



***Dissertation on***

**PerfectCrop**  
**The right crop for your soil**

*Submitted in partial fulfilment of the requirements for the award of degree of*

**Bachelor of Technology**  
**in**  
**Computer Science & Engineering**

**UE18CS390A – Capstone Project Phase - 1**

***Submitted by:***

<b>Srish Srinivasan</b>	<b>PES1201800051</b>
<b>Akash Kumar Rao</b>	<b>PES1201800089</b>
<b>Vishruth P Reddy</b>	<b>PES1201800102</b>
<b>Ishan Agarwal</b>	<b>PES1201800291</b>

*Under the guidance of*

**Prof. Raghu B A**  
Associate  
Professor  
PES University  
**January - May 2021**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**  
**FACULTY OF ENGINEERING**  
**PES UNIVERSITY**

(Established under Karnataka Act No. 16 of 2013)  
100ft Ring Road, Bengaluru – 560 085, Karnataka, India



**PES UNIVERSITY**  
(Established under Karnataka Act No. 16 of 2013)  
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

**FACULTY OF ENGINEERING**

## **CERTIFICATE**

*This is to certify that the dissertation entitled*

**‘PerfectCrop-The right crop  
for your soil’**

*is a bonafide work carried out by*

**Srish Srinivasan  
Akash Kumar Rao  
Vishruth P Reddy  
Ishan Agarwal**

**PES1201800051  
PES1201800089  
PES1201800102  
PES1201800291**

in partial fulfilment for the completion of seventh semester Capstone Project Phase - 1 (UE18CS390A) in the Program of Study - Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period Jan. 2021 – May. 2021. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The dissertation has been approved as it satisfies the 6<sup>th</sup> semester academic requirements in respect of project work.

Signature  
Prof. Raghu B A  
Associate Professor

Signature  
Dr. Shylaja S S  
Chairperson

Signature  
Dr. B K Keshavan  
Dean of Faculty

**External Viva**

**Name of the Examiners**

**Signature with Date**

1. \_\_\_\_\_

\_\_\_\_\_

2. \_\_\_\_\_

\_\_\_\_\_

## DECLARATION

We hereby declare that the Capstone Project Phase - 1 entitled “**PerfectCrop-The right crop for your soil**” has been carried out by us under the guidance of Prof. Raghu B A, Associate Professor and submitted in partial fulfilment of the course requirements for the award of degree of **Bachelor of Technology in Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester January – May 2021. The matter embodied in this report has not been submitted to any other university or institution for the award of any degree.

PES1201800051

Srish Srinivasan



PES1201800089

Akash Kumar Rao



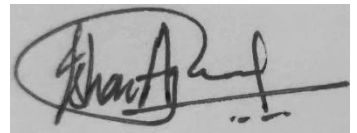
PES1201800102

Vishruth P Reddy



PES1201800291

Ishan Agarwal



## **ACKNOWLEDGEMENT**

I would like to express my gratitude to Prof. Raghu B A, Department of Computer Science and Engineering, PES University, for his continuous guidance, assistance, and encouragement throughout the development of this UE18CS390A - Capstone Project Phase – 1.

I am grateful to the project coordinators, Prof. Silviya Nancy J, and Prof. Sunitha R for organizing, managing, and helping with the entire process.

I take this opportunity to thank Dr. Shylaja S S, Chairperson, Department of Computer Science and Engineering, PES University, for all the knowledge and support I have received from the department. I would like to thank Dr. B.K. Keshavan, Dean of Faculty, PES University for his help.

I am deeply grateful to Dr. M. R. Doreswamy, Chancellor, PES University, Prof. Jawahar Doreswamy, Pro Chancellor – PES University, Dr. Suryaprasad J, Vice-Chancellor, PES University for providing to me various opportunities and enlightenment every step of the way. Finally, this project could not have been completed without the continual support and encouragement I have received from my family and friends.

# **ABSTRACT**

Our project comes under the domain of Precision Agriculture. It helps farmers make informed decisions with regards to the kind of crop they must invest in to get good returns. The aim of this project is to build a predictive model to recommend the most suitable crop to grow based on the various parameters that influence the fertility of the soil with the help of machine learning algorithms. A farmer or a horticulturist toils all day and keeps himself really busy throughout the year by looking after his crops and providing them with the right amount of water and nutrients expecting only one thing in return. And this is good crop yield. But if the farmer or the horticulturist makes a mistake in the very first step by choosing the wrong crop to cultivate, then the all the remaining steps in the process is absolutely useless. So, our main goal in this project is to give the farmer or the horticulturist a sensible start to their cropping season by helping them chose the right crop to grow in their lands so that they are able to obtain a very good yield at the end of the harvest season. The focus will be restricted to a very small part of India and we will be making use of the historic crop yield data for that particular region and try to incorporate machine learning algorithms that can ingest this data in order to build accurate models that could recommend the most ideal crop to be grown when fed with the properties associated with the soil and atmosphere of that region.

# TABLE OF CONTENT

Chapter No.	Title	Page No.
1.	INTRODUCTION	
2.	PROBLEM STATEMENT	
3.	LITERATURE REVIEW	
	3.1 Paper1: Random Forest Algorithm for Soil Fertility Prediction and Grading Using Machine Learning	
	3.2 Paper2: Crop Selection Method to maximize crop yield rate using machine learning technique.	
	3.3 Paper3: An approach for prediction of crop yield using machine learning and bigdata techniques.	
	3.4 Crop Prediction using Machine Learning Algorithms.	
4.	PROJECT REQUIREMENTS SPECIFICATION	
5.	HIGH LEVEL DESIGN DOCUMENT	
6.	IMPLEMENTATION AND PSEUDOCODE	
7.	CONCLUSION OF CAPSTONE PROJECT PHASE-1	
8.	PLAN OF WORK FOR CAPSTONE PROJECT PHASE-2	
	REFERENCE/BIBLIOGRAPHY	
	APPENDIX A DEFINITIONS, ACRONYMS AND ABBREVIATIONS	

---

## CHAPTER 1

### INTRODUCTION

Our project comes under the domain of Precision Agriculture. Therefore, it is very important to understand what Precision Agriculture means at its core. Precision Agriculture is a system to manage farms that is based on the use of advanced technologies at every step of the agriculture process. It is based on observation, measurement and response to the variability in crops. The aim of precision farming is to develop a decision support system for the management of a farm with the goal being maximization of returns with the efficient and judicious use of resources. In simpler terms, the goal is to increase the harvest through the efficient usage of seeds, fertilizers and pesticides. With the involvement of advanced technology, decisions are made purely based on data thereby reducing the risk of failure to a large extent as decisions are no longer made based on intuition and luck. And by making use of resources in an efficient way, we are saving the environment from the rapid depletion of its natural resources.

Precision agriculture originated in the 1980s in the United States of America. Precision agriculture was the most important development of the third wave of modern agricultural revolutions. In the first agricultural revolution that took place between the 1900 to 1930, people saw a large increase in mechanized agriculture. This resulted in each farmer producing food that was more than sufficient to satisfy the hunger of 26 people. Later in the 1960s, people witnessed the second agricultural revolution in the form of Green Revolution which gave birth to genetic modification. At this stage, farmers were able to produce food that was more than sufficient to satisfy the hunger of 156 people. And finally, with rapid technological advancements and its integration in the field of agriculture gave birth to the third agricultural revolution which was widely known as Precision Agriculture. The others synonyms for Precision Agriculture include “Precision Farming”, “Satellite Farming” and “Site Specific Crop Management”. Input recommendation map for fertilizers was the first outcome in this domain and this was based on the grid soil sampling that was done. However, it was in its very early stages and was not practiced much. But with the advent of smartphones, high speed networks and enormous amount of satellite data, precision agriculture has become very popular and has seen a steep growth in the last 5 years.

---

As we all know, a field has heterogeneous zones and with the aid of technology, we can identify these zones and manage their variability. Therefore, it is important to have some knowledge about the technologies being used. The GPS has been one of the most important enabling factors for the practice of precision farming. The farmer's ability to precisely locate his/her position in the field led to the development of spatial variability maps for variables such as crop yield, nutrient levels, humus content and soil moisture content. Data similar to the ones mentioned in the previous sentence are extracted using sensor arrays mounted on GPS-equipped combine harvesters.

