**Yield Predict Recommendation**

Table Of Contents

[ABSTRACT 4](#_Toc123845940)

[INTRODUCTION 7](#_Toc123845941)

[Project Scope 11](#_Toc123845942)

[Purpose 11](#_Toc123845943)

[Objectives 12](#_Toc123845944)

[Motivation 12](#_Toc123845945)

[Literature Survey 13](#_Toc123845946)

[Paper 1 13](#_Toc123845947)

[Paper 2 16](#_Toc123845948)

[Paper 3 18](#_Toc123845949)

[Paper 4 21](#_Toc123845950)

[Literature Survey Comparison 25](#_Toc123845951)

[Data Visualization 25](#_Toc123845952)

[Columns description 26](#_Toc123845953)

[Exploratory Data Analysis 27](#_Toc123845954)

[Comparative Study 27](#_Toc123845955)

[Univariate Analysis 28](#_Toc123845956)

[Kernel Density Estimation 30](#_Toc123845957)

[Boxplot 31](#_Toc123845958)

[Data Preprocessing 33](#_Toc123845959)

[Random Forest 33](#_Toc123845960)

[METHODOLOGY 35](#_Toc123845961)

[System Architecture 35](#_Toc123845962)

[Datasets 36](#_Toc123845963)

[Model Selection and Performance Evaluation 38](#_Toc123845964)

[Crop Recommendation 38](#_Toc123845965)

[Architecture Diagram 39](#_Toc123845966)

[Tools and Technologies 39](#_Toc123845967)

[CODE 40](#_Toc123845968)

[Deployed website 43](#_Toc123845969)

[Results 44](#_Toc123845970)

[Decision Tree 44](#_Toc123845971)

[AdaBoost 45](#_Toc123845972)

[Gradient Boost 46](#_Toc123845973)

[XGBoost 47](#_Toc123845974)

[Logistic Regression 48](#_Toc123845975)

[Gaussian Naive Bayes 48](#_Toc123845976)

[K-nearest Neighbor 49](#_Toc123845977)

[Stochastic Gradient Descent 50](#_Toc123845978)

[Support Vector Machine 51](#_Toc123845979)

[Accuracy Comparison: 53](#_Toc123845980)

[High Accuracies 55](#_Toc123845981)

[Conclusion 56](#_Toc123845982)

[Future Scope 57](#_Toc123845983)

[SCREENSHOT: 57](#_Toc123845984)

[Appendix: Pilot Project – 1 59](#_Toc123845985)

[Abstract 59](#_Toc123845986)

[Introduction 60](#_Toc123845987)

[Motivation 61](#_Toc123845988)

[Problem   Statement 61](#_Toc123845989)

[Objective 62](#_Toc123845990)

[Background 63](#_Toc123845991)

[Criteria/Constraints: 63](#_Toc123845992)

[Crop Recommendation 63](#_Toc123845993)

[Predictions 63](#_Toc123845994)

[The capability of Application: 65](#_Toc123845995)

[Users Handling: 65](#_Toc123845996)

[Methodology 65](#_Toc123845997)

[Dataset 65](#_Toc123845998)

[Flowchart Steps: 66](#_Toc123845999)

[Idea: 66](#_Toc123846000)

[Existing System 66](#_Toc123846001)

[Proposed Solution 67](#_Toc123846002)

[Application 67](#_Toc123846003)

[EXPERIMENTAL INVESTIGATIONS: 68](#_Toc123846004)

[Tools/ Technologies used 68](#_Toc123846005)

[Software requirements: 68](#_Toc123846006)

[Languages: 68](#_Toc123846007)

[Advantages 68](#_Toc123846008)

[Disadvantages 69](#_Toc123846009)

[Appendix: Pilot Project - 2 70](#_Toc123846010)

[METHODOLOGY 70](#_Toc123846011)

[Gathering the Dataset 70](#_Toc123846012)

[Data Analysis 70](#_Toc123846013)

[Visualizing the soil's nitrogen-and-phosphorus ratio 70](#_Toc123846014)

[Visualizing the Ratio of Potassium and Temperature in the soil 71](#_Toc123846015)

[Visualizing the Humidity and pH in the soil 72](#_Toc123846016)

[Visualizing the Rainfall 72](#_Toc123846017)

[Module workflow 73](#_Toc123846018)

[Methods 74](#_Toc123846019)

[Decision Tree 75](#_Toc123846020)

[Gaussian Naive Bayes 77](#_Toc123846021)

[Support Vector Machine (SVM) 78](#_Toc123846022)

[Logistic Regression 80](#_Toc123846023)

[Random Forest 81](#_Toc123846024)

[XGBoost 83](#_Toc123846025)

[Accuracy Comparison 85](#_Toc123846026)

[Accuracy differentiates between all the methods: 85](#_Toc123846027)

[Result 86](#_Toc123846028)

[Making a prediction 86](#_Toc123846029)

[Conclusion 87](#_Toc123846030)

[REFERENCES 88](#_Toc123846031)

# ABSTRACT

Since there have been climate changes that have resulted in an increasing amount of unexpected rainfalls, par below temperatures, and heatwaves in the region, which result in significant loss of ecosystem. Machine learning has helped develop various utility tools to tackle world problems. This problem of agriculture can be solved by using various ML algorithms. This project aims to create two things.

Agriculture has been severely impacted by climate change due to unpredictable weather conditions. Pick- the right crop to grow is one of the most critical decisions a farmer needs to make to gain maximum yield. Traditional techniques are not very effective due to the current uncertainties involved. However, a modern farming technique called Precision agriculture leverages the Internet of Things and Machine Learning techniques to collect real-time data and further analyze the data to make better decisions.

A crop recommendation system applies classification techniques using Machine learning algorithms on real-time data such as soil parameters, temperature, humidity, etc. to recommend a crop for farmers to sow, helping them directly. Gaussian Naive Bayes, Neural Networks, Logistic Regression, Support Vector Machine, etc, are amongst the list of algorithms used to develop models for this crop recommendation system. The performance of these models is compared based on their accuracy level, Time, and space complexity. The algorithm that provides the best results is chosen for the application.

India has a large agricultural sector supporting the majority of the population for their livelihood. It employs almost half of India’s working-class people. Every year huge resources in terms of land and fertilizer are used for the production of food. Most of the fertilizer being used in farms today is being wasted due to the incorrect use of a type of fertilizer. We are developing a machine learning model that analyses various soil features like Calcium, Iron concentration, pH of the soil, and various other features to recommend the type of fertilizer to use and also the type of crop needed for a particular area for maximum benefit.

Agriculture is one of the main sources of supply for human needs. It is a key employment and one of the most important industrial segments in many countries. When making farming decisions, farmers frequently use conventional approaches based on naked-eye inspections. But, due to globalization an increase in population, and climate change a lot of uncertainty is incorporated. Traditional techniques may not consider and cope with such rapid changes and drastic effects. Hence, a more robust technique is necessary where the data about the features affecting farming are collected periodically and monitored in real-time.

Precision agriculture is a modern farming technique that leverages the Internet of Things (IoT) and Machine-Learning techniques. It applies IoT methods to collect real-time data on several features such as soil parameters, temperature, humidity, etc, these parameters are collected using sensor nodes and are stored in the cloud for monitoring. Several supervised and unsupervised machine learning algorithms can be applied to this data to further analyze and perform predictive analysis to help farmers make informed decisions.

One of the most important decisions a farmer makes is to pick the right crop to sow. We can use the Precision agriculture techniques discussed above to help with this decision by developing a Crop Recommendation System.

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 includes seeders, sprayers, etc. to achieve optimal distribution of resources. In order Toe, 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 the soil can wirelessly transmit data without the need for 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 for any prior experience. These drones are equipped with multispectral cameras that can capture multiple images of the field.

The Basic idea behind this system is to collect field data from which features are extracted and processed. On this processed data, several Machine learning algorithms are used to develop several prediction models. Comparing these predictive models, the best model is used for our Crop recommendation system. This system will then recommend the best crop to sow under the given conditions in real time.

A crop recommendation system and a Plant disease identification system are embedded into a single website. The datasets were publicly available over the internet. Once the features for task one are extracted, then the dataset is trained on five different algorithms - logistic regression, decision tree, support-vector-machine (SVM), multi-layer perceptron, and random forest. Random forest achieved an accuracy of 99.31%.

# INTRODUCTION

Machine learning can help humans solve problems that are not easily solvable by humans. Machine learning is to be used to solve tasks like classification, prediction, identification, etc., and can be applied to a wide range of other fields such as Agriculture, sports, trade and business, and so on. This project aims at building a website focused on the agriculture sector, solving two significant issues crop recommendation and crop disease identification. The method used to solve these problems is by training models on datasets available over the internet and comparing them. Models with reasonable accuracy are embedded into the website, which can be then deployed on the cloud.

Our Model will help in predicting the farm yield and it will also be giving the recommendation for the crop for farming. It will help the farmers and give better guidance to add to the traditional farming techniques that are currently in use. It will help the farmers to get the desired results and help them decide which crop is the best for cultivation according to the available resources. Other than giving the right crop it will also be focusing on providing the farmers with the right number of seeds, water, pesticides, and fertilizers that should be used for maximum production.

Agriculture is one of the oldest and noblest professions in India. It plays a critical role in the global economy and is certainly the largest livelihood provider in India. The production of Agroecology is minimal. As the demand for food is growing exponentially, brilliant minds throughout the world are working on finding new methods to help farmers to fulfill the ever-rising need for agricultural production.

This project aims to take in features like annual rainfall, pH of the soil, humidity, and the temperature of a particular land area and train a machine learning model for accurate prediction of the type of crop to be sown to get a good yield from the crop. We also build a model to analyze soil features and recommend the type of fertilizer to be used.

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 are 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. Precision farming aims to develop a decision support system for the management of a farm with the goal being the 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.

Agriculture is practiced across India as an occupation and is a source of livelihood for 58% of the population of India. This is a huge figure. But many farmers in India are considered poor as agriculture is not a profitable field for many. The reasons for this are various but mostly it is because of practicing traditional methods of agriculture. Most of them depend on the traditional seasonal farming that is continued over the years to grow crops. But now the times have changed the cycle of traditional seasons is changing because of climate change and global warming. Unpredicted rainfall, excess or lack of heat, and excess or lack of cold are a few parameters that are destroying the crops. The wrong choice of crops at unsuited times with unsuited conditions ruins the entire yield. Thus, causing huge losses. Some farmers also commit suicide in India because of losses incurred. AI-assisted farming for crop recommendation and yield prediction will assist farmers in recommending to take a particular crop based on various parameters like current climate, soil conditions, rainfall, etc. It will predict the approximate yield of that crop and also the approximate revenue.

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 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 the 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, technological advancements and their integration into the field of agriculture gave birth to the third agricultural revolution which was widely known as Precision Agriculture. The other 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 amounts of satellite data, precision agriculture has become very popular and has seen 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.

A lot of sensors have been predominantly wireless ones that have been used 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 to come meaningful insights that can be used to manage farms efficiently.

Coming to the complexity involved in practicing precision farming, is a little complex because most of the technologies that are being used are new and therefore require a skilled workforce 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, technology has simple technological solutions 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 of precision farming in your farms or agricultural fields. At present, some of the sophisticated equipment and software are quite expensive and therefore precision farming technologies are mostly seen only in large farms owned by affluent farmers.

But it is a well-known fact that with Agriculture 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 to make use of it because it is extremely profitable. In one of

the blog the name onesoil.ai, we learned that African farmers can save 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.

Farmers being a crucial part of our society are not given their due credit. They face a lot of hardships while using the traditional methods of farming in today’s technologically centric world. Farmers find it difficult to understand the technicality of the advanced science used to ease their hardships. The parameters generally used in other research papers focus more on factors like soil type and the nutrients they possess which are complicated for the farmers to comprehend. So, in this paper, we have used parameters like district, rainfall, temperature, and area which are easier for the farmers to understand.

In addition to crop yield, we are also helping them in predicting the right crop for cultivation. Selecting a crop for farming is one of the essential decisions farmers have to make as their whole revenue and yield depend on this decision. We have used the above-mentioned methods to minimize the errors and give better predictions which in turn would help the farmers.

# Project Scope

This project aims 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, and K contents and pH and atmospheric conditions like temperature, humidity, and rainfall. This results in a greater yield of crops and therefore, stabilizes their financial status.

In this project, the focus is on analyzing the existing data and employing suitable models 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 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 for the product more runabout to eastbound power.

