etc. under check and gives a crystal- clear real-time observation. Use of appropriate Machine learning algorithms on the sensed data can help to improve the productivity and quality of recommendation to crops in the agriculture sector. High-performance computing capability in ML opens up new opportunities in Agriculture sector

1.4 Overview of the Project

Smart Agriculture System using IOT & ML is a smart irrigation system to water the plants using ESP8266 (NodeMCU). In this project, the soil moisture sensor will check the moisture levels in the soil. If the soil is dry then the pump will be turned on automatically to supply water. If the soil contains water or moisture then pump will be turned off automatically to stop supply of water. It sends the data to Think Speak Dashboard through which the user can see the moisture levels in the soil, temperature, humidity level. With ML we will be doing crop yield prediction using Decision Tree Regression with help of FAO Dataset.

1.5 Organization of the Report

The report is structured in 5 chapters where, chapter 1 consists of existing Agriculture sector problems and we believe creating a model for crop yield prediction is the issue we should focus on. Chapter 2 i.e., literature review consists of a brief summary of study of different smart agriculture systems using IoT & ML, applications and their use cases. Chapter 3 consisting of Problem Statement, Circuit Diagram, Components used, Flowchart, Algorithm & DTR Model Architecture. Chapter 4 comprises of the implementation of IoT Model. Chapter 5 Comprises of implementation of Crop Yield Prediction (ML Model) using Decision Tree Regression. Chapter 6 consists of Conclusions and future scope that can be add on to the project.

Chapter 2

Literature Survey

This chapter presents a brief about the literature and overview of major writings and other sources on the selected topic.

In "Automated Irrigation System-IoT Based Approach" [1] the aim of the investigation presented in this research paper is to decrease the loss of water, labour and improve the productivity. Moisture sensor is used to sense the content of moisture in soil and sends moisture sensor information to Arduino. Moisture sensor is used to detect moisture in the soil. It works on the principle of open and short circuit. When the soil is dry the circuit behaves like an open circuit and close if the soil is wet. Wi-Fi module is used for communication to transmit data from sensor layer to the cloud. Data collected from moisture sensor is fed into Arduino and Arduino upload this information or values on cloud by using Wi-Fi. Threshold value is set according to the crop's need. Moisture level checked with respect to predefined threshold value. The threshold value is different for different crops. If the moisture value is less than the reference value pump is ON otherwise remains OFF. This helps in reduction of water usage.

In "Smart Irrigation System using Iot Approach" [2] the system is used to turn the valves ON or OFF automatically as per the water requirement of the plants. The system is used for sensing, monitoring, controlling and for communication purpose. Different sensors are used to detect the different parameters of the soil like moisture, temperature, humidity, pH of soil and nitrogen content of the soil. Depending upon the sensors output the ARM9 processor will take the necessary action. The moisture sensor output will help to determine whether to irrigate the land or not depending upon the moisture content. Along with moisture sensor the temperature sensor output can also be taken into consideration while irrigating the land. If the moisture content of soil is very low and the temperature is very high then there is need of irrigation for plants, but the time for which irrigation will be provided is different for different temperature range. Because if the temperature is very high then the evaporation rate is also very high and hence we have to provide water for more time in order to attain the proper moisture level in the soil. Hence for different temperature range and moisture content level in the soil the land will be irrigated for different time interval.

In “IoT Based Low Cost Smart Irrigation System” [3] the system which aimed at precise water supplement to crops. The system manages waters supplement by monitoring and analysis of ambient parameters that includes humidity and temperature, moisture and ultrasonic sensor. Ultrasonic sensor is used to monitor the level of water in the reservoir. The proposed system consists of PX28015 ultrasonic sensors, soil moisture sensor, DHT22/AM2302 Humidity and Temperature Sensor and ESP8266 microcontroller. Sensors connected to microcontroller are used to sense the amount of water as a combined effort. The system also determines the quantity of water required per day. A threshold value known as reference value is set. Motor pump will be ON/OFF each time when the amount of water is insufficient with respect to the reference value. A notification is sent to server and mobile via MQTT protocol.

In “Design and Implementation of Automatic Plant Irrigation System” [4] proposed to design an irrigation system which is automated by using controllable parameter such as temperature, soil moisture and air humidity. According to paper the farmer automatically can monitor irrigation system for the crops along with controlling humidity and temperature in his field. The designed system is Power-efficient, cost- effective and user friendly that is efficient enough to monitor the crop condition and remotely controlthe peripherals of the irrigation system which will make the job of the user easier. This system is scalable, as it allows any number of devices to be added with no major changes in its core architecture.

In “Smart Management of Crop Cultivation using IOT and Machine Learning” [5] Machine Learning Algorithm (KNN) calculates the parameter to suggest the crop which is best to grow in the particular field based on the values received at real time. A standardized dataset containing the minimum requirements for a particular crop is maintained and is used for the prediction of the crop. The sensors are added to the field for which the readings are needed to be calculated. The DHT11, MQ2, Soil Moisture Sensor, Light Intensity Sensor sends the readings in real time to the cloud server.

In “IoT based Agro Automation System using Machine Learning Algorithms” [6] evaluates the crop quality factor based on pre-established weather conditions and nature of soil using the trained set of data and implementing Supervised and Reinforcement models of machine learning. If any unfavorable conditions are perceived ahead of time, the alternative and precautious measures are adopted so as to ensure the wellbeing of the planted crops and agricultural land. Specific measures are also taken to predict the right period of sowing, reaping and harvesting for the overall enhancement of the production which can also be foreseen as a part of the modern agricultural revolution.

In “Crop Prediction System using Machine Learning Algorithms” [7] the system takes the necessary weather and soil properties data for a given coordinate automatically from an appropriate source. This provides users with the capability to perform strategy changes, like choosing a more robust genetic variation before planting or even changing the crop type, in order to accommodate for extreme climatic variations further ahead in the crop cycle. The algorithm developed introduces a data-driven model to predict and forecast crop yield using joint dependencies of soil and climate features. Use of naïve Bayes and decision tree makes the model very efficient in terms of computation. The system is scalable as it can be used to test on different crops. Also, the optimal and worst environmental condition can also be incurred. The model focuses on all type of farms, and smaller farmers can also be benefitted. This model can be further enhanced to find the yield of every crop, and for pesticide recommendation. Also, it can be modified to suggest about the fertilizers and irrigation need for crops.

- ## Objectives of the Project

This project includes an automated irrigation system for efficient water management after the continuous monitoring of the soil conditions during the entire growth of the crops. These include soil parameters like temperature and humidity, soil moisture content etc. Also by using ML system aims at predicting or forecasting the crop yield by learning the past data of the farming land. By considering various factors such as soil conditions, rainfall, temperature, yield and other entities the system builds a predicting model using machine learning techniques. Here we make use of different machine learning techniques such as Gradient Boosting Regressor Random Forest Regressor, SVM, Decision Tree Regressor. Performance is evaluated based on predicted accuracy.

Chapter 3

Project Design

This chapter presents the details about the design of the project. A basic introduction is given which is followed by the problem statement. It also provides the Circuit Diagram, Flowchart & Algorithm of model along with DTR Model Architecture.

3.1 Introduction

The project developed for automatically watering the crops makes use of a microcontroller ESP8266 (NodeMCU) and various other sensors and parts for moisture measurement and watering of the plants. A Soil Moisture Sensor is used for measuring the soil moisture and a DHT11 Sensor is used for measuring the atmospheric temperature and humidity. A Pump is used for pumping out water from the tank, into the other pot. Based on the soil moisture values, the pump can be turned On/Off through a relay connection.. All the data/values are communicated to the user using an API. The API used in the project is Think Speak Dashboard.

3.2 Problem Statement

People spend a lot of time in watering plants or farms by checking soil status. Also, they have to ensure that excess water should not be supplied. To resolve this issue Smart Agriculture System using IOT & ML has been developed which will work to minimize the number of workers in a crop field, control and save water. All these features make this research sustainable option to be considered to improve the agriculture and irrigation efficiency. With the help of Machine Learning along with IoT in the field of agriculture, we can increase the efficiency of crop production. Different parameters are taken into consideration with which the best suitable crop to be grown are predicted with the help of supervised learning like Decision Tree Regression.