These sensors work in real time thereby collecting a wide spectrum of information ranging from pigment levels to water status, along with multispectral images. This data is used in combination with satellite images through variable rate technology which include seeders, sprayers, etc. in order to achieve optimal distribution of resources. In order to avoid the possibility of an overlap or an underlap during the process of sowing seeds or the application of fertilizers and pesticides, onboard computers and GPS-navigators are used in vehicles. Digital maps and variable rate applications are being used for fields based on variable characteristics and the calculation of fertilizer dosage for every individual zone respectively. With the help of recent technological advancements, real time sensors placed in soil can wirelessly transmit data without the need of human intervention post sensor installment in the soil. For the remote monitoring of fields, drones and satellites have been put to use. Unmanned aerial vehicles are not very expensive and can be controlled by pilots without the need of any prior experience. These drones are equipped with multispectral cameras that can capture multiple images of the field

A lot of sensors have been, predominantly wireless ones have been used in order to numerically capture the influence of field indicators such as moisture, pressure, rainfall, temperature, etc. And finally, with the help of applications that are hosted either as a mobile application or a web application, all the data that has been captured will be analyzed in order to obtain some meaningful insights that can be used to manage farms in an efficient manner.

Coming to the complexity involved in practicing precision farming, it is a little complex because most of the technologies that are being used are new and therefore requires a skilled workforce in order to make good use of this technology. For instance, a person who lacks the required skills will find it difficult to analyze a satellite image or repair an onboard computer. But at the same time, technological solutions that are simple do exist and can be accessed by every farmer. Some of these simple solutions include weather sensors, wireless modems, etc.



The next very important thing to look into is the cost involved in the incorporation precision farming in your farms or agricultural fields. At present, some of the sophisticated equipment and software are quite expensive and therefore the precision farming technologies are mostly seen only in large farm owned by affluent farmers.

But it is a well-known fact that with increasing developments in technology, it only becomes more affordable and easier to use. With this natural dynamic, the focus is on enabling easy access to these technologies with very little cost involved.

Stepping into the future, precision agriculture is something that will be unavoidable and it makes absolutely no sense in not making use of it because it is extremely profitable. In one of the blogs by the name onesoil.ai, we learnt that the American farmers are able to save on a large amount of money that they spend on agriculture through the incorporation of precision farming in their fields.

With precision farming up and running, farmers will be able to

1. **Make improved decisions**
2. **Improve the inherent quality of the farm products**
3. **Enhance marketing of farm products**
4. **Improve relationships with landlords and local money lenders**

So, our focus in this project is to improve the decision making involved in the process of crop selection with the help of machine learning algorithms by taking into account the soil properties and the surrounding atmospheric conditions.

---

## CHAPTER 2

### PROBLEM STATEMENT

Our aim in this project is to build a recommendation system that recommends the most suitable crop to be grown given the properties of the soil and the surrounding atmospheric conditions of a specific location. We will be narrowing down our focus onto a specific part of India. The properties of the soil that are to be taken into consideration could include its nutrient contents and their quantities. The nutrients could include both macronutrients as well as micronutrients. Macronutrients are those nutrients that are produced by the soil in relatively larger quantities. Some examples of macronutrients include N, P, K, Ca, S, Mg, C, O, and H. On the other hand, micronutrients are those nutrients that are produced by the soil in relatively smaller quantities. Some examples of micronutrients include Fe, B, Cl, Mn, Zn, Cu, Mo, and Ni. These elements are present in the form of salts beneath the soil and are absorbed by the plants in the form of ions. Other soil properties include pH, EC, moisture, temperature, etc. The atmospheric conditions to be taken into consideration could include rainfall, temperature, humidity, etc. On collecting the required data, we plan to use machine learning algorithms in order to build robust models that can make forecasts with a very good accuracy level. Our project comes under the domain of precision farming. Precision farming can reduce the quantity of nutrients and other requirements used by a large extent while boosting the yield by a large margin. Farmers can thus obtain great returns on their investments and can also save big on fertilizer, pesticide, water and other resources. The next big advantage of practicing precision farming is the prevention of malefic impact on our environment by using the right quantity of chemicals and thereby conserving the quality of soil and ground water.

---

## CHAPTER 3

### LITERATURE SURVEY

#### Paper1

The authors of [1] have proposed a machine learning based solution for the analysis of imperative soil parameters and their influence on the kind of crops that could be suitably grown in a given soil. The various soil nutrients are treated as the independent variables and the grade of the soil is the target variable. The regression algorithm along with RMSE value were employed to predict the rank of a soil and on applying a few classification algorithms for the purpose of crop recommendation, they found that Random Forest was the most accurate model.

In order to have a good yield, it is important that the soil is rich in the required nutrients. So, the main goal in this project was to rank a soil sample by examining its nutrient contents (Macronutrients and Micronutrients) and then recommend the most suitable crop that could be grown in this soil.

In the first part of the project, the contents of various soil nutrients such as EC, K, pH, Mn, Zn, S, P, and B are considered as the independent variables and the grade of the soil is considered as the dependent variable. So, a Multi-Variate Linear Regression model was built to predict the fertility of soil on a scale of 1-5.

A linear combination of the independent variables was chosen as the hypothesis function. The cost function chosen was:

$$J(\theta) = \frac{1}{2m} \sum (h_{\theta}(X_i) - Y_i)^2$$

$X_i$  = vector of independent variables

$h_{\theta}$  = hypothesis function

$Y_i$  = True value of the response variables

$m$  = normalizing parameter

---

The Gradient Descent algorithms was adopted to minimize the cost function. Then hypothesis testing was carried out on the test dataset in order to check for the model's correctness and efficiency and the RMSE value was used to determine accuracy of the model.

In the second component of the project, the authors attempted to recommend crops using machine learning algorithms such as Support Vector Machines, Random Forest Classification and Decision Tree and based on the RMSE value the best model was chosen. The true accuracy of the model will be obtained when real-time data would be passed to this model.

The most important feature is used to split a node and then recursively the next most important features is looked for from the subset of the remaining features thereby generating a highly accurate classifier with wide diversity.

In order to split a node, only a selective group of features are selected among all the features. An element of randomness is introduced through the use of random thresholds for the feature set. A Random Forest Classifier applies a technique known as bootstrap aggregation or bagging to the tree learners. From the training set, random sampling with replacement was performed and to each of these samples, trees were fit.

Then a voting is performed among all the predictions output by all the trees in order to arrive at the final result. To ensure that the variance is low and at the same time the bias is also kept low, the bootstrapping procedure was applied. If the trees are not related to each other, then the average of the outputs produced by these trees are more robust to noise but in the case of a single tree, the prediction made can be very easily influenced by the noise.

Therefore, the idea behind using different samples from the training sets was to develop trees that are highly uncorrelated. The number of trees used in a random forest classifier is usually in the range of a few hundreds to several thousands and this number is heavily dependent on characteristics of the training data set.

### **Learnings from [1] are**

- The Random Forest Algorithm is based on ensemble learning and proved to be a very effective algorithm for classification.
- The basic idea is to build multiple decision trees from randomly selected subsets of the data. And then when a new data instance comes in, it is put through all these decision trees and a majority vote is taken in order to give the instance its final classification.

- 
- Each tree as individual entities might not be ideal, but as a group they can perform really well.
  - Since there are numerous trees, the existence of any errors or uncertainties associated with any of the trees are taken care off by this algorithm.

## Paper 2

Yield rate of crops is dependent on two broad factors, the first being genetic development of seeds which lies more on the fields of Bio-Technology and the second is crop selection management. The latter factor is more algorithmic and thus an algorithm can be developed for this job specifically. This concept drives this paper towards building an algorithm for just this purpose. It is provided with the necessary inputs and information, and a selection pattern is expected in return. Some factors that go as input to this algorithm include the predicted yield of the respective crop. This prediction is done by various different machine learning models over various different factors like soil properties such as Nitrogen, Phosphate and Potassium contents, pH properties of the soil, weather conditions, rainfall predictions or forecasts. The machine learning models used to perform this task are the most prominent ones that have proved their value in various other fields of research. The newer models, some that use Boosting techniques had not been tried into this field of research until this paper was published, and the paper aimed at trying them out as well and place a detailed comparative analysis for the readers. These techniques included GDBT (Gradient Boosted Decision Tree) and RGF (Regularized Greedy Forest).

The method is composed of 2 significantly differentiable parts that work together. The first part uses machine learning models like Artificial Neural Network (ANN), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Decision Tree Learning, Random Forest, Gradient Boosted Decision Tree (GBDT), Regularized Greedy Forest (RGF) to predict yield rate of crop before the season arrives. This is possible due to the fact that season of entire year in India depends on summer rainfall data of said year. Therefore, it is possible to predict a season in advance, provided that the rainfall data is available.