# Purpose

There are many problems currently associated with the agriculture sector. The demand for products is only going up every day as the population increases. The crop production should also increase at the same rate. However, because of the unpredictable nature of the atmosphere, crop growth and production get hampered. Unwanted rain washes away the crops hence the loss of productivity. Even excessive heat damages the crops. All of this could be changed to an extent if we have the predicted nature of the atmosphere over a certain period. Considering these predicted values of the atmosphere like temperature, rainfall, humidity, etc. mapping it with the atmospheric conditions required by a crop, we can find out which crop will be the best suitable to be grown in that particular region where the farmer lives in. Even the soil parameters are equally important for the growth of any plant. If we know the current soil conditions, we can map it with the soil conditions required for a certain crop and find out which crop is the best suited for the farmer. Such cases are an example of how we can benefit the farmer by trying our best to make use of Artificial Intelligence to recommend crops that be profitable to the farmers. Such systems will help to pre-empt the weather and soil conditions that will recommend a crop that will not because any loss listlessness even in adverse weather conditions. This will also indirectly reduce farmer suicides as life is very precious. Hence such systems have a huge purpose to be played in the agricultural industry.

# Objectives

* Data Preprocessing
* Data Visualization
* Using various algorithms and comparing the accuracy
* Participants must make use of IBM services (Watson studio / Auto ai/ build machine learning model) and can use any SDK to create a web Interface.
* Build a Machine Learning model that predictand its yield, and revenue and also recommend crops using IBM capabilities.
* Integrate the model with UI (User interface)
* Build a Machine Learning model that predicts yield, and revenue and also recommends crops using IBM capabilities.

# Motivation

Climate change has been quite effective over the past five years. Less knowledge about scientific ways of farming also leads to wrong decisions in the selection of crops. Farmers often tend to rely on experiences that are limited and also full of errors. Overall, the agriculture industry suffers a huge loss due to improper usage of knowledge such as soil constituents present, pH of the soil, and early detection of diseases of plants. This problem can be solved by making proper use of technology. With the help of machine learning and the web, this solution can reach every individual having access to a mobile phone with an internet connection on it. Our application provides a digital application to the farmers to help them evaluate the crop yield and decide which crop to grow.

Farmers, who are an important component of our society, are undervalued. In today's technologically focused society, they confront several challenges while employing traditional farming practices. Farmers struggle to comprehend the complexity of the modern technology employed to alleviate their sufferings. The criteria commonly employed in other study articles place a greater emphasis on aspects like soil type and nutrient content, which are difficult for farmers to grasp. So, in this study, we've chosen factors like district, rainfall, temperature, and area that are more easily understood by farmers.

# Literature Survey

There are many attempts made to tackle the problem of crop recommendation and plant disease classification. G Chauhan and A. Chaudhary proposed ways to recommend crops based on soil type and used random forest and decision trees to make the prediction. This showed that the task of predicting crops based on land patterns would be helpful via a random forest classifier.

We are going to give a brief idea of the existing farming system that uses conventional methods of farming, the drawbacks that come along with it due to its manual setting and labor, and how our proposed system can overcome them. We also talk about the limitations that our system possesses.

## Paper 1

The authors of [1] have proposed a machine-learning 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 values was employed to predict the rank of soil and on applying a few classification algorithms for crop recommendation, they found that Random Forest was the most accurate model.

To have a good yield, the soil must be rich in the required nutrients. So, the main goal of 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 the independent variables, and the grade of the soil is considered the dependent variable. So, a Multivariate 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:

1. Xi = vector of independent variables
2. he = hypothesis function
3. Yi = True value of the response variables
4. m = normalization parameter

The Gradient Descent Algorithm was adopted to minimize the cost function. Then hypothesis testing was carried out on the test dataset to check for the model’s correctness and efficiency and the RMSE value was used to determine the 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 Trees, 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 feature is looked for from the subset of the remaining features thereby generating a highly accurate classifier with a wide diversity.

To split a node, only a select 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 for each of these samples, trees were fit.

Then voting is performed among all the predictions output by all the trees 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 hundred to several thousand and this number is heavily dependent on the characteristics of the training data set.

**Learnings from [1] are**

1. The Random Forest Algorithm is based on ensemble learning and proved to be a very effective algorithm for classification.
2. 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 to give the instance its final classification.
3. Each tree as an individual entity might not be ideal, but as a group, they can perform well.
4. Since there are numerous trees, the existence of any errors or uncertainties associated with any of the trees is taken care of by this algorithm.

## Paper 2

The yield rate of crops is dependent on two broad factors, the first being the genetic development of seeds which lies more in 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 toward 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 machine learning models over various divariousike soil properties such as Nitrogen, Phosphate, and Potassium contents, pH properties of the soil, weather conditions, and 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 in this field of research until this paper was published, and the paper aimed at trying them out as well and placing detailed comparative analysis for the readers. These techniques included GDBT (Gradient Boosted Decision Tree) and RGF (Regularized Greedy Forest).

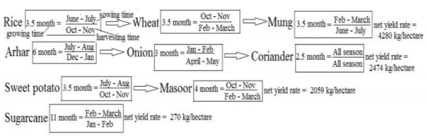
The method is composed of 2 significantly differentiable parts that work together. The first part uses machine learning models like Artificial Neural Network (ANN), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Decision Tree Learning, Random Forest, Gradient Boosted Decision Tree (GBDT), Regularized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

Crops can be classified as:

* Seasonal crops
* Whole-year crops
* Short-time plantation crops
* Long-time plantation crops

A 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 a waste of farmers’ resources. Below is the algorithm used in this part of the research work.

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 from. The algorithm is provided with all of this data, that is, each crop’s seasonal information, as well as the 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 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 in detail. The first one is the limitation of the geographical diversity of the data collected. The data is collected from a single farmer residing in the Patna district of Bihar, India. This data and results might be appropriate for the said district, but cannot be extrapolated to the 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, and climatic and weather conditions. Therefore, including these factors as well into a future study holds a lot of potentials

## 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 a 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 may be 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. The use of precision farming can give more accurate results due to the high inspection and efforts in a small area. Since we can get almost accurate values of soil parameters, if weather conditions are known more accurately beforehand, farmers can adapt accordingly and try growing a suitable crop.

Employing machine learning can give us insights into 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 issue of changing the weights to obtain the desired value as 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 is the metrics employed by the authors to validate the outcomes.

|  |  |
| --- | --- |
| **Metrics** | **Formula** |
| Root Mean Squared Error |  |
| Mean Squared Error |  |
| Mean Absolute Error |  |

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 the non-availability of data and the 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 crop and weed growth in the surrounding areas.

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 are 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 to improve the results. They have compared the prediction of the ideal crop by using different models to get a better understanding of how to use ML

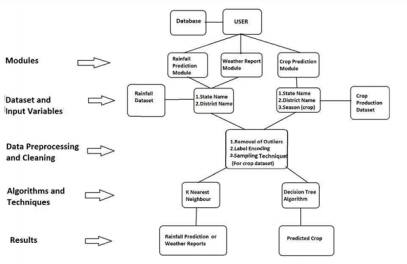
techniques in the future.

To increase the 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 fertilizers, 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:

1. Seasonal
2. Whole Year
3. Short plantation period
4. Long-time plantation

The data 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 to create soil 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 are also taken into account. This is shown in their work flowchart below.

The data must include:

* Soil Parameters
* Soil Type
* Soil pH
* Humidity
* Temperature
* Wind
* Rainfall
* Production
* Cost of cultivation
* Previous year yield results

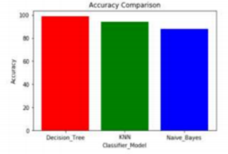
The data is pre-processed and fed into KNN, Decision tree, and Naive Bayes classifier and

the results from all of these are compared.

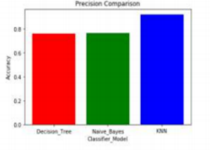
(The selection attributes of the Decision tree are the Gini index, entropy, and information gain.)

The results were as follows:

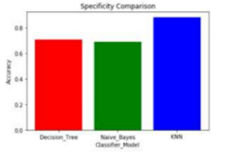
* + - * Accuracy:



* + - * Precision:



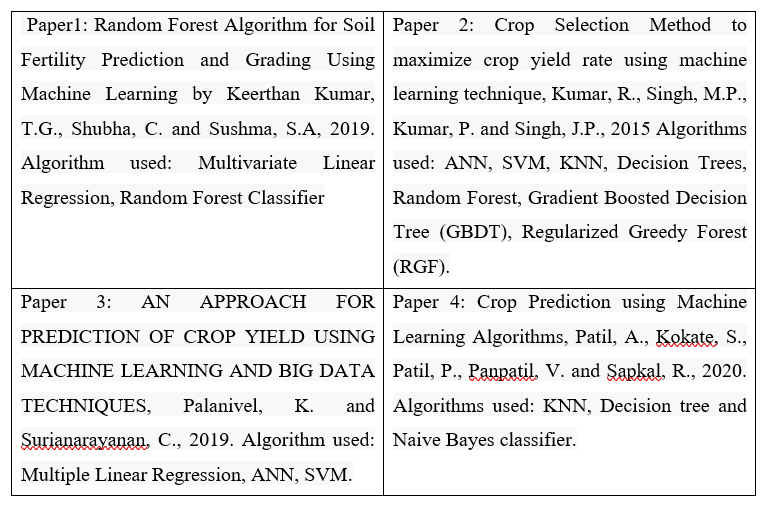
* + - * Specificity



After studying the results from the three models and their comparisons, the conclusion obtained is that the Decision tree shows poor performance when the dataset is having more variations but naïve Bayes provides better results than decision trees for such datasets. The combination classification algorithms like naïve Bayes and decision tree classifiers are better performing than the use of a single classifier model. Thus, we can make use of this study and also cross-check the findings when using

different models.

## Literature Survey Comparison



# Data Visualization

Exploratory data analysis is the most important and the first step in any machine learning or data-related project as it determines the properties and underlying concepts present in the data. We have used data analysis to determine the distribution of the features’ data points for various target labels. This is a supervised machine learning project, wherein we have used a dataset comprising details regarding the soil composition and weather conditions of the area and its corresponding crop as the target variable.

Using panda’s library, we read the .csv into a data frame and used the *read CSV* function. The data consists of 2200 entries with 3 columns of int datatype, 4 of float datatype (7 feature columns), and the target column of object datatype.

## Columns description

As previously mentioned, the dataset consists of 2200 rows and 7 columns:

* ” **N**” - Entries of integer datatype containing the ratio of Nitrogen content in the soil.
* ” **P**” - Entries of integer datatype containing the ratio of Phosphorus content in the soil.
* ” **K**” - Entries of integer datatype containing a ratio of Potassium content in the soil.
* ” **temperature**” - Entries of float datatype containing temperature in degrees Celsius.
* ” **humidity**” - Entries of float datatype containing relative humidity in %.
* ” **ph.**” - Entries of float datatype containing the ph value of the soil.
* ” **rainfall**” - Entries of float datatype containing rainfall in mm.

Here, we choose 7 attributes, namely N, P, K, Temperature, Humidity, ph, and Rainfall collectively as the input and the label as the output. This label contains 22 unique values which specify the name of different crops.

The described method of the data frame gives the summary statistics of the data frame, consisting of minimum, And maximum values of each feature as well as the mean and standard deviation. For our data, it was evident that the overall spread of data for each feature is not uniform, as the minimum value of the N column was 0 whereas that of the rainfall column was 20.211267.

We also used panda’s library to generate the profile report for our data to visualize any shortcomings and determine the null values. From the report, we concluded that the data is clean, i.e., there is no null value and all the feature columns had an equal number of entries.

## Exploratory Data Analysis

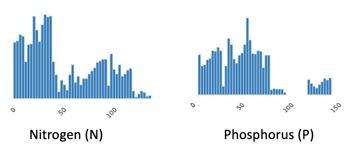
The basic statistical measures are calculated from the dataset to get an overall notion of the dataset. The number of instances corresponding to each unique crop is 100. From the heat map drawn, we could observe that there exists no null value in the dataset. The distance plot is drawn from the attributes of Temperature and ph. conveys that both these attributes are normally distributed. A bar plot is drawn for comparing the N-P-K values which converts the data capable of easier analysis through observation alone. Thus, after subjecting the original data to some statistical and visual analysis, we were able to obtain a clear image of the dataset and gather a comprehensible idea about the structure of the dataset. Therefore, we could conclude that the 'crop recommendation dataset' is a befitting choice for building a crop recommendation system.

