

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belgaum-590018



A PROJECT REPORT (15CSP85) ON

## “CROP YIELD PREDICTION BASED ON WEATHER USING MACHINE LEARNING”

Submitted in Partial fulfillment of the Requirements for the Degree of Bachelor of  
Engineering in Computer Science & Engineering

By

**ANMOL (1CR16CS022)**

**R AISHWARYA (1CR16CS031)**

**KARUNA ROHILLA (1CR16CS069)**

**KUMAR SHAURYA (1CR16CS073)**

Under the Guidance

of,

**Dr. Prem Kumar Ramesh**

**Professor & Head, Dept. of CSE**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CMR INSTITUTE OF TECHNOLOGY**

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI, BANGALORE-560037

# CMR INSTITUTE OF TECHNOLOGY

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI, BANGALORE-560037

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



## CERTIFICATE

Certified that the project work entitled “**CROP YIELD PREDICTION BASED ON WEATHER USING MACHINE LEARNING**” carried out by **Ms. ANMOL**, USN **1CR16CS022**, **Ms. BR AISHWARYA**, USN **1CR16CS031**, **Ms. KARUNA ROHILLA**, USN **1CR16CS069**, **Mr. KUMAR SHAURYA**, USN **1CR16CS073**, bonafide students of CMR Institute of Technology, in partial fulfillment for the award of **Bachelor of Engineering** in Computer Science and Engineering of the Visveswaraiah Technological University, Belgaum during the year 2019-2020. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library.

The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

\_\_\_\_\_  
**Dr. Prem Kumar Ramesh**  
Professor & Head  
Dept. of CSE, CMRIT

\_\_\_\_\_  
**Dr. Prem Kumar Ramesh**  
Professor & Head  
Dept. of CSE, CMRIT

\_\_\_\_\_  
**Dr. Sanjay Jain**  
Principal CMRIT

# DECLARATION

We, the students of Computer Science and Engineering, CMR Institute of Technology, Bangalore declare that the work entitled "**CROP YIELD PREDICTION BASED ON WEATHER USING MACHINE LEARNING**" has been successfully completed under the guidance of Dr. Prem Kumar Ramesh, Computer Science and Engineering Department, CMR Institute of technology, Bangalore. This dissertation work is submitted in partial fulfillment of the requirements for the award of Degree of Bachelor of Engineering in Computer Science and Engineering during the academic year 2019 - 2020. Further the matter embodied in the project report has not been submitted previously by anybody for the award of any degree or diploma to any university.

Place:

Date:

**Team members:**

**ANMOL (1CR16CS022)**

---

**BR AISHWARYA (1CR16CS031)**

---

**KARUNA ROHILLA (1CR16CS069)**

---

**KUMAR SHAURYA (1CR16CS073)**

---

## **ABSTRACT**

India is an Agriculture based economy whose most of the GDP comes from farming. The motivation of this project comes from the increasing suicide rates in farmers which may be due to low harvest in crops. Climate and other environmental changes have become a major threat in the agriculture field. Machine learning is an essential approach for achieving practical and effective solutions for this problem. Predicting yield of the crop from historical available data like weather, soil, rainfall parameters and historic crop yield. We achieved this using the machine learning algorithm. We did a comparative study of various machine learning algorithms, i.e., ANN, K Nearest Neighbour, Random Forest, SVM and Linear Regression and chose Random Forest Algorithm which gave an accuracy of 95%.

In this project a mobile application has been developed which predicts the crop yield in general and also for a particular crop. Along with that, it also suggests the user if it is the right time to use the fertilizer or not.

## ACKNOWLEDGEMENT

We take this opportunity to express my sincere gratitude and respect to **CMR Institute of Technology, Bengaluru** for providing me a platform to pursue my studies and carry out my final year project.

We have great pleasure in expressing my deep sense of gratitude to **Dr. Sanjay Jain**, Principal, CMRIT, Bangalore, for his constant encouragement.

We would like to thank **Dr. Prem Kumar Ramesh**, Professor and Head, Department of Computer Science and Engineering, CMRIT, Bangalore, who has been a constant support and encouragement throughout the course of this project.

We consider it a privilege and honor to express our sincere gratitude to **Mrs. Shilpa Pande**, Professor, Department of Information Science and Engineering, for the valuable guidance throughout the tenure of this project.

We also extend my thanks to all the faculty of Computer Science and Engineering who directly or indirectly encouraged me.

Finally, we would like to thank my parents and friends for all their moral support they have given me during the completion of this work.

# TABLE OF CONTENTS

	<b>Page No.</b>
<b>Certificate</b>	<b>ii</b>
<b>Declaration</b>	<b>iii</b>
<b>Abstract</b>	<b>iv</b>
<b>Acknowledgement</b>	<b>v</b>
<b>Table of contents</b>	<b>vi</b>
<b>List of Figures</b>	<b>viii</b>
<b>List of Tables</b>	<b>x</b>
<b>List of Abbreviations</b>	<b>xi</b>
<b>1. INTRODUCTION</b>	<b>1</b>
<b>1.1 Relevance of the Project</b>	<b>2</b>
<b>1.2 Problem Statement</b>	<b>2</b>
<b>1.3 Objective</b>	<b>3</b>
<b>1.4 Scope of the Project</b>	<b>4</b>
<b>1.5 Methodology</b>	<b>5</b>
<b>2. LITERATURE SURVEY</b>	<b>19</b>
<b>2.1 Machine learning approach for forecasting crop yield based on climatic parameters</b>	<b>19</b>
<b>2.2 Prediction of Crop Yield Using Machine Learning</b>	<b>20</b>
<b>2.3 Predicting Yield of the Crop Using Machine Learning Algorithm</b>	<b>22</b>
<b>2.4 A Survey on Crop Prediction using Machine Learning Approach</b>	<b>23</b>
<b>2.5 Heuristic Prediction of Crop Yield using Machine Learning Technique</b>	<b>25</b>
<b>3. SYSTEM REQUIREMENTS SPECIFICATION</b>	<b>28</b>
<b>3.1 Functional Requirements</b>	<b>29</b>
<b>3.2 Non-Functional Requirements</b>	<b>31</b>
<b>3.3 Hardware Requirements</b>	<b>33</b>
<b>3.4 Software Requirements</b>	<b>36</b>

<b>4. SYSTEM ANALYSIS AND DESIGN</b>	<b>40</b>
<b>4.1 System Architecture</b>	<b>40</b>
<b>4.2 Flowchart</b>	<b>41</b>
<b>4.3 Use Case</b>	<b>41</b>
<b>4.4 Activity Diagram</b>	<b>42</b>
<b>4.5 Sequence Diagram</b>	<b>43</b>
<b>5. IMPLEMENTATION</b>	<b>45</b>
<b>5.1 Crop Yield Prediction</b>	<b>45</b>
<b>5.2 Fertilizers Module</b>	<b>45</b>
<b>5.3 Experimental Setup</b>	<b>46</b>
<b>6. RESULTS AND DISCUSSION</b>	<b>49</b>
<b>7. TESTING</b>	<b>53</b>
<b>7.1 Functionality Testing</b>	<b>53</b>
<b>7.2 Usability Testing</b>	<b>53</b>
<b>7.3 Interface Testing</b>	<b>54</b>
<b>7.4 Compatibility Testing</b>	<b>54</b>
<b>7.5 Performance Testing</b>	<b>54</b>
<b>8. CONCLUSION AND FUTURE SCOPE</b>	<b>55</b>
<b>REFERENCES</b>	

## LIST OF FIGURES

	Page No.
<b>Fig 1.1: Types of Machine Learning</b>	7
<b>Fig 1.2: Supervised Learning</b>	7
<b>Fig 1.3: Unsupervised Learning</b>	8
<b>Fig 1.4: Reinforcement Learning</b>	8
<b>Fig 1.5: General Process of Machine Learning</b>	9
<b>Fig 1.6: Soil and Crop data sample</b>	10
<b>Fig 1.7: Rain and Temperature data sample</b>	10
<b>Fig 1.8: Merged Dataset Sample for Crop Yield Prediction</b>	10
<b>Fig 1.9: Merged Dataset Sample for Weather Prediction</b>	11
<b>Fig 1.10: Correlation matrix Example</b>	12
<b>Fig 1.11: EDA Code</b>	
<b>Fig 1.12: Output of describe() function</b>	
<b>Fig 1.13: Correlation Matrix of Proposed System</b>	
<b>Fig 1.14: Random Forest Flow</b>	
<b>Fig 2.1: Modular Diagram</b>	20
<b>Fig 2.2: Experimental Results</b>	22
<b>Fig 3.1: Types of Requirements in SRS</b>	28
<b>Fig 3.2: Processor</b>	32
<b>Fig 3.3: Wifi</b>	33
<b>Fig 3.4: Hard drive</b>	33
<b>Fig 3.5: RAM</b>	34
<b>Fig 3.6: Jupyter Notebook</b>	35
<b>Fig 3.7: Python</b>	35
<b>Fig 3.8: PyCharm</b>	36
<b>Fig 3.9: Ionic</b>	36
<b>Fig 3.10: Flask</b>	37
<b>Fig 4.1 Architecture</b>	38
<b>Fig 4.2 Flowchart of Random Forest Algorithm</b>	39



**Fig 4.3 Flowchart of Crop Yield Prediction System**

**40**

**Fig 4.4 Use Case Diagram**

**41**

<b>Fig 4.5 Activity Diagram</b>	<b>42</b>
<b>Fig 4.6 Sequence Diagram</b>	<b>43</b>
<b>Fig 5.1 Block Diagram of Experimental Implementation</b>	<b>45</b>
<b>Fig 6.1 Register Screen</b>	<b>48</b>
<b>Fig 6.2 Login Screen</b>	<b>48</b>
<b>Fig 6.3 Yield Prediction Screen</b>	<b>49</b>
<b>Fig 6.4 Yield Predicted Screen</b>	<b>49</b>
<b>Fig 6.5 Crop Prediction Screen</b>	<b>50</b>
<b>Fig 6.6 Crops are their production predicted</b>	<b>50</b>
<b>Fig 6.7 Fertilizer Module</b>	<b>51</b>

## **LIST OF TABLES**

	<b>Page No.</b>
<b>Table 1.1 Summary of Approaches</b>	<b>13</b>

## **LIST OF ABBREVIATIONS**

<b>ANN</b>	<b>Artificial Neural Networks</b>
<b>KNN</b>	<b>K nearest Neighbors</b>
<b>ML</b>	<b>Machine Learning</b>
<b>SRS</b>	<b>System Requirements Specification</b>
<b>SVM</b>	<b>Support Vector Machine</b>

## CHAPTER 1

### INTRODUCTION

India is ranked 2nd worldwide in farm output [9]. Agriculture and allied sectors like forestry and fisheries accounted for 16.6 percent of the GDP 2009, about 50 percent of the overall workforce [10]. The monetary contribution of agriculture to India's GDP is regularly declining. The crop yield of plants relies on different factors like on climatic, geographical, organic, political and financial elements. For farmers, it is difficult when there is more than one crop to grow especially when the market prices are unknown to them. Citing the Wikipedia statistics, the farmer suicide rate in India has ranged between 1.4 and 1.8 per 100000 total population, over a 10-year period through 2005. While 2014 saw 5650 farmer suicides, the figure crossed 8000 in 2015 [11].

In recent times, it has become inevitable to use technology to create awareness about cultivation. The seasonal climatic conditions are also being changed against the fundamental assets like soil, water and air which lead to insecurity of food. In a scenario, crop yield rate is falling short of meeting the demand consistently and there is a need for a smart system which can solve the problem of decreasing crop yield. Therefore, to eliminate this problem, we propose a system which will provide crop selection based on economic and environmental factors to reap the maximum yield out of it for the farmers which will sequentially help meet the elevating demands for the food supplies in the country. The proposed system uses machine learning to make the predictions. The system will provide crop yield and crop selection based on weather attributes suitable for the crop to get the maximum yield out of it for the farmers. The system makes predictions of the productions of crops by studying the factors such as rainfall, temperature, area (in hectares), season, etc. The system also helps in suggesting whether a particular time is the right one to use fertilizers.

Crop yield prediction is an important agricultural problem. Every farmer always tries to know how much yield will be produced and whether it meets their expectations. In the past, yield prediction was calculated by analyzing a farmer's previous experience on a particular crop. The Agricultural yield is primarily dependent on weather conditions pests and planning of harvest operation. Accurate information about the history of crop yield is an important thing for making decisions related to agricultural risk management.

## **1.1 Relevance of the Project**

In recent years, India has been shaken by economic and social forces related to higher suicide rates amongst small and marginal farmers [11]. Our aim is to offer assistance and tools to help such farmers and communities and address these issues. Generally, they face challenges accessing and trusting educational outreach and training to better understand how to increase crop yields and improve financial standing. Because of the serious nature of the issues at stake and general hesitance to trust help from outside the community, any service or product meant to help must be carefully designed and tested in order to ensure positive outcomes and successful adoption.

There is no existing software solution which recommends crops based on multiple factors such as type of the soil and weather components which include temperature and rainfall. And the systems that already exist are hardware based which makes them expensive and difficult to maintain. The proposed system suggests a Mobile based application, which can precisely predict the most profitable crop to the farmer by predicting the yield. The user location is identified with the help of GPS and the Area & soil type are taken as user input. According to user location, the crop yield in the respective location is identified from the soil and weather database. After the processing is done at the server side, the result is sent to the user's application. The previous production of the crops is also taken into account which in turn leads to precise crop yield results. Depending on the numerous scenarios and additional filters according to the user requirement, the most producible crop is suggested based on the yield.

While there are many ways to contribute to improvements in the lives of our target audience, our task was to leverage data to predict a valuable result so that farmers and aid workers would be able to make informed planning decisions. Ultimately, the focus of the work during this project was to both conduct audience research that would direct the design of the product and design a data model that would produce the desired results.

## **1.2 Problem Statement**

The Problem Statement revolves around prediction of crop yield using Machine Learning Techniques. The goal of the project is to help the users choose a suitable crop to grow in order to maximize the yield and hence the profit.