Crops can be classified as:

- a) Seasonal crops, and
- b) Whole year crops
- c) Short time plantation crops, and
- d) Long time plantation crops

Second major part of the research work includes appropriate crop selection. Once the yield of all the types of crops is calculated, it is to be followed by selection among these crops that maximize the yield and also keep seasonal rotations in consideration along with minimizing crop-less days that are waste of farmers' resources. Below is the algorithm used in this part of the research work.

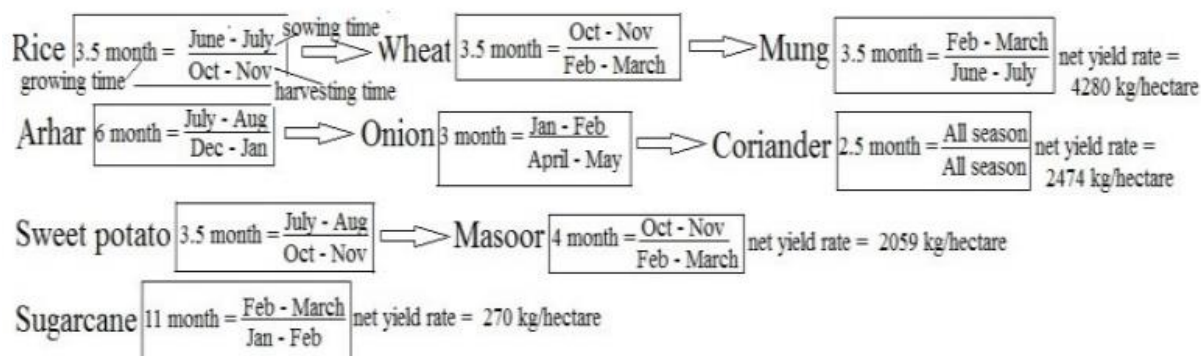
---

### Algorithm: Crop Selection Method(CSM)

```
cropSelector(presentTime)
    if presentTime ≥ End of Season then return 0
    end if else
        if presentTime = sowingTime then
            return cropSelector(presentTime + 1)
        end if else
            cropSowingTable ← cropInputTable(presentTime)
            L: crop ← max{crop → prRate / crop → plantationDay}
                crop ∈ cropSowingTable

            if (presentTime + crop → plantationDay) ≥ End of Season then
                //remove crop from cropSowingTable
                cropSowingTable ← cropSowingTable - crop
                if cropSowingTable is NULL then
                    return cropSelector(presentTime + 1)
                end if else
                    go to L
                end else end if
            else
                update(OutputcropTable, crop)
                npr ← (crop → prRate + cropSelector(presentTime + crop → plantationDay))
                return npr
            end else end else
        end else
    end cropSelector
```

Taking an example to further explain the work done by the authors makes things easier to understand. Consider the below image where four different crop rotation options are available for the farmer to choose. The algorithm is provided with all of this data, that is, each crop's seasonal information, as well as yield rate for the year provided by the machine learning model from the first part of the work.



The algorithm chooses the first option, that is, Rice followed by Wheat and Mung in respective order. The net yield rate for this option is the highest among all the options and can be grown in the said order without any seasonal conflicts.

In Conclusion, the final sequence of crops is the end result of the Crop Selection Method (CSM) with inputs such as rainfall of the year and all the crops' seasonal information. The work done is remarkable and has been cited multiple times for further research work and enhancements. One of many important points made in the paper is the yield in the upcoming season can be predicted using historical rainfall data. The machine learning models used were never tried before for this particular application and the results turned out to be very satisfactory.

There are some cons to the work as well which should be mentioned with details. The first one being, the limitation of the geographical diversity of the data collected. The data is collected from a single farmer residing in Patna district of Bihar, India. This data and results might be appropriate for the said district, but cannot be extrapolated to entire country's Agriculture Sector. The yield of the crops uses rainfall data alone, whereas in reality it depends on various other factors like soil parameters, climatic and weather conditions. Therefore, including these factors as well into a future study holds a lot of potential.



### Paper 3

In our country, due to the lack of accessibility to technology, farmers grow crops purely based on the history of crops grown in that region and based on intuition. It may work out sometimes but they go under heavy losses mostly. Putting in months of effort to see their crop not prosper is very sad and is a huge loss to the farmers. Due to lack of awareness and knowledge, they might perform certain practices extensively or may not be doing enough which results in poor yield. Overuse of fertilizers, insecticides and pesticides can also lead to poor production. Even if the farmer doesn't factor into the atmospheric conditions, he will go under loss due to poor planning. Educating the farmer maybe helpful but it is a tedious job. To make their work simpler and more profitable, we can use machine learning techniques to predict the best crop a farmer or horticulturist must grow by factoring in all the parameters to increase his profits.

This early prediction can help farmers plan for either annual or seasonal crops. Use of precision farming can give more accurate results due to the high inspection and efforts on a small area. Since we can get almost accurate values of soil parameters, if weather conditions are known more accurately before-hand, farmers can adapt accordingly and try growing a suitable crop.

Employing machine learning can give us insights about the soil fertility, the elements in soil and atmospheric conditions which can be used to precisely predict what crop can be grown in that particular field. In this paper, the authors have employed a variety of machine learning methods such as supervised, reinforcement and unsupervised learning. Techniques such as regression, clustering, classification etc are used to predict the perfect crop for the provided conditions.

Linear regression is a technique used when we have 2 parameters of interest. When one parameter is dependent on the other, linear regression comes into play. The only drawback is that it works only for linear data and not for non-linear or complex data.

---

Artificial Neural Network and especially Back Propagation Neural Network is used when we have multiple parameters of interest which decide the crop yield. Here, the 3 layers namely, the input, the hidden and the output

layer decide the crop yield values. Weights can be adjusted to get a better recommendation. ANN not only works for linear data but complex data as well.

Support Vector Machines is another technique that gives very accurate results. The problem of overfitting doesn't affect SVM. SVM is used when we have lots of parameters to consider and especially atmospheric conditions. This removes the issues of changing the weights to obtain a desirable value as that was the case in ANN.

The authors have used several metrics to validate the output of the predictions. These metrics give us the accuracy of the predictions. Mean Squared Error, Mean Absolute Error and Root Mean Squared Error are the metrics employed by the authors to validate the outcomes.

Metrics	Formula
Root Mean Squared Error	$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \bar{y}_i)^2}$
Mean Squared Error	$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \bar{y}_i)^2$
Mean Absolute Error	$MAE = \frac{1}{N} \sum_{i=1}^N  y_i - \bar{y}_i $

The paper also throws light on various techniques used for predicting various crops.

Multiple Linear Regression gives the best results for tea crop yield as it is only dependent on the soil conditions such as being acidic, well drained and light soil. Pepper, potato and tomato are predicted using MLR as well.

ANN is used for predicting Maize and wheat crops due to non-availability of data and presence of non-linear data. SVM is also used to predict the yield of Maize crops as they are unaffected by overfitting. It also distinguished between a crop and weed growing in the surrounding areas. Rice is also

---

The inputs to the models will be broadly categorized into weather and non-weather inputs.

Weather Parameters	Non-weather Parameters
Temperature	Soil Moisture
Rainfall	pH
Humidity	Crop Type
	Seed Variety
	Salts such as N, P, K, C, Ca, Mg, Mn, S etc.

The pros of this paper is that they have incorporated different machine learning techniques for prediction and validated them using different performance metrics. The cons of the paper are that they didn't work on any big data models for prediction but mentioned that further work can be done along big data lines. So, they should have used an appropriate title for their research work as it was a little misleading.