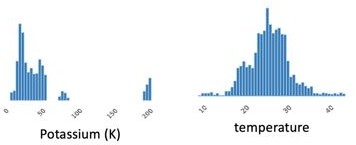
## Comparative Study

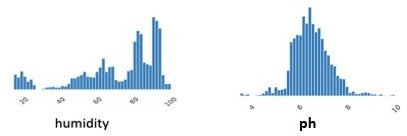
After the exploratory data analysis of the Crop Recommendation Dataset, it is submitted to the comparative study of classical supervised learning algorithms. These commonly used algorithms of machine learning are chosen one by one and they are trained using the available predefined models and their accuracies corresponding to their default parameters are noted down. Then each of these algorithms is provided with normalized data which is developed using different Normalization techniques. Afterward, the data is submitted to hyperparameter tuning which enables us to choose the best parameters to yield an optimized accuracy for the model. Research is the function we use for hyperparameter tuning. All the supervised algorithms we have chosen for analysis and normalization techniques are available in the sci-kit-learn library in Python. The algorithms used for the comparative study are K Nearest Neighbour Algorithm, Decision Tree, Random Forest, Naive Bayes Algorithm, Logistic Regression, Multi-Layer Perceptron, and Support Vector Machine.  
We succeeded in optimizing the accuracies of these algorithms which they had from training the model using default parameters. Now, we move on to another advanced neural network algorithm to build the proposed model. Extreme Learning Machine algorithm is used in our project to build the crop recommendation system with optimized accuracy and enhanced performance.

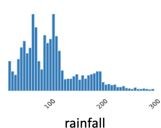
## Univariate Analysis

The analysis carried out by taking one of the feature column values concerning the target column values is called univariate analysis. This analysis gives the visual representation of the data spread for each feature value concerning the lump.

the







**Histogram representation of the feature columns**

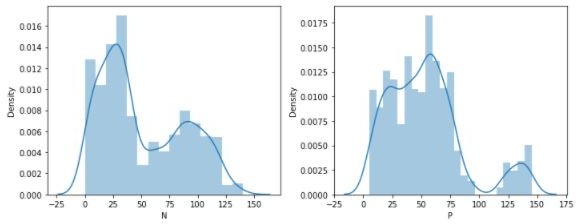
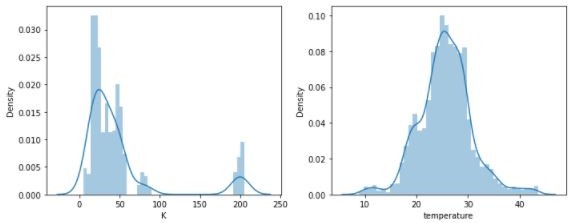
From the univariate distribution of the feature variables, we conclude that the data is not uniformly spread, as already mentioned in the summary statistics.

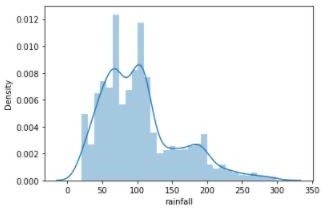
The data has 22 classes, thus making this a multi-class classification problem. The classes are:

1. rice mango
2. maize grapes
3. chickpea watermelon
4. pigeon peas apple
5. moth beans orange
6. Mungbam papaya
7. black gram coconut
8. lentil cotton
9. banana coffee
10. pomegranate jute
11. kidney-beans muskmelon

## Kernel Density Estimation

A kernel density estimation is an important tool for plotting the shape of the distribution. Like the histogram plotted in the univariate analysis, the KDE plot encodes the density of feature values vs the height of the feature values.



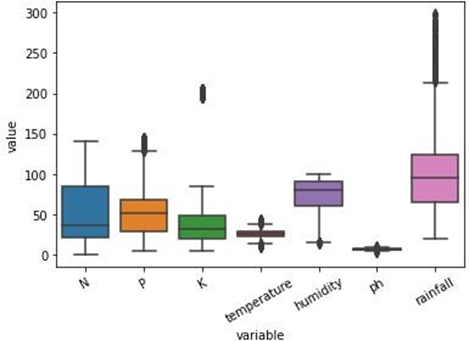


Kernel Density Estimation of the feature columns

From the KDE plots, we observed that the histograms are right and left-skewed and hence we need to transform the data to make it linear.

## Boxplot

A box plot, also called a box and whisker plot shows the distribution of quantitative data in a way that helps to compare variables. The box represents the quartiles in the dataset whereas the whiskers show the rest of the distribution. Using this we can find the features which can be removed.



Boxplot representation of the dataset

Bivariate Analysis

To determine if any 2 features are related, we used the bivariate analysis. As our feature set consists of all numerical values, this analysis will be Numeric vs Numeric. If any 2 columns are highly correlated, then any one of the columns can be dropped as it would not affect the accuracy of the model and hence reduce the complexity of the model. The below figure shows the correlation between the feature columns of our dataset.



Correlation of the features

From the correlation matrix, we observe that columns P and K are highly correlated. But as we have only 7 feature columns, we decided not to drop one of these columns at this stage.

# Data Preprocessing

Before we fit our model with our data, we need to clean and trim the data so that the model generates effective results. From the initial analysis, we concluded that our data does not have any null values or missing values. But from the boxplot we found our data to have outliers, which if not removed, would affect the performance of the model

where N = Total number of observations in the feature column

*xi* = I the observation

The mean takes every point into account these considering outliers as regular data points. So, we need to drop the outliers. This process includes methods to remove any null values or infinite values which may affect the accuracy of the system. The values are also rescaled for faster training of the models. Steps in preprocessing include outlier detection, missing value treatment, and rescaling. The cleaning process is used for the removal or fixing of some missing data there may be incomplete data.

We have dropped the outliers for each feature based on the quartile values for each column and then replaced this missing outlier values with the median of the column. We replace it with median because our motive is to generalize the data as the data is skewed due to the outliers present. Below are the boxplot, kernel density estimation, and the corresponding correlation matrix of the features dataset after replacing the outliers present in the data with the median values.

# Random Forest

Random forest is built using multiple decision trees. The decisions from each tree are used and then the maximum outcome from all is chosen to give as the output. Due to this most of the time, Random Forest works as the best classifier model.

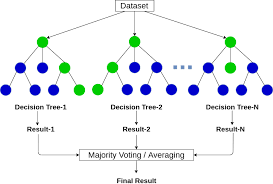


Fig Random Forest

Random forest work on the same principle as a decision tree. A random forest consists of several decision trees. Each decision tree is constructed using a few randomly selected features. The goodness of the split of each tree is measured with an impurity measure – The Gini index or entropy.

While classifying testing samples, each set of feature values is run down all the trees. The class which is chosen by the majority of the trees is assigned to that particular feature list.

In our implementation, we use the Random Forest - Classifier model available in the sklearn library to build our random forest Classifier.

# METHODOLOGY

## System Architecture

The overall system can be divided into two parts a) a Web server and b) an ML container. The whole web server can be connected to a database most probably a SQL database. The architecture is designed in a way to allow a large number of requests. It is easily scalable and deployable.

The technologies used for this whole project are:

* + - Docker: Docker helps in the containerization process. It creates a whole development environment that is easy to use and easily deployable across any server
    - Keras: Keras is a deep-learning library that provides powerful deep-learning solutions
    - MySQL: A relational database system
    - Flask: A light-weight python web server

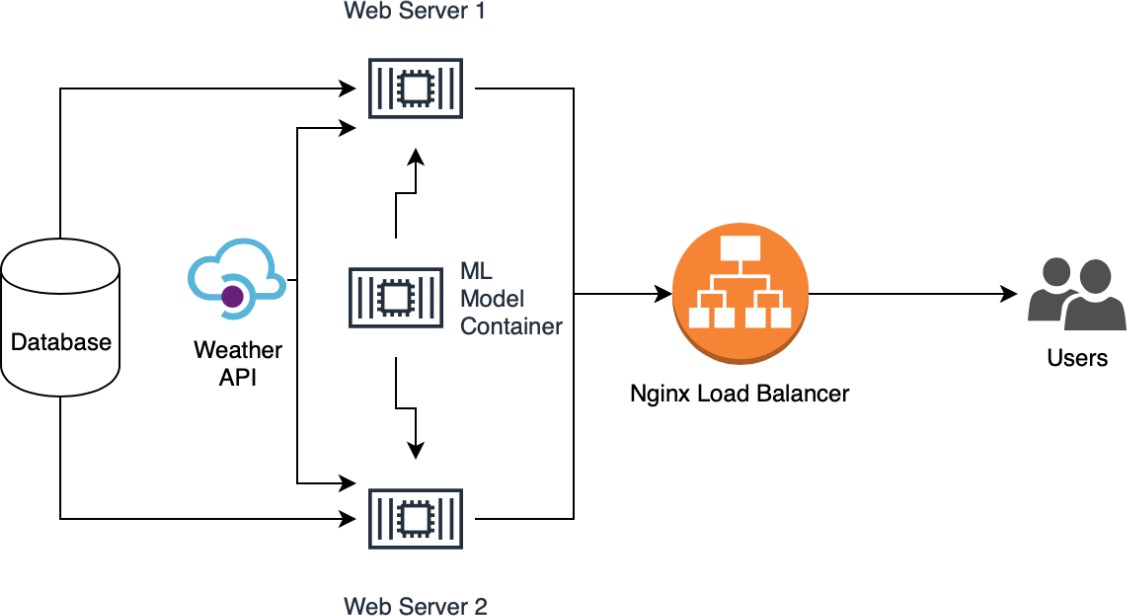
Since previous works have used these technologies/methods to predict the crops individually, there is a potential of achieving better results when these methods are combined. Therefore, we provide our models with all the 3 types of data, that is, soil properties such as N-P-K, and atmosphere properties such as moisture and vegetation indices obtained from satellite image data.

A prototype code for the generation of NDVI using LANDSAT data from USGS (United States Geological Survey) was run successfully. The index value obtained now has to be given as an input to our model that will be designed to verify the results. The code is attached in Appendix A.

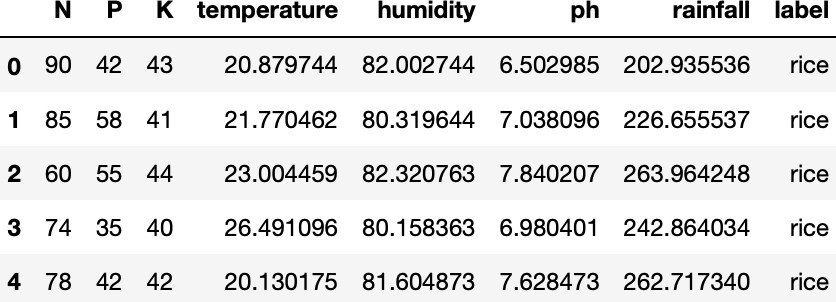
Fig.2 (Appendix A) shows the NDVI index calculated for Shimura. The data used to obtain this is collected from the USGS website and is dated 14th April 2021. We see that the values are generously towards the positive side, making it very cultivable. Using this quantitative data regarding the vegetation, we aim to predict qualitative data regarding the type of crop to be grown.

## Datasets

The datasets for both tasks were available online and were open-sourced under an open license to be used by anybody. The dataset for the first task consists of a CSV file that contains 2200 entries of the various factors such as soil condition, temperature, pH, humidity, and rainfall and label output as the type of crop well produced in that type of conditions. Refer figure:9[for](#_bookmark17) peak into the dataset.



**System Architecture**



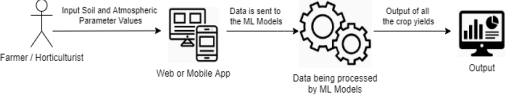
Random data from the dataset for crop recommendation

These datasets were used to train all the further mentioned ML algorithms and the one with the best accuracy was chosen. A crop requires specific climatic as well as regional conditions to give the optimal yield. So, it becomes crucial to consider these factors for predicting the crop and its yield. While predicting a crop or its yield numerous features are taken into consideration. Thus, the features we have taken into account for predicting “Crop Yield” are District, Crop, Average Temperature, Rainfall, and Area. Similarly, while predicting the “Crop” the features we have taken into consideration are District, Average Temperature, Rainfall, Area, and Production (Yield). Temperature is one of the most important factors as humidity and moisture affect the production of the crop as high air temperature reduces the growth of shoots and in turn, reduces root growth. In the same way, Rainfall plays a vital role in the growth and nourishment of the crop as excess or minimal amount of rainfall would destroy the crop. According to the Area under cultivation, the yield of the crop is estimated.

The design approach we have planned on using is as follows:

1. Choose a specific geographic location like a district or state, for eg: Karnataka.
2. Gather past data for the chosen area.
3. Allow the users to enter real-time data into the application.
4. The application will predict and display a list of crops along with their yield percentage which will be ranked in a hierarchy.

In the backend, the machine learning models such as SVM, Decision Trees, ANN, etc will process the data provided and will predict the range of crops.