3.3 Block Diagram/ Circuit diagram

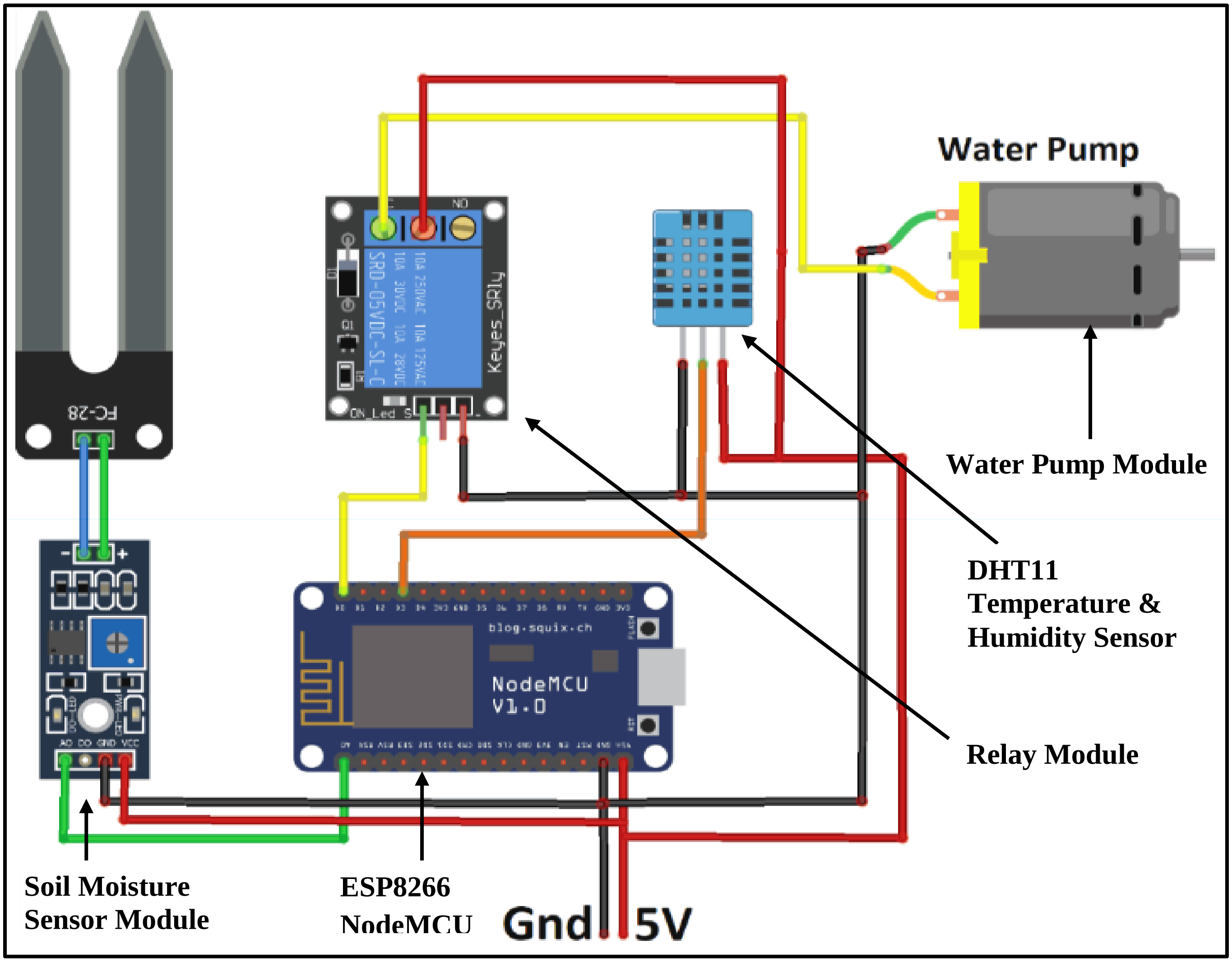

Figure no.3.3 Circuit Diagram

3.4 Components Used

1) ESP8266 Node MCU:

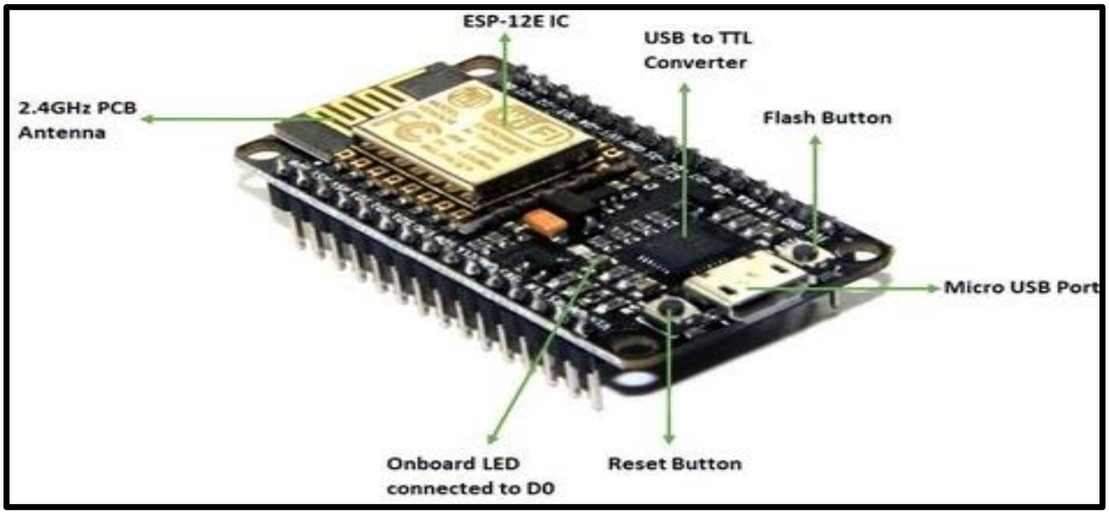

Figure no. 3.4.1 ESP8266 NodeMCU

- **Purpose:**

  ESP8266 NodeMCU will be used to collect the data from the sensors. It will be useful to exchange the data and wifi module will act as web server.

Item Name	ESP8266 NodeMCU CP2102 Development Board
Analog Input Pins	1
RAM Size	128 KB
Digital I/O Pins	16
Input Voltage	7-12 Volts
Operating Voltage	3.3 Volts

Table no. 3.4.1 ESP8266 NodeMCU Specifications

2) Soil Moisture Sensor Module:

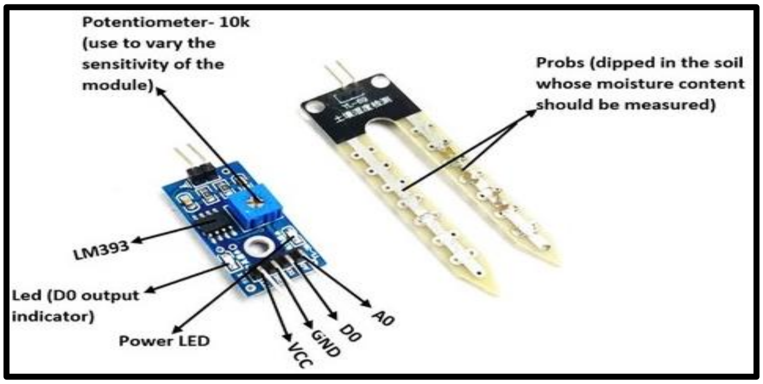

Figure no. 3.4.2 Soil Moisture Sensor

- **Purpose:**

  The soil moisture sensor is commonly used in smart agriculture or other garden automation projects to measure the moisture content present in the soil.

Operating Voltage	3.3V ~ 5V
Dimensions	60 x 30 mm
Weight	11 grams

Table no. 3.4.2 Soil Moisture Sensor Specifications

3) DHT11 Temperature & Humidity Sensor:

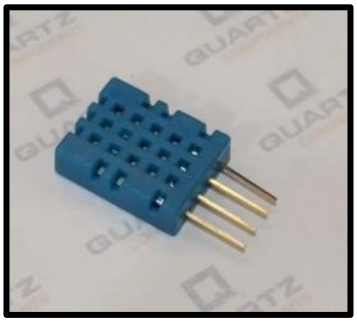

Figure no. 3.4.3 DHT11 Temperature & Humidity Sensor

- **Purpose:**

DHT11 Digital Temperature And Humidity Sensor is a basic and low-cost digital temperature and humidity sensor.

Operating Voltage	3.3V ~ 5V
Dimensions	23 x 12 x 5 mm
Weight	11 grams
Temperature Range	0 – 50^0 C
Humidity Range	20 – 90 % Relative Humidity

Table no. 3.4.3 DHT11 Temperature & Humidity Sensor Specifications

4) Relay Module:

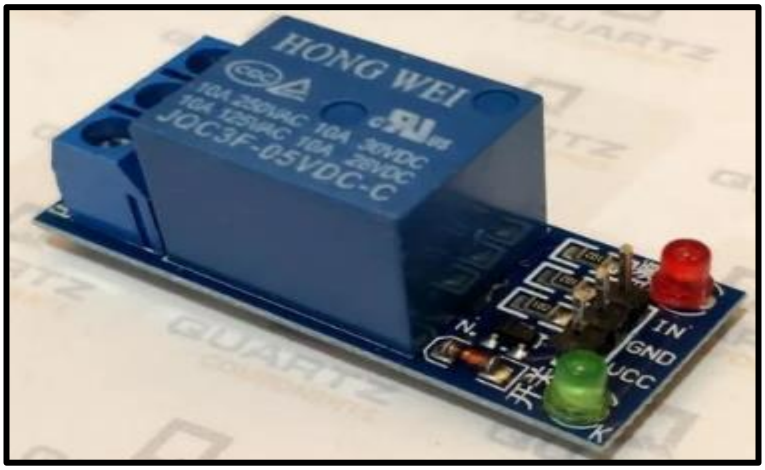

Figure no. 3.4.4 Relay Module

- ## Purpose:

This 5V Relay module helps to switch (control) AC/DC loads from a microcontroller like ESP8266 Node MCU, Arduino etc. You can turn on or turn off loads that consume upto 10A using Relay module.