The system proposed tries to overcome the drawbacks of existing systems and make

predictions by analyzing structured data. The solution we are proposing is to design a system taking into consideration the most influencing parameters to grow a crop and to get a better selection of crops which can be grown over the season. This would help reduce the difficulties faced by the farmers in selecting the crop to get high yield and thus maximize profits which in turn will reduce the suicide rates.

The system consists of two main modules:

- i. Yield Prediction Module - In this module, the user is given two options where they can either select a particular crop & get the yield for it or they can view top 5 crops that have the highest yield among all other crops.
- ii. Fertiliser Module - This module helps the user decide whether a particular time is right for using fertilisers.

### 1.3 Objectives

This project aims at predicting the crop yield at a particular weather condition and thereby recommending suitable crops for that field. It involves the following steps.

- Collect the weather data, crop yield data, soil type data and the rainfall data and merge these datasets in a structured form and clean the data. Data Cleaning is done to remove inaccurate, incomplete and unreasonable data that increases the quality of the data and hence the overall productivity.
- Perform Exploratory Data Analysis (EDA) that helps in analyzing the complete dataset and summarizing the main characteristics. It is used to discover patterns, spot anomalies and to get graphical representations of various attributes. Most importantly, it tells us the importance of each attribute, the dependence of each attribute on the class attribute and other crucial information.
- Divide the analysed crop data into training and testing sets and train the model using the training data to predict the crop yield for given inputs.
- Compare various Algorithms by passing the analysed dataset through them and calculating the error rate and accuracy for each. Choose the algorithm with the highest accuracy and lowest error rate.
- Implement a system in the form of a mobile application and integrate the algorithm at the

- back end.
- Test the implemented system to check for accuracy and failures.

## 1.4 Scope of the Project

Integrating farming and machine learning, we can lead to further advancements in agriculture by maximizing yield and optimizing the use of resources involved. Previous year's production data is an essential element for predicting the current yield.

The goal of this project is to help the farmers by combining agriculture and technology. The end result is an application that is available on the web as well as mobile. The application has the following features:

- i. Login/Register: The user can register themselves by providing a username of their choice and a password. After the registration, they can login to use the application further and view all the options that are provided to them.
- ii. Yield Prediction: This is one of the modules available in the application that enables the user to view the yield predictions of crops. The user is given two choices here:
  - 'I know what to plant': This option is for those users who already have a crop in their mind that they want to grow. When chosen, the user will be given choices of crops that they must select along with other inputs .i.e., Area and the soil type. After processing the inputs, the application will return the predicted yield on the user's screen.
  - 'Yet to decide the crop': This option is when the user is not sure between some crops or has no crop in mind. The user has to input the soil type and the area. The input is again processed at the back end by the modelled algorithm and the predicted yield is returned to the user.
- iii. Fertiliser: This is the second module available. The functionality that this module provides revolves around whether using fertilisers at a certain point of time would be recommended or not. As farmers mostly use water soluble fertilisers, it is important that it doesn't rain for 14-15 days after they use the fertilisers as they may wash off with rain and the use of the fertilisers will go in vain.
- iv. Sign Out: The user may sign out at the end which will take them back to the login/register page.

Agriculture is one of the main sources of income in India and there is an enormous need to



maintain agricultural sustainability with the increasing rate of farmer suicides. Hence it is a significant contribution towards the economic and agricultural welfare of the countries across the world.

## 1.5 Methodology

The system uses machine learning to make predictions of the crop and Python as the programming language since Python has been accepted widely as a language for experimenting in the machine learning area. Machine learning uses historical data and information to gain experiences and generate a trained model by training it with the data. This model then makes output predictions. The better the collection of dataset, the better will be the accuracy of the classifier. It has been observed that machine learning methods such as regression and classification perform better than various statistical models [2].

Crop production is completely dependent upon geographical factors such as soil chemical composition, rainfall, terrain, soil type, temperature etc. These factors play a major role in increasing crop yield. Also, market conditions affect the crop(s) to be grown to gain maximum benefit. We need to consider all the factors altogether to predict the yield.

Hence, using Machine Learning techniques in the agricultural field, we build a system that uses machine learning to make predictions of the production of crops by studying the factors such as rainfall, temperature, area, season, etc.

### 1.5.1 Machine Learning

Machine Learning is undeniably one of the most influential and powerful technologies in today's world. Machine learning is a tool for turning information into knowledge. In the past 50 years, there has been an explosion of data [10]. This mass of data is useless; we analyse it and find the patterns hidden within. Machine learning techniques are used to automatically find the valuable underlying patterns within complex data that we would otherwise struggle to discover. The hidden patterns and knowledge about a problem can be used to predict future events and perform all kinds of complex decision making. To learn the rules governing a phenomenon, machines have to go through a learning process, trying different rules and learning from how well they perform. Hence, why it's known as Machine Learning.

*“Traditionally, software engineering combined human created rules with data to create answers to a problem. Instead, machine learning uses data and answers to discover the rules behind a problem.” - Chollet, 2017*

### 1.5.1.1 Basic Terminology

- Dataset: A set of data examples, which contain features important to solving the problem.
- Features: Important pieces of data that help us understand a problem. These are fed into a Machine Learning algorithm to help it learn.
- Model: The representation (internal model) of a phenomenon that a Machine Learning algorithm has learnt. It learns this from the data it is shown during training. The model is the output you get after training an algorithm. For example, a decision tree algorithm would be trained and produce a decision tree model.

### 1.5.1.2 Types of Machine Learning

There are multiple forms of Machine Learning; supervised, unsupervised, semi-supervised and reinforcement learning. Each form of Machine Learning has differing approaches, but they all follow the same underlying process and theory.

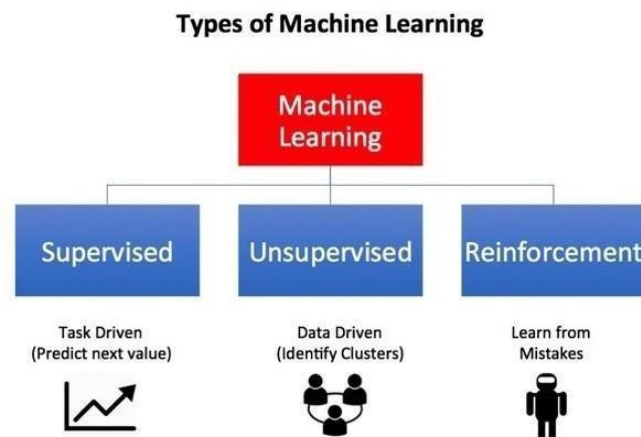


Fig 1.1 Types of Machine Learning

- **Supervised Learning:** It is the most popular paradigm for machine learning. Given data in the form of examples with labels, we can feed a learning algorithm these example-label pairs one by one, allowing the algorithm to predict the label for each example, and giving it feedback as to whether it predicted the right answer or not. Over time, the algorithm will learn to approximate the exact nature of the relationship between examples and their labels. When fully-trained, the supervised learning algorithm will be able to observe a new, never-before-seen example and predict a good label for it.

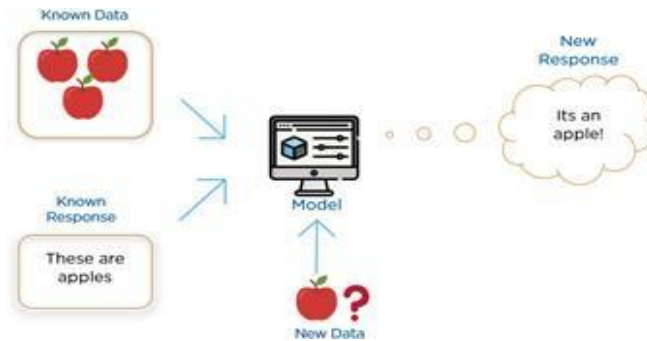


Fig 1.2 Supervised Learning

- **Unsupervised learning:** It is very much the opposite of supervised learning. It features no labels. Instead, the algorithm would be fed a lot of data and given the tools to understand the properties of the data. From there, it can learn to group, cluster, and organize the data in a way such that a human can come in and make sense of the newly organized data. Because unsupervised learning is based upon the data and its properties, we can say that unsupervised learning is data- driven. The outcomes from an unsupervised learning task are controlled by the data and the way it's formatted.

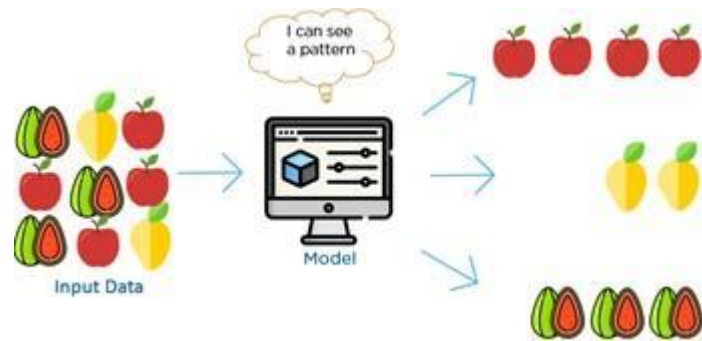


Fig 1.3 Unsupervised Learning

- **Reinforcement learning:** It is fairly different when compared to supervised and unsupervised learning. Reinforcement learning is very behaviour driven. It has influences from the fields of neuroscience and psychology. For any reinforcement learning problem, we need an agent and an environment as well as a way to connect the two through a feedback loop. To connect the agent to the environment, we give it a set of actions that it can take that affect the environment. To connect the environment to the agent, we have it continually issue two signals to the agent: an updated state and a reward (our reinforcement signal for behaviour).

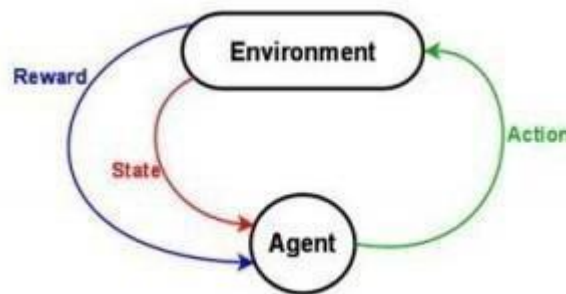


Fig 1.4 Reinforcement Learning

### 1.5.1.3 Basic Process

- Data Collection:** Collect the data that the algorithm will learn from.
- Data Preparation:** Format and engineer the data into the optimal format, extracting important features and performing dimensionality reduction.

- iii. **Training:** Also known as the fitting stage, this is where the Machine Learning algorithm actually learns by showing it the data that has been collected and prepared.
- iv. **Evaluation:** Test the model to see how well it performs.
- v. **Tuning:** Fine tune the model to maximize its performance.

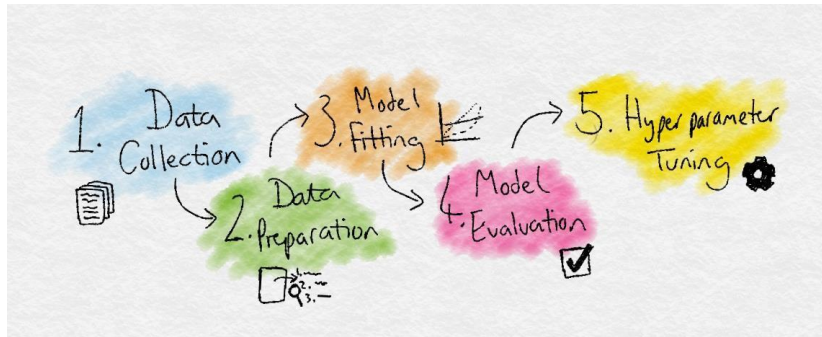


Fig 1.5 General Process

## 1.5.2 Datasets

Machine Learning depends heavily on data. It's the most crucial aspect that makes algorithm training possible. It uses historical data and information to gain experiences. The better the collection of the dataset, the better will be the accuracy.

The first step is Data Collection. For this project, we require two datasets. One for modelling the yield prediction algorithm and other for predicting weather .i.e. Average Rainfall and Average Temperature. These two parameters are predicted so as to be used as inputs for predicting the crop yield. The sources of our datasets are: '<https://en.tutiempo.net/>' for weather data and '<https://www.kaggle.com/srinivas1/agriculture-crops-production-in-india>' for crop yield data.

The yield prediction module dataset requires the following columns: State, District, Crop, Season, Average Temperature, Average Rainfall, Soil Type, Area and Production as these are the major factors that crops depend on. 'Production' is the dependent variable or the class variable. There are eight independent variables and 1 dependent variable. We achieved this by merging the datasets. The datasets were merged taking the location as the common attribute in both. We are considering only two states here, Maharashtra & Karnataka as the suicide rates in farmers in these two States were found to be very high.