---

#### **Paper 4**

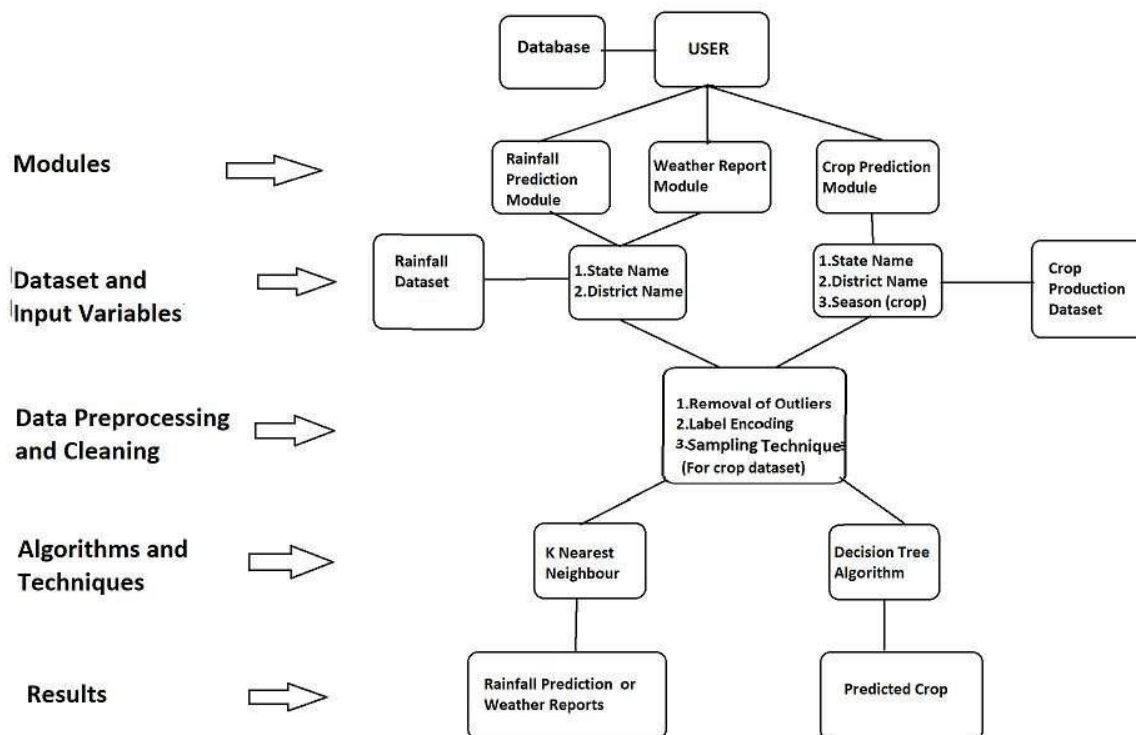
In this paper, the authors have taken into account the different ML algorithms which are used in crop prediction over various other studies and have tried to add more attributes to the system in order to improve the results. They have compared the prediction of the ideal crop from using different models to get a better understanding of how to use ML techniques in the future.

In order to increase yield rate of the crops, several biological and chemical approaches have been implemented over the years like better quality seeds, proper use of insecticides and pesticides, use of fertilisers, etc. The method of crop prediction identified by the authors based on previous work done i.e. the crop selection method (CSM) distributes crops into

- Seasonal
- Whole Year
- Short plantation period
- Long-time plantation

The data of these were then taken for a particular selected region (as agriculture depends on the type of place and various factors like climate, soil, etc.) and then the farmers could be given a list of crops they would choose from along with the desired sequence in which the crops could be planted so as to increase the total yield throughout the season. This may also improve land reusability and hence the resources available thus further improving the farmers' profit. Thus, the already existing systems can give the suitable crop keeping in mind the yield over a particular selected region.

The previous work the authors surveyed made use of ML algorithms with one attribute and thus they made a system to add more attributes to it so that along with crop, the time of the year and the weather prediction is also taken into account. This is shown in their work flowchart below.



The data must include:

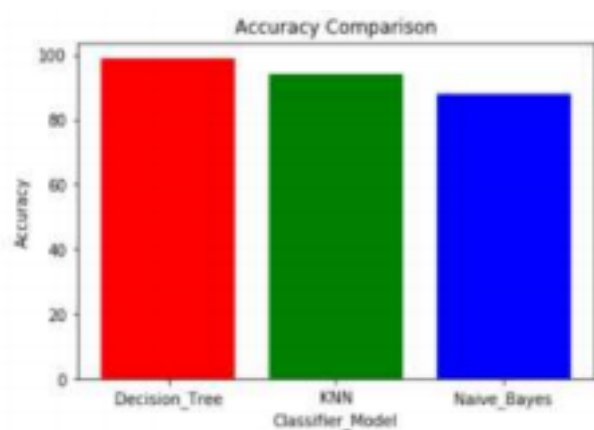
- Soil Parameters
  1. Soil Type
  2. Soil Ph
- Climatic Parameters like humidity, temperature, wind, rainfall
  1. Humidity
  2. Temperature
  3. Wind
  4. Rainfall
- Production
- Cost of cultivation
- Previous year yield results

The data is pre-processed and fed into KNN, Decision tree and Naive Bayes classifier and the results from all of these are compared.

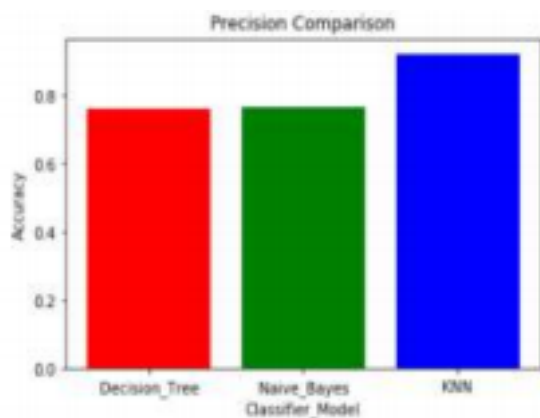
(The selection attributes of the Decision tree being Gini index, entropy and information gain.)

The results were as follows:

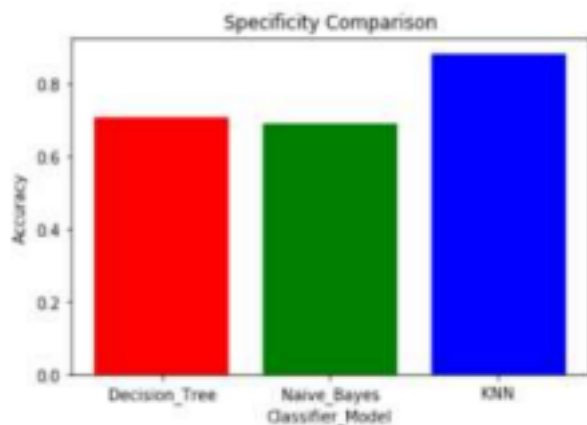
Accuracy:



Precision:



Specificity:



After studying the results from the three models and their comparisons, the conclusion obtained is that Decision tree shows poor performance when dataset is having more variations but naïve bayes provides better result than decision tree for such datasets. The combination classification algorithm like naïve bayes and decision tree classifier are better performing than use of single classifier model.

Thus, we can make use of this study and also cross check the findings when using different models.

## Literature Survey Comparison

<p><b>Paper1: Random Forest Algorithm for Soil Fertility Prediction and Grading Using Machine Learning by</b> Keerthan Kumar, T.G., Shubha, C. and Sushma, S.A, 2019.</p> <p><b>Algorithm used:</b> Multivariate Linear Regression, Random Forest Classifier</p>	<p><b>Paper 2: Crop Selection Method to maximize crop yield rate using machine learning technique,</b> Kumar, R., Singh, M.P., Kumar, P. and Singh, J.P., 2015</p> <p><b>Algorithms used:</b> ANN, SVM, KNN, Decision Trees, Random Forest, Gradient Boosted Decision Tree (GBDT), Regularized Greedy Forest (RGF).</p>
<p><b>Paper 3: AN APPROACH FOR PREDICTION OF CROP YIELD USING MACHINE LEARNING AND BIG DATA TECHNIQUES,</b> Palanivel, K. and Surianarayanan, C., 2019.</p> <p><b>Algorithm used:</b> Multiple Linear Regression, ANN, SVM.</p>	<p><b>Paper 4: Crop Prediction using Machine Learning Algorithms,</b> Patil, A., Kokate, S., Patil, P., Panpatil, V. and Sapkal, R., 2020.</p> <p><b>Algorithms used:</b> KNN , Decision tree and Naive Bayes classifier.</p>



---

## CHAPTER 4

### PROJECT REQUIREMENTS SPECIFICATION

#### Introduction

Our project comes under the domain of Precision Agriculture. It helps farmers make informed decisions with regards to the kind of crop they must invest in to get good returns.

#### Project Scope

The aim of this project is to build a predictive model to recommend the most suitable crop to grow based on the various parameters that influence the fertility of the soil.

This project enables the farmers to grow the most suitable crop by factoring in various soil characteristics like N, P, K contents and pH and atmospheric conditions like temperature, humidity and rainfall. This results in greater yield of crop and therefore, stabilizing their financial status.

In this project, the focus is on analysing the existing data and employing suitable models in order to give the best recommendations possible to the farmers. On the other hand, we will not be diving too deep into the implementation of how the data will be extracted but we will be researching about the methods used to collect the same. One of our data sources is only limited to 22 crops but we will make an effort to find more data in order to make this product more robust with regard to its recommendation power.