Voltage	5V
Current Consumption	20 mA
Load Current	10 mA
AC Load Voltage	250 V
DC Load Voltage	30 V

Table no. 3.4.4 Relay Module Specifications

5) **Water Pump Module:**

Figure no. 3.4.5 Water Pump Module

- ## Purpose:

This DC operated mini submersible water pump is ideal for small vending machines and other applications, in which a small amount of water has to be pumped.

Operating Voltage	3v to 12v

Table no. 3.4.5 Water Pump Specifications

3.5 Flow Chart

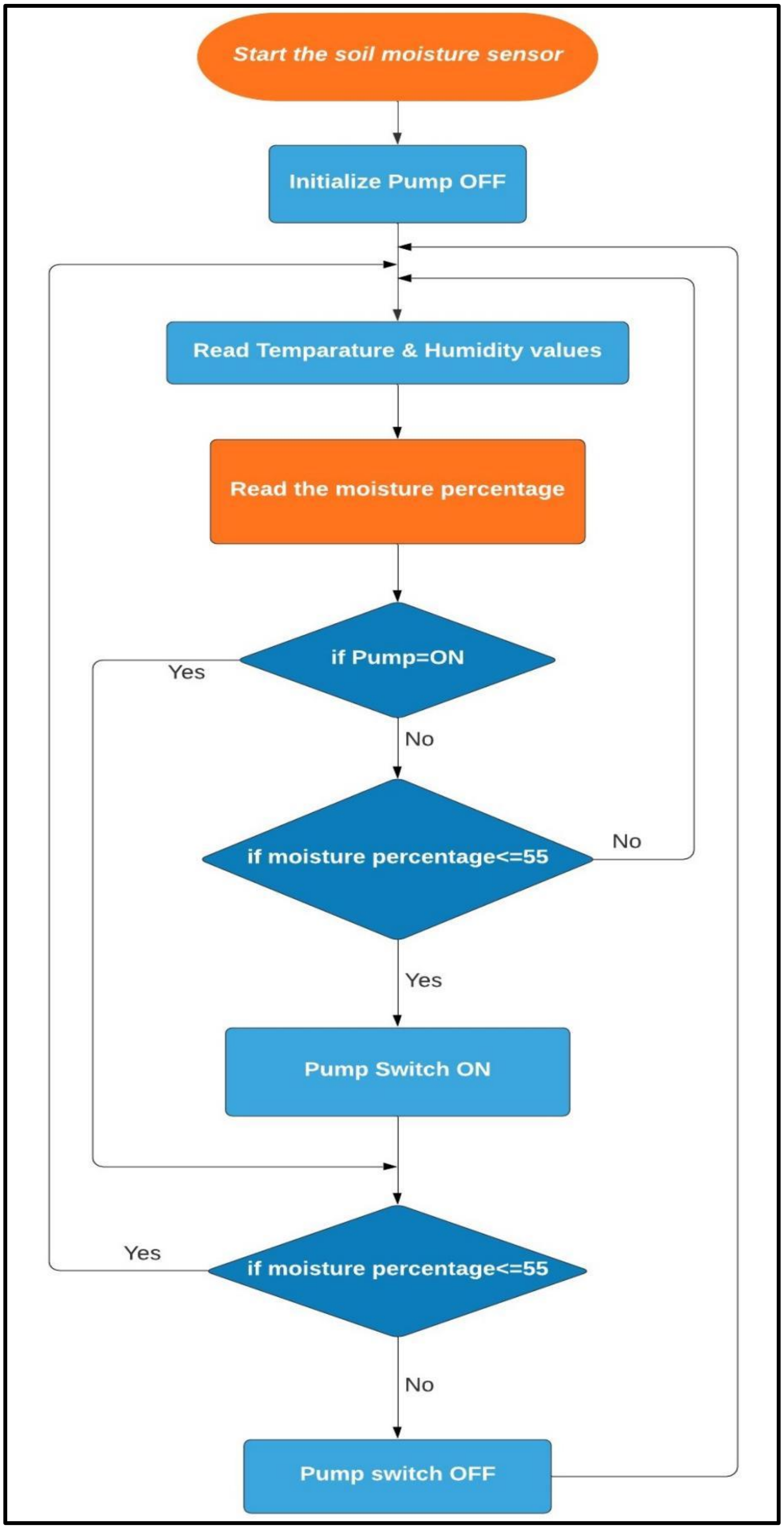

Figure no. 3.5 Flowchart

3.6 Algorithm

Step 1:- Starting the soil moisture sensor.

Step 2:- Initializing the pump OFF

Step 3:- Reading the temperature and humidity of the atmosphere with the help of sensor (DHT11) into the sample.

Step 4:- Inserting the soil moisture sensor into the sample.

Step 5:- Once the sensor is inserted, Fetching the moisture percentage value from the sensor and sending the value to ESP8266 NodeMCU.

Step 6:- If moisture percentage is less than 56% than the pump is switched on. (55% is selected because at this value soil deterioration happens at lower rate).

Step 7:- Every 10 secs Step 5 and 6 are repeated to ensure that the moisture percentage does not exceeds 56%.

Step 8:- Once the moisture percentage crosses the threshold of 56 then the pump is switched off.

3.7 DTR Model Architecture

3.7.1 Overview of Decision Tree Algorithm

Decision Tree is one of the most commonly used, practical approaches for supervised learning. It can be used to solve both Regression and Classification tasks with the latter being put more into practical application.

It is a tree-structured classifier with three types of nodes. The Root Node is the initial node which represents the entire sample and may get split further into further nodes. The Interior Nodes represent the features of a data set and the branches represent the decision rules. Finally, the Leaf Nodes represent the outcome. This algorithm is very useful for solving decision-related problems

With a particular data point, it is run completely through the entirely tree by answering True/False questions till it reaches the leaf node. The final prediction is the average of the value of the dependent variable in that particular leaf node. Through multiple iterations, the Tree is able to predict a proper value for the data point.

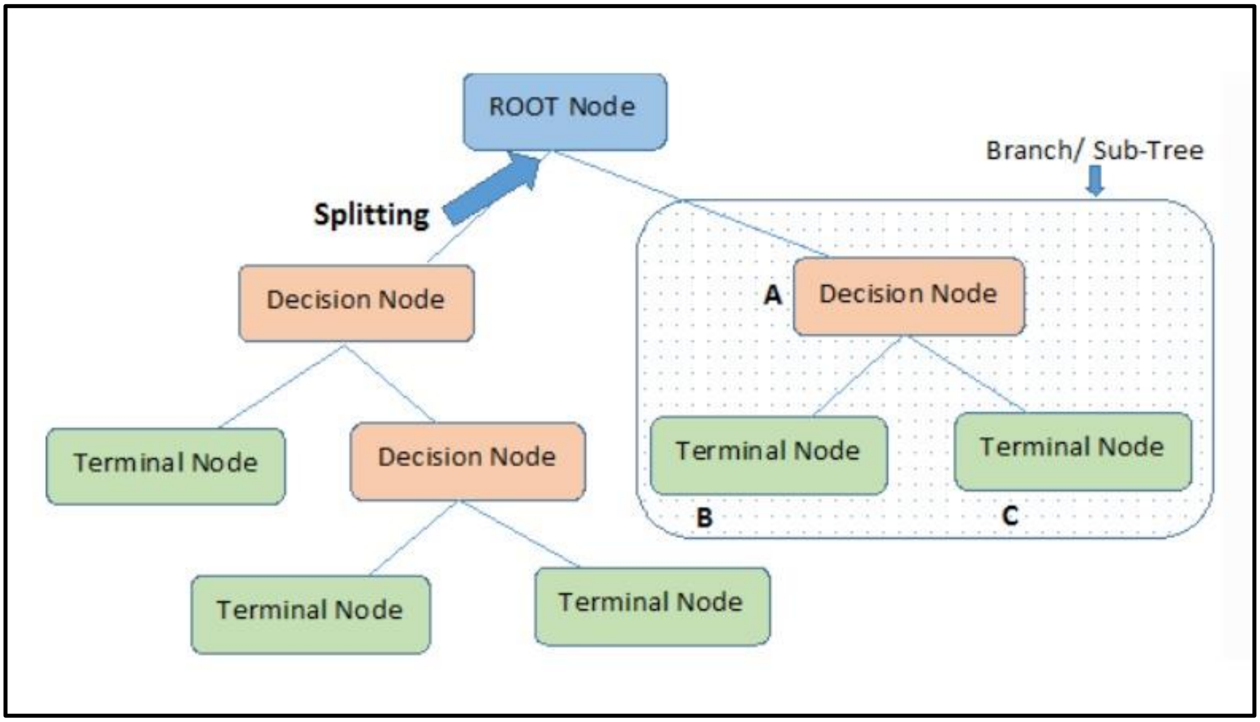

Figure no. 3.7.1 (DTR) Flowchart

Important Terminology:

- **Root Node**: It represents entire population or sample and this further gets divided into two or more homogeneous sets.
- **Splitting**: It is a process of dividing a node into two or more sub-nodes.
- **Decision Node**: When a sub-node splits into further sub-nodes, then it is called decision node.
- **Leaf/Terminal Node**: Nodes do not split is called Leaf or Terminal node.
- **Pruning**: When we remove sub-nodes of a decision node, this process is called pruning. You can say opposite process of splitting.
- **Branch / Sub-Tree:** A sub section of entire tree is called branch or sub-tree.
- **Parent and Child Node**: A node, which is divided into sub-nodes is called parent node of sub-nodes whereas sub-nodes are the child of parent node.