1	Crop	SoilType
2	Maize	Sandy
3	Arhar/Tur	Loamy
4	Bajra	Black
5	Gram	Loamy
6	Jowar	Loamy
7	Moong(Green Gram)	Loamy
8	Pulses total	Loamy
9	Ragi	Sandy
10	Rice	Loamy

Fig 1.6 Soil and Crop data sample

1	Year	Season	Avg Rainfall(mm)	Avg Temperature
2	1997	Rabi	42.35	27.7
3	1998	Rabi	46.2	27.8
4	1999	Rabi	44.4	27.7
5	2000	Rabi	15.42	27.6
6	2001	Rabi	34.02	27.3
7	2002	Rabi	10.97	27.7
8	2003	Rabi	8.47	27.5
9	2004	Rabi	12.57	27
10	2005	Rabi	16.57	27.5

Fig 1.7 Rain and Temperature data sample

The figures 1.6 and 1.7 are small examples of the dataset before merging to achieve the main dataset which is Fig 1.8 Merged Dataset Sample for Crop Yield Prediction

1	State	District	Year	Season	Crop	Area	Production	SoilType	Avg Rainfall	Avg Temperature
2	Maharashtra	AHMEDNAGAR	1997	Kharif	Arhar/Tur	17600	6300	Loamy	184.63	22.6
3	Maharashtra	AKOLA	1997	Kharif	Arhar/Tur	81200	64400	Loamy	184.63	22.6
4	Maharashtra	AMRAVATI	1997	Kharif	Arhar/Tur	83400	61300	Loamy	184.63	22.6
5	Maharashtra	AURANGABAD	1997	Kharif	Arhar/Tur	37100	3700	Loamy	184.63	22.6
6	Maharashtra	BEED	1997	Kharif	Arhar/Tur	44200	7200	Loamy	184.63	22.6
7	Maharashtra	BHANDARA	1997	Kharif	Arhar/Tur	10200	2700	Loamy	184.63	22.6
8	Maharashtra	BULDHANA	1997	Kharif	Arhar/Tur	63000	29700	Loamy	184.63	22.6
9	Maharashtra	CHANDRAPUR	1997	Kharif	Arhar/Tur	30300	9700	Loamy	184.63	22.6
10	Maharashtra	DHULE	1997	Kharif	Arhar/Tur	29400	18500	Loamy	184.63	22.6
11	Maharashtra	GADCHIROLI	1997	Kharif	Arhar/Tur	2100	600	Loamy	184.63	22.6
12	Maharashtra	JALGAON	1997	Kharif	Arhar/Tur	24100	7300	Loamy	184.63	22.6
13	Maharashtra	JALNA	1997	Kharif	Arhar/Tur	37700	5600	Loamy	184.63	22.6
14	Maharashtra	KOLHAPUR	1997	Kharif	Arhar/Tur	3700	900	Loamy	184.63	22.6
15	Maharashtra	LATUR	1997	Kharif	Arhar/Tur	65000	2800	Loamy	184.63	22.6
16	Maharashtra	NAGPUR	1997	Kharif	Arhar/Tur	51900	2500	Loamy	184.63	22.6
17	Maharashtra	NANDED	1997	Kharif	Arhar/Tur	46000	6000	Loamy	184.63	22.6
18	Maharashtra	NASHIK	1997	Kharif	Arhar/Tur	8800	4300	Loamy	184.63	22.6
19	Maharashtra	OSMANABAD	1997	Kharif	Arhar/Tur	69200	2900	Loamy	184.63	22.6
20	Maharashtra	PARBHANI	1997	Kharif	Arhar/Tur	80000	4000	Loamy	184.63	22.6

Fig 1.8 Merged Dataset

State	District	Year	Season	Avg Rainfall	Avg Temperature
Karnataka	BAGALKOT	1998	Kharif	816.27	20
Karnataka	BANGALORE RURAL	1998	Kharif	816.27	20
Karnataka	BELGAUM	1998	Kharif	816.27	20
Karnataka	BELLARY	1998	Kharif	816.27	20
Karnataka	BENGALURU URBAN	1998	Kharif	816.27	20
Karnataka	BIDAR	1998	Kharif	816.27	20
Karnataka	BIJAPUR	1998	Kharif	816.27	20
Karnataka	CHAMARAJANAGAR	1998	Kharif	816.27	20
Karnataka	CHIKMAGALUR	1998	Kharif	816.27	20
Karnataka	CHITRADURGA	1998	Kharif	816.27	20
Karnataka	DAVANGERE	1998	Kharif	816.27	20
Karnataka	DHARWAD	1998	Kharif	816.27	20
Karnataka	GADAG	1998	Kharif	816.27	20
Karnataka	GULBARGA	1998	Kharif	816.27	20

Fig 1.9 Merged Dataset for Weather Prediction

### 1.5.3 Exploratory Data Analysis

It is an approach to analysing datasets to summarize their main characteristics, often with visual methods. It is also about knowing your data, gaining a certain amount of familiarity with the data, before one starts to extract insights from it. The idea is to spend less time coding and focus more on the analysis of data itself. After the data has been collected, it undergoes some processing before being cleaned and EDA is then performed. After EDA, go back to processing and cleaning of data, i.e., this can be an iterative process. Subsequently, use the cleaned dataset and knowledge from EDA to perform modelling and reporting. Exploratory data analysis is generally cross-classified in two ways. First, each method is either non-graphical or graphical. And second, each method is either univariate or multivariate (usually just bivariate). It is a good practice to understand the data first and try to gather as many insights from it. EDA is all about making sense of data in hand. EDA can give us the following:

- Preview data
- Check total number of entries and column types using in built functions. It is a good practice to know the columns and their corresponding data types.
- Check any null values.
- Check duplicate entries

- Plot distribution of numeric data (univariate and pairwise joint distribution)
- Plot count distribution of categorical data.

Using various built in functions, we can get an insight of the number of values in each column which can give us information about the null values or the duplicate data. We can also find the mean, standard deviation, minimum value and the maximum value. This is the basic procedure of EDA. To get a better insight of the data that is being used, we can plot graphs like the correlation matrix which is one of the most important concept which gives us a lot of information about how variables (columns) are related to each other and the impact each of them have on the other. A few other graphs like box plot and distribution graphs can be plotted too.

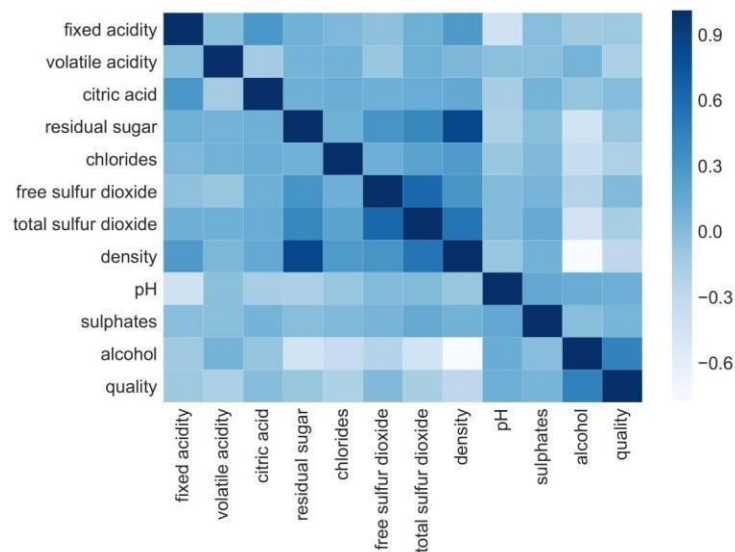


Fig 1.10 Correlation Matrix Example

Dark shades represent positive correlation while lighter shades represent negative correlation.

Hence, we can make the following inferences from the above example:

- Here we can infer that “density” has strong positive correlation with “residual sugar” whereas it has strong negative correlation with “alcohol”.
- “Free sulphur dioxide” and “citric acid” have almost no correlation with “quality”.
- Since correlation is zero we can infer there is no linear relationship between these two predictors.

**EDA Performed:**



```
print("Columns : " , df_data.shape[1])
print("\nFeatures : \n" , df_data.columns.tolist())
print("\nMissing values : " , df_data.isnull().sum().values.sum())
print("\nUnique values : \n", df_data.nunique())
print("\nInfo : \n")
print(df_data.info())
```

Fig 1.11 EDA Code

```
df_data.describe()
```

	Year	Area	Production	Avg Rainfall(mm)	Avg Temperature
<b>count</b>	11739.000000	11739.000000	1.161300e+04	11739.000000	10395.000000
<b>mean</b>	2005.280262	26318.246443	5.249072e+04	128.414517	24.936479
<b>std</b>	5.029136	60475.364022	4.157544e+05	97.293491	3.804528
<b>min</b>	1997.000000	1.000000	0.000000e+00	1.100000	19.600000
<b>25%</b>	2001.000000	500.000000	2.890000e+02	16.570000	22.300000
<b>50%</b>	2005.000000	3500.000000	2.400000e+03	166.200000	23.200000
<b>75%</b>	2010.000000	24200.000000	1.990000e+04	208.950000	27.600000
<b>max</b>	2014.000000	726300.000000	1.600010e+07	315.100000	39.900000

Fig 1.12 Output of describe function

Fig 1.10 shows the simple code written in python to perform the initial steps in EDA .i.e., finding the number of columns, the features, the missing values, etc. Fig 1.11 shows the details of each attribute in tabular form which helps in getting a deeper insight of the attributes.

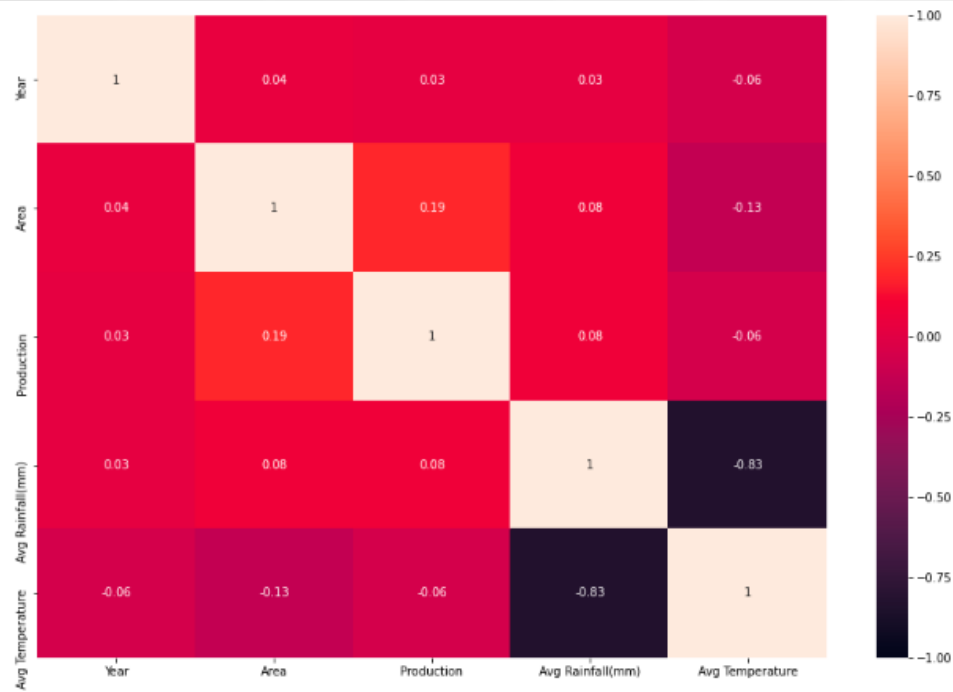


Fig 1.13 Correlation Matrix of the Proposed System

Fig 1.12 shows how each attribute is related to the other .i.e., the correlation matrix.

### 1.5.4 Algorithms Used

Machine Learning offers a wide range of algorithms to choose from. These are usually divided into classification, regression, clustering and association. Classification and regression algorithms come under supervised learning while clustering and association comes under unsupervised learning.

- Classification: A classification problem is when the output variable is a category, such as “red” or “blue” or “disease” and “no disease”. Example: Decision Trees
- Regression: A regression problem is when the output variable is a real value, such as “dollars” or “weight”. Example: Linear Regression
- Clustering: A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behaviour. Example: k means clustering
- Association: An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.

## Example: Apriori Algorithm

A few algorithms can come under multiple types. Considering the problem statement and the desired output of the project, the most suitable type of algorithm would come under regression. Before choosing an algorithm and working with it further, many algorithms were explored and the error rates and accuracy were checked for each. The table 1.1 summarizes the various algorithms that were explored [3].

**Table 1.1 Summary of the Approaches**

Sno.	Algorithm	Accuracy
1.	Artificial Neural Networks (ANN)	86%
2.	Support Vector Machine (SVM)	75%
3.	Multivariate linear Regression	60%
4.	Random Forest	95%
5.	K Nearest Neighbours (KNN)	90%

From the above table, we can conclude that the Random Forest Algorithm gives the best accuracy for our dataset.

- ANN vs. Random Forest: Random Forest is less computationally expensive and does not require a GPU to finish training. A random forest can give you a different interpretation of a decision tree but with better performance. Neural Networks will require much more data than an everyday person might have on hand to actually be effective. The neural network will simply decimate the interpretability of your Features to the point where it becomes meaningless for the sake of performance.
- SVM vs. Random Forest: Random forests are probably THE “worry-free” approach. There are no real hyper parameters to tune (maybe except for the number of trees; typically, the more trees we have the better). On the contrary, there are a lot of knobs to be turned in SVMs: Choosing the “right” kernel, regularization penalties, the slack variable, etc. Random forests are much simpler to train and easier to find a good, robust model. In SVMs, we typically need to do a fair amount of parameter tuning, and in addition to that, the computational cost grows linearly with the number of classes as well.
- Linear Regression vs. Random Forest: Random forests very often outperform linear

regression. Random forests fit data better from the get-go without transforms. They're more forgiving in almost every way. You don't need to scale your data, you don't need to do any monotonic transformations (log, etc.). You often don't even need to remove outliers. You can throw in categorical features, and it'll automatically partition the data if it aids the fit. You don't have to spend any time generating interaction terms. And perhaps most important: in most cases, it'll probably be notably more accurate.

- KNN vs. Random Forest: Random Forest is faster due to KNN's expensive real time execution. 'K' should be wisely selected while there is no such decision to be made in Random Forest. KNN has large computation cost during runtime if sample size is large.

### 1.5.5 Random Forest

Random forest is a flexible, easy to use machine learning algorithm that produces, even without hyper-parameter tuning, a great result most of the time. It is also one of the most used algorithms, because of its simplicity and diversity. It can be used for both classification and regression tasks. Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction. One big advantage of random forest is that it can be used for both classification and regression problems, which form the majority of current machine learning systems. Another great quality of the random forest algorithm is that it is very easy to measure the relative importance of each feature on the prediction. Sklearn provides a great tool for this that measures a feature's importance by looking at how much the tree nodes that use that feature reduce impurity across all trees in the forest. It computes this score automatically for each feature after training and scales the results so the sum of all importance is equal to one.

The hyper parameters in random forest are either used to increase the predictive power of the model or to make the model faster. Python offers some built in random Forest functions which have the following hyper parameters.

#### **i To increase the predictive power:**

- Firstly, there is the `n_estimators` hyper parameter, which is just the number of trees the algorithm builds before taking the maximum voting or taking the averages of predictions.
- Another important hyper parameter is `max_features`, which is the maximum number of features random forest considers to split a node.
- The last important hyper parameter is `min_sample_leaf`. This determines the

minimum number of leafs required to split an internal node.