* Currently, we have 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 will attempt to implement some of the **ensemble machine learning algorithms such as AdaBoost and XGBoost** and also an **Artificial Neural Networks** model to the recommendation of crops with a good accuracy score.

## Model Selection and Performance Evaluation

Following feature extraction, the dataset is broken down into xtrain, ytrain, and xtest, ytest pairs. Sklearn is used to import the algorithm model. A model is used to construct the model. Fit (xtrain, ytrain).

Crop type and fertilizer type are our output (target). Since we are going to classify using continuous features, we are going to implement using Decision tree regression, support vector regression, and also a neural network-based classifier.

Following the construction of the model, prediction is performed using the model. predict (xtest). The accuracy is determined using metrics accuracy score shipped from statistics (ytest, predicted).

# Crop Recommendation

The crop recommendation is a classification task. Standard ML algorithms were used to classify various plants. The features trained for the classification were the NPK value of the soil, temperature of the surroundings, pH of the soil, and rainfall in a particular area.

This dataset was trained on the following algorithms:

* + - Logistic-Regression
    - Decision Tree
    - Support-Vector-Machine (SVM)
    - Multi-Layer Perceptron
    - Random Forest

These five algorithms were chosen because, given the features and labeled dataset, these algorithms can run faster and provide good results on the labeled dataset for classification problems.

The accuracy of the mentioned algorithms is given in the observations section.

## Architecture Diagram

# Tools and Technologies

As the project had two parts, web and ML two spectrums of tools are been used. Those are listed below:

* + - Python
    - Flask
    - HTML5
    - CSS3
    - JS

# CODE

from flask import Flask, redirect, url\_for, render\_template, request JSON on

import pandas as pd

import NumPy as np

from sklearn.preprocessing import LabelEncoder

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.neighbors import KNeighborsClassifier

# From crop import classifier

app=Flask(\_\_name\_\_)

app.secret\_key="Teams"

@app.route("/")

def home():

    return render\_template("index.html",cont="User")

@app.route("/login",methods=['GET','POST'])

def login():

    if request.method=='POST':

        N=float(request.form['Nitrogen'])

        P=float(request.form['Phosphorous'])

        K=float(request.form['Potassium'])

        Temperature=float(request.form['Temperature'])

        Humidity=float(request.form['Humidity'])

        PH=float(request.form['PH'])

        Rainfall=float(request.form['Rainfall'])

        # object = Crop.classifier.predict(np.array([N, P, K, Humidity, PH, Rainfall]))

        # model=pickle.load(open("model.pkl",'rb'))

        data= pd.read\_csv('data/Crop\_recommendation.csv')

        data['label']=LabelEncoder().fit\_transform(data['label'])

        data['label'].unique()

        X = data.drop(['label'],axis=1)

        Y = data.label

        X\_train, X\_test, y\_train, y\_test = train\_test\_split (X, Y, test\_size=0.20)

        scaler = StandardScaler()

        scaler.fit(X\_train)

        X\_train = scaler.transform(X\_train)

        X\_test = scaler.transform(X\_test)

        classifier = KNeighborsClassifier(n\_neighbors=5)

        classifier.fit(X\_train, y\_train)

        predict1=classifier.predict(np.array([N, P, K, Temperature, Humidity, PH, Rainfall]).reshape(1,-1) )

        if predict1==0:

            crop\_name = 'Apple'

        elif predict1 == 1:

            crop\_name = 'Banana'

        elif predict1 == 2:

            crop\_name = 'Blackgram'

        elif predict1 == 3:

            crop\_name = 'Chickpea'

        elif predict1 == 4:

            crop\_name = 'Coconut'

        elif predict1 == 5:

            crop\_name = 'Coffee'

        elif predict1 == 6:

            crop\_name = 'Cotton'

        elif predict1 == 7:

            crop\_name = 'Grapes'

        elif predict1 == 8:

            crop\_name = 'Jute'

        elif predict1 == 9:

            crop\_name = 'Kidneybeans'

        elif predict1 == 10:

            crop\_name = 'Lentil'

        elif predict1 == 11:

            crop\_name = 'Maize'

        elif predict1 == 12:

            crop\_name = 'Mango'

        elif predict1 == 13:

            crop\_name = 'Mothbeans'

        elif predict1 == 14:

            crop\_name = 'Mungbeans'

        elif predict1 == 15:

            crop\_name = 'Muskmelon'

        elif predict1 == 16:

            crop\_name = 'Orange'

        elif predict1 == 17:

            crop\_name = 'Papaya'

        elif predict1 == 18:

            crop\_name = 'Pigeonpeas'

        elif predict1 == 19:

            crop\_name = 'Pomegranate'

        elif predict1 == 20:

            crop\_name = 'Rice'

        elif predict1 == 21:

            crop\_name = 'Watermelon'

        if float(Humidity) >=1 and float(Humidity)<= 33 :

            humidity\_level = 'Low Humid'

        elif float(Humidity) >=34 and float(Humidity) <= 66:

            humidity\_level = 'Medium Humid'

        else:

            humidity\_level = 'High Humid'

        if float(Temperature) >= 0 and float(Temperature)<= 6:

            temperature\_level = 'Cool'

        elif float(Temperature) >=7 and float(Temperature):

            temperature\_level = 'Warm'

        else:

            temperature\_level= 'Hot'

        if float(Rainfall) >=1 and float(Rainfall) <= 100:

            rainfall\_level = 'Less'

        elif float(Rainfall) >= 101 and float(Rainfall) <=200:

            rainfall\_level = 'Moderate'

        elif float(Rainfall) >=201:

            rainfall\_level = 'Heavy Rain'

        if float(N) >= 1 and float(N) <= 50:

            N\_level = 'Less'

        elif float(N) >=51 and float(N) <=100:

            N\_level = 'Not too less and Not to High'

        elif float(N) >=101:

            N\_level = 'High'

        if float(P) >= 1 and float(P) <= 50:

            P\_level = 'Less'

        elif float(P) >= 51 and float(P) <=100:

            P\_level = 'Not too less and Not to High'

        elif float(P) >=101:

            P\_level = 'High'

        if float(K) >= 1 and float(K) <=50:

            potassium\_level = 'Less'

        elif float(K) >= 51 and float(K) <= 100:

            potassium\_level = 'Not too less and Not to High'

        elif float(K) >=101:

            potassium\_level = 'High'

        if float(PH) >=0 and float(PH) <=5:

            phlevel = 'Acidic'

        elif float(PH) >= 6 and float(PH) <= 8:

            phlevel = 'Neutral'

        elif float(PH) >= 9 and float(PH) <= 14:

            phlevel = 'Alkaline'

        # return render\_template ("index.html", cont=[crop\_name,humidity\_level,temperature\_level,rainfall\_level,N\_level,P\_level,potassium\_level,phlevel])

        return render\_template("Display.html", cont=[N\_level,P\_level,potassium\_level,humidity\_level,temperature\_level,rainfall\_level,phlevel],values=[N,P,K,Humidity,Temperature,Rainfall,PH],cropName=crop\_name,)

    return render\_template("index.html")

@app.route("/user/<usr>")

def user(usr):

    return f"<h1> Hi {usr} !</h1>"

# @app.route("/user")

# def user():

#     if "user" in session:

#         user=session['user']

#         return f"<h1> hi {user} </h1>"

#     return render\_template("login.html")

if \_\_name\_\_=="\_\_main\_\_":

   app.run(debug=True)

# Deployed website

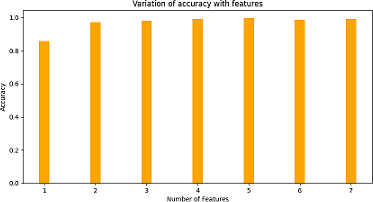
This section shows the various outputs of the deployed application. The web application is created in Flask and has an ML backend to it. It will later be deployed on a cloud service platform.

The landing page of the website. The pages are built using bootstrap and flask. The landing page of the crop recommendation system and the corresponding result.

# Results

## Decision Tree

The graph shows how the accuracy varies with several features used while classifying.



Variation of accuracy with features

Hyperparameter tuning was run on all other parameters as well. The following parameter values provided the best results for the decision tree for our dataset.

|  |  |
| --- | --- |
| Impurity measure | Gini Index |
| Depth of tree | 100 |
| Number of features | 5 |
| Minimum number of samples on leaf | 1 |
| Minimum sample split | 5 |

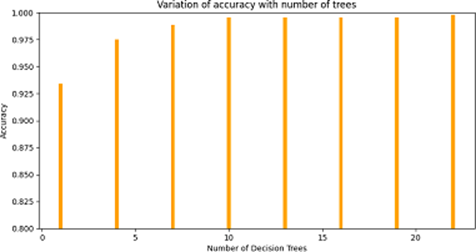
The accuracy obtained was close to 98.86 percent.

**Random Forest**

Hyperparameter tuning is done on Random forest classification to find the best fit for our dataset.

The other tuned parameters of the random forest classifier are mentioned below.

|  |  |
| --- | --- |
| Impurity measure | Entropy |
| Depth of tree | 200 |
| Number of features | 3 |
| Minimum number of samples on leaf | 5 |
| Minimum sample split | 10 |
| Number of estimators | 25 |

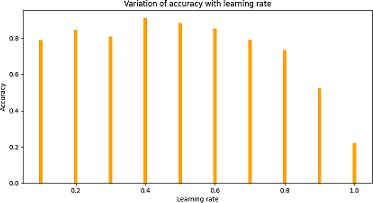


Variation of accuracy with the number of decision trees

The accuracy obtained using random forest was close to 99.09 percent.

## AdaBoost

The graph indicates how the accuracy of the boosting technique with a decision tree as a base estimator varies with the learning rate.



Variation of accuracy with learning rate

It is important to note that these accuracies vary for the same learning rate with the different models as a base estimator. The tuned parameters of the AdaBoost are mentioned below.

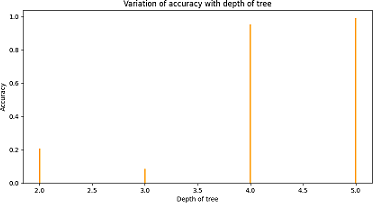
Base estimator Random Forest Learning Rate 0.6

The accuracy obtained using random forest was close to

99.24 percent. We can observe that boosting has enhanced the accuracy of its base estimator.

## Gradient Boost

The graph indicates how the accuracy of the boosting technique with decision tree regressor as base estimator varies with the depth of trees.



Variation of accuracy with a depth of trees Likes in AdaBoost,

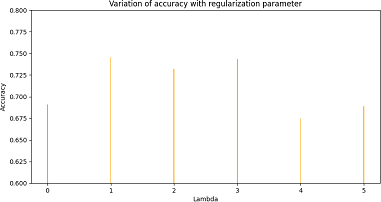
these accuracies further vary when other parameters are defined/tuned. Here a tree depth of 5 is suitable. However, when the other parameters are tuned, a tree depth of 5 may or may not be the best-suited solution. The best combination of parameters for gradient boost is as mentioned below.

|  |  |
| --- | --- |
| Depth of tree | 5 |
| Criterion | Friedman Use |
| Learning rate | 1 |
| Number of estimators | 100 |

The accuracy obtained using gradient boost is close to 98.78 percent.

## XGBoost

The graph below shows how the accuracy of the XGBoost algorithm varies with *λ* values. We know that as *λ* values increases, pruning increases. However, over-pruning a tree is also not efficient as indicated in the figure below. From the figure, we can observe that initially, the accuracy increases with the in- crease in pruning. However, as we increase lambda further, the accuracies fall due to over-pruning.



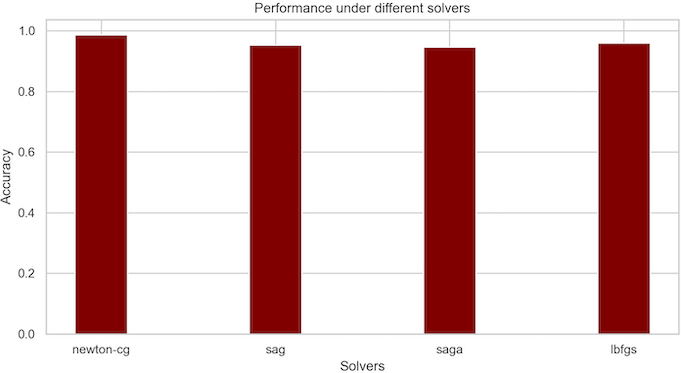
Variation of accuracy with lambda values

Similarly, other parameters of XGBoost were tuned using hyperparameter tuning. The following are the parameter values for which XGBoost provides the highest accuracy.

|  |  |
| --- | --- |
| Depth of tree | 8 |
| Booster | dart |
| gamma | 5 |
| Learning rate | 0.2 |
| lambda | 1 |

The accuracy obtained using XGBoost is close to 96.5 percent.