3.7.2 Model Working:

1) The decision of making strategic splits heavily affects a tree's accuracy. The decision criteria is different for classification and regression trees. Decision trees regression normally use mean squared error (MSE) to decide to split a node in two or more sub-nodes.
2) Suppose we are doing a binary tree the algorithm first will pick a value, and split the data into two subset. For each subset, it will calculate the MSE separately. The tree chooses the value with results in smallest MSE value.
3) We need to pick a variable and the value to split on such that the two groups are as different from each other as possible. For each variable, for each possible value of the possible value of that variable see whether it is better.
4) How to determine if it is better? Take weighted average of two new nodes (mse*num_samples) To sum up, we now have: A single number that represents how good a split is which is the weighted average of the mean squared errors of the two groups that create. A way to find the best split which is to try every variable and to try every possible value of that variable and see which variable and which value gives us a split with the best score.

Chapter 4

Implementation and Experimentation (IOT)

This chapter presents the details about the project implementation and working of the model which is developed along with the circuit simulation and outputs of the model.

4.1 Circuit Simulation

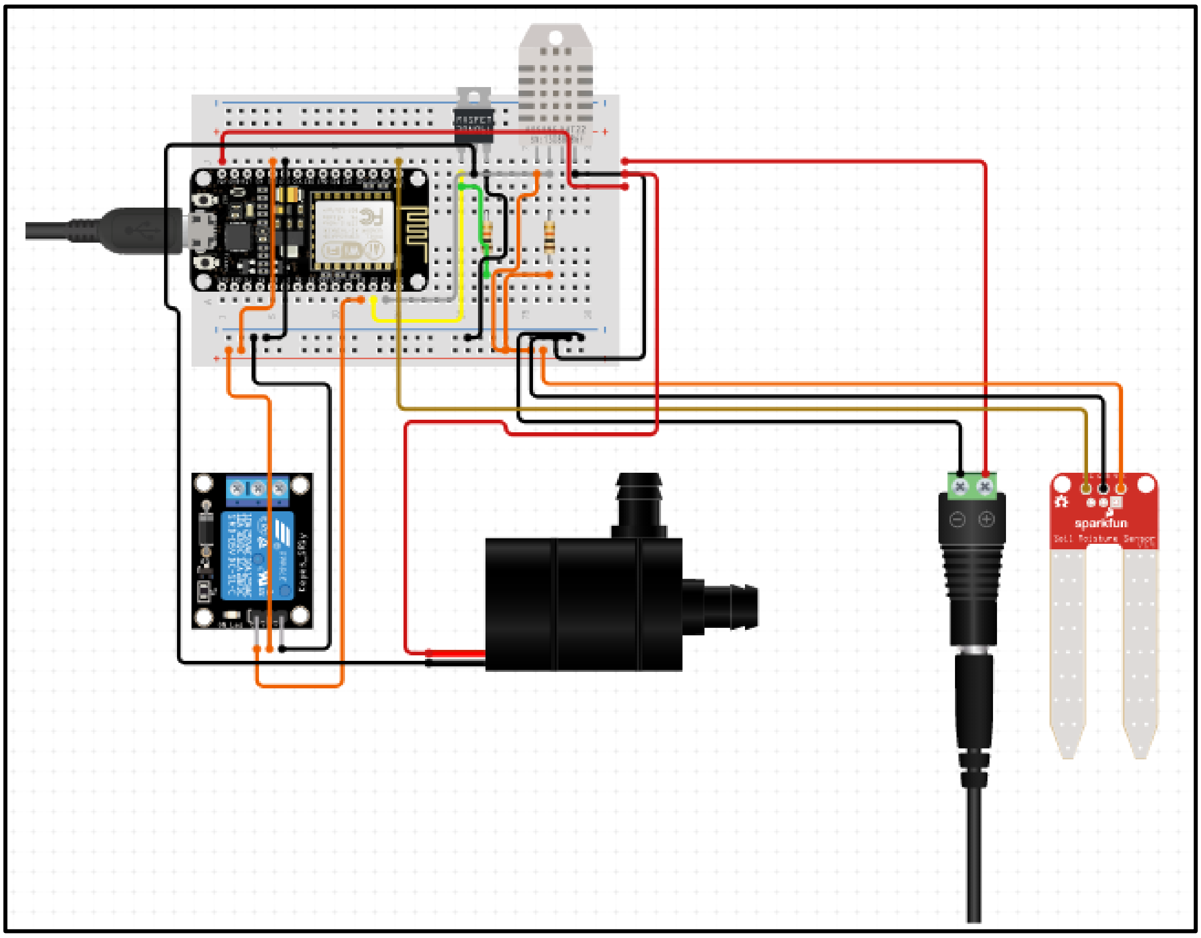

Figure no.4.1 Circuit Simulation

4.2 NodeMCU Program

nodeMCU

```
#include <DHT.h>
#include <ESP8266WiFi.h>
String apiKey = "X5AQ3EGIKMBYW31H";     //  Enter your Write API key here
const char* server = "api.thingspeak.com";
const char *ssid =  "miniproject";     // Enter your WiFi Name
const char *pass =  "ask@123"; // Enter your WiFi Password
#define DHTPIN D3          // GPIO Pin where the dht11 is connected
DHT dht(DHTPIN, DHT11);
WiFiClient client;

const int moisturePin = A0;             // moisteure sensor pin
const int motorPin = D0;
unsigned long interval = 10000;
unsigned long previousMillis = 0;
unsigned long interval1 = 1000;
unsigned long previousMillis1 = 0;
float moisturePercentage;              //moisture reading
float h;                  // humidity reading
float t;                  //temperature reading

void setup()
{
  Serial.begin(115200);
  delay(10);
  pinMode(motorPin, OUTPUT);
  digitalWrite(motorPin, LOW); // keep motor off initally
  dht.begin();
  Serial.println("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");              // print ... till not connected
  }
```

Figure no.4.2.1 NodeMCU Program

nodeMCU

```
  }
  Serial.println("");
  Serial.println("WiFi connected");
}

void loop()
{
  unsigned long currentMillis = millis(); // grab current time

  h = dht.readHumidity();     // read humiduty
  t = dht.readTemperature();     // read temperature

  if (isnan(h) || isnan(t))
  {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  moisturePercentage = ( 100.00 - ( (analogRead(moisturePin) / 1023.00) * 100.00 ) );

  if ((unsigned long)(currentMillis - previousMillis1) >= interval1) {
    Serial.print("Soil Moisture is  = ");
    Serial.print(moisturePercentage);
    Serial.println("%");
    previousMillis1 = millis();
  }

if (moisturePercentage < 50) {
  digitalWrite(motorPin, HIGH);         // tun on motor
}
if (moisturePercentage > 50 && moisturePercentage < 55) {
  digitalWrite(motorPin, HIGH);        //turn on motor pump
}
if (moisturePercentage > 56) {
  digitalWrite(motorPin, LOW);          // turn off mottor
}
```

Figure no.4.2.2 NodeMCU Program

nodeMCU

```
if ((unsigned long)(currentMillis - previousMillis) >= interval) {

  sendThingspeak();           //send data to thing speak
  previousMillis = millis();
  client.stop();
}

}

void sendThingspeak() {
  if (client.connect(server, 80))
  {
    String postStr = apiKey;              // add api key in the postStr string
    postStr += "&field1=";
    postStr += String(moisturePercentage);    // add mositure readin
    postStr += "&field2=";
    postStr += String(t);                 // add tempr readin
    postStr += "&field3=";
    postStr += String(h);                  // add humidity readin
    postStr += "\r\n\r\n";

    client.print("POST /update HTTP/1.1\n");
    client.print("Host: api.thingspeak.com\n");
    client.print("Connection: close\n");
    client.print("X-THINGSPEAKAPIKEY: " + apiKey + "\n");
    client.print("Content-Type: application/x-www-form-urlencoded\n");
    client.print("Content-Length: ");
    client.print(postStr.length());           //send lenght of the string
    client.print("\n\n");
    client.print(postStr);                      // send complete string
    Serial.print("Moisture Percentage: ");
    Serial.print(moisturePercentage);
    Serial.print("%. Temperature: ");
    Serial.print(t);
    Serial.print(" C, Humidity: ");
```