**ii To Increase the model's speed:**

- The `n_jobs` hyper parameter tells the engine how many processors it is allowed to use. If it has a value of one, it can only use one processor. A value of “-1” means that there is no limit.
- The `random_state` hyper parameter makes the model’s output replicable. The model will always produce the same results when it has a definite value of `random_state` and if it has been given the same hyper parameters and the same training data.
- The `oob_score` (also called oob sampling), which is a random forest cross-validation method. In this sampling, about one-third of the data is not used to train the model and can be used to evaluate its performance. These samples are called the out-of-bag samples.

The working of Random Forest is as follows:

- **Step 1** – First, start with the selection of random samples from a given dataset.
- **Step 2** – Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.
- **Step 3** – In this step, voting will be performed for every predicted result.
- **Step 4** – At last, select the most voted prediction result as the final prediction result.

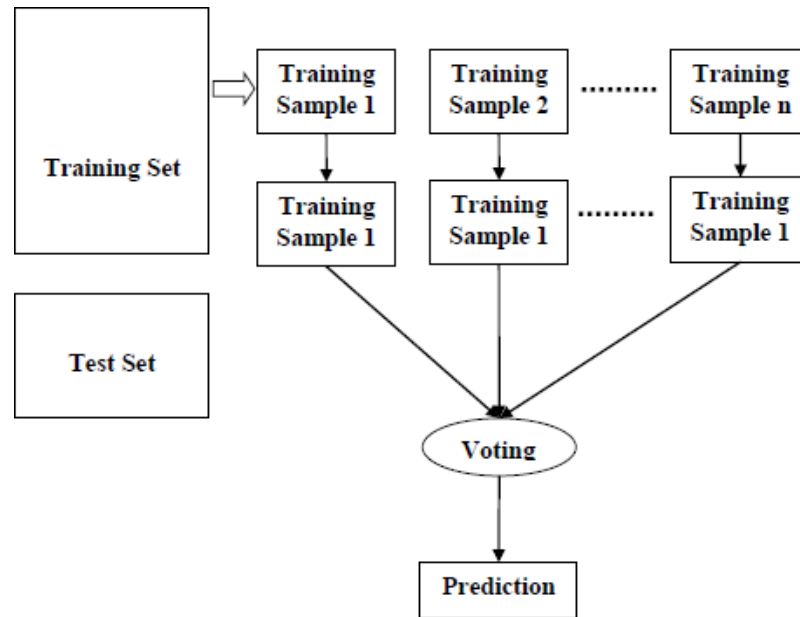


Fig 1.14 Random Forest Flow

**Another Reason for choosing Random Forest:**

The data of a particular crop was taken and passed through two algorithms .i.e., Random Forest and another Algorithm that is said to give best results for that crop. The accuracy achieved in both the algorithms were compared. Rice and Groundnut were chosen based on the research papers that were found.

- Rice: According to the paper, the best algorithm for rice yield prediction is Linear Regression [14]. After running both the algorithms, we found a very high difference between actual value and predicted value in Linear Regression while Random Forest continued to maintain an accuracy of 90+.
- Groundnut: The Research Paper stated that KNN works best for Groundnut yield prediction [13]. On running the algorithms, we did not find much difference in the results in both the algorithms.

Hence, we can conclude that Random Forest can be used as a general algorithm which gives a considerably high accuracy with Good predictions.

## CHAPTER 2

### LITERATURE SURVEY

Literature Survey is a systematic and thorough search of all types of published literature as well as other sources including dissertation, these in order to identify as many items as possible that are relevant to a particular topic.

Predicting agricultural products plays a very important role in agriculture. It helps in increasing net produce, better planning and gaining more profits. To achieve better results, we studied a few research papers related to our project topic.

#### 2.1 Machine learning approach for forecasting crop yield based on climatic parameters

**Authors:** S.Veenadhari, Dr. Bharat Misra & Dr. CD Singh

**Publication:** International Conference on Computer Communication and Informatics (ICCCI - 2014), Jan, 2014

In this paper, the study was aimed to develop a website for finding out the influence of climatic parameters on crop production in selected districts of Madhya Pradesh. The selection of districts has been made based on the area under that particular crop. Based on this criteria first top five districts in which the selected crop area is maximum were selected. The crops selected in the study were based on the predominant crops in the selected district. The selected crop included: Soybean, Maize, Paddy and Wheat. The yield of these crops was tabulated for continuous 20 years by collecting the information from secondary sources. Similarly for the corresponding years climatic parameters such as Rainfall, Maximum & Minimum temperature, Potential Evapotranspiration, Cloud cover, Wet day frequency were also collected from the secondary sources. The methodology adopted for analysis includes for values above the threshold were considered as one child and the remaining as another child. It also handles missing attribute values. In pseudo code, the general algorithm for building decision trees is:

- For each attribute  $a$  : find the normalized information gain from splitting on  $a$
- Let  $a_{best}$  be the attribute with the highest normalized information gain
- Create a decision node that splits on  $a_{best}$
- Recurse on the sublists obtained by splitting on  $a_{best}$ , and add those nodes as children of

node.

In this approach to relevance analysis, they computed the information gain for each of the attributes defining the samples in  $S$ . The attribute with the highest information gain was considered the most discriminating attribute of the given set. By computing the information gain for each attribute, they obtained a ranking of the attributes. This ranking can be used for relevance analysis to select the attributes to be used in concept description. A web based software has been developed in C# language in .net platform. The backend used is sql server 2008.

Conclusions made were that Out of 20 years of data the predictions were correct in 18 years and were incorrect in two years indicating the prediction accuracy of the developed model at 90% in case of soybean in Dewas district. The prediction accuracy of the developed model varied from 76 to 90% for the selected crops and selected districts. Based on these observations the overall prediction accuracy of the developed model is 82.00%.

This paper focuses on relevance approach analysis to ensure accurate prediction. This is achieved by calculating information gain of each attribute and comparing them. However, it does not include the analysis of other supervised machine learning algorithms like random forest and linear regression.

## 2.2 Prediction of Crop Yield Using Machine Learning

**Author:** Rushika Ghadge, Juilee Kulkarni, Pooja More, Sachee Nene, Priya R L

**Publication:** International Research Journal of Engineering and Technology (IRJET) Volume 05, Issue 02, Feb-2018

This paper states, most of the existing systems are hardware based which makes them expensive and difficult to maintain and lack to give accurate results. Some systems suggest crop sequence depending on yield rate and market price. In this paper, the system proposed tries to overcome these drawbacks and predicts crops by analyzing structured data. Being a totally software solution, it does not allow maintenance factor to be considered much. Also the accuracy level would be high as compared to hardware based solutions, because components like soil composition, soil type, pH value, weather conditions all come into picture during the prediction process.

It can be achieved using unsupervised and supervised learning algorithms, like Kohonen



Self Organizing Map (Kohonen's SOM) and BPN (Back Propagation Network). Dataset will then be trained by learning networks. It compares the accuracy obtained by different network learning techniques and the most accurate result will be delivered to the end user.

This paper proposes a system that will check soil quality and predict the crop yield accordingly along with it providing fertiliser recommendation if needed depending upon the quality of soil. The system takes inputs pH value and location from the user and result processing is done by two controllers. The result of the controller 1 and controller 2 are compared with a predefined "nutrients" data store. These compared results are supplied to controller 3 wherein the combination of the above results and the predefined data set present in the crop data store is compared. Finally, the results are displayed in the form of bar graphs along with accuracy percentage.

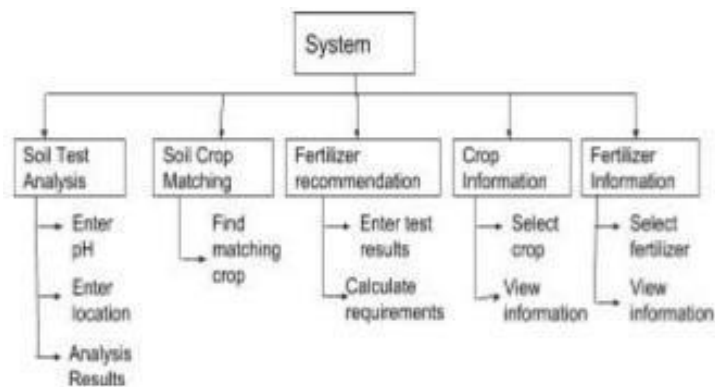


Fig 2.1 Modular diagram

The Fig 2.1, represents the architecture of the system. The system includes the following modules: soil test analysis, soil crop matching, fertilizer recommendation, crop information and fertilizer information. Each of the modules performs a specific function i.e., soil test analysis module on taking the ph and location as inputs analyses the soil and is categorized using the soil crop matching module. The fertilizer recommendation module analyses the test results obtained from the soil test analysis and soil crop matching modules and calculates the requirements. The crop information module on taking the crop as input returns the information of the crop i.e., the production predicted. The fertilizer information module on taking the fertilizer as input returns the information of the fertilizer.

The paper concludes that the system uses supervised and unsupervised Machine learning

algorithms and gives best results based on accuracy. The results of the two algorithms will be compared and the one giving the best and accurate output will be selected. Thus the system will help reduce the difficulties faced by the farmers and stop them from attempting suicides. It will act as a medium to provide the farmers efficient information required to get high yield and thus maximize profits which in turn will reduce the suicide rates and lessen his difficulties.

This paper includes the study of soil as well as fertilizers. The system proposed predicts the yield for the given crop. It also returns the information of the given fertilizer and also recommends one after calculating the requirements based on the soil properties and location. However, this paper does not take into consideration the climatic conditions of the location which have quite a large impact on the yield of the crops.

## 2.3 Predicting Yield of the Crop Using Machine Learning

### Algorithm

**Author:** P.Priya, U.Muthaiah & M.Balamurugan

**Publication:** International Journal of Engineering Sciences & Research Technology (IJESRT), April, 2018

This paper uses R programming with Machine Learning techniques. R is the leading tool for statistics, data analysis, and machine learning. It is more than a statistical package; it's a programming language, so you can create your own objects, functions, and packages. It's platform-independent, so it can be used on any operating system and it's free. R programs explicitly document the steps of our analysis and make it easy to reproduce and/or update analysis, which means it can quickly try many ideas and/or correct issues. All the datasets used in the research were sourced from the openly accessible records of the Indian Government. This was sourced for the years 1997 to 2013 for different seasons like Kharif and Rabi of rice production. From the vast initial dataset, only a limited number of important factors which have the highest impact on agricultural yield were selected for the present research. The dataset contains the following parameters: rainfall, season, and temperature and crop production. This paper also compares the two machine learning algorithms: decision trees and Random Forest.

- **Decision Tree:** The Decision tree classifiers uses greedy approach hence an attribute chosen at the first step can't be used anymore which can give better classification if used in later steps. Also it over fits the training data which can give poor results for unseen

data. So, to overcome this limitation ensemble model is used. In ensemble models results from different models are combined. The result obtained from an ensemble model is usually better than the result from any one of individual models.

- **Random Forest:** Random Forests is an ensemble classifier which uses many decision tree models to predict the result. A different subset of training data is selected, with replacement to train each tree. A collection of trees is a forest, and the trees are being trained on subsets which are being selected at random, hence random forests. This can be used for classification and regression problems. Class assignment is made by the number of votes from all the trees and for regression the average of the results is used.

In this paper, the procedure they followed was as given below.

- Split the loaded data sets into two sets such as training data and test data in the split ratio of 67% and 33%.
- Then calculate Mean and Standard Deviation for needed tuples and then summarize the data sets. Compare the summarized data list and the original data sets & calculate the probability.
- Based on the result the largest probability produced is taken for prediction. The accuracy can be predicted by comparing the resultant class value with the test data set. The accuracy can range from 0% to 100%.

The paper concludes that the Results show that we can attain an accurate crop yield prediction using the Random Forest algorithm. The Random Forest algorithm achieves a largest number of crop yield models with the lowest models. It is suitable for massive crop yield prediction in agricultural planning. The dataset used for modelling here includes the climatic factors as well i.e., rainfall and temperature. The author did a comparative study of decision trees and random forest algorithms. But other algorithms were not considered and the dataset includes very few attributes that would not give accurate predictions.

## 2.4 A Survey on Crop Prediction using Machine Learning

### Approach

**Authors:** Sriram Rakshith.K, Dr. Deepak.G, Rajesh M, Sudharshan K S, Vasanth S & Harish Kumar

**Publications:** International Journal for Research in Applied Science & Engineering Technology

This paper is mainly focused on the techniques and measures taken to improve farming by inculcating the technical knowledge and developments in order to make the agricultural sector more reliable and easy for the farmers by predicting the suitable crop by using Machine learning techniques by sensing parameters like-- soil, weather and market trends. Parameters considered are PH, Nitrogen-phosphate-potassium contents of soil, temperature, rainfall and humidity. They consider Artificial Neural Network, Information Fuzzy Network and other Data Mining Techniques.

1. **Artificial Neural Network:** In Neural networks it consists of input, output and hidden layers, where the neurons are the input given to the ANN and it is performed by hidden layers by some units and it is used by output layers to produce output. The accuracy of the neural networks increases as the data increases. In artificial intelligence and machine learning algorithms like ID3 and other optimizing algorithms are used in tomato crop detection. To design an expert system for Tomato crops. In maize cultivation, the machine learning techniques are used. Corn is also a popular crop and a main source of cereals along with rice genotypes which is adapted and well suited in drought situations which has to be grown under controlled situations and marginal law has to be implemented.
2. **Information Fuzzy Network:** Crop prediction and analysis is done through neural networks. The inputs are Soil moisture content, ground biomass and repository organ for Neuro-fuzzy Inference system. There are other problems like forecasting yield in a remote sensing area and goes long behind in time. The design of the algorithm is in such a way that it leaves behind a year and uses the rest of the data. The deviation is determined by comparing the yield with the one that is left out. The study conducted to consolidate the aspect of incomplete information, used the web based decision support system where the fuzzy logic advancement of an agricultural used because the effect of climate development is considered to be a major part in the field of agriculture and crop prediction in Malaysia. The detailed information about soil rainfall arrangement will be taken into consideration to bring about accomplished results.
3. **Data Mining Techniques:** Data Mining is used to analyse the collected data to give insights to make decisions .By using the information or result obtained by the data mining techniques we can reduce the risk associated with the agriculture by predicting

the crop yield more precisely to harvest. To provide efficient results the Data mining technique required large amounts of data to analyse such a way in yield prediction it requires more data that related crop yield. Weather data, soil properties and agricultural statistics etc. Data mining techniques are divided into two groups:

- Classification
- Clustering.