#### Product Perspective

Usually, farmers and horticulturists don't have a firm idea as to what is the best crop to be grown due to the limited knowledge of the soil parameters and the surrounding conditions. This often results in poor yield of crops which impacts the farmers financially thereby instigating the farmers to take extreme measures like committing suicide.

---

Therefore, our project is a return of favour to all those hard-working men and women who slog all day at the fields so that we are able to consume nutritious food.

### **Product Features**

Our product ingests parameters that describe the soil and its surrounding atmospheric conditions such as N, P, K, temperature, humidity, pH and rainfall as input and outputs the name of the most suitable crop that could be grown in order to achieve maximum yield and have a successful harvest season.

### **Operating Environment**

Our plan is to build a web application that provides a simple user interface for the farmers to interact with in order to make informed decisions with regards to crop selection. We will also be developing a mobile version of this web application in order to increase the usability of this application with all the hardware limitations of each person accounted for. This mobile adaptation will be for android devices only.

### **General Constraints, Assumptions and Dependencies**

- Good quality network connection to use the web application
- Good quality network connection to install the mobile application
- After the installation is complete, network connection is not required to make use of the application.
- A minimum of 4 GB of RAM is a must for the smooth functioning of the application.
- The mobile must have Android as its operating system.

### **Risks**

The data which we are utilizing must be from a reliable source as farmers will be investing their time, efforts and resources in growing the crop recommended by our model with the aim of maximizing their profits. We must also ensure that the mobile application is lightweight so that it can function efficiently even if there's a fluctuation in the network connectivity.



### **Functional Requirements**

- After finishing with all the installation and setup, the user needs to input the soil and atmospheric parameters requested by the application.
- The application validates the parameters input by the user and raises an exception in case of an erroneous input. It then prompts the user to change the value and this continues until all the parameters are correctly input by the user.
- The application passes these values to the machine learning models and returns to the user the name and details of the crop that is most suitable to be grown based on the results obtained from the analytics.

### **Hardware Requirements**

A good quality network connection is necessary for using the web application and also for downloading the mobile application. However, on successfully installing the mobile application, network connection is no longer required to use the same. The mobile running the application is required to have a minimum of 4 GB of RAM and must have android as its operating system.

### **Performance Requirement**

#### **1. Smartphone**

- Android Operating System
- 4 GB of RAM (Minimum)
- 5.5 inch display (Minimum)
- Good quality network connection (Wi-Fi or Cellular Data)

#### **2. PC**

- Windows 10
- 4 GB of RAM (Minimum)
- Good quality network connection (Wi-Fi or Ethernet)
- Web Browser like Chrome, Firefox

## **Security Requirements**

The client will have to create an account in order to make use of our application. The client's login credentials will be stored in an encrypted format and it will be made sure that no other user is able to compromise any other fellow user's account.

---

## CHAPTER 5

# HIGH LEVEL DESIGN DOCUMENT

### 1. Introduction

The main goal of our project is to predict the most suitable crop to a farmer or a horticulturist based on the atmospheric and soil parameter values that are entered in the mobile or web application. This document gives a detailed description of our mobile and web application and the machine learning models being used to predict the right crops.

### 2. Current System

On reading literature surveys that are relevant to our problem statement, we understand that some of the common machine learning techniques that have been employed are logistic regression, support vector machines, naive bayes classifier, and decision trees. Therefore, we will make an attempt to implement other machine learning algorithms that have been left out and also try and incorporate other advanced algorithms such as ensemble models, artificial neural networks bagging and boosting.

### 3. Design Considerations

#### 3.1 Design Goals

- The existing applications require the farmer to create an account using email IDs and by providing card details.
- Some applications even charge for using their product.
- Our application is very simple as we do not collect any personal information from the user. All they have to do is enter their phone number and create a password.
- Once they are logged in, all that farmer has to do is to just input the soil and atmospheric details and let the machine learning models do their job.
- No payment or addition of personal details is required.

- We respect the privacy of our users so we do not collect any personal details. They can use the app directly by just using their phone number to login once and use it indefinitely until they logout.
- Since we aren't collecting any personal information like name, age, salary, card details and other details like farm location or farm ID, so security is also maintained.
- It can be easily downloaded as it is a small application and needs only a minimum of 2GB RAM which is available in almost all the smartphones.
- As soon as the user enters all the values asked for, the application immediately starts to process the data and predicts the perfect crop within a few seconds.

### **3.2 Architecture Choices**

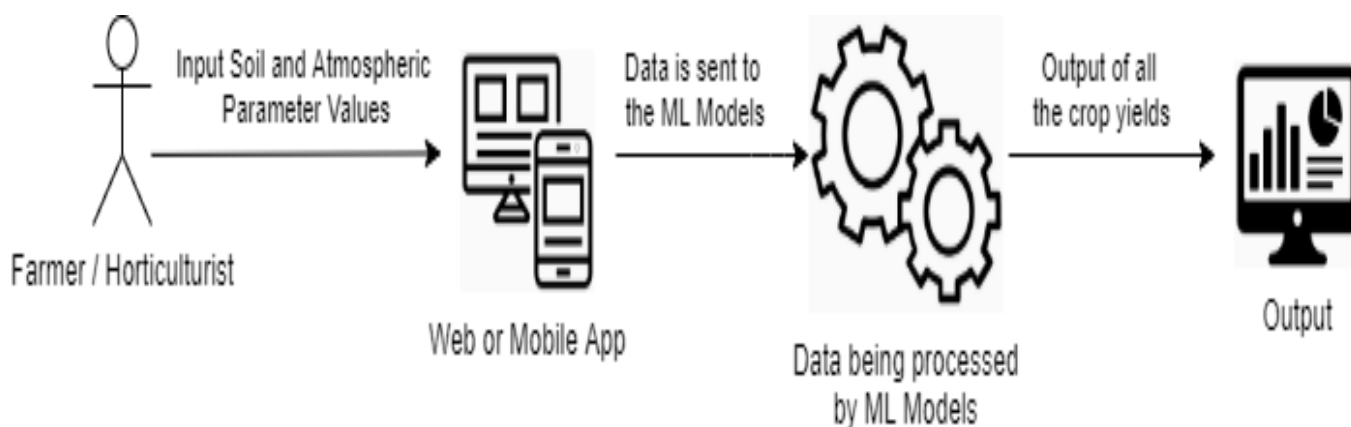
One way was to use the NPK sensors to get the N, P and K contents from the soil. These details would either be displayed using a 7-segment display or it could be displayed directly into the user's mobile application. The model used to extract the soil details would have been robust but the problem was that a crop is not only dependent on Nitrogen, Phosphorus and Potassium but other elements such as Iron, Copper, Manganese, Magnesium etc. All these elements along with fertilizers, insecticides, pesticides and water supply matter a lot when it comes to growing a particular crop. So, the simple NPK sensor can't determine the ideal crop to be grown.

Our application gives the user the flexibility of inserting the values of not only the N, P and K values but also other elements like pH value of soil, Fe, Mn, Mg, Zn, Cu, Ca etc. They also need to enter other parameters like average rainfall, humidity and temperature range of that location. After adding the necessary values, all they have to do is click the submit button and wait for the application to display all the possible crops that can be grown along with their yield percentage.

### 3.3 Constraints, Assumptions and Dependencies

- **Interoperability requirements:** There are no interoperability requirements in our project.
- **Interface/protocol requirements:** All that is necessary is the mobile or the web application to add the soil and atmospheric data.
- **Data repository and distribution requirements:** All the soil and atmospheric details added by users will be stored in the firebase for training the models and improving their predictive capacity.
- There will not be any platform related issues as it is a very simple application and easy to operate.
- **End-user environment:** In the end-user environment, the output of the prediction will be very clear and self-explanatory. It needs no technical knowledge.
- **Hardware or software environment:**
  - There is no hardware component for our project.
  - The software component will be the mobile or web application that the users will use for prediction.