## Logistic Regression

The graph indicates the impact of parameters on the ac- curacy of a logistic regression model. Here, the accuracy changes as the solver used for the classification varies. Based on the graph it is clear that the model gives the best results when ’newton-cg’ is its solver.

Variation of accuracy with solvers

Hyper-parameter tuning is performed on parameters of logistic regression classification to find the best fit for our dataset. The best combination of parameters for our logistic regression model is:

Penalty l2

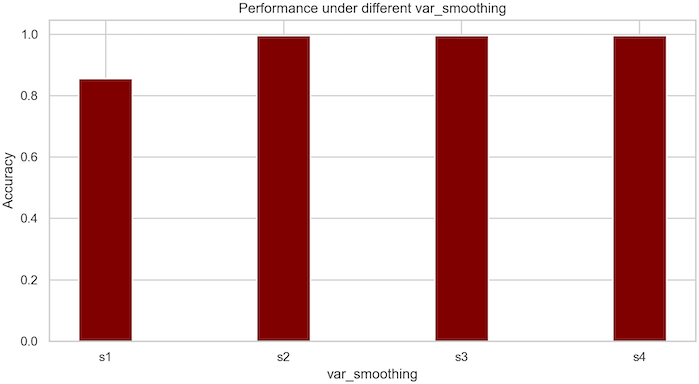
Solver newton-CG Class weight balanced

The accuracy obtained using logistic regression is 98.69 percent.

## Gaussian Naive Bayes

The graph indicates the impact of the parameters on the accuracy of the gaussian naive Bayes model. Here, the accuracy changes as the var-smoothing parameter are changed for the classification. Value-at-risk-at-risk inserts a consumer number to the volatility of the dispersion, thus expanding the curve and allowing for additional samples that are located further away from either the distribution average.

Hyperparameter tuning is performed on parameters of the Gaussian Naive Bayes classifier to find the best fit for our dataset. The best combinations of feature parameters that are given for our gaussian-naive Bayes model are

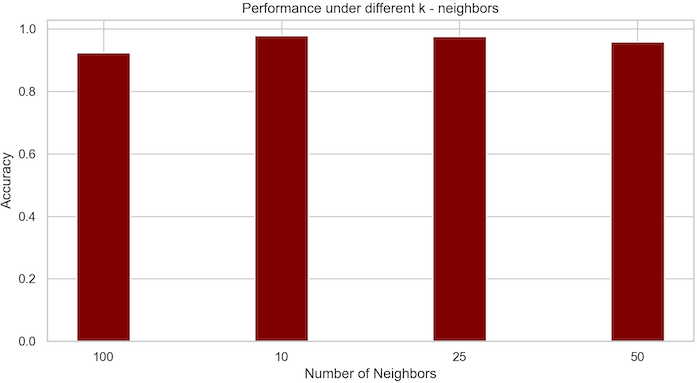


Variation of accuracy with var-smoothing var smoothing: 2.848035868435799e-05

The accuracy obtained using Gaussian Naive Bayes is 99.09 percent.

## K-nearest Neighbor

The graph indicates the impact of parameters on the accuracy of the k-nearest neighbor model. Here, the accuracy changes as the number of neighbors used for the classification varies. Based on the graph it is clear that the model gives the best results when the n-neighbor is 10.



Variation of accuracy with n-neighbors

N-neighbors 10

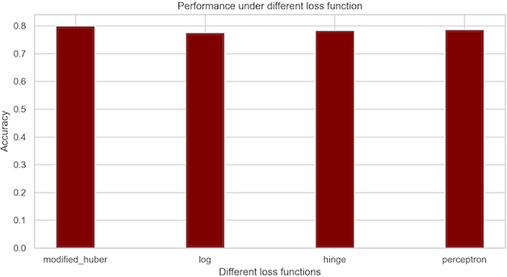
Weight Uniform

Metric Manhattan Algorithm auto

The accuracy obtained using K-nearest Neighbor is 97.95 percent.

## Stochastic Gradient Descent

The stochastic Gradient Descent model is unpredictable and the accuracy of its classification cannot be determined because it varies in time. This model will not be the best for our application due to its uncertainty. However, conducting hyper-parameter tuning indicates the impact of parameters on the accuracy stochastic gradient model. Here the accuracy is plotted for varying loss functions used for classification.



Variation of accuracy with loss functions

Hyper-parameter tuning is performed on parameters of stochastic gradient descent classification to find the best fit for our dataset. The best combination of parameters for our stochastic gradient descent model are:

Loss modified-Huber Penalty l1

Class weight balanced

Learning rate optimal

The accuracy obtained using stochastic gradient descent is 80.00 percent.

## Support Vector Machine

Hyperparameter tuning was run on all the parameters to find the best-fit parameter for our data. The following parameter values provided the best results for SVM:

N-neighbors 10

Weight Uniform

Metric Manhattan Algorithm auto

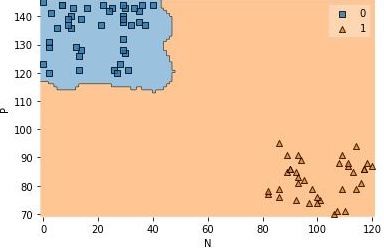
The accuracy obtained using K-nearest Neighbor is 97.95 percent.

C 1

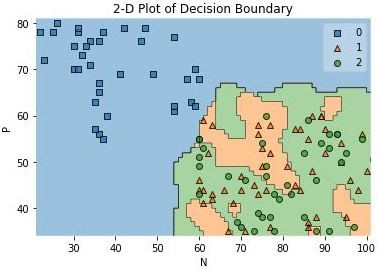
γ 0.1

kernel rbf

Below is the plot of binary classification (by taking 2 classes, namely apple and banana) vs 2 input features N and P.



Binary classifiers constructed for the OVA approach



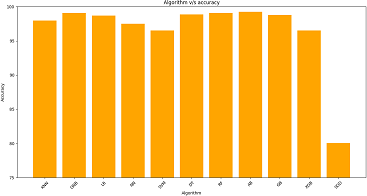
A 2-D plot of the Decision Boundary

The above 2-D plot is concerning 3 target classes. The decision boundary for all 22 classes in 2-Dimensions is shown below.

The accuracy obtained using the Support Vector Machine was close to 96.59 %.

## Accuracy Comparison:

The following graph shows the accuracy produced by various classification techniques for the crop recommendation System.

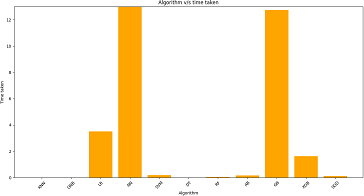


Accuracy of various classification techniques

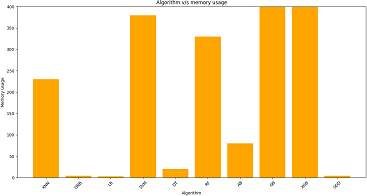
Most of our algorithms provide an accuracy greater than 95 percent. Stochastic gradient descent provides a low accuracy of close to 80 percent and is hence not suitable for this classification problem.

Since most of our algorithms work well, the best way to choose the most suitable algorithm is to compare the space and time complexity of these algorithms.

The bar graphs show the time and space complexity of various classification algorithms used in this project.



The time complexity of classification algorithms



The space complexity of classification algorithms

Though Gradient Boost and XGBoost provide an accuracy greater than 95 percent, these algorithms have a very high time complexity. We have other algorithms with similar or better accuracy and smaller runtime. Hence Gradient Boost and XGBoost are not suitable algorithms for our crop recommendation system.

The AI-assisted crop recommendation system and yield prediction system consider various parameters related to the atmosphere and soil conditions to give a recommendation of the best crop for the farmer. The farmer inputs the soil condition parameters viz. N, P, K and pH levels. It is one of the biggest boons to the farmer.

Apart from these most of our algorithms provide an accuracy greater than or close to 97 percent. Hence an algorithm with the top accuracy, least time, and space complexity is chosen. From our analysis, we choose Gaussian Naive Bayes as the most suited algorithm for crop recommendation systems. This algorithm outperforms all other algorithms. Gaussian Naive Bayes has an accuracy of 99.90 percent with time complexity of 68ms and space complexity of a mere 4kb.

## High Accuracies

As we can see from the results, we have high accuracy for almost all our algorithms. Our methods work so well because the dataset used is very clean. There are no missing values or null values in our dataset. If there were any missing or null values, we would have to replace those with the median values. Moreover, we have also replaced the outliers in the data with the median value of each feature, which has helped to remove any skewness present in the feature column. With a harder dataset, meaning having a larger dataset with more features then depending on how clean the data is, our model accuracy would also vary. If the dataset is skewed due to a large number of outliers present, then dropping the outliers would reduce the data thus affecting the effect of certain features on the overall model. This would have tuned our classifier parameters differently and would certainly affected the accuracy of the model.

Apart from that one more point to be noted is the fact that our dataset is very small. However, we can’t be certain on how a larger dataset would affect the accuracy. There are chances of the accuracy decreasing. increasing or remaining the same based on the dataset.

Also, we tried generating data for a few of our classes and tested them on some of our models. What we observed was that the accuracies of those models reduced slightly with this new data and the model was not able to classify every input correctly.

# Conclusion

This project solves the problem of the agricultural industry by providing a solution to a major problem of harvesting. A crop or yield recommendation system is developed in which the most suitable crop is predicted based on the soil and weather parameters. This system maps 7 soil and weather parameters (features) to 22 crops (classes).

The recommendation system is designed with various classifiers such as linear models, ensemble models, and neural nets. After removing the outliers and normalizing the feature values, hyperparameter tuning has been applied to each of the classification models to increase the accuracy of crop prediction. Decision tree, Random Forest, and Gaussian Naive Bayes are the best-performing algorithms for this crop dataset. Based on time complexity and accuracies we conclude that Gaussian Naive Bayes is the most suited classification algorithm for the recommendation system. Gaussian Naive Bayes provides an accuracy of 99.09 with time complexity of 68ms and space complexity of 4kb. Having a clean dataset has helped us achieve high accuracy.

The AI-assisted crop recommendation system and yield prediction application can successfully predict the crop based on the atmospheric parameters and soil conditions given after soil testing by the farmer. At any given time one crop is more suited to be grown than other based on the above conditions. The two models of crop and yield are successfully trained in Auto AI which then predicts appropriate crop and yield respectively.

We select the models with the highest possible accuracy and the accuracy is measured with various metrics like K-fold, precision, and recall. We intend to develop this model so that it is beneficial for farmers in decision-making for good farm produce.