The paper concluded that the task of prediction of crops can be achieved using different methodologies as discussed above. We can also say that ANN's give us better, accurate predictions. Hence by using the soil, weather and market prices, we can build a model, having a workflow as shown above, that can provide accurate predictions about the crop yield suitable for a particular region. These are performed by using Artificial Neural Network (ANN).

This paper includes a combination of machine learning algorithms and data mining techniques to give more accurate results. Other machine learning algorithms like KNN, Linear Regression and SVM are not analysed.

## **2.5 Heuristic Prediction of Crop Yield using Machine Learning**

### **Technique**

**Author:** S. Pavani, Augusta Sophy Beulet P.

**Publications:** International Journal of Engineering and Advanced Technology (IJEAT) Volume-9, December 2019

The paper says, vast research has been done and several attempts are made for application of Machine learning in agricultural fields. Major challenge in agriculture is to increase the production in the farm and deliver it to the end customers with best possible price and good quality. It is found that at least 50 percent of the farm produce never reaches the end consumer due to wastage and high-end prices. Machine learning based solutions developed to solve the difficulties faced by the farmers are being discussed in this work. The real time environmental parameters of Telangana District like soil moisture, temperature, rainfall, humidity are collected and crop yield is being predicted using KNN Algorithm.

There is a profound need to raise the farmer's income and ensure sustainable growth in Telangana to reduce poverty. Prediction of the yield of the crop in advance for a particular region depending on the climatic conditions and other factors which contribute to more yield is

important, which will make farmers select the crop or to get better yield of their particular crops in the particular region. The use of data mining in rural areas has brought advantages in the field of research. This application's evaluations help farmers predict crop yields. Support Vector Machine (SVM), Regression analysis, K Nearest Neighbor (KNN), clustering and various types of techniques are used for prediction. KNN is used in this work. KNN is a language learning algorithm that is non-parametric. The aim is to use a model where information focuses are clustered in a few groups in order to predict the classification of another instance. KNN Algorithm depends on the closeness of the feature: After our training set, how firmly out of test highlights determine how we manage a given point of data. KNN can be utilized for classification- the yield or the output is a class membership (predicts a class-a discrete esteem or value). An element is distinguished by a larger proportion of its neighbours' vote, with the entity being divided among its closest neighbours to the most regular class. It can also be used for regression-output, which is the item's reward (predicts unceasing qualities). This calculation is the average (or middle) of its nearest neighbours' estimates of  $k$ . A commonly used metric for distance calculation in KNN algorithms is Euclidean distance.

This technique can be used to predict crop yield with more accuracy for years of data together. The main principle reason of the work is to predict the crop yield in prior according to the details of the factors required for good crop yield.

In this paper, the data sets of different districts of the Telangana state are collected from Telangana State Development Planning Society. The important factors that determine the crop yield are temperature, humidity, soil moisture and rainfall. These samples were taken for the month of May 2019. Among the data set available maximum of the data was used for training and the remaining data was used for testing. The machine learning technique KNN algorithm was used for prediction of crop yield.

So with respect to the application of KNN algorithm towards prediction of crop yield, the nearest neighbours of a particular point (crop yield) like temperature, humidity, rain fall and soil moisture were considered. The implementation steps were:

1. Load the data set.
2. Initialize the 'k' value.
3. For getting the anticipated class, repeat from one to all the numbers of the training data set.
4. Compute the distance between test data and each line of training information. Here the

- Euclidean distance is utilized, since it's the most prominent technique.
5. Classify the computed distances in ascending order based on distance values.
  6. Get top 'k' rows from the classified array.
  7. Get the most continuous class of these lines.
  8. Return the anticipated class.

The paper concludes that a model of machine learning to predict plant yield is proposed and gives reasonable crop yield suggestions for particular districts in Telangana. The research has been done on soil moisture, temperature, humidity and rainfall datasets of all the districts of Telangana State. By applying machine learning algorithms the model has been tested. K-NN suggests suitable accuracy in crop yield prediction. The well-constructed data set and the machine learning algorithm supports the proposed model. In future, providing other factors that greatly influence the crop yield is our concern, also more data of all these parameters of different seasons in the state will be added to make this model more accurate and efficient.

This paper includes a comparative study of the KNN, SVM and Linear Regression giving KNN as the most appropriate one with maximum accuracy. The climatic as well as soil properties are analysed to predict the yield. This paper does not include recommendation of fertilizers or crops based on the soil, climate and location.

## CHAPTER 3

### SYSTEM REQUIREMENTS SPECIFICATION

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and nonfunctional requirements, and may include a set of use cases that describe user interactions that the software must provide. It is very important in a SRS to list out the requirements and how to meet them. It helps the team to save upon their time as they are able to comprehend how are going to go about the project. Doing this also enables the team to find out about the limitations and risks early on.

A SRS can also be defined as a detailed description of a software system to be developed with its functional and non-functional requirements. It may include the use cases of how the user is going to interact with the software system. The software requirement specification document is consistent with all necessary requirements required for project development. To develop the software system we should have a clear understanding of Software system. To achieve this we need continuous communication with customers to gather all requirements.

A good SRS defines how the Software System will interact with all internal modules, hardware, and communication with other programs and human user interactions with a wide range of real life scenarios. It is very important that testers must be cleared with every detail specified in this document in order to avoid faults in test cases and its expected results.

#### **Qualities of SRS**

- Correct
- Unambiguous
- Complete
- Consistent
- Ranked for importance and/or stability
- Verifiable
- Modifiable
- Traceable





Fig 3.1 Types of Requirements in SRS

Some of the goals an SRS should achieve are to:

- Provide feedback to the customer, ensuring that the IT Company understands the issues the software system should solve and how to address those issues.
- Help to break a problem down into smaller components just by writing down the requirements.
- Speed up the testing and validation processes.
- Facilitate reviews.

### 3.1 Functional Requirements

A Functional Requirement is a description of the service that the software must offer. It describes a software system or its component. A function is nothing but inputs to the software system, its behaviour, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform. In software engineering and systems engineering, a Functional Requirement can range from the high-level abstract statement of the sender's necessity to detailed mathematical functional requirement specifications. Functional software requirements help you to capture the intended behaviour of the system.

### **Benefits of functional requirements:**

- Helps you to check whether the application is providing all the functionalities that were mentioned in the functional requirement of that application
- A functional requirement document helps you to define the functionality of a system or one of its subsystems.
- Functional requirements along with requirement analysis help identify missing requirements. They help clearly define the expected system service and behavior.
- Errors caught in the Functional requirement gathering stage are the cheapest to fix.
- Support user goals, tasks, or activities

### **3.1.1 Basic Requirements**

1. **Data collection:** The dataset used in this project is the data collected from reliable websites and merged to achieve the desired data set. The sources of our datasets are: <https://en.tutiempo.net/> for weather data and <https://www.kaggle.com/srinivas1/agriculture-crops-production-in-india> for crop yield data. It consists of names of the crops, production, area, average temperature, average rainfall (mm), season, year, name of the states and the districts. 'Production' is the dependent variable or the class variable. There are eight independent variables and 1 dependent variable.
2. **Data Preprocessing:** The purpose of preprocessing is to convert raw data into a form that fits machine learning. Structured and clean data allows a data scientist to get more precise results from an applied machine learning model. The technique includes data formatting, cleaning, and sampling. Here, data pre-processing focuses on finding the attributes with null values or invalid values and finding the relationships between various attributes as well. Data Pre-processing also helps in finding out the impact of each parameter on the target parameter. To preprocess our datasets we used EDA methodology. All the invalid and null values were handled by removing that record or giving the default value of that particular attribute based on its importance.
3. **Dataset splitting:** A dataset used for machine learning should be partitioned into two subsets — training and test sets. We split the dataset into two with a split ratio of 80% i.e., in 100 records 80 records were a part of the training set and remaining 20 records

- were a part of the test set.
4. **Model training:** After a data scientist has preprocessed the collected data and split it into train and test can proceed with a model training. This process entails “feeding” the algorithm with training data. An algorithm will process data and output a model that is able to find a target value (attribute) in new data an answer you want to get a predictive analysis. The purpose of model training is to develop a model. We trained our model using the random forest algorithm. On training the model it predicts the yield on giving the other attributes of the dataset as input.
  5. **Model evaluation and testing:** The goal of this step is to develop the simplest model able to formulate a target value fast and well enough. A data scientist can achieve this goal through model tuning. That’s the optimization of model parameters to achieve an algorithm’s best performance.

### 3.1.2 Application Requirements

1. Users must be able to register as a new user.
2. Users should be able to login if they already have an account.
3. The location inputs must be read correctly by the application.
4. The weather Prediction algorithm must give accurate prediction of the average rainfall and average temperature.
5. The user should be able to give the required inputs like soil type and area.
6. All the modules of the application must work in a proper manner.
7. The yield prediction module should provide two options. One is if the user is familiar with what crop is to be grown and other is when the user is not sure.
8. The predictions must be accurate.
9. Users must be able to access the Fertilisers module as well.
10. The fertiliser module must help the farmers decide whether to use the fertilisers or not.
11. The user must be able to logout.

### 3.2 Non-Functional Requirements

Non-Functional Requirement (NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and

other non-functional standards that are critical to the success of the software system. Failing to meet non-functional requirements can result in systems that fail to satisfy user needs. Non-functional Requirements allows you to impose constraints or restrictions on the design of the system across the various agile backlogs. Example, the site should load in 3 seconds when the number of simultaneous users are  $> 10000$ . They specify the criteria that can be used to judge the operation of a system rather than specific behaviours. They may relate to emergent system properties such as reliability, response time and store occupancy. Non-functional requirements arise through the user needs, because of budget constraints, organizational policies, the need for interoperability with other software and hardware systems or because of external factors such as:- Product Requirements, Organizational Requirements, User Requirements, Basic Operational Requirement, etc.

**Benefits of Non Functional Requirements:**

- The nonfunctional requirements ensure the software system follows legal and compliance rules.
- They ensure the reliability, availability, and performance of the software system.
- They ensure good user experience and ease of operating the software.
- They help in formulating security policy of the software system.

**1.2.1 Requirements**

1. The access permissions for system data may only be changed by the system's data administrator.
2. Passwords shall never be viewable at the point of entry or at any other time.
3. Apps should be able to adapt themselves to increased usage or be able to handle more data as time progresses.
4. Application should be responsive to the user Input or to any external interrupt which is of highest priority and return back to the same state.
5. Users should be able to understand the flow of the App easily .I.e. users should be able to use the App without any guideline or help from experts/manuals.
6. All the app data should be secured and be encrypted with minimum needs so that it's protected.
7. There should be a common plan where the user can access the application to install and look for regular updates to give feedback.

8. The application should be able to render it's layout to different screen sizes. Along with automatic adjustment of Font size and image rendering.
9. The application should run at a speed that is desirable by the users. A slow application can lead to frustration and hence, will not be preferred over other faster applications.
10. The application must be stable. It should never crash or force close in the case of many users using it simultaneously.
11. The application must be easy to maintain.
12. It must be user-friendly. Having a user-friendly application is of key importance for the success of the application.

### 3.3 Hardware Requirements

The hardware requirements include the requirements specification of the physical computer resources for a system to work efficiently. The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. The Hardware Requirements are listed below:

1. **Processor:** A processor is an integrated electronic circuit that performs the calculations that run a computer. A processor performs arithmetical, logical, input/output (I/O) and other basic instructions that are passed from an operating system (OS). Most other processes are dependent on the operations of a processor. A minimum 1 GHz processor should be used, although we would recommend S2GHz or more. A processor includes an arithmetical logic and control unit (CU), which measures capability in terms of the following:
  - Ability to process instructions at a given time
  - Maximum number of bits/instructions
  - Relative clock speed



Fig 3.2 Processor

The proposed system requires a 2.4 GHz processor or higher.

**2. Ethernet connection (LAN) OR a wireless adapter (Wi-Fi):** Wi-Fi is a family of radio technologies that is commonly used for the wireless local area networking (WLAN) of devices which is based around the IEEE 802.11 family of standards. Devices that can use Wi-Fi technologies include desktops and laptops, smartphones and tablets, TV's and printers, digital audio players, digital cameras, cars and drones. Compatible devices can connect to each other over Wi- Fi through a wireless access point as well as to connected Ethernet devices and may use it to access the Internet. Such an access point (or hotspot) has a range of about 20 meters (66 feet) indoors and a greater range outdoors. Hotspot coverage can be as small as a single room with walls that block radio waves, or as large as many square kilometres achieved by using multiple overlapping access points.



Fig 3.3 Wi-Fi

**3. Hard Drive:** A hard drive is an electro-mechanical data storage device that uses magnetic storage to store and retrieve digital information using one or more rigid rapidly rotating disks, commonly known as platters, coated with magnetic material. The platters are paired with magnetic heads, usually arranged on a moving actuator arm, which reads and writes data to the platter surfaces. Data is accessed in a random-access manner, meaning that individual blocks of

data can be stored or retrieved in any order and not only sequentially. HDDs are a type of non-volatile storage, retaining stored data even when powered off. 32 GB or higher is recommended for the proposed system.



Fig 3.4 Hard Drive

**4. Memory (RAM):** Random-access memory (RAM) is a form of computer data storage that stores data and machine code currently being used. A random-access memory device allows data items to be read or written in almost the same amount of time irrespective of the physical location of data inside the memory. In today's technology, random-access memory takes the form of integrated chips. RAM is normally associated with volatile types of memory (such as DRAM modules), where stored information is lost if power is removed, although non-volatile RAM has also been developed. A minimum of 2 GB RAM is recommended for the proposed system.



Fig 3.5 RAM

### 3.4 Software Requirements

The software requirements are description of features and functionalities of the target system. Requirements convey the expectations of users from the software product. The requirements can be obvious or hidden, known or unknown, expected or unexpected from client's point of view.