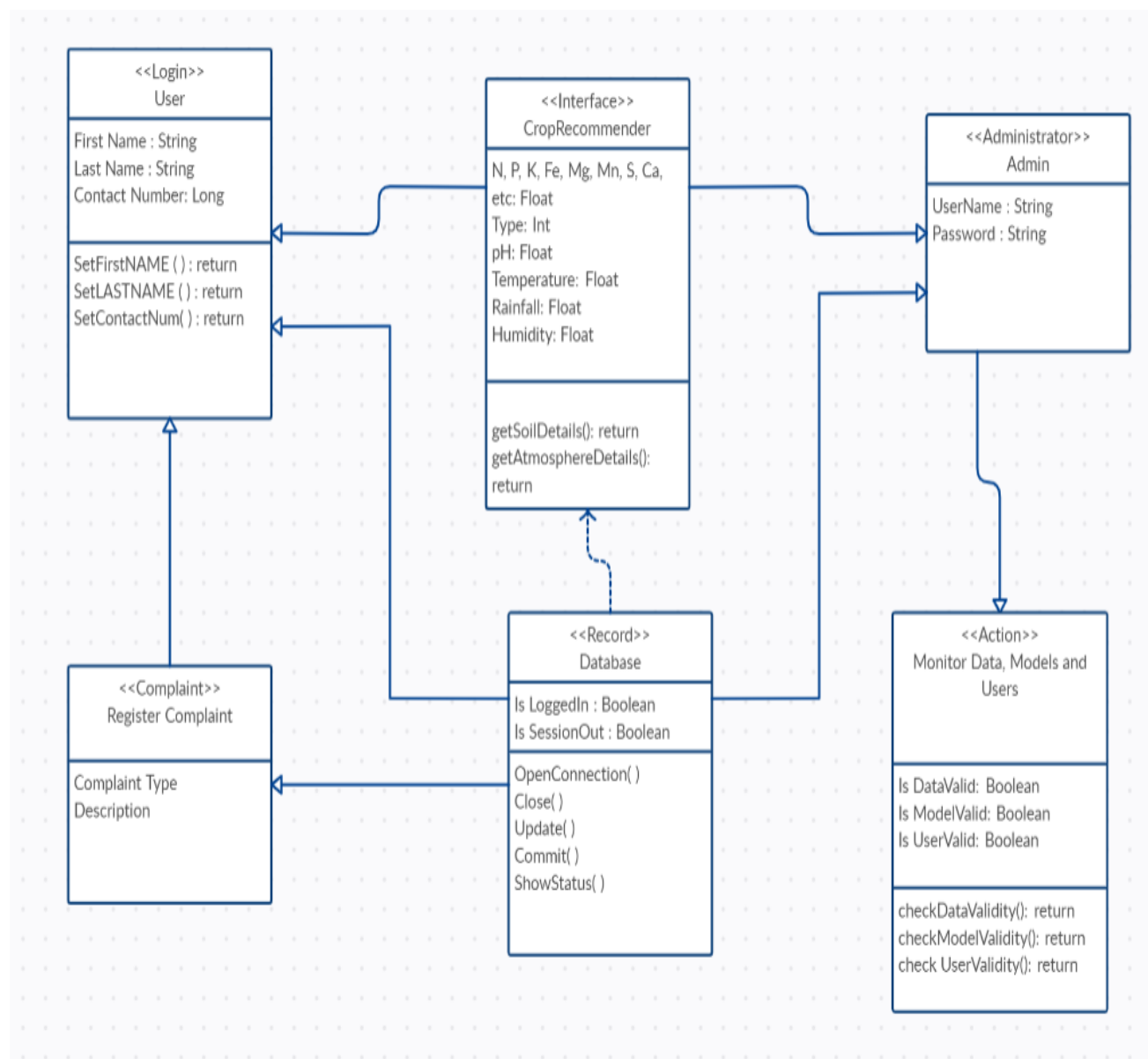
## 4. High Level System Design





## 5. Design Description

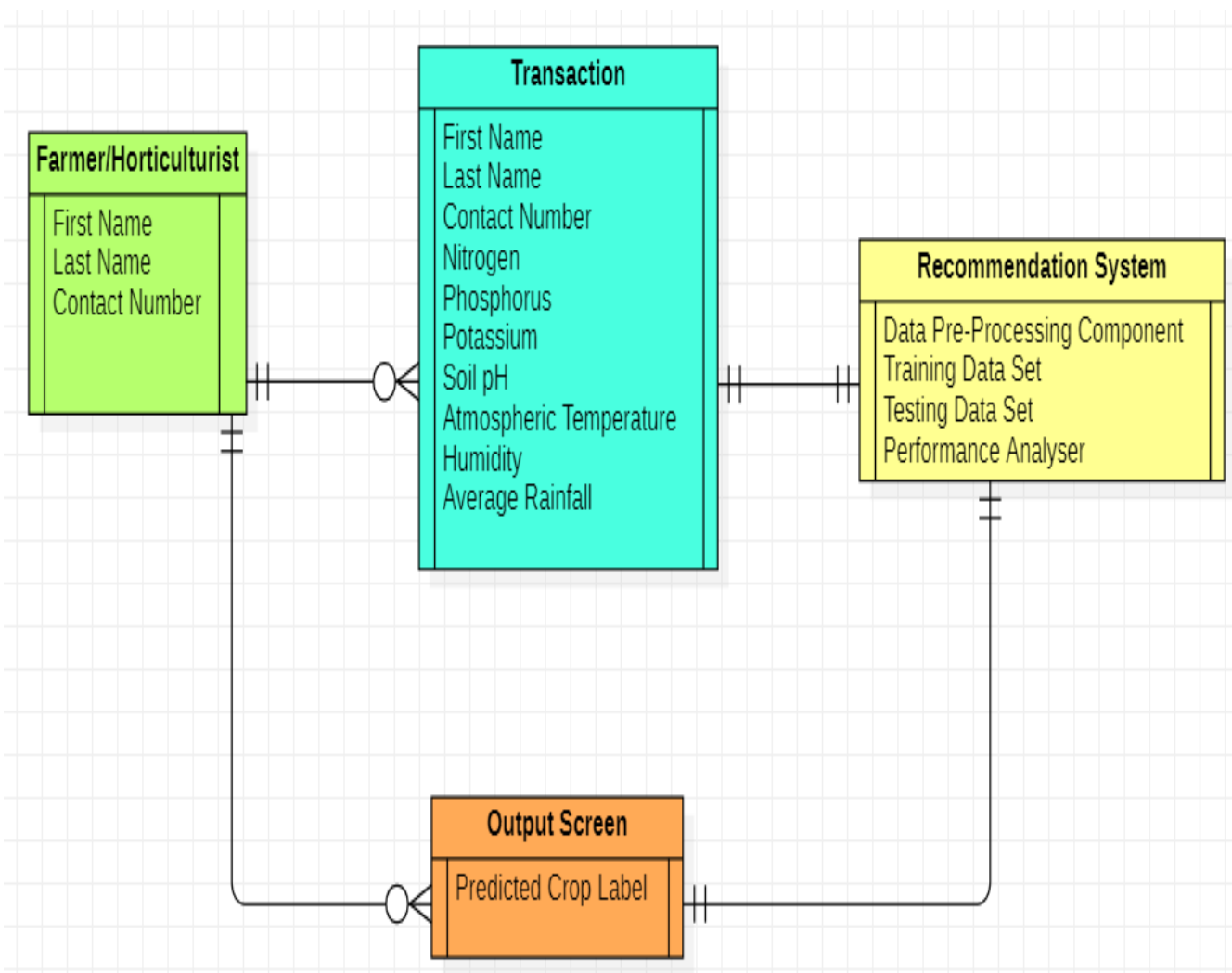
### 5.1 Master Class Diagram




## 5.2 Reusability Considerations

- The machine learning models that will be built in order to make crop recommendations are built using the scikit learn API which happens to be the reusable component in the project.
- The web application and android mobile application that we will be building from scratch will be the non-reusable components of the project.

## 6. ER Diagram

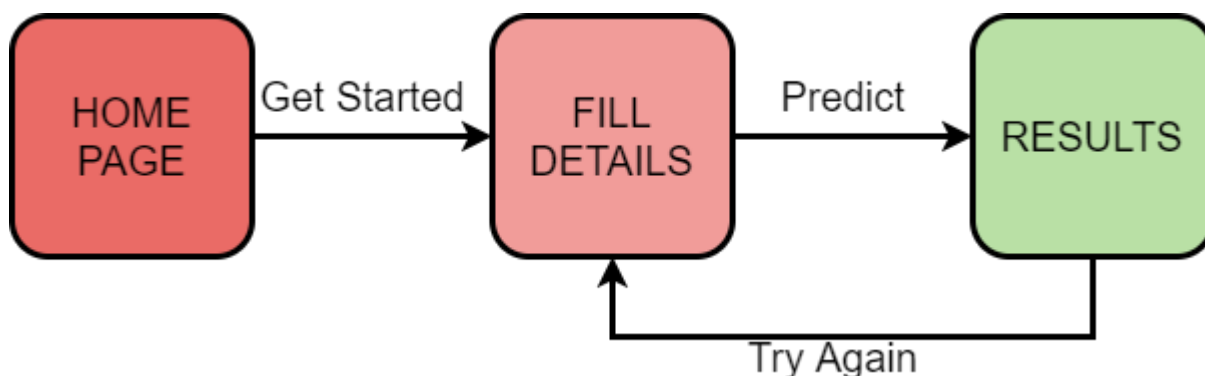


#	Entity	Name	Definition	Type  
<b>ENTITIES</b>				
1.	Client	Farmer/ Horticulturist	The person making use of our application.	Strong
2.	Database	Transactions	Details provided by the user.	Strong
3.	Server	Recommendation System	Data processing is done and the predicted crop is being generated.	Strong
4.	Output	Output Screen	The predicted crop is being displayed.	Strong
#	Attribute	Name	Definition	Type (size)
<b>DATA ELEMENTS</b>				
1.	Client	First Name	Provide the first name of the user.	String
2.		Last Name	Provide the last name of the user.	String