Figure no.4.2.3 NodeMCU Program

4.3 Think Speak Dashboard(Output)

1) **Temperature:**

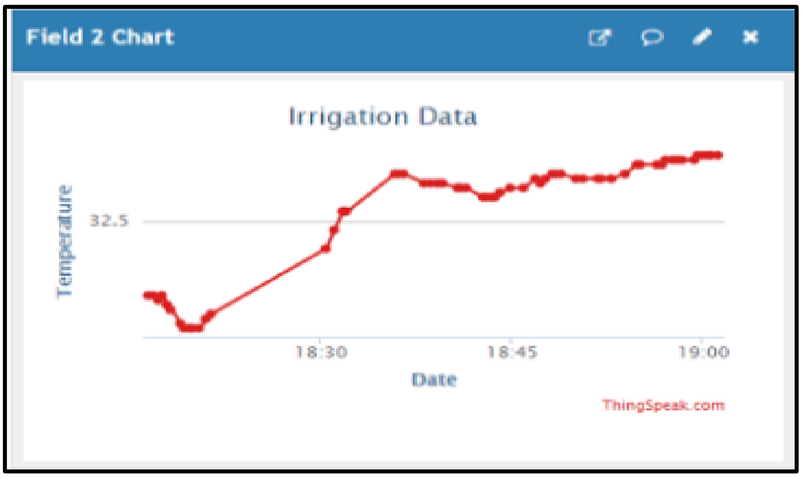

Figure no.4.3.1 Graph of temperature

2) **Humidity:**

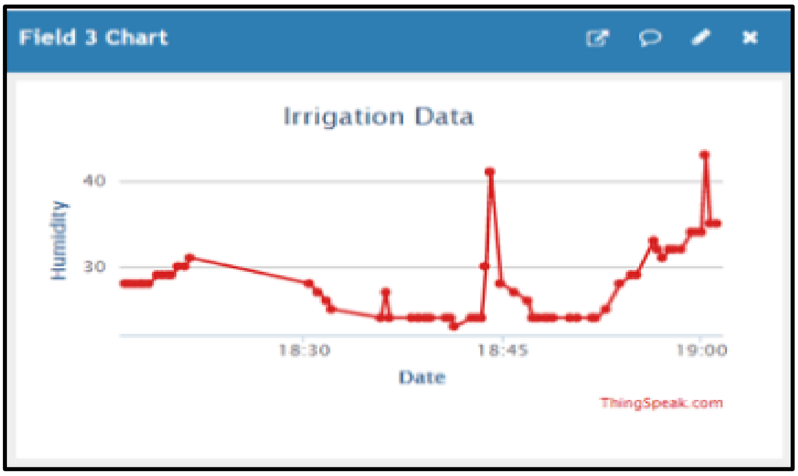

Figure no.4.3.1 Graph of humidity

3) **Soil Moisture:**

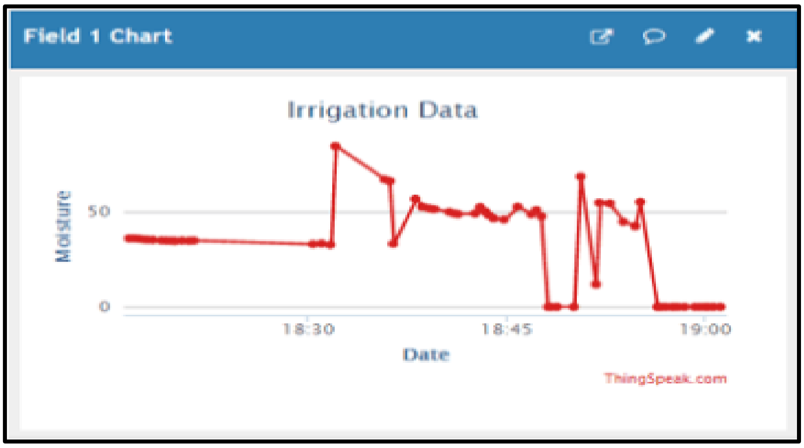

Figure no.4.3.1 Graph of Soil Moisture

Chapter 5

Implementation and Experimentation (ML)

This chapter presents the details about the project implementation and working of the ML model (Crop Yield Prediction).

5.1 Crop Yield Prediction

Crop yield prediction is an important agricultural problem. The Agricultural yield primarily depends on weather conditions (rain, temperature, etc), pesticides. Accurate information about history of crop yield is important for making decisions related to agricultural risk management and future predictions.

In this project the prediction of top 10 most consumed yields all over the world is established by applying machine learning techniques.

These crops include :

- Cassava
- Maize
- Plantains and others
- Potatoes
- Rice, paddy
- Sorghum
- Soybeans
- Sweet potatoes
- Wheat
- Yams

5.2 Gathering & Cleaning Data

Data collection is the process of gathering and measuring information on variables of interest. FAOSTAT provides access to over 3 million time-series and cross sectional data relating to food and agriculture. The FAO data can be found in csv format. FAOSTAT contains data for 200 countries and more than 200 primary products and inputs in its core data set. It offers national and international statistics on food and agriculture. The first thing to get is the crops yield for each country.

- Importing Data set of actual yield:

```
[ ] from google.colab import drive
    drive.mount('/content/drive')

    Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

[ ] import numpy as np
    import pandas as pd

[ ] df_yield = pd.read_csv('/content/drive/MyDrive/crop yield/yield.csv')
    df_yield.shape

    (56717, 12)
```

- Data frame of yield (Head):

```
df_yield.head()
```

	Domain Code	Domain	Area Code	Area	Element Code	Element	Item Code	Item	Year Code	Year	Unit	Value
0	QC	Crops	2	Afghanistan	5419	Yield	56	Maize	1961	1961	hg/ha	14000
1	QC	Crops	2	Afghanistan	5419	Yield	56	Maize	1962	1962	hg/ha	14000
2	QC	Crops	2	Afghanistan	5419	Yield	56	Maize	1963	1963	hg/ha	14260
3	QC	Crops	2	Afghanistan	5419	Yield	56	Maize	1964	1964	hg/ha	14257
4	QC	Crops	2	Afghanistan	5419	Yield	56	Maize	1965	1965	hg/ha	14400

- Data frame of yield (Tail):

```
df_yield.tail()
```

	Domain Code	Domain	Area Code	Area	Element Code	Element	Item Code	Item	Year Code	Year	Unit	Value
56712	QC	Crops	181	Zimbabwe	5419	Yield	15	Wheat	2012	2012	hg/ha	24420
56713	QC	Crops	181	Zimbabwe	5419	Yield	15	Wheat	2013	2013	hg/ha	22888
56714	QC	Crops	181	Zimbabwe	5419	Yield	15	Wheat	2014	2014	hg/ha	21357
56715	QC	Crops	181	Zimbabwe	5419	Yield	15	Wheat	2015	2015	hg/ha	19826
56716	QC	Crops	181	Zimbabwe	5419	Yield	15	Wheat	2016	2016	hg/ha	18294

Looking at the columns in the csv, we can rename **Value** to **hg/ha_yield** to make it easier to recognise that this is our crops yields production value. In addition to removal of unnecessary coloumns like Area Code, Domain, Item Code, etc.

- Renaming columns:

```
df_yield = df_yield.rename(index=str, columns={"Value": "hg/ha_yield"})
df_yield.head()
```

	Domain Code	Domain	Area Code	Area	Element Code	Element	Item Code	Item	Year Code	Year	Unit	hg/ha_yield
0	QC	Crops	2	Afghanistan	5419	Yield	56	Maize	1961	1961	hg/ha	14000
1	QC	Crops	2	Afghanistan	5419	Yield	56	Maize	1962	1962	hg/ha	14000
2	QC	Crops	2	Afghanistan	5419	Yield	56	Maize	1963	1963	hg/ha	14260
3	QC	Crops	2	Afghanistan	5419	Yield	56	Maize	1964	1964	hg/ha	14257
4	QC	Crops	2	Afghanistan	5419	Yield	56	Maize	1965	1965	hg/ha	14400

- Removing unwanted columns :

	Area	Item	Year	hg/ha_yield
0	Afghanistan	Maize	1961	14000
1	Afghanistan	Maize	1962	14000
2	Afghanistan	Maize	1963	14260
3	Afghanistan	Maize	1964	14257
4	Afghanistan	Maize	1965	14400

- Basic description:

```
df_yield.describe()
```

	Year	hg/ha_yield
count	56717.000000	56717.000000
mean	1989.669570	62094.660084
std	16.133198	67835.932856
min	1961.000000	0.000000
25%	1976.000000	15680.000000
50%	1991.000000	36744.000000
75%	2004.000000	86213.000000
max	2016.000000	1000000.000000

From cell above, we know the dataframe starts at 1961 and ends at 2016, this is all the available data up to date from FAO.