**1. Jupyter Notebook:** The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use. The Jupyter Notebook combines three components:

- **The notebook web application:** An interactive web application for writing and running code interactively and authoring notebook documents.
- **Kernels:** Separate processes started by the notebook web application that runs users' code in a given language and returns output back to the notebook web application. The kernel also handles things like computations for interactive widgets, tab completion and introspection.
- **Notebook documents:** Self- contained documents that contain a representation of all content visible in the notebook web application, including inputs and outputs of the computations, narrative text, equations, images, and rich media representations of objects. Each notebook document has its own kernel.



Fig 3.6 Jupyter Notebook



**2. Python:** It is an object-oriented, high-level programming language with integrated dynamic semantics primarily for web and app development. It is extremely attractive in the field of Rapid Application Development because it offers dynamic typing and dynamic binding options. Python is relatively simple, so it's easy to learn since it requires a unique syntax that focuses on readability. Developers can read and translate Python code much easier than other languages. In turn, this reduces the cost of program maintenance and development because it allows teams to work collaboratively without significant language and experience barriers. Additionally, Python supports the use of modules and a package, which means that programs can be designed in a modular style and code can be reused across a variety of projects.



Fig 3.7 Python

- 2. Pycharm:** PyCharm is the most popular IDE for Python, and includes great features such as excellent code completion and inspection with advanced debugger and support for web programming and various frameworks. The intelligent code editor provided by PyCharm enables programmers to write high quality Python code. The editor enables programmers to read code easily through colour schemes, insert indents on new lines automatically, pick the appropriate coding style, and avail context-aware code completion suggestions. At the same time, the programmers can also use the editor to expand a code block to an expression or logical block, avail code snippets, format the code base, identify errors and misspellings, detect duplicate code, and auto-generate code. PyCharm offers some of the best features to its users and developers in the following aspects
- Code completion and inspection
  - Advanced debugging

- Support for web programming and frameworks such as Django and Flask



Fig 3.8 PyCharm

**3. Ionic Development Tools:** Ionic makes it easy to build high-performance mobile and Progressive Web Apps (or PWAs) that look and feel beautiful on any platform or device. Ionic's open source Framework and developer-friendly tools and services power apps for some of the world's best-known brands - from highly successful consumer apps like Sworkit, Untappd and Dow Jones MarketWatch, to mission-critical apps supporting Nationwide, Amtrak, and NASA. The open source Ionic Framework features a rich library of front-end building blocks and UI components that make it easy to design beautiful, high-performance mobile and Progressive Web Apps (or PWAs) using web technologies like HTML, CSS, and JavaScript.

*"Write Once, Run Anywhere"*



Fig 3.9 Ionic

**4. Flask:** Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja and has become one of the most popular Python web application frameworks. It offers suggestions, but doesn't enforce any dependencies or project layout. It is up to the developer to choose the tools and libraries they want to use. There are many extensions provided by the community that make adding new functionality easy.



Fig 3.10 Flask

## CHAPTER 4

### SYSTEM ANALYSIS AND DESIGN

Systems development is a systematic process which includes phases such as planning, analysis, design, deployment, and maintenance.

System Analysis is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components. System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose. Analysis specifies what the system should do.

System Design is a process of planning a new business system or replacing an existing system by defining its components or modules to satisfy the specific requirements. Before planning, you need to understand the old system thoroughly and determine how computers can best be used in order to operate efficiently. System Design focuses on how to accomplish the objective of the system.

#### 4.1 System Architecture

Architecture diagrams can help system designers and developers visualize the high-level, overall structure of their system or application for the purpose of ensuring the system meets their users' needs. They can also be used to describe patterns that are used throughout the design. It's somewhat like a blueprint that can be used as a guide for the convenience of discussing, improving, and following among a team.

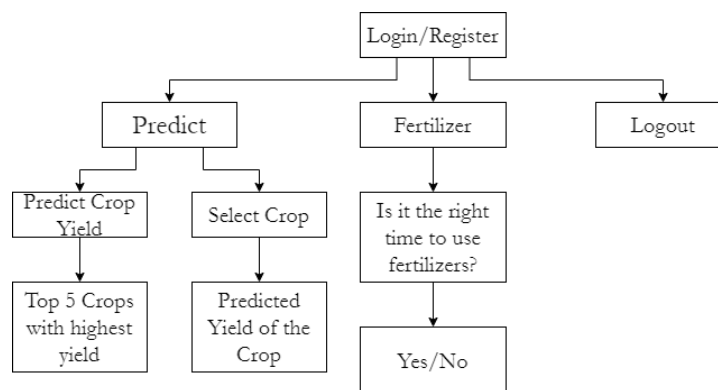


Fig 4.1 System Architecture

Figure 4.1 shows the System Architecture. The first step is to login or register to the application. At the next step, three options are available i.e., Predict, Fertiliser and Logout. The user may select one of the three options and proceed further. Under Predict, the system offers two options that depend on whether the user knows what to plant already or is yet to decide the crop. The inputs are taken from the user in either case and the predicted value is given to the user. When the Fertilizer Module is selected, the user gets a pop up message that says whether or not they can use the fertilizer and it may or may not rain for the next 15 days. Last is the Logout that logs the user out and takes them back to the login/Register Page.

## 4.2 Flowchart

A flowchart is simply a graphical representation of steps. It shows steps in sequential order and is widely used in presenting the flow of algorithms, workflow or processes. Typically, a flowchart shows the steps as boxes of various kinds, and their order by connecting them with arrows. It originated from computer science as a tool for representing algorithms and programming logic but had extended to use in all other kinds of processes. Nowadays, flowcharts play an extremely important role in displaying information and assisting reasoning. They help us visualize complex processes, or make explicit the structure of problems and tasks. A flowchart can also be used to define a process or project to be implemented.

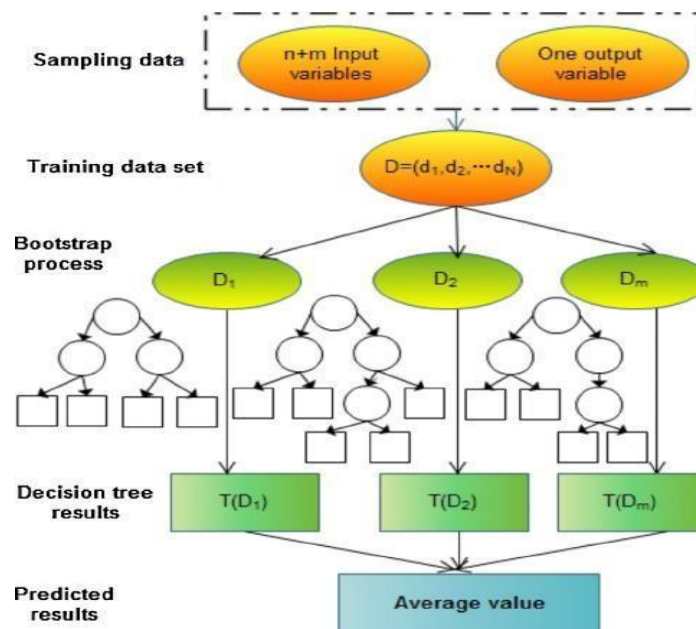


Fig 4.2 Flowchart of Random Forest Algorithm

The figure 4.2 is the graphical representation of sequential steps of Random Forest Algorithm. This algorithm uses a sample dataset with n input variables and 1 output variable. First, it starts with the selection of random samples from a given dataset. Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree. Finally the average value of all the predictions is considered as the result.

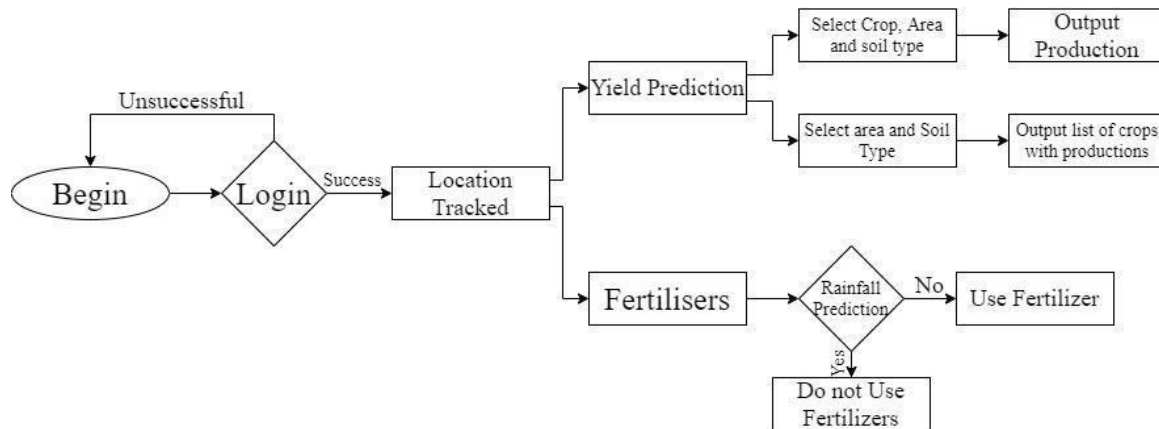


Fig 4.3 Flowchart of Crop Yield Prediction System

The figure 4.3 is the graphical representation of the system designed by us to predict the crop yield based on weather conditions. Working of the system can be understood by the following steps (as depicted in the flowchart):

- Step 1: The user logs in to the system.
- Step 2: If login is successful, the location of the user is tracked.
- Step 3: The system now provides user with the following two paths:
  - Prediction Module: The user can choose to know the prediction of a particular crop or know the list of crops with their corresponding productions.
    - Yield Prediction: The user needs to provide crop, soil type and area as inputs. The system returns the production of the crop given.
    - Crops Prediction: The user needs to provide soil type and area as inputs. The system returns a list of crops along with their production values.
  - Fertiliser Module: The user can choose this module to know if it is the right time to use the fertiliser. This is done by predicting the rainfall for the next 15 days. The system returns “yes” if it is likely to rain else “no”
    - If it returns “yes” the user is suggested not to use the fertiliser.
    - Else the user is suggested to use the fertiliser.

### 4.3 Use Case Diagram

A use case is a methodology used in system analysis to identify, clarify and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. A use case document can help the development team identify and understand where errors may occur during a transaction so they can resolve them.

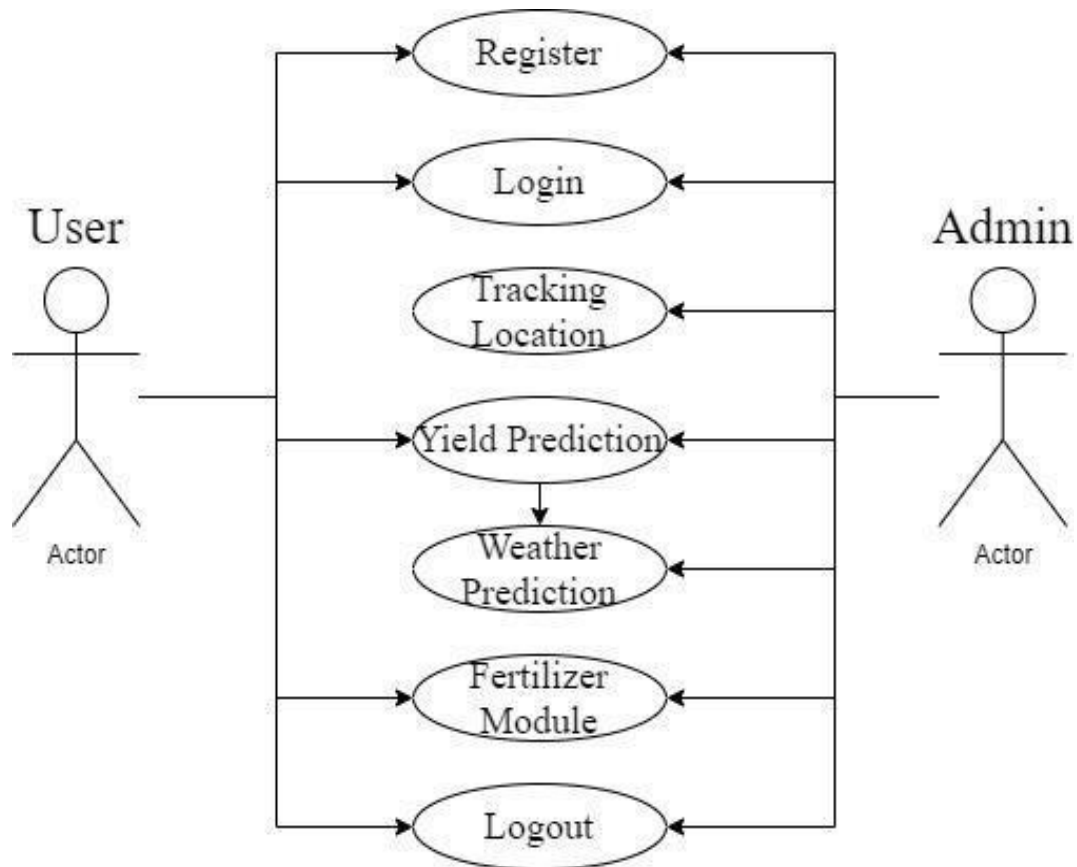


Fig 4.4 Use Case Diagram

The figure 4.4 represents the actors (users) and their functional requirements provided by the system. The system involves two actors – End user (Farmer) and Admin. The functionalities provided by the system are represented in ovals. The arrows represent the dependencies and visibility of the functionalities.

The user has the privilege to access only a few of the functionalities like the Login, Register, Yield Prediction, Fertilizer Module and the Logout whereas the Admin can access two more functionalities that is the location of the user and the results of the weather Prediction.

## 4.4 Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. It is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all types of flow control by using different elements such as fork, join, etc. It captures the dynamic behaviour of the system.