3.		Contact Number	Provide the phone number of the user.	Long
4.	Database	N,P,K,Ca,Mg etc	Salts present in the soil.	Float
5.		pH	Provide the acidic or alkaline contents of the soil.	Float
6.		Temperature	Provide the atmospheric temperature.	Float
7.		Humidity	Provide the atmospheric humidity.	Float
8.		Rainfall	Provide the annual/seasonal average rainfall.	Float
9.	Server	Data pre-processing	Input data is processed.	-
10.		Training phase	70% of the data is trained.	-
11.		Testing phase	30% of the data is tested.	-
12.		Performance Analyzer	Provides the accuracy score.	-
13.	Output	Output Screen	Shows the final output of the predicted crop.	String

## 7. User Interface Diagram

The UI of the system is kept as minimal as possible with no unnecessary Log-In, Sign-Up implementation and recording farmer's data. The farmer shall simply open the website, fill in the soil and location details, hit the submit button and will be presented with our algorithm's best options.



## 8. Report Layouts

Not Applicable.

## 9. External Interfaces

Please refer to **4. High Level System Design, page 6.**

## 10. Packaging and Deployment Diagram

Not Applicable.

## 11. Help

Since the targeted audience for this project comprises mostly farmers from rural as well as urban backgrounds, a User Manual describing the usage of the product becomes very essential which must not be limited by any sorts of language or financial barriers. Thus, a detailed video guide explaining the usage must be made along with written guides in several regional languages.

## **12. Design Details**

### **1. Novelty**

The problem statement that we are solving here is not new and a few solutions have already been built in the past. But the difference in the approaches arises due to the different machine learning algorithms that have been employed to solve the problem. We will be focusing on a few machine learning algorithms, ensemble learning algorithms and artificial neural networks.

### **2. Innovativeness**

We plan to incorporate innovativeness into our project by employing the most accurate machine learning technique and at the trying to minimize any computational overhead to the greatest extent possible. Our goal is also to make the user interface extremely simple and user friendly.

### **3. Interoperability**

We will be ensuring that the exchange of information between the machine learning models and the web application/mobile application is extremely smooth and quick.

### **4. Performance**

We will be making sure that both the mobile and web applications will be functioning very efficiently and thereby producing the required results in quick time.

### **5. Security**

The application requires a very minimal set of user data, which includes First Name, Last Name, Contact Number and City. We will make sure that the user data is stored in an encrypted format and is denied access to any other person apart from the user himself.

## **6. Reliability**

We will make sure that the results generated by the application are very reliable by making use of extremely reliable data in order to build and train the machine learning models.

## **7. Maintainability**

We will be designing our application in a modular approach. Once every component of the application is a module in itself, then the application becomes very maintenance friendly and in the case of any modifications or additions, their implementations can be integrated with the existing application with absolute ease.

## **8. Portability**

Our application could be used both on PCs as well as android smartphones. Therefore, our application is portable.

## CHAPTER 7

### IMPLEMENTATION AND PSEUDOCODE

Models Implemented:

- 1) **Decision Tree Classifier**
- 2) **Naïve Bayes Classifier**
- 3) **K-Nearest Neighbors Classifier**
- 4) **Random Forest Classifier**

#### 1. Decision Tree Algorithm Classifier

### Decision Tree

### Importing the libraries

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

### Importing the dataset

```
In [2]: dataset = pd.read_csv('Crop_recommendation_dataset.csv')
```

```
In [3]: dataset.head()
```

```
Out[3]:
```

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



## Extracting Target Variable and independent variables

```
In [5]: X = dataset.iloc[:, :-1].values  
        y = dataset.iloc[:, -1].values
```

## Splitting the dataset into training set and test set

```
In [6]: crop_labels = list(dataset.label)  
        from sklearn.model_selection import train_test_split  
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.4, random_state = 0, stratify = crop_labels)
```

## Feature Scaling

```
In [7]: from sklearn.preprocessing import StandardScaler  
        sc = StandardScaler()  
        X_train = sc.fit_transform(X_train)  
        X_test = sc.transform(X_test)
```

## Training the Decision Tree model on the Training set

```
In [8]: from sklearn.tree import DecisionTreeClassifier  
        classifier = DecisionTreeClassifier(criterion = 'entropy', random_state = 0)  
        classifier.fit(X_train, y_train)
```

```
Out[8]: DecisionTreeClassifier(criterion='entropy', random_state=0)
```

## Predicting the Test set results

```
In [9]: y_pred = classifier.predict(X_test)
```

## Accuracy Score

```
In [10]: from sklearn.metrics import accuracy_score, confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print(cm, '\n')
ac = accuracy_score(y_test, y_pred)
print("Accuracy Score: ", ac)
```

```
[[40  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0 40  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0 36  0  0  0  0  0  0  0  0  4  0  0  0  0  0  0  0  0]
 [ 0  0  0 40  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0 40  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0 40  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0 40  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0 40  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  1  0  0 37  0  0  0  0  0  0  0  0  0  0  2]
 [ 0  0  0  0  0  0  0  0  0 40  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0 40  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0 40  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0 40  0  0  0  0  0  0  0]
 [ 0  0  2  0  0  0  0  0  0  0  1  3  0 34  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0 40  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0 40  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 40  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 40  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 40  0  0]
 [ 0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0 40  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 38  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0 38]]
```

Accuracy Score: 0.9806818181818182

## Applying k-Fold Cross Validation

```
In [11]: from sklearn.model_selection import cross_val_score
accuracies = cross_val_score(estimator = classifier, X = X_train, y = y_train, cv = 10)
print("Accuracy: {:.2f} %".format(accuracies.mean()*100))
print("Standard Deviation: {:.2f} %".format(accuracies.std()*100))
```

Accuracy: 98.03 %

Standard Deviation: 1.28 %

## 2. Naïve Bayes Classifier

# Naive Bayes

## Importing the libraries

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

## Importing the dataset

```
In [2]: dataset = pd.read_csv('Crop_recommendation_dataset.csv')
```

```
In [3]: dataset.head()
```

```
Out[3]:
```

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

## Extracting Target Variable and independent variables

```
In [5]: X = dataset.iloc[:, :-1].values  
        y = dataset.iloc[:, -1].values
```

## Splitting the dataset into training set and test set

```
In [6]: crop_labels = list(dataset.label)  
        from sklearn.model_selection import train_test_split  
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.4, random_state = 0, stratify = crop_labels)
```

## Feature Scaling

```
In [7]: from sklearn.preprocessing import StandardScaler  
        sc = StandardScaler()  
        X_train = sc.fit_transform(X_train)  
        X_test = sc.transform(X_test)
```

## Training the Naive Bayes model on the Training set

```
In [8]: from sklearn.naive_bayes import GaussianNB  
        classifier = GaussianNB()  
        classifier.fit(X_train, y_train)
```

```
Out[8]: GaussianNB()
```

## Predicting the Test set results

```
In [9]: y_pred = classifier.predict(X_test)
```

## Accuracy Score

```
In [10]: from sklearn.metrics import accuracy_score, confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print(cm, '\n')
ac = accuracy_score(y_test, y_pred)
print("Accuracy Score: ", ac)
```

```
[[40  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0 40  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0 40  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0 40  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0 40  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0 40  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0 40  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0 40  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0 40  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0 40  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0 40  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  1  0  0  0  0 39  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0 40  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0 40  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0 40  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 40  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 40  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 40  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 40  0  0]
 [ 0  0  0  0  0  0  0  0  3  0  0  0  0  0  0  0  0  0  0 37  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 40]]
```

Accuracy Score: 0.9954545454545455

## Applying k-Fold Cross Validation

```
In [12]: from sklearn.model_selection import cross_val_score
accuracies = cross_val_score(estimator = classifier, X = X_train, y = y_train, cv = 10)
print("Accuracy: {:.2f} %".format(accuracies.mean()*100))
print("Standard Deviation: {:.2f} %".format(accuracies.std()*100))
```