- Climate Data: Rainfall

The climatic factors include rainfall and temperature. They are abiotic components, including pesticides and soil, of the environmental factors that influence plant growth and development. Rainfall has a dramatic effect on agriculture. For this project rain fall per year information was gathered from World Data Bank.

- Importing Climate data rainfall:

```
df_rain = pd.read_csv('/content/drive/MyDrive/crop yield/rainfall.csv')
df_rain.head()
```

	Area	Year	average_rain_fall_mm_per_year
0	Afghanistan	1985	327
1	Afghanistan	1986	327
2	Afghanistan	1987	327
3	Afghanistan	1989	327
4	Afghanistan	1990	327

- Description of Rainfall:

```
df_rain.describe()
```

	Year
count	5953.000000
mean	2001.352091
std	9.531763
min	1985.000000
25%	1993.000000
50%	2001.000000
75%	2010.000000
max	2017.000000

- Pesticides Data:

```
df_pes = pd.read_csv('/content/drive/MyDrive/crop yield/pesticides.csv')
df_pes.head()
```

	Domain	Area	Element	Item	Year	Unit	Value
0	Pesticides Use	Albania	Use	Pesticides (total)	1990	tonnes of active ingredients	121.0
1	Pesticides Use	Albania	Use	Pesticides (total)	1991	tonnes of active ingredients	121.0
2	Pesticides Use	Albania	Use	Pesticides (total)	1992	tonnes of active ingredients	121.0
3	Pesticides Use	Albania	Use	Pesticides (total)	1993	tonnes of active ingredients	121.0
4	Pesticides Use	Albania	Use	Pesticides (total)	1994	tonnes of active ingredients	201.0

```
df_pes = df_pes.rename(index=str, columns={"Value": "pesticides_tonnes"})
df_pes = df_pes.drop(['Element','Domain','Unit','Item'], axis=1)
df_pes.head()
```

	Area	Year	pesticides_tonnes
0	Albania	1990	121.0
1	Albania	1991	121.0
2	Albania	1992	121.0
3	Albania	1993	121.0
4	Albania	1994	201.0

- Description of pesticides:

```
df_pes.describe()
```

	Year	pesticides_tonnes
count	4349.000000	4.349000e+03
mean	2003.138883	2.030334e+04
std	7.728044	1.177362e+05
min	1990.000000	0.000000e+00
25%	1996.000000	9.300000e+01
50%	2003.000000	1.137560e+03
75%	2010.000000	7.869000e+03
max	2016.000000	1.807000e+06

- Loading temperature dataset:

```
avg_temp=  pd.read_csv('/content/drive/MyDrive/crop yield/temp.csv')
```

```
avg_temp.head()
```

	year	country	avg_temp
0	1849	Côte D'Ivoire	25.58
1	1850	Côte D'Ivoire	25.52
2	1851	Côte D'Ivoire	25.67
3	1852	Côte D'Ivoire	NaN
4	1853	Côte D'Ivoire	NaN

- Description of temperature data:

```
avg_temp.describe()
```

	year	avg_temp
count	71311.000000	68764.000000
mean	1905.799007	16.183876
std	67.102099	7.592960
min	1743.000000	-14.350000
25%	1858.000000	9.750000
50%	1910.000000	16.140000
75%	1962.000000	23.762500
max	2013.000000	30.730000

- Merging all required data frames together to predict yield:

```
yield_df = pd.merge(yield_df,avg_temp, on=['Area','Year'])
yield_df.head()
```

	Area	Item	Year	hg/ha_yield	average_rain_fall_mm_per_year	pesticides_tonnes	avg_temp
0	Albania	Maize	1990	36613	1485	121.0	16.37
1	Albania	Potatoes	1990	66667	1485	121.0	16.37
2	Albania	Rice, paddy	1990	23333	1485	121.0	16.37
3	Albania	Sorghum	1990	12500	1485	121.0	16.37
4	Albania	Soybeans	1990	7000	1485	121.0	16.37

- Whole required data Description:

```
yield_df.describe()
```

	Year	hg/ha_yield	pesticides_tonnes	avg_temp
count	28248.000000	28248.000000	28248.000000	28248.000000
mean	2001.541950	77047.863282	37069.136973	20.543722
std	7.052997	84950.194454	59954.787836	6.311828
min	1990.000000	50.000000	0.040000	1.300000
25%	1995.000000	19918.750000	1695.710000	16.710000
50%	2001.000000	38295.000000	17517.760000	21.510000
75%	2008.000000	104598.250000	48687.880000	26.000000
max	2013.000000	501412.000000	367778.000000	30.650000

5.3 Data Exploration

Grouping by countries and items to have a better understanding of the dataset. There are 101 countries in the dataframe, with India having the highest crop yield production. In addition, India is the highest for production of cassava and potatoes. Potatoes seem to be the dominant crop in the dataset, being the highest in 4 countries.

```
yield_df.groupby('Item').count()
```

Item	Area	Year	hg/ha_yield	average_rain_fall_mm_per_year	pesticides_tonnes	avg_temp
Cassava	2047	2047	2047	2047	2047	2047
Maize	4123	4123	4123	4123	4123	4123
Plantains and others	556	556	556	556	556	556
Potatoes	4276	4276	4276	4276	4276	4276
Rice, paddy	3388	3388	3388	3388	3388	3388
Sorghum	3039	3039	3039	3039	3039	3039
Soybeans	3223	3223	3223	3223	3223	3223
Sweet potatoes	2892	2892	2892	2892	2892	2892
Wheat	3857	3857	3857	3857	3857	3857
Yams	847	847	847	847	847	847

```
yield_df.describe()
```

	Year	hg/ha_yield	pesticides_tonnes	avg_temp
count	28248.000000	28248.000000	28248.000000	28248.000000
mean	2001.541950	77047.863282	37069.136973	20.543722
std	7.052997	84950.194454	59954.787836	6.311828
min	1990.000000	50.000000	0.040000	1.300000
25%	1995.000000	19918.750000	1695.710000	16.710000
50%	2001.000000	38295.000000	17517.760000	21.510000
75%	2008.000000	104598.250000	48687.880000	26.000000
max	2013.000000	501412.000000	367778.000000	30.650000

- Top 10 countries with highest crop yield with india on top of the list:

```
yield_df.groupby(['Area'],sort=True)['hg/ha_yield'].sum().nlargest(10)
```

```
Area
India             327420324
Brazil            167550306
Mexico            130788528
Japan             124470912
Australia         109111062
Pakistan           73897434
Indonesia          69193506
United Kingdom     55419990
Turkey             52263950
Spain              46773540
Name: hg/ha_yield, dtype: int64
```

- Grouping yield by crops:

```
yield_df.groupby(['Item','Area'],sort=True)['hg/ha_yield'].sum().nlargest(10)
```

```
Item            Area
Cassava         India             142810624
Potatoes        India              92122514
                Brazil             49602168
                United Kingdom     46705145
                Australia          45670386
Sweet potatoes  India              44439538
Potatoes        Japan              42918726
                Mexico             42053880
Sweet potatoes  Mexico             35808592
                Australia          35550294
Name: hg/ha_yield, dtype: int64
```

5.4 Data Preprocessing

Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.

In the final dataframe there are two categorical columns in the dataframe, categorical data are variables that contain label values rather than numeric values. The number of possible values is often limited to a fixed set, like in this case, items and countries values. Many machine learning algorithms cannot operate on label data directly, they require all input variables and output variables to be numeric.

- Encoding Categorical Variables:

```
from sklearn.preprocessing import OneHotEncoder
```

```
yield_df_onehot = pd.get_dummies(yield_df, columns=['Area',"Item"], prefix = ['Country',"Item"])
features=yield_df_onehot.loc[:, yield_df_onehot.columns != 'hg/ha_yield']
label=yield_df['hg/ha_yield']
features.head()
```

	Year	average_rain_fall_mm_per_year	pesticides_tonnes	avg_temp	Country_Albania	Country_Algeria	Country_Angola	Country_Argentina	Country_Armenia	Country_Australia	Country_Austria	Country_Azerbaijan	Country_Bahamas
0	1990	1485	121.0	16.37	1	0	0	0	0	0	0	0	0
1	1990	1485	121.0	16.37	1	0	0	0	0	0	0	0	0
2	1990	1485	121.0	16.37	1	0	0	0	0	0	0	0	0
3	1990	1485	121.0	16.37	1	0	0	0	0	0	0	0	0
4	1990	1485	121.0	16.37	1	0	0	0	0	0	0	0	0

5 rows × 115 columns

- Training Data:

The dataset will be split to two datasets, the training dataset and test dataset. The data is usually tend to be split inequality because training the model usually requires as much data-points as possible. The common splits are 70/30 or 80/20 for train/test.