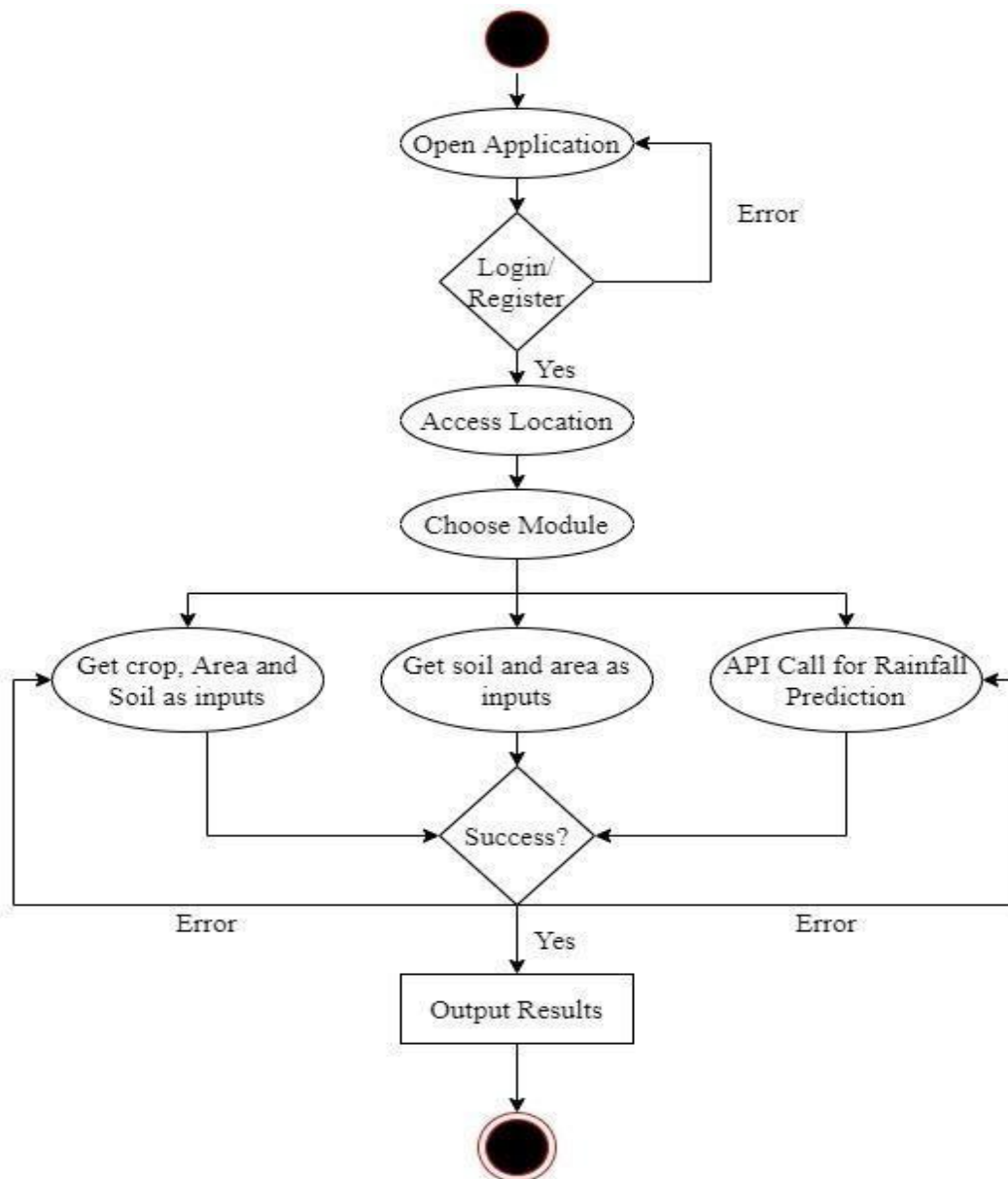


Fig 4.5 Activity Diagram



The diagram in Fig 4.5 represents the flow of operations in the system. As seen in the diagram the operational flow in the system is sequential until the location is tracked and is then branched as it provides three different functionalities i.e., to predict the yield of a given crop, to return a list of crops along with their yield based on the weather and soil conditions and to suggest whether it is the ideal time to use fertiliser.

## 4.5 Sequence Diagram

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

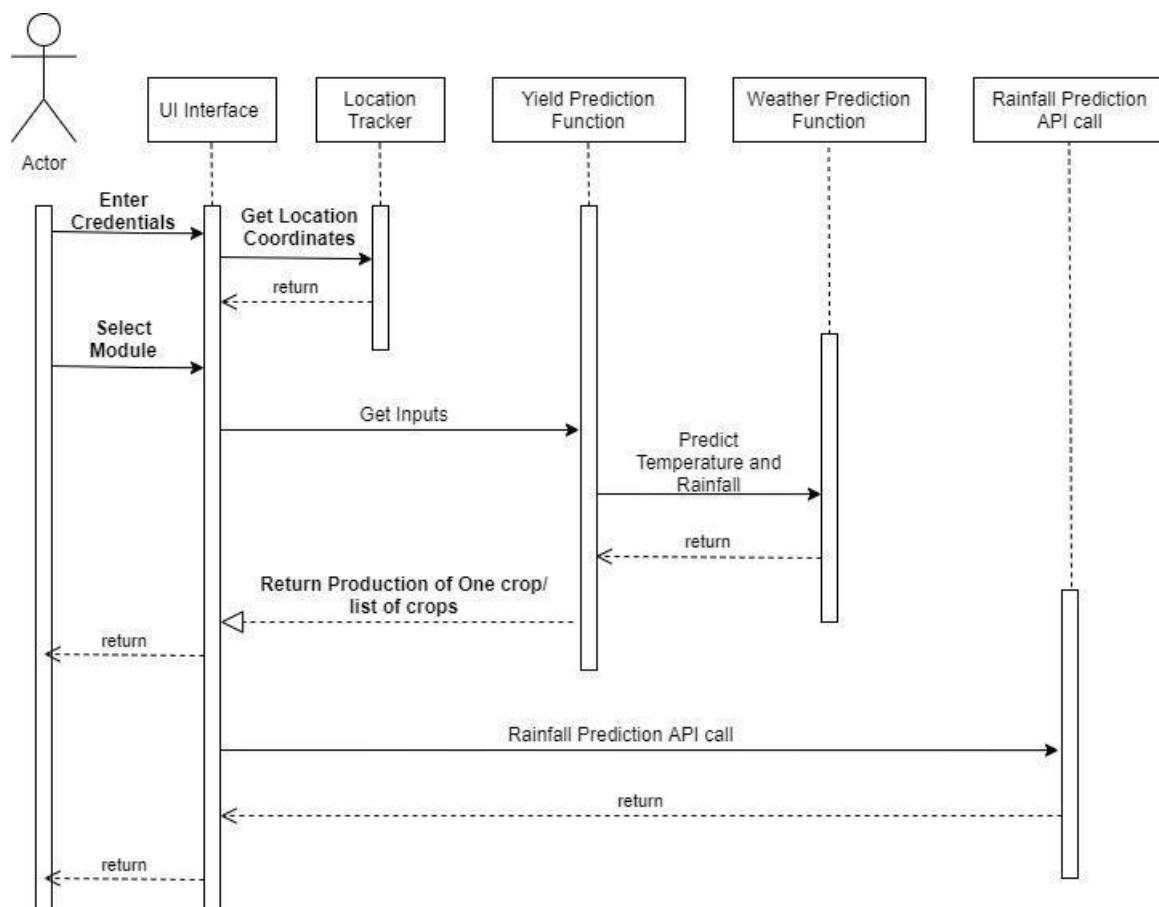


Fig 4.6 Sequence Diagram

The Fig 4.6 represents the various interactions between the user and the objects involved in the system. The various objects or modules involved in the system are UI, location tracker, yield prediction function, weather prediction function and rainfall prediction API call. Here, the various modules are UI interface, Location Tracker, Yield Prediction, Weather Prediction and Rainfall Prediction. The user interacts with each of the modules to give appropriate results. The location tracker returns the current location of the user. Yield Prediction returns the production of the given crop or a list of crops along with their productions. Weather prediction module returns the temperature and rainfall. Rainfall prediction module predicts the rainfall for the next 15 days.

## CHAPTER 5

### IMPLEMENTATION

The implementation of the project was divided into two .i.e. crop yield prediction and rainfall prediction (for fertilizers module).

#### 5.1 Crop Yield Prediction

This module returns the predicted production of crops based on the user's input. If the user wants to know the production of a particular crop, the system takes the crop as the input as well. Else, it returns a list of crops along with their production as output.

These are the following steps of the algorithm implemented:

- **Step 1** : Choose the functionality i.e., crop prediction or yield prediction.
- **Step 2** : If the user chooses crop prediction:-
  - Take soil type and area as inputs.
  - These values are given as input to the random forest implementation in the backend and the corresponding predictions are returned.
  - The algorithm returns a list of crops along with their production predicted.
- **Step 3** : If the user chooses yield prediction:-
  - Take crop, soil type and area as inputs.
  - These values are given as input to the random forest implementation in the backend and the corresponding crop yield prediction is returned.
  - The algorithm returns the predicted production of the given crop.

#### 5.2 Fertilizers Module

This module is used to suggest the farmer on usage of fertilizer based on the rainfall in next few days. To predict the rainfall for the next 15 days we are using an API service provided by 'OpenWeather'. If it is likely to rain we suggest the farmer not to use the fertilizer.

These are the following steps of the algorithm implemented:

- **Step 1**: On selection of this module, API call is made to the 'OpenWeather' Services.
- **Step 2**: The rainfall for the next 14 days is read from the result of the API call.
- **Step 3**: If rainfall is above 1.25 the farmer is suggested not to use the fertilizer. Else, it is

safe to use the fertilizer.

### 5.3 Experimental Implementation

The implementation of the system can be divided into two, i.e., frontend and backend implementation.

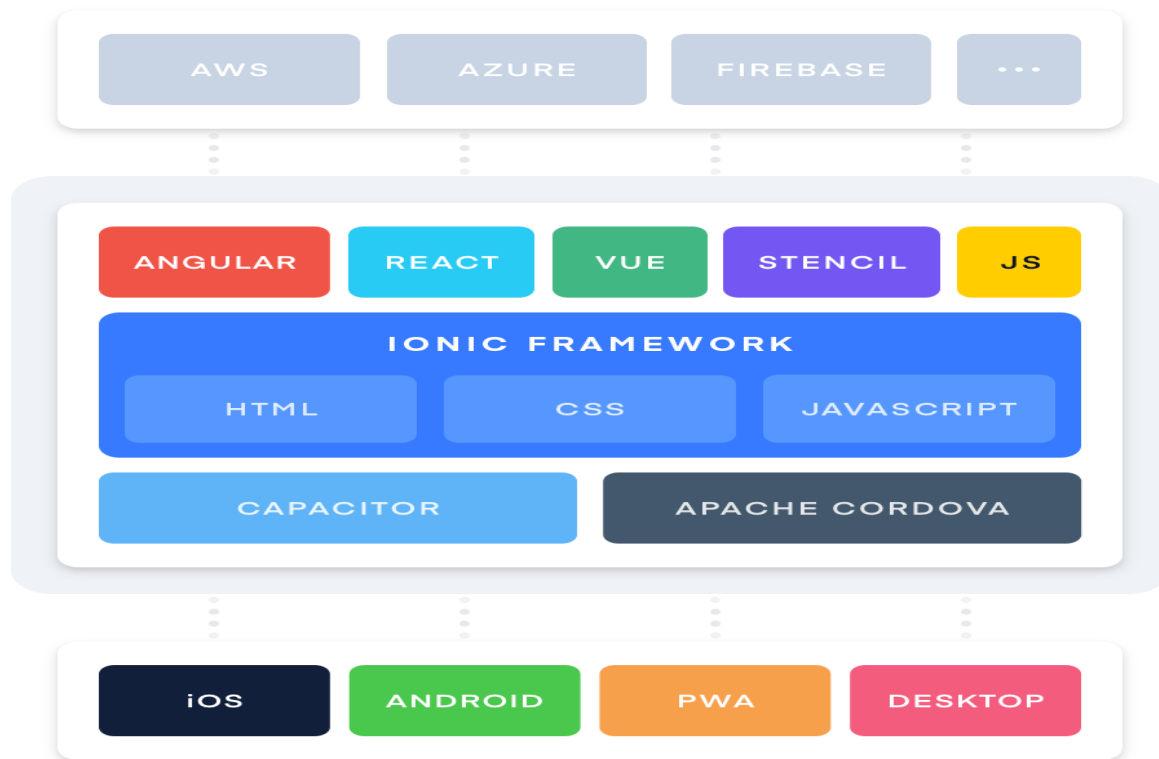


Fig 5.1 Block diagram of Experimental Implementation

The frontend is implemented using the ionic development tools. Ionic Framework is an open source UI toolkit for building performance, high-quality mobile and desktop apps using web technologies — HTML, CSS, and JavaScript — with integrations for popular frameworks like Angular and React. Ionic Framework focuses on the frontend UX and UI interaction of an app — UI controls, interactions, gestures, animations. It integrates with other libraries or frameworks, such as Angular, React, or Vue and thus can be used on any platform.

Ionic is the only mobile app stack that enables web developers to build apps for all major app stores and the mobile web from a single codebase. And with Adaptive Styling, Ionic apps look and feel at home on every device. Thus, the performance of the system is enhanced. Ionic is built

to perform and behave great on the latest mobile devices with best practices like efficient hardware accelerated transitions, and touch-optimized gestures. Ionic is designed to work and display beautifully on all current mobile devices and platforms. With ready-made components, typography, and a gorgeous (yet extensible) base theme that adapts to each platform, you'll be building in style. Ionic emulates native app UI guidelines and uses native SDKs, bringing the UI standards and device features of native apps together with the full power and flexibility of the open web. Ionic uses Capacitor (or Cordova) to deploy natively, or runs in the browser as a Progressive Web App. Thus, our system is web optimized.

The system can be built and deployed across multiple platforms, such as native iOS, android, desktop, and the web as a Progressive Web App - all with one code base. Using ionic we can make the core components to work standalone in a web page with just a script tag (using javascript). @ionic/angular package makes integration with the Angular ecosystem a breeze. @ionic/angular includes all the functionality that Angular developers would expect coming from Ionic 2/3, and integrates with core Angular libraries, like the Angular router. To build apps that target iOS, Android, the web, and the desktop we use ionic react. The official Ionic CLI, or Command Line Interface, is a tool that quickly scaffolds Ionic apps and provides a number of helpful commands to Ionic developers. We use the Ionic CLI to perform cloud builds and deployments, and administer our account.