Accuracy: 99.47 %  
Standard Deviation: 0.59 %

### 3. K-Nearest Neighbors Classifier

## K-Nearest Neighbors

## Importing the libraries

```
In [1]: import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt
```

## Importing the dataset

```
In [2]: dataset = pd.read_csv('Crop_recommendation_dataset.csv')
```

```
In [3]: dataset.head()
```

```
Out[3]:
```

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

## Extracting Target Variable and independent variables

```
In [5]: X = dataset.iloc[:, :-1].values  
        y = dataset.iloc[:, -1].values
```

## Splitting the dataset into training set and test set

```
In [6]: crop_labels = list(dataset.label)  
        from sklearn.model_selection import train_test_split  
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.4, random_state = 0, stratify = crop_labels)
```

## Feature Scaling

```
In [7]: from sklearn.preprocessing import StandardScaler  
        sc = StandardScaler()  
        X_train = sc.fit_transform(X_train)  
        X_test = sc.transform(X_test)
```

## Training the K-NN model on the Training set

```
In [8]: from sklearn.neighbors import KNeighborsClassifier  
        classifier = KNeighborsClassifier(n_neighbors = 5, weights = 'distance', metric = 'minkowski', p = 2)  
        classifier.fit(X_train, y_train)
```

```
Out[8]: KNeighborsClassifier(weights='distance')
```

## Predicting the Test set results

```
In [9]: y_pred = classifier.predict(X_test)
```

## Accuracy Score

```
In [10]: from sklearn.metrics import accuracy_score, confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print(cm, '\n')
ac = accuracy_score(y_test, y_pred)
print("Accuracy Score: ", ac)
```

```
[[40  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0 40  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0 37  0  0  0  0  0  0  0  3  0  0  0  0  0  0  0  0  0]
 [ 0  0  0 40  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0 40  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0 40  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0 40  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0 40  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  1  0  0 39  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0 40  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  1  0  0  0  0  0  0  0 39  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  2  0  0  0  0 38  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0 40  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  3  0  0 37  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0 40  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 40  0  0  0  0]
 [ 0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0 37  0  0  2  0]
 [ 0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0 36  0  0  3  0]
 [ 0  0  0  0  0  0  0  0  0  2  0  0  2  1  0  0  0  0 35  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 40  0  0]
 [ 0  0  0  0  0  0  0  0  8  0  0  0  0  0  0  0  0  0  0 32  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0 39]]
```

Accuracy Score: 0.9647727272727272

## Applying k-Fold Cross Validation

```
In [11]: from sklearn.model_selection import cross_val_score
accuracies = cross_val_score(estimator = classifier, X = X_train, y = y_train, cv = 10)
print("Accuracy: {:.2f} %".format(accuracies.mean()*100))
print("Standard Deviation: {:.2f} %".format(accuracies.std()*100))
```

Accuracy: 97.12 %

Standard Deviation: 1.21 %



#### 4. Random Forest Classifier

## Random Forests

### Importing the libraries

```
In [1]: import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt
```

### Importing the dataset

```
In [2]: dataset = pd.read_csv('Crop_recommendation_dataset.csv')
```

```
In [3]: dataset.head()
```

```
Out[3]:
```

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

## Extracting Target Variable and independent variables

```
In [5]: X = dataset.iloc[:, :-1].values  
        y = dataset.iloc[:, -1].values
```

## Splitting the dataset into training set and test set

```
In [6]: crop_labels = list(dataset.label)  
        from sklearn.model_selection import train_test_split  
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.4, random_state = 0, stratify = crop_labels)
```

## Feature Scaling

```
In [7]: from sklearn.preprocessing import StandardScaler  
        sc = StandardScaler()  
        X_train = sc.fit_transform(X_train)  
        X_test = sc.transform(X_test)
```

## Training the Random Forests model on the Training set

```
In [8]: from sklearn.ensemble import RandomForestClassifier  
        classifier = RandomForestClassifier(n_estimators = 15, criterion = 'entropy', random_state = 0)  
        classifier.fit(X_train, y_train)
```

```
Out[8]: RandomForestClassifier(criterion='entropy', n_estimators=15, random_state=0)
```

## Predicting the Test set results

```
In [9]: y_pred = classifier.predict(X_test)
```

## Accuracy Score

```
In [10]: from sklearn.metrics import accuracy_score, confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print(cm, '\n')
ac = accuracy_score(y_test, y_pred)
print("Accuracy Score: ", ac)
```

```
[[40  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0 40  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0 40  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0 40  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0 40  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0 40  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0 40  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0 40  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0 40  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0 40  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0 40  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0 40  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0 40  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0 40  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0 40  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 40  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 40  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 40  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 40  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 40]]
```

Accuracy Score: 0.9943181818181818

## Applying k-Fold Cross Validation

```
In [11]: from sklearn.model_selection import cross_val_score
accuracies = cross_val_score(estimator = classifier, X = X_train, y = y_train, cv = 10)
print("Accuracy: {:.2f} %".format(accuracies.mean()*100))
print("Standard Deviation: {:.2f} %".format(accuracies.std()*100))
```

Accuracy: 99.17 %  
Standard Deviation: 0.79 %

On applying K-Fold Cross Validation to all the 4 models, it is observed that the **Naïve Bayes Classifier** has the **highest accuracy score of 99.47 %**

---

## CHAPTER 8

### CONCLUSION OF CAPSTONE PROJECT PHASE-1

Through the literature survey that we conducted, we learnt about the various machine learning algorithms that have been employed to recommend the most suited crop given the required soil and atmospheric properties of a specific location. We have also implemented 4 machine learning algorithms namely Decision Trees, K-Nearest Neighbors, Naïve Bayes Classifier and Random Forest Classifier. Among these models, the Naïve Bayes Classifier gave the maximum accuracy of 99.47 %. We got in touch with **Dr.K Ganesha Raj, the General Manager of RRSC South, ISRO** seeking for useful data related to our project. They assigned **Dr.Ramasubramoniam, Soil and Agri Scientist** to guide us through our project. The scientist will be providing us with area specific soil and crop data gathered by ISRO over the past years to carry out our project. We received the dataset for the state of Kerala from Dr. Ramasubramoniam and have performed some basic analysis in order to understand its usage. On thorough searching, we also came across a relevant dataset for the state of Andhra Pradesh covering 13 districts.

---

## **CHAPTER 9**

### **PLAN OF WORK FOR CAPSTONE PROJECT PHASE-2**

In phase-2 of this project, we will make an attempt to implement some of the ensemble machine learning algorithms such as AdaBoost and XGBoost and also an Artificial Neural Networks model in order to achieve the recommendation of crops with a good accuracy score. On completing the implementation of these models, we will be building a website and an associated mobile application in order to enable access to this crop recommendation system with user friendly and elegant graphical user interface.

---

## REFERENCE / BIBLIOGRAPHY

- [1] Keerthan Kumar, T.G., Shubha, C. and Sushma, S.A., **Random Forest Algorithm for Soil Fertility Prediction and Grading Using Machine Learning**. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 2019.
- [2] Kumar, R., Singh, M.P., Kumar, P. and Singh, J.P., 2015, May. **Crop Selection Method to maximize crop yield rate using machine learning technique**. In 2015 international conference on smart technologies and management for computing, communication, controls, energy and materials (ICSTM) (pp. 138-145). IEEE.
- [3] Palanivel, K. and Surianarayanan, C., 2019. **An approach for prediction of crop yield using machine learning and big data techniques**. *International Journal of Computer Engineering and Technology*, 10(3), pp.110-118.
- [4] Patil, A., Kokate, S., Patil, P., Panpatil, V. and Sapkal, R., 2020. **Crop Prediction using Machine Learning Algorithms**. *International Journal of Advancements in Engineering & Technology*, 1(1), pp.1-8.

---

## APPENDIX A DEFINITIONS, ACRONYMS AND ABBREVIATIONS

GPS: Global Positioning System

pH: power of Hydrogen

EC: Electrical Conductivity

Mg: Magnesium

K: Potassium

AI: Artificial Intelligence

ANN: Artificial Neural Networks

N: Nitrogen

P: Phosphorus

K: Potassium

Ca: Calcium

S: Sulphur

Mg: Magnesium

C: Carbon

O: Oxygen

H: Hydrogen

Fe: Iron

B: Boron

Cl: Chlorine

Mn: Manganese

Zn: Zinc

Cu: Copper

Mo: Molybdenum

Ni: Nickel