The training dataset is the initial dataset used to train ML algorithm to learn and produce right predictions. (70% of dataset is training dataset)

The test dataset, however, is used to assess how well ML algorithm is trained with the training dataset. You can’t simply reuse the training dataset in the testing stage because ML algorithm will already “know” the expected output, which defeats the purpose of testing the algorithm. (30% of dataset is testing dataset

```
]  from sklearn.model_selection import train_test_split
   train_data, test_data, train_labels, test_labels = train_test_split(features, label, test_size=0.3, random_state=42)
```

```
]  #write final df to csv file
   yield_df.to_csv('yield_df.csv')
```

```
]  from sklearn.model_selection import train_test_split
   train_data, test_data, train_labels, test_labels = train_test_split(features, label, test_size=0.3, random_state=42)
```

5.5 Model Comparison & Selection

Before deciding on an algorithm to use, first we need to evaluate, compare and choose the best one that fits this specific dataset.

Usually, when working on a machine-learning problem with a given dataset, we try different models and techniques to solve an optimization problem and fit the most accurate model that will neither over fit nor under fit the model.

The evaluation metric is set based on **R^2 (coefficient of determination)** regression score function, that will represent the proportion of the variance for items (crops) in the regression model. **R^2** score shows how well terms (data points) fit a curve or line.

For this project, we'll compare between the following models :

- Gradient Boosting Regressor
- Random Forest Regressor
- SVM
- Decision Tree Regressor

```
print(*model_train, sep = "\n")

['GradientBoostingRegressor', 0.8965731164462923]
['RandomForestRegressor', 0.6842532317855172]
['SVR', -0.20353376480360752]
['DecisionTreeRegressor', 0.9590644980817136]
```

From results viewed above, **Decision Tree Regressor** has the highest R^2 score of **96%**, **Gradient Boosting Regressor** comes second with **89%.**

we'll also calculate Adjusted R^2 , where it also indicates how well terms fit a curve or line, but adjusts for the number of terms in a model. If you add more and more useless variables to a model, adjusted r-squared will decrease. If you add more useful variables, adjusted r-squared will increase. Adjusted R^2will always be less than or equal to R^2

- Setting test data to columns from dataframe and excluding 'hg/ha_yield' values where ML model should be predicting:

```
test_df['Country']=countries
test_df['Item']=items
test_df.head()
```

	average_rain_fall_mm_per_year	pesticides_tonnes	avg_temp	Country	Item
0	0.183443	0.110716	0.542078	Spain	Rice, paddy
1	0.458451	0.000413	0.627257	Madagascar	Wheat
2	0.183443	0.106159	0.518228	Spain	Sorghum
3	1.000000	0.224154	0.890971	Colombia	Potatoes
4	0.458451	0.000355	0.625213	Madagascar	Sweet potatoes

```
clf=DecisionTreeRegressor()
model=clf.fit(train_data,train_labels)

test_df["yield_predicted"]= model.predict(test_data)
test_df["yield_actual"]=pd.DataFrame(test_labels)["hg/ha_yield"].tolist()
test_group=test_df.groupby("Item")
test_group.apply(lambda x: r2_score(x.yield_actual,x.yield_predicted))
```

```
Item
Cassava                 0.926622
Maize                   0.894963
Plantains and others    0.817372
Potatoes                0.910895
Rice, paddy             0.896925
Sorghum                 0.802025
Soybeans                0.847838
Sweet potatoes          0.840751
Wheat                   0.922150
Yams                    0.928241
dtype: float64
```

- Actual vs. predicted:

```
fig, ax = plt.subplots()

ax.scatter(test_df["yield_actual"], test_df["yield_predicted"],edgecolors=(0, 0, 0))

ax.set_xlabel('Actual')
ax.set_ylabel('Predicted')
ax.set_title("Actual vs Predicted")
plt.show()
```

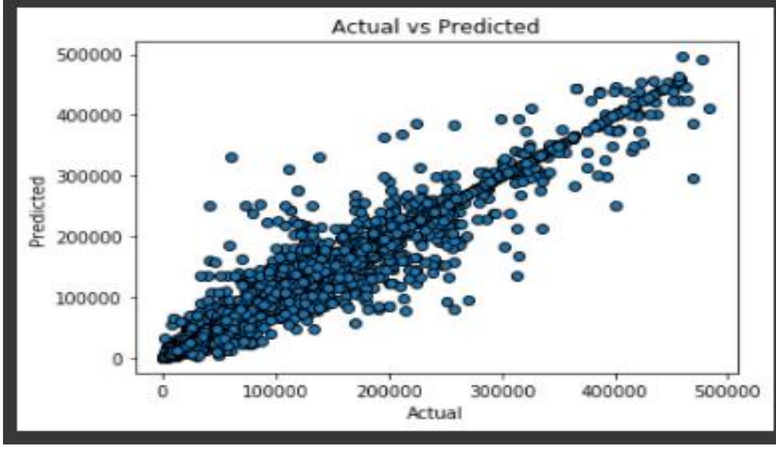

Figure no. 5.5.1 actual vs predicted

5.6 Model Results

The most common **interpretation** of **R²** is how well the regression model fits the observed data. For **example**, an **R²** of 60% reveals that 60% of the data fit the regression model. Generally, a higher **R²** indicates a better fit for the model. From the obtained results, it's clear that the model fits the data to a very good measure of 96%.

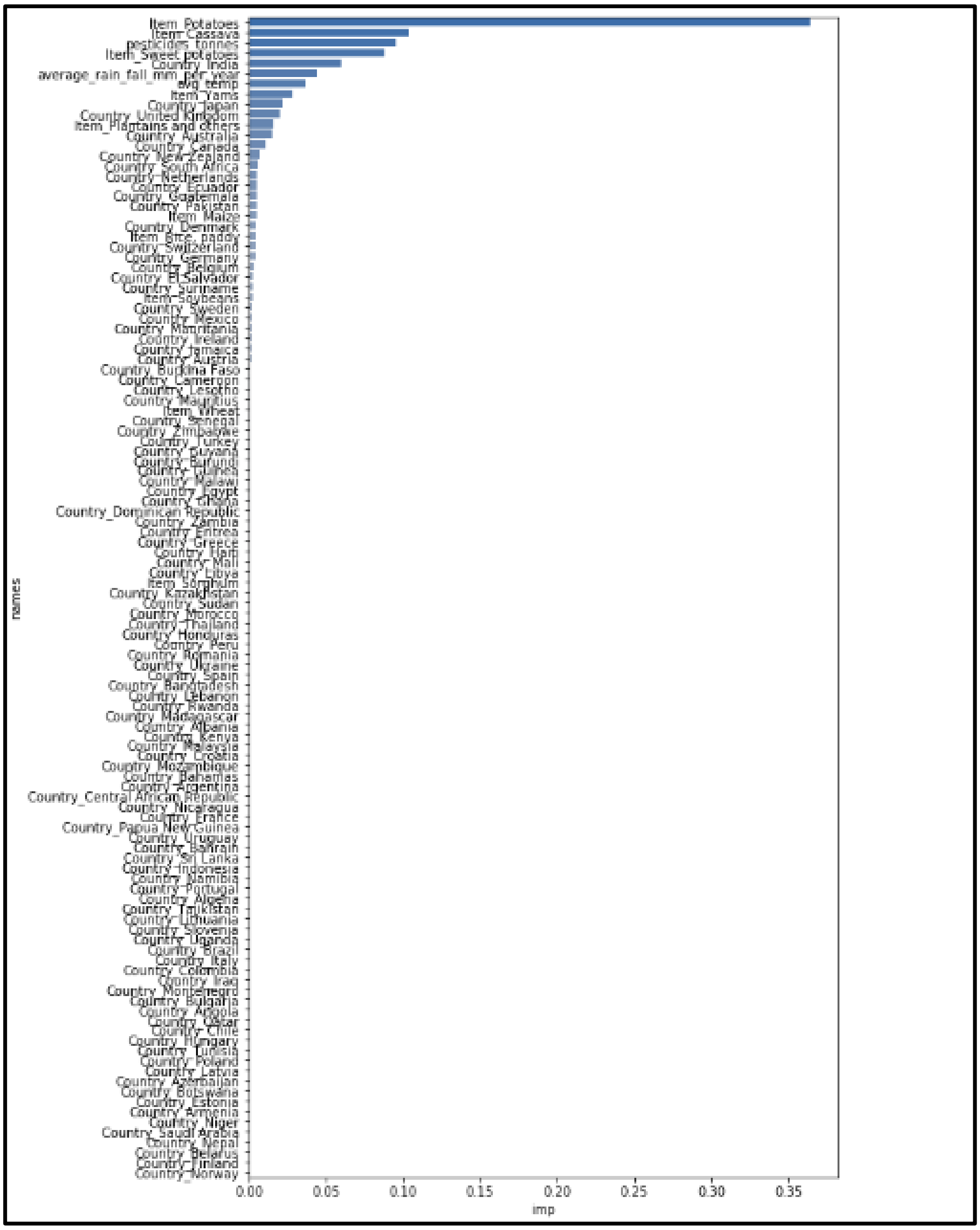

Figure no. 5.6.1 model results

- Getting only top 7 of features importance in the model:

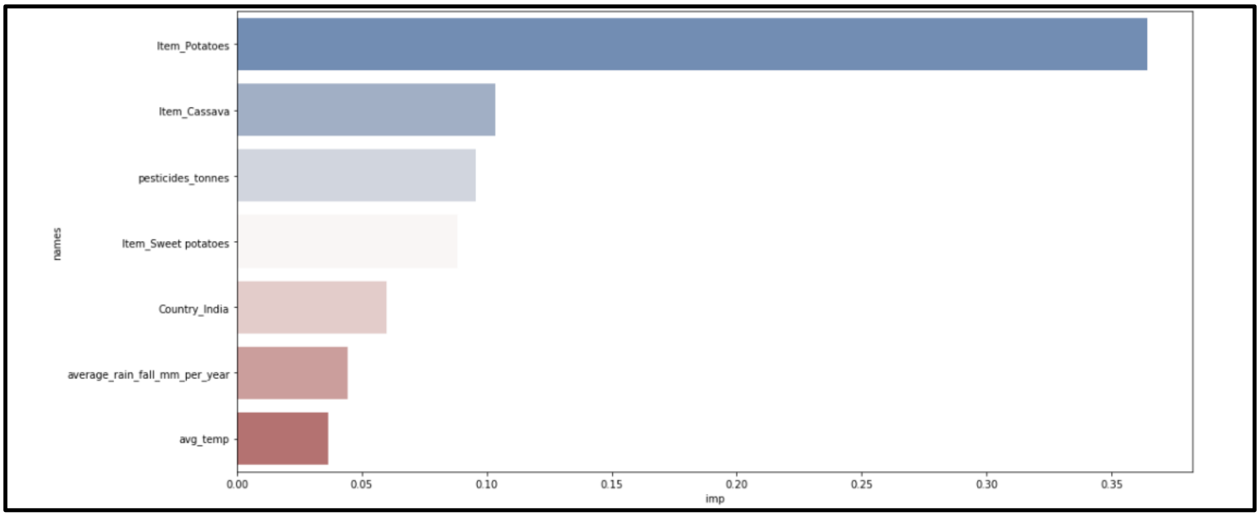

Figure no. 5.6.2 Top 7 features

The crop being potatoes has the highest importance in the decision making for the model, where it's the highest crops in the dataset. Cassava too, then as expected we see the effect of pesticides, where its the third most important feature, and then if the crop is sweet potatoes, we see some of the highest crops in features importance in dataset.

If the crop is grown in India, makes sense since India has the largest crops sum in the dataset. Then comes rainfall and temperature. The first assumption about these features were correct, where they all significantly impact the expected crops yield in the model.

- Boxplot that shows yield for each item

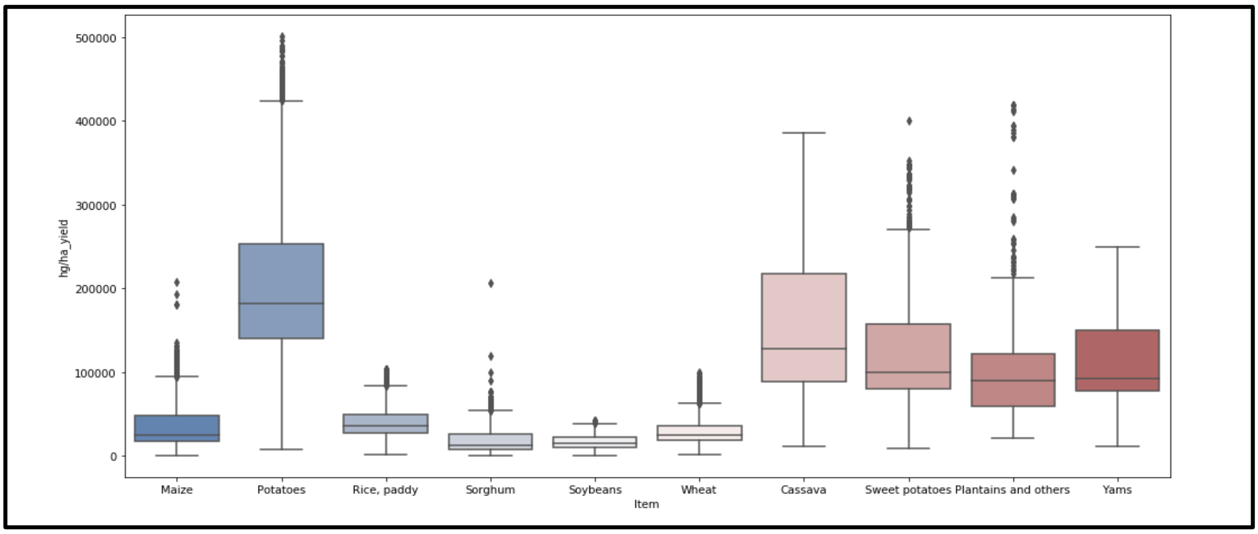

Figure no. 5.6.3 Boxplot of yield

Chapter 6

Conclusion and Future Work

This Chapter presents the conclusion & future scope of the implemented project.

6.1 Conclusion

In this project we strived to bring efficiency and accuracy in the field of agriculture by automating the entire agricultural process with the help of Internet of Things and Machine Learning. In IOT we proposed a model on NodeMCU with embedded ESP8266 Wi-Fi module controls the monitoring system and the transfer of data to and from the sensors to the Think Speak server without human intervention. With the help of historical data we predicted the crop yield using ML by comparing multiple ML algorithms and by comparing we got to know that Decision Tree Regressor is the best algorithm for this as it is a best fitted model. Thus this system aims at empowering farmers with the decision-making tools and automation technologies that seamlessly integrate products, knowledge and services for better productivity, quality, and profit.

6.2 Future Scope

Considering this project will definitely help the farmer for the production of crops and will solve their hassle so we have proposed some future work which we can implement:

1. We can also add PH sensor which will sense the PH of the soil at a regular interval and predict whether that soil needs more water or not.
2. We can also add more features, like climate data wind and pollution, the economical situation of a given country and so on which will probably enhance the model's predictions
3. As we have proposed one prediction model which is based on crop yield prediction we can further add more models which will based on Crop Classification.
4. We can also develop a mobile/web app for better mobility and better and simpler user interface for under educated farmers.
5. The model can be improved by integrating this with other departments like horticulture, sericulture, and others towards the agricultural development of our country.

Bibliography

1. Mishra D., Khan A., Tiwari R and Upadhay S. (2018). "Automated Irrigation System-IoT Based Approach". 3rd International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU). Available: https://ieeexplore.ieee.org/document/8519886 [Accessed 25 November 2019].

2. **"**Smart Irrigation System using Iot Approach" Dr.S.Jothi Muneeswari1 , Merlin Janet E2 , Rajeshwari3 , G.Selvarani4 Department of ECE, DMI College of Engineering, Chennai

3. Pernapati K. (2019). IoT Based Low Cost Smart Irrigation System. Retrieved 25 November 2019

4. "Design and Implementation Of Automatic Plant Irrigation System" by Dr.P.Sengottuvel, Dr.J.Hameed Hussain BIST, BIHER, Bharath University, Chennai-73

5. T Raghav Kumar, Bhagavatula Aiswarya, Aashish Suresh, Drishti Jain, Natesh Balaji, Varshini Sankaran, "Smart Management of Crop Cultivation using IOT and Machine Learning," International Research Journal of Engineering and Technology (IRJET) Nov 2018, pp. 845- 850

6. Akash Raj N, Balaji Srinivasan, Deepit Abhishek D, Sarath Jeyavanth J, Vinith Kannan A, "IoT based Agro Automation System using Machine Learning Algorithms", International Journal of Innovative Research in Science, Engineering and Technology November 2016, pp. 19938-19342

7. Design And Implementation Of Crop Yield Prediction Model In Agriculture http://www.ijstr.org/final-print/jan2020/Design-And-Implementation-Of-Crop-Yield-Prediction-Model-In-Agriculture.pdf

8. Crop Prediction System using Machine Learning Algorithms https://www.irjet.net/archives/V7/i2/IRJET-V7I2163.pdf

9. https://circuitdigest.com/microcontroller-projects/iot-based-smart-irrigation-system-using-esp8266-and-soil-moisture-sensor

10. https://towardsdatascience.com/predicting-crops-yield-machine-learning-nanodegree-capstone-project-e6ec9349f69

11. http://www.fao.org/home/en/

12. https://data.worldbank.org/

13. https://gdcoder.com/decision-tree-regressor-explained-in-depth/

Acknowledgements

Firstly, we would like to express our gratitude for our Faculty advisor Dr. Anudeepa Kholapure, EXTC Department whose relentless efforts helped us to maintain a perfect timeline for the execution of our mini project. We would also like to thank our EXTC department and K.J. Somaiya College of Engineering to help us with the execution of the Mini Project. We would like to thank our HOD Sir, Dr. Ameya Naik and our principal ma'am Dr. Shubha Pandit for giving us this opportunity. This practical knowledge we gained will help us for our Final year project as well as other projects assigned during corporate working. This project has helped us in developing our cognitive skills and team management abilities.