To build the frontend resources we use npm. npm is the world's largest Software Registry. The registry contains over 800,000 code packages. Open-source developers use npm to share software. Many organizations also use npm to manage private development. To build the system and install ionic we execute the following command:

```
npm install -g @ionic/cli --save
```

The data needed for the system or resources is hosted on firebase. Firebase Storage provides secure file uploads and downloads for Firebase apps, regardless of network quality, to be used for storing images, audio, video, or other user-generated content. All the files such as CSS, HTML, JavaScript and other files are supported by firebase.

We can add the platforms where we want our app to run using the following command:

```
$ cordova platform add ios
```

```
$ cordova platform add android
```

To build the app we run the following command:

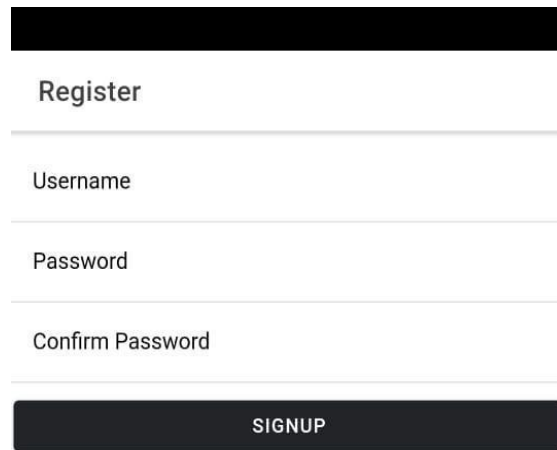
*\$ cordova build*

On successful build we can run the application on the configured host and use the system.

## CHAPTER 6

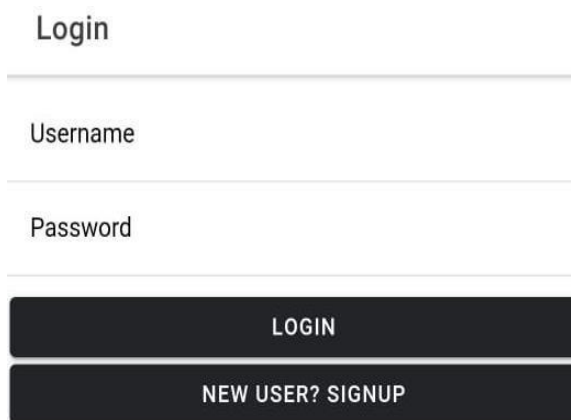
# RESULTS AND DISCUSSION

In the final implementation of the application the first screen the user can view is the login page. Here, the user can register or login using his/her credentials into the application as seen in the Fig 6.1 and Fig 6.2.



The Register screen features a dark header bar at the top. Below it, the word "Register" is centered. There are three input fields: "Username", "Password", and "Confirm Password", each with a horizontal line below it. At the bottom, there is a dark button with the text "SIGNUP" in white.

Fig 6.1 Register Screen



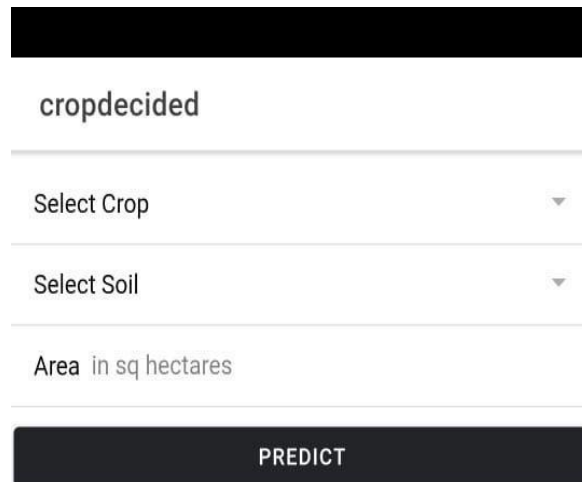
The Login screen features a dark header bar at the top. Below it, the word "Login" is centered. There are two input fields: "Username" and "Password", each with a horizontal line below it. At the bottom, there are two dark buttons: the top one has "LOGIN" in white, and the bottom one has "NEW USER? SIGNUP" in white.

Fig 6.2 Login Screen

The system provides three main functionalities:

- i) **Yield Prediction:** The system takes the required inputs to predict the yield of the given crop.

The inputs to be given are our crop type, soil type and area as shown in the Fig 6.3. The system returns a screen with the predicted yield as seen in the Fig 6.4.



cropdecided

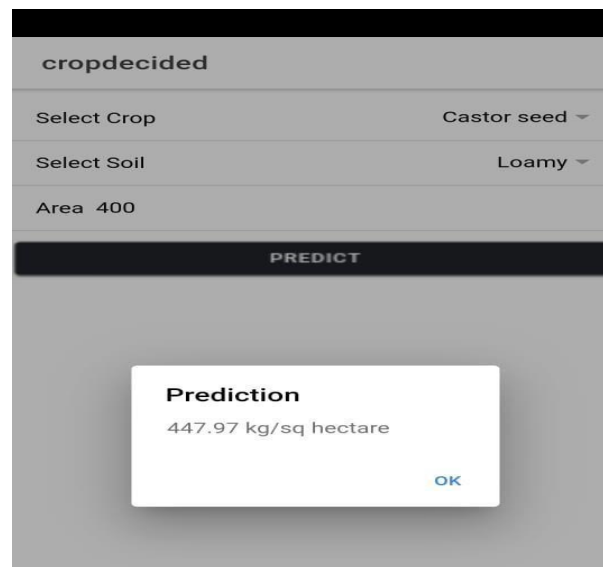
Select Crop ▼

Select Soil ▼

Area in sq hectares

PREDICT

Fig 6.3 Yield Prediction Screen



cropdecided

Select Crop Castor seed ▼

Select Soil Loamy ▼

Area 400

PREDICT

**Prediction**  
447.97 kg/sq hectare  
OK

Fig 6.4 Yield Predicted Screen

**ii) Crop Prediction:** For this module the system takes the required inputs i.e., soil type and area as seen in the Fig 6.5. The system returns a screen with the list of crops with their productions as seen in the Fig 6.6.



The screenshot shows a web interface for crop prediction. At the top, there is a black header bar. Below it, the text 'cropundecided' is displayed. There are two input fields: 'Select Soil' with a dropdown arrow and 'Area in sq hectares'. At the bottom, there is a large black button labeled 'PREDICT'.

Fig 6.5 Crop Prediction Screen

This screenshot shows the same interface as Fig 6.5, but with the 'PREDICT' button clicked. The 'Select Soil' dropdown is now set to 'Black' and the 'Area' field contains '290'. Below the 'PREDICT' button, a table of predictions is displayed.

Predictions	
Arhar/Tur	292.01 kg/sq hectare
Bajra	288.86 kg/sq hectare
Black pepper	288.86 kg/sq hectare
Castor seed	292.52 kg/sq hectare
Cowpea(Lobia)	289.03 kg/sq hectare
Dry chillies	347.07 kg/sq hectare
Dry ginger	288.86 kg/sq hectare
Gram	289.4 kg/sq hectare
Groundnut	301.48 kg/sq hectare
Horse-gram	302.2 kg/sq hectare
Jowar	371.98 kg/sq hectare
Linseed	288.86 kg/sq hectare
Maize	373.12 kg/sq hectare

Fig 6.6 Crops and their productions predicted

iii) **Fertiliser Module:** On choosing this module the system returns a screen with a pop up as seen in Fig 6.7 suggesting the user to use the fertiliser or not.

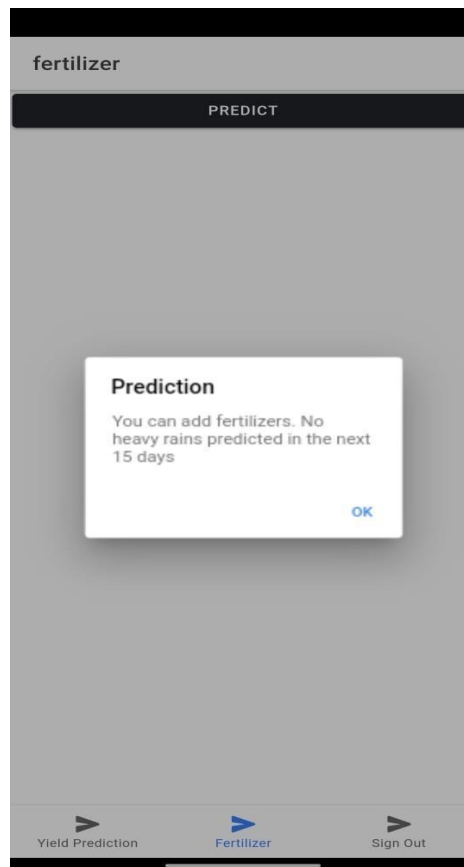


Fig 6.7 Fertiliser Module

## CHAPTER 7

### TESTING

Software testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying that the software product is fit for use.

Software testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test:

- meets the requirements that guided its design and development,
- responds correctly to all kinds of inputs,
- performs its functions within an acceptable time,
- it is sufficiently usable,
- can be installed and run in its intended environments, and
- Achieves the general result its stakeholder's desire.

#### 7.1 Functionality Testing

- Database connection is successfully established.
- The flow of the application from one page to another is correct, accurate and quick.
- All the forms included in the application are working as expected.
- Proper alert messages are displayed in case of wrong inputs.
- After every action on the application the appropriate data is fetched from the backend.

#### 7.2 Usability Testing

- The application enables smooth navigation, hence gives a user friendly experience.
- The inputs taken from the user are via dropdown hence correct inputs are provided to the system.
- Wrong inputs given by the system are handled effectively.

- The content provided by the application is verified and is taken by the trusted sources.
- The datasets trained for prediction of the crop yield are accurate and balanced.

### **7.3 Interface Testing**

- The application connects correctly with the server. In case of failure an appropriate message is displayed.
- Interruptions by the server or by the user are handled efficiently.
- If the user enters wrong credentials or invalid email id, the application handles it efficiently by displaying appropriate messages.
- The interaction with the user is smooth and easy.

### **7.4 Compatibility Testing**

- This application is compatible with all the browsers enabled with javascript.
- It is compatible with all the mobile devices and desktop.

### **7.5 Performance Testing**

- It works fine with moderate internet speed.
- The connection is secured and user details are stored in a secured manner.
- The switch from one screen to another is quick and smooth.
- The inputs from users are taken correctly and response is recorded quickly.

## CHAPTER 8

### CONCLUSION

This system is proposed to deal with the increasing rate of farmer suicides and to help them to grow financially stronger. The Crop Recommender system helps the farmers to predict the yield of a given crop and also helps them to decide which crop to grow. Moreover, it also tells the user the right time to use the fertiliser.

Appropriate datasets were collected, studied and trained using machine learning tools. The system tracks the user's location and fetches needed information from the backend based on the location. Thus, the user needs to provide limited information like the soil type and area.

This system contributes to the field of agriculture. One of the most important and novel contributions of the system is suggesting the user the right time to use the fertiliser, this is done by predicting the weather of the next 14 days. Also, the system provides a list of crops with their productions based on the climatic conditions.

The future work is focused on providing the sequence of crops to be grown depending on the soil and weather conditions and to update the datasets time to time to produce accurate predictions. The Future Work targets a fully automated system that will do the same. Another functionality that we are trying to implement is to provide the correct fertiliser for the given crop and location. To implement this through study of fertilisers and their relationship with soil and climate is required. We are also aiming to predict the crisis situation in advance like the recent hike of onion prices.

## REFERENCES

- [1] <https://towardsdatascience.com/machine-learning-basics-part-1-a36d38c7916>
- [2] <https://healthcare.ai/machine-learning-versus-statistics-use/>
- [3] <https://medium.com/fintechexplained/machine-learning-algorithm-comparison-f14ce372b855>
- [4] <https://dataaspirant.com/2017/05/22/random-forest-algorithm-machine-learning/>
- [5] <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>
- [6] [https://openweathermap.org/api/one-call-api?gclid=CjwKCAjwt-L2BRA\\_EiwAacX32SzjPujH29VetWiBl-xcyG9Dw17xto02vSwkxdTFeIR18tr5GF-8vx0C2-kQAvD\\_BwE](https://openweathermap.org/api/one-call-api?gclid=CjwKCAjwt-L2BRA_EiwAacX32SzjPujH29VetWiBl-xcyG9Dw17xto02vSwkxdTFeIR18tr5GF-8vx0C2-kQAvD_BwE)
- [7] <https://ukdiss.com/examples/rainfall-prediction-machine-learning.php>
- [8] <https://iopscience.iop.org/article/10.1088/1748-9326/aae159>
- [9] [https://en.wikipedia.org/wiki/Agriculture\\_in\\_India](https://en.wikipedia.org/wiki/Agriculture_in_India)
- [10] <https://www.agricultureinformation.com/forums/threads/agriculture-in-india.101537/>
- [11] [https://en.wikipedia.org/wiki/Farmers%27\\_suicides\\_in\\_India](https://en.wikipedia.org/wiki/Farmers%27_suicides_in_India)
- [12] <https://towardsdatascience.com/machine-learning-an-introduction-23b84d51e6d0>
- [13] Groundnut Crop Yield Prediction Using Machine Learning Techniques by Vinita Shah and Prachi Shah.
- [14] Developing regression model to forecast the rice yield at Raipur condition by A Jain, JL Chaudhary, MK Beck and Love Kumar
- [15] Machine learning approach for forecasting crop yield based on climatic parameters by S.Veenadhari, Dr. Bharat Misra & Dr. CD Singh, International Conference on Computer Communication and Informatics (ICCCI -2014), Jan, 2014
- [16] Predicting Yield of the Crop Using Machine Learning Algorithm by P.Priya, U.Muthaiah & M.Balamurugan, International Journal of Engineering Sciences & Research Technology (IJESRT), April, 2018
- [17] A Survey on Crop Prediction using Machine Learning Approach by Sriram Rakshith.K,

Dr. Deepak.G, Rajesh M, Sudharshan K S, Vasanth S & Harish Kumar, International Journal for Research in Applied Science & Engineering Technology (IJRASET) Volume 7, Issue IV, Apr 2019

- [18] Heuristic Prediction of Crop Yield using Machine Learning Technique by S. Pavani, Augusta Sophy Beulet P, International Journal of Engineering and Advanced Technology (IJEAT) Volume-9, December 2019