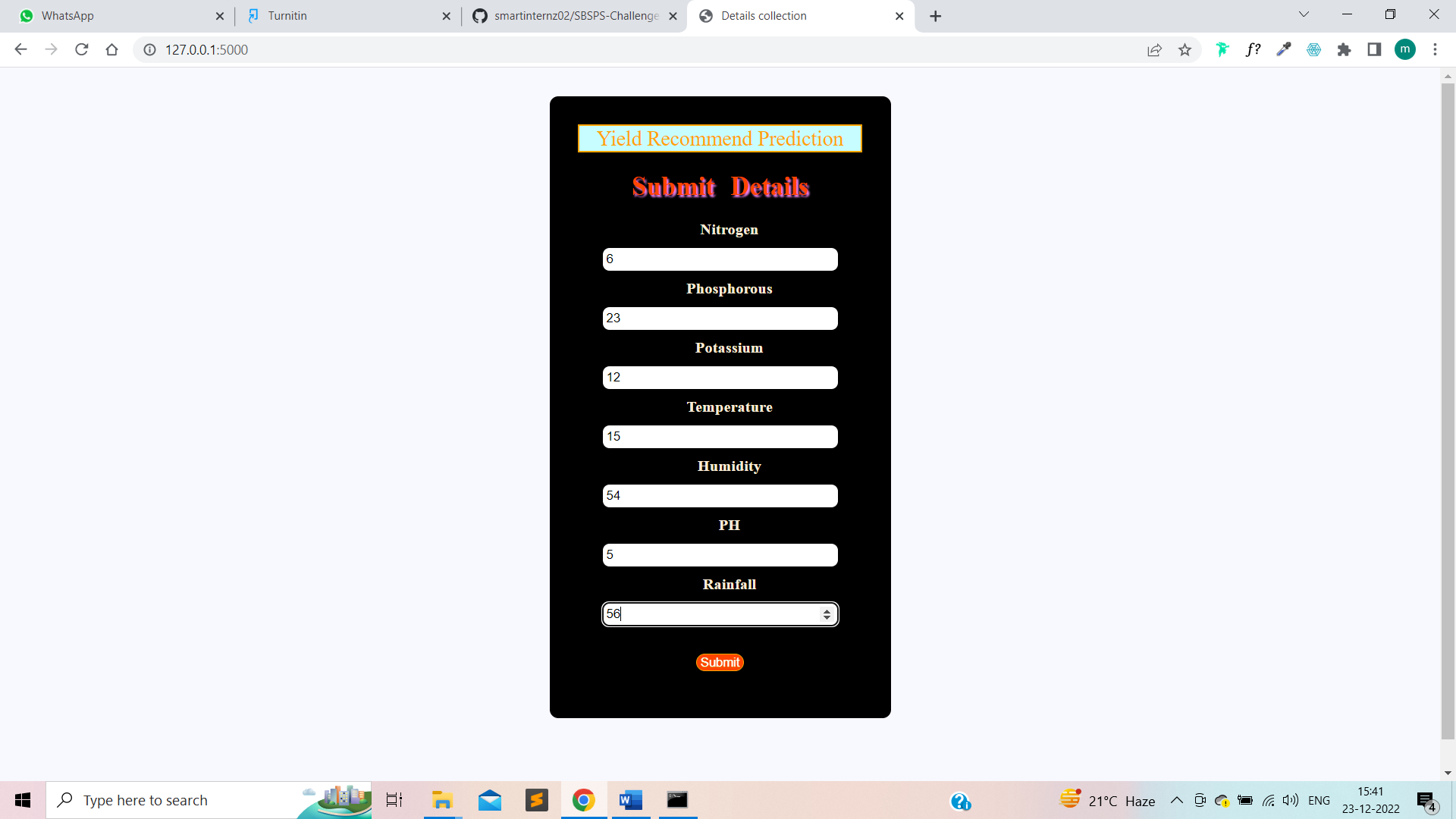
The proposed work helps the farmer make an informed de- decision on choosing the right crop based on soil and weather conditions. This ensures maximum yield which in turn im- proves the livelihood and economy. We studied 5 different algorithms for task 1 and concluded that Random Forest is the best suited for the selected dataset. Random Forest achieved an overall accuracy of 99.3%.

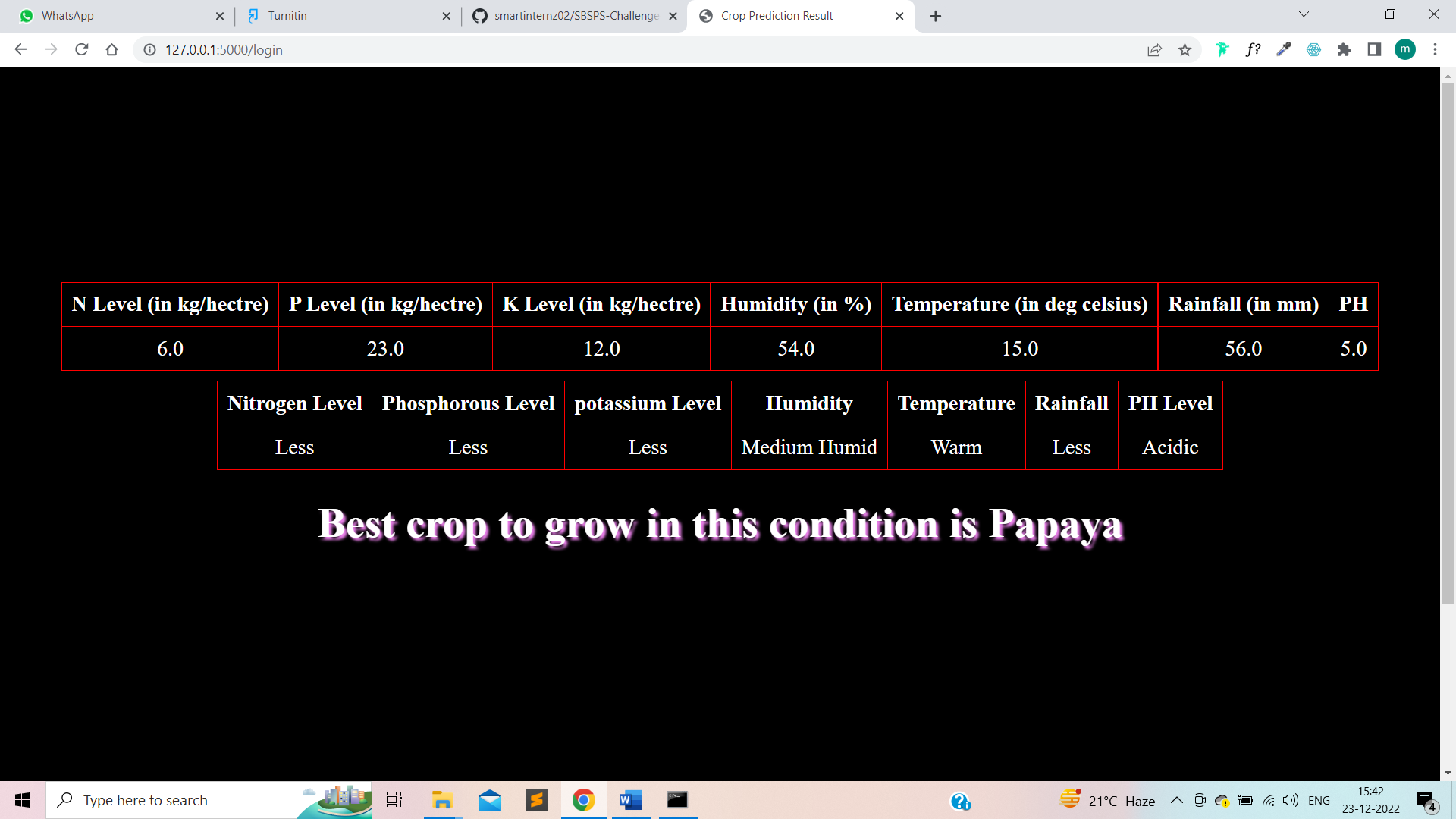
# Future Scope

One big future scope is increasing the scope of the project. Currently, it is restricted just to Maharashtra. It could be expanded to the entire India. We can also include more crops. Fruits can also be included in the system. Furthermore, we can also add parameters where the system can recommend fertilizers for a crop based on current soil conditions. Crops based on particular types based on the availability of water could be another feature than can be added.

# SCREENSHOT:

First, filling the details they want to be owned ones of the details System Recommend the crops we grow that give the best result.





# Appendix: Pilot Project – 1

# Abstract

Agriculture is a job that combines interaction with nature and life. We are all aware of how random nature can be. Farmers are thus at the mercy of nature. The potential risk is quite high since it may rain correctly, there may be dryness, or there may be a flood. Even if they work hard, they may not see the fruits of their labor. As a result, if nature has other intentions, its efforts may be in vain.

Right present, there are 7.2 billion people on the earth. In a few decades, it will reach 10 billion. Agriculture must be modernized to provide for 10 000 000 million more people. Traditional approaches are no longer viable if crops cultivated on soil are not fit for such circumstances. Because there is a finite amount of soil fertility, it needs to be regularly examined and maintained.

One of India's oldest and most honorable professions is agroecology. In today's technologically-centered world, farmers confront various challenges when adopting conventional farming practices. Compared to conventional agriculture methods, precision farming is a new strategy.

Producers were incapable of benefiting from their farm areas due to a lack of information and forecast skills. Furthermore, urbanization is rising as the world's population rises. Purchasing habits are changing, and discretionary cash is rising. Farmers must devise a strategy to increase output because they are under intense pressure to fulfill growing demand. There will be more individuals to feed in thirty years, and because the number of productive soils is restricted, it will also be essential to forsake traditional farming.

We must look for ways to reduce or, at the absolute least, limit the risks that farmers confront.

.

# Introduction

With a population of nearly 1.3 billion people, India is the world's second-biggest country. Many people rely on agriculture but the industry, particularly in our nation, lacks efficiency and technology. Effective crop cultivation may be done by creating a bridge between conventional agriculture and data science.

Agriculture was one of the first practices of civilization. Traditional farming is built on ancestral wisdom and first-hand knowledge. The existing level of the food supply is inadequate and does not offer farmers a greater level of revenue due to incorrect crop cultivation in line with agricultural components such as soil, climate, and rainfall.

Farming and associated sectors account for around 18% of India's Total Additional Value (GVA). A large-scale, poorly educated agricultural choice might have a significant influence on the overall economy of an area. We should develop technologies that can provide Indian farmers with forecasted information, allowing them to select the best crops to produce. In most situations, a farmer's decision to plant a crop is affected by his intuition, as well as other irrelevant variables such as generating quick money, being oblivious of market demand, overestimating a soil's capacity, and so on. The farmer's unwise decision might put his family's finances in jeopardy.

Computational modeling and deep learning are used to create a Python model for a genuine crop suggestion tool. This work depicts a system in the form of a webpage. Techniques for machine learning are used in Python business logic to predict the most lucrative crop in the expected climate and soil characteristics at a certain site. The proposed system would incorporate weather reports crop repositories, and soil data to predict the best-suited crops depending on the present condition of the environment. This is accomplished through the use of a machine learning approach known as Linear Regression Analysis. This provides a farmer with a diverse selection of crops to consider planting.

# Motivation

The motivation behind developing Crop It was to study the application of Machine Learning techniques to solve real-life problems. With the increasing use of technology in the agriculture sector and the availability of facilities to test the soil for NPK and pH values, this application aims to help farmers make an informed decision based on Machine Learning algorithms.

The team members have a good grasp of Python programming language and are interested in exploring Machine Learning.

As we researched more about the concept of machine learning, we came across its types, techniques, and applications. Hence, we took up this concept as the main implementation for our Project – “Crop It”. Not only it helped strengthen our knowledge of programming but also helped us in exploring the new field of Machine Learning and its practical applications.

Producers may use our programmer to measure crop yields and determine which crops to grow by utilizing program a digital tool.

Farmers, who play an important part in our economy, are underappreciated. In today's advanced world, they encounter a lot of challenges while adopting traditional farming practices. Farmers struggle to comprehend the complexities of modern procedures employed to alleviate their pain. Other study articles typically utilize criteria that place more emphasis on variables that are difficult for farmers to grasp, such as the type of soil and nutrient concentration.

As a result, we selected factors for this study those producers can grasp better, such as district, rain, temp, and size.

# Problem   Statement

India is an agrarian economy; hence farming and related activities are of great importance. Many times, farmers are unaware of which crops are best to grow in their region which leads to excessive use of fertilizers and irrigation. This may lead to the degradation of soil and affect profits earned by the farmers. The application must accept soil parameters (N-P-K and pH) and climate parameters (Temperature, Rainfall, and Humidity). Using these parameters, the model should predict the crop that is suitable for the given conditions. The main challenge in this project is to implement, and analyze several classification algorithms and select the best algorithm for the problem which gives the most accurate results.

Every moment the surrounding weather changes, most crops are killed because the surrounding weather is not suited. We did not alter the ambient climate, but we did identify which crops are appropriate in this climate.

I'm developing a crop suggestion system to assist farmers in picking the best crop to put in their fields and estimating yield and income, which might be the solution to this problem. The majority of farmers employ traditional farming methods when determining which crops to grow on a field. However, with the use of digitized farming and accurate agriculture, harvests may be watered, fertilized, and pesticides applied exactly when needed to boost output, quality, and yields.

## Objective

The foremost objective is to design a robust machine-learning model that would predict the optimal crop for the given soil and climate conditions. The model should be trained with a dataset having adequate information about a reasonable variety of crops over a good range of parameters. The algorithm that we choose to use needs to have high accuracy for the project. The selected algorithm must consider all the parameters for choosing the optimal crop.

# Background

## Criteria/Constraints:

This suggested system product would primarily identify four different sorts of crops based on the environmental parameters of the chosen piece of land. The soil type or any other modifications in the specified land would be the explanation for obtaining a probability of greater than 90% for the mentioned crops. However, to keep these influences from influencing crop forecasts, the system includes a farmer feedback mechanism.

Following the crop type suggestion, the farmer is frequently prompted for facts and comments via the Web-application to educate him or her on the necessary procedures. The feedback mechanism is utilized to provide relevant feedback after selecting the crop variety in the mobile application. This improves the overall correctness of the product.

## Crop Recommendation

Models required

* Crop recommendation model
* By using various parameters like Temperature, Humidity, and Rainfall the crop which is suitable for that particular environment is predicted.
* NPK prediction model
* It is used to predict the amount of nitrogen, phosphorus, and potassium that is required for crop production.
* Fertilizer prediction model
* If the amount of NPK is not sufficient for the user’s desired crop yield, then the user can use this particular fertilizer to make the yield better.

### Predictions

* Crop
* Features required for prediction
* Temperature
* Humidity
* Rainfall
* These features can be automatically predicted by knowing the user's location, so no user input is required.
* Output
* Crop Name
* NPK Prediction
* Features required for prediction
* Temperature
* Humidity
* Rainfall
* Crop
* Here Temperature, Humidity, and Rainfall are automatically determined using location-based weather API, so the crop is one of the required user inputs.
* Output
* Nitrogen content
* Phosphorus content
* Potassium content
* Fertilizer Recommendation
* Features required for prediction
* Nitrogen
* Phosphorus
* Potassium
* Soil Type
* Crop Type
* Temperature
* Humidity
* Here Soil type is required user input, other things are automatically known using the model and the API.
* Output
* Fertilizer Name
* This project also focuses on Fertilizer recommendations, whenever required.

Algorithm used

* KNN(K-Nearest Neighbours).

In the above phase, the prediction is done in the following manner,

* If the user is not interested in using Fertilizers, then the user has to provide the NPK values, then this system would recommend a crop based on these NPK values.
* Or on the other hand, if the user has no problems using Fertilizers, then the crop recommendation is done by knowing the user's location. After that, the NPK values are predicted using the chosen crop and environmental factors. Using these NPK values, crop type, and soil type, this system would predict the fertilizer which is required to maximize the production.

## The capability of Application:

Machine Learning recognizes patterns and themes in data by analyzing massive amounts of data and producing results in secs.

Machine Learning has the power to give much more in the Agriculture industry. This structure benefits both the nations and indeed the individual groups that make it up. Because it is a cloud-based system, it can be swiftly set up and connected with a wide range of hardware alternatives.

## Users Handling:

Estimation: The user may examine the modules they wish to test here. They wish to forecast by filling out all of the essential data, and the outcome should be shown as a number value (this value contains how much production is there in a particular crop).

# Methodology

## Dataset

In the dataset we have the following columns:

**Nitrogen (N)** – the ratio of Nitrogen content in the soil

**Phosphorus (P)** – the ratio of Phosphorous content in the soil

**Potassium (K)** – the ratio of Potassium content in the soil

**Temperature** – the temperature in degrees Celsius

**Humidity** - relative humidity in %

**PH** - ph. value of the soil

**Rainfall** - rainfall in mm

## Flowchart Steps:

1. Get an Official Dataset from Govt. Site
2. Train and test a best ML Model
3. Deploy Auto Al Model
4. Create Flask Framework
5. Integrate ML Model in Crop Recommendation App
6. Launch the App

## Idea:

* Producers may use the solution to decrease risk in their area of work and obtain the best results with the smallest amount of effort. The concept is made up of three parts:
* An artificial intelligence (AI) algorithm that proposes products to farmers and offers income statistics.
* Real-Time Forecast Analysis Using an Online Application and Device. Producers may use a Flask Web Server gateway to access these Resources.

## Existing System

The existing system predicts the crop through analysis made on just Climatic attributes (rainfall, temperature, humidity). This may not give the most accurate result as several parameters are not considered. As the application is aimed at farmers and related people, it is necessary to have a simple yet effective user interface for such users. Various projects use different algorithms for prediction, in this project the team has used the Random Forest algorithm.

## Proposed Solution

We propose an efficient crop guidance system that uses machine learning to estimate crop compatibility while considering all relevant parameters such as temperature, rainfall, geography, and soil conditions. The primary purpose of this system is to carry out Agro-Consultant's primary obligation, which is to offer crop advice to farmers.

After brainstorming a lot on how to deal with the problem at hand we decided to follow certain steps. We defined the scope of the project to the state of Maharashtra and the various districts in it. We then tackled the root cause of the problem statement. We found out that the atmospheric parameters and the soil conditions are the main factors behind the growth of any crop. That was the beginning of our steps. We determined the various atmospheric conditions required for the main 15 different crops grown around Maharashtra. We found favorable conditions viz. rainfall, temperature, humidity, and wind speed for these 15 crops. A dataset of the same was created which had all these values and the weather API key was given to the model which gave real-time district-wise weather conditions. The farmer just has to mention his district and all the weather parameters are automatically fetched.

Next, we found out the favorable soil conditions required for each of these crops. Various conditions like nitrogen(N), phosphorous(P), and potassium(K) and were determined for each crop. Even favorable pH levels were obtained. The dataset was formed. The model was trained on this dataset. However, the farmer will have to test the soil on his farm and put the values of NPK and pH as input to the model.

The farming sector may experience a paradigm shift thanks to AI solutions. With less effort and greater outcomes, these solutions can lessen the physical and emotional strain farmers experience, resulting in Best Crop being suited for the area right now.

# Application

* This solution can be applied in the modern farming sector.
* Accurate Crop and its production can be predicted.
* Both small-scale and large-scale farms can use the Crop Speak and Crop Sense Systems. Even urban gardens can employ the Crop Sense IoT device. The Crop Sense Device can automate irrigation systems. Combining Hardware and Software Solutions using the IBM IoT Platform enables the automatic watering of plants.
* This app can assist newcomers to farm or garden by explaining the times for planting and harvesting various crops.
* Agriculture's modernization can benefit society as a whole. If irrigation and fertilizer systems are automated, it is no longer required to stand in the sun for hours each day.
* An emerging sector that has the potential to improve employment, GDP, and productivity of the country is automated farming. If quick modernization can be applied to this industry, crops can be grown in excess and exported to other nations.

## EXPERIMENTAL INVESTIGATIONS:

90% of the data was used for training the algorithm and the rest 10% for test purposes. In the case of crop production data, regression prediction type.

# Tools/ Technologies used

## Software requirements:

1. Jupyter Notebook (Anaconda Navigator
2. Sublime Text (Version: 3.2)

## Languages:

1. Machine Learning
2. Python 3.7
3. Flask
4. JavaScript
5. HTML5 and CSS3

# Advantages

The advantages of this system are many. The major advantage is that it recommends crops suitable to the conditions hence avoiding damage to crops that would have otherwise been planted. It uses 2 models to predict crop and yield. It recommends crops based on real-time weather conditions. It also considers the soil conditions for recommending the crop. It also predicts the yield, production, and even revenue.

* Using these automatically generated recommendations based on one of their soil efficiency that was 99.1% accurate.
* The dashboard shows real-time predictive analysis provided by IoT sensors.
* If consumers don't have a crop sense sensor, they can enter their sensor value data on an interactive website.
* You can use the Website without paying anything.
* This dataset is compiled by using data from an oﬃcial government site which proves its authenticity.
* Farmers can directly ﬁnd out if the crop they are about to sow will result in proﬁt after cultivation.

# Disadvantages

The disadvantage is that the scope is limited to the region of Maharashtra and only to the main crops. Farmers have to do soil testing every time before taking up a new crop. The weather data is limited only to a few days.

* Crop Sense Sensors must have access to the Internet to connect to the Cloud. It's post something maybiblee in India's interior.
* Farmers with lesser acreage may not be able to afford the AI-Assisted Solution's additional cost for Crop Sense Sensors.

# Appendix: Pilot Project - 2

# METHODOLOGY

## Gathering the Dataset

A healthy or productive vegetation or crop needs a precise temperature, humidity, soil pH, sunlight, and moisture in the soil range. Several requirements must be fulfilled too to obtain a satisfied. However, according to the plant species, such criteria may vary. The original data set is provided by the Department of Agriculture, numerous agricultural publications, agricultural sites, and other publications and journal articles. The categorization cropping system was trooping utilizing this first set of data to increase accuracy.

I utilized this 1000-row dataset to create an experimental setup and find the optimal machine learning method that will produce the greatest results or accuracy. The code was created in Python.

## Data Analysis

Computer modeling is the preparatory process of performing an investigation on the information to uncover patterns, stomal, Bayesian inference, as well as double hypotheses using summary statistics and visualizations.

### Visualizing the soil's nitrogen-and-phosphorus ratio

Organic soil matter that has decomposed releases nitrogen into the environment. Phosphorus is produced through the breakdown of soil organic matter and minerals.

Factor1**=** plt**.**figure(figsize**=**(20,5))

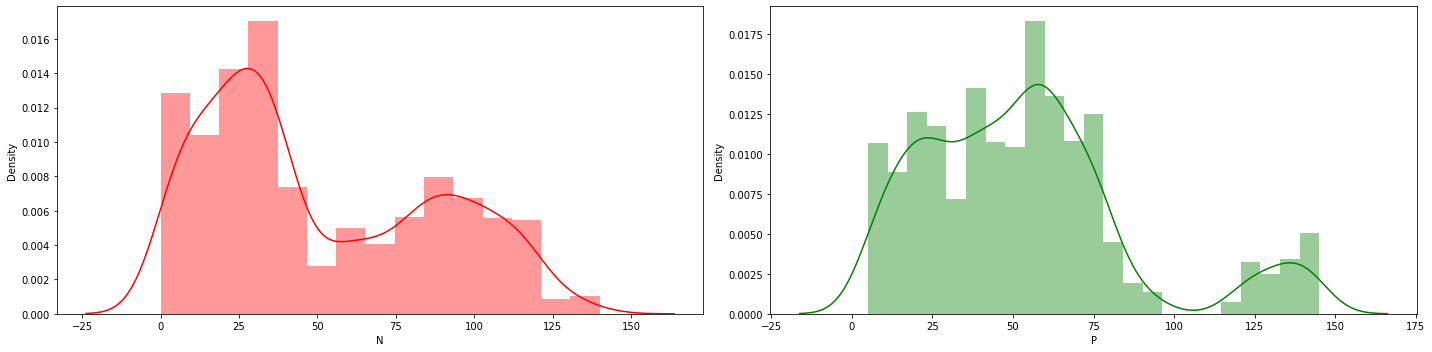
axis1**=**Factor1**.**add\_subplot(122)

sns**.**distplot(data['N'] , color **=**'red',axis**=**axis1)

axis2**=**Factor1**.**add\_subplot(121)

sns**.**distplot(data['P'] , color **=**'green' , axis **=** axis2)

plt**.**tight\_layout()



### Visualizing the Ratio of Potassium and Temperature in the soil

Potassium is a critical nutrient that plants absorb from the soil and fertilizer. It increases disease resistance, helps stalks to grow upright and sturdy, improves drought tolerance, and helps plants get through the winter.

Factor2**=** plt**.**figure(figsize**=**(20,5))

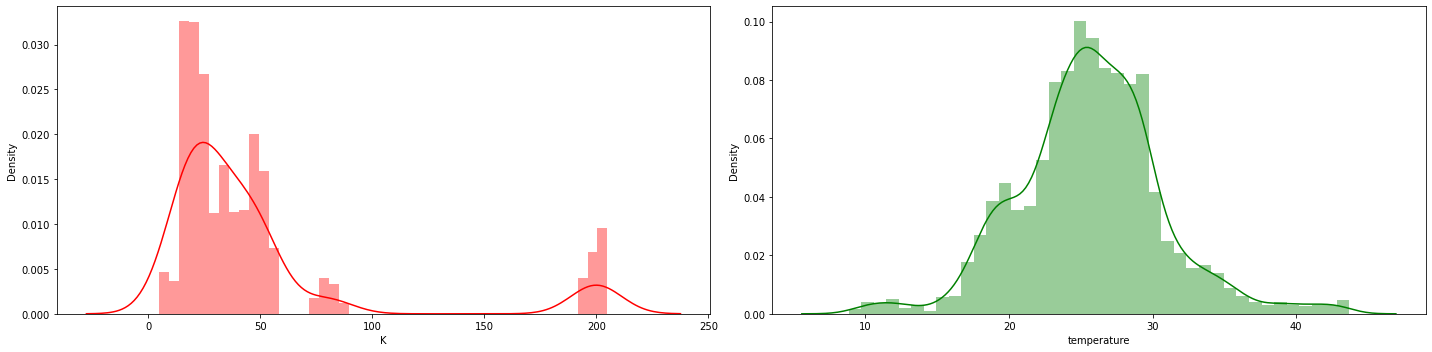
axis3**=**f**.**add\_subplot(123)

sns**.**distplot(data['K'] , color **=**'red',axis3**=**axis3)

axis4**=**Factor2**.**add\_subplot(121)

sns**.**distplot(data['temperature'] , color **=**'green' , axis4 **=** axis4)

plt**.**tight\_layout()



### Visualizing the Humidity and pH in the soil

As moisture increased, pH increased, whereas redox potential (Eh) decreased, and consequently, soil Eh and pass were negatively correlated.

Factor3**=** plt**.**figure(figsize**=**(20,5))

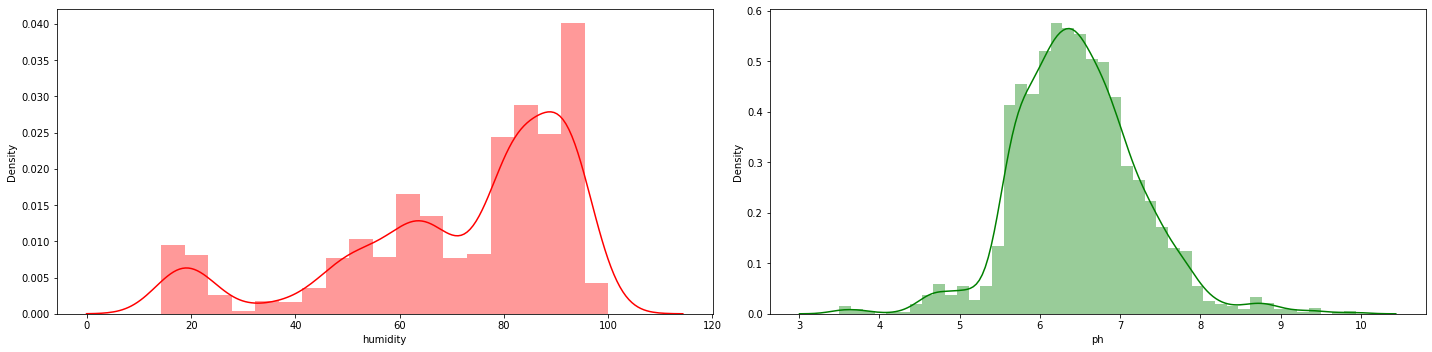
axis5**=**Factor3**.**add\_subplot(122)

sns**.**distplot(data['humidity'] , color **=**'red',axis5**=**axis5)

axis6**=**Factor3**.**add\_subplot(122)

sns**.**distplot(data['ph'] , color **=**'green' , axis6 **=** axis6)

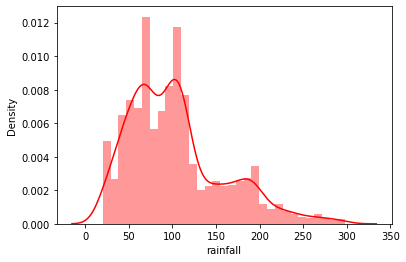
plt**.**tight\_layout()



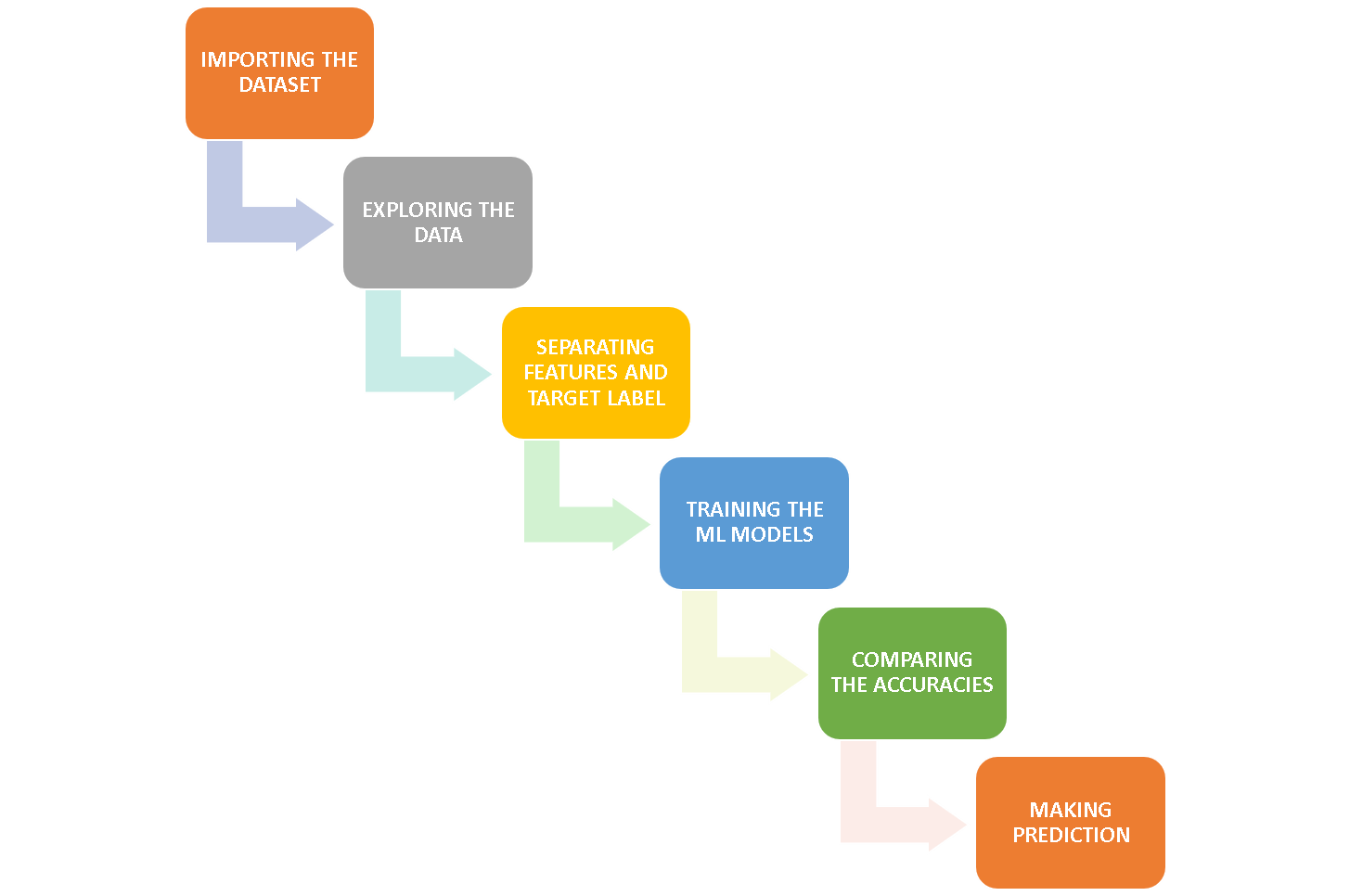
### Visualizing the Rainfall

Rainfall also has a significant impact on soil. If the soil is excessively moist or too dry, nutrients in the soil might wash off and not reach the roots of the plants, resulting in poor development and general health. Furthermore, as previously said, overwatering or excessive rain can promote the growth of bacteria, fungi, and mold in the soil.

sns.distplot(data['rainfall'],color ='red')



## Module workflow



## Methods

**df1 = pd.read\_csv('crop\_recommendations.CSV)**

**df1.head()**

This function calls the excel dataset to get the data in a compiler.

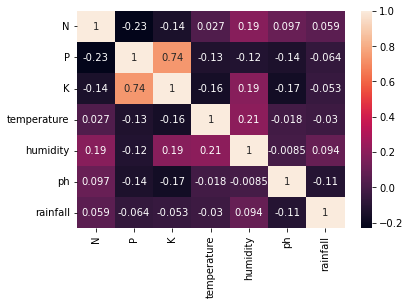
**df1['label'].value\_counts()**

This function counts the total number of rows for a specified category in a label

|  |  |
| --- | --- |
| rice 100 | rice 100 |
| maize 99 | maize 98 |
| jute 99 | jute 100 |
| cotton 98 | cotton 99 |
| coconut 100 | coconut 100 |
| papaya 100 | papaya 100 |
| orange 95 | orange 99 |
| apple 97 | apple 99 |
| muskmelon 98 | muskmelon 97 |
| watermelon 100 | watermelon 96 |
| grapes 99 | grapes 100 |
| mango 99 | mango 100 |
| banana 96 | banana 97 |
| pomegranate 95 | pomegranate 98 |

**sns.heatmap(df.corr(),annot = True)**

Heatmaps use color changes like hue, saturation, or brightness to depict the data as 2-D colored maps. Instead of using numbers to represent relationships between variables, heatmaps use colors. It shows the correlation between different features.



**Separating features and target label**

features = df[['N', 'P', 'K',' temperature', 'humidity', 'ph', 'rainfall']]

target = df['label']

#features = df[['temperature', 'humidity', 'ph', 'rainfall']]

labels = df['label']

**# Initializing empty lists to append all model's names and corresponding name**

Accuracy1 = []

model = []

**# Splitting into train and test data**

**Test and train the dataset with 20% of the test and 80%the of the trained dataset**

from sklearn.model\_selection import train\_test\_split

Xtrain, Xtest, Ytrain, Ytest = train\_test\_split(features, target, test\_size = 0.2,random\_state =2)

### Decision Tree

from sklearn.tree import DecisionTreeClassifier

DecisionTree = DecisionTreeClassifier(criterion="entropy",random\_state=2,max\_depth=5)

DecisionTree.fit(Xtrain, Ytrain)

Decision\_predicted\_values = DecisionTree.predict(Xtest)

Decision\_x = metrics.accuracy\_score(Ytest, Decision\_predicted\_values)

Accuracy.append(Decision\_x)

model.append('Decision Tree')

print("DecisionTrees's Accuracy is: ", Decision\_x\*100)

print(classification\_report(Ytest,Decision\_predicted\_values))

**OUTPUT:**

DecisionTrees's **Accuracy is: 90.0**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **precision** | **recall** | **f1-score** | **support** |
|  |  |  |  |  |
| apple | 1 | 1 | 1 | 13 |
| banana | 1 | 1 | 1 | 17 |
| Black gram | 0.59 | 1 | 0.74 | 16 |
| chickpea | 1 | 1 | 1 | 21 |
| coconut | 0.91 | 1 | 0.95 | 21 |
| coffee | 1 | 1 | 1 | 22 |
| cotton | 1 | 1 | 1 | 20 |
| grapes | 1 | 1 | 1 | 18 |
| jute | 0.74 | 0.93 | 0.83 | 28 |
| Kidney beans | 0 | 0 | 0 | 14 |
| lentil | 0.68 | 1 | 0.81 | 23 |
| maize | 1 | 1 | 1 | 21 |
| mango | 1 | 1 | 1 | 26 |
| Moth beans | 0 | 0 | 0 | 19 |
| mungbean | 1 | 1 | 1 | 24 |
| muskmelon | 1 | 1 | 1 | 23 |
| orange | 1 | 1 | 1 | 29 |
| papaya | 1 | 0.84 | 0.91 | 19 |
| Pigeon peas | 0.62 | 1 | 0.77 | 18 |
| pomegranate | 1 | 1 | 1 | 17 |
| rice | 1 | 0.62 | 0.77 | 16 |
| watermelon | 1 | 1 | 1 | 15 |
|  |  |  |  |  |
| **Accuracy** | 0.9 | 440 |  |  |
| **Macro-average** | 0.84 | 0.88 | 0.85 | 440 |
| **Average** | 0.86 | 0.9 | 0.87 | 440 |

#### Cross-validation Score (Decision Tree)

Accurate score = cross\_val\_score (DecisionTree, features, target, cv=5)

Accurate score

**OUTPUT:**

array([0.9363, 0.9090, 0.9181, 0.8704, 0.9363])

### Gaussian Naive Bayes

from sklearn.naive\_bayes import GaussianNB

Naïve\_Bayes = GaussianNB()

Naïve\_Bayes.fit(Xtrain,Ytrain)

Naïve\_Bayes\_predicted\_values = Naïve\_Bayes.predict(Xtest)

Naïve\_Bayes\_x = metrics.accuracy\_score(Ytest, Naïve\_Bayes\_predicted\_values)

Accuracy.append(Naïve\_Bayes\_x)

model.append('Naive\_Bayes')

print("Naive Bayes's Accuracy is: ", Naïve\_Bayes\_x)

print(classification\_report(Ytest, Naïve\_Bayes\_predicted\_values))

**OUTPUT:**

Naive Bayes's **Accuracy is: 0.990909090909091**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1-score** | **Support** |
| Apple | 1 | 1 | 1 | 13 |
| Anana | 1 | 1 | 1 | 17 |
| Black gram | 1 | 1 | 1 | 16 |
| Chickpea | 1 | 1 | 1 | 21 |
| Coconut | 1 | 1 | 1 | 21 |
| Coffee | 1 | 1 | 1 | 22 |
| Cotton | 1 | 1 | 1 | 20 |
| Grapes | 1 | 1 | 1 | 18 |
| Jute | 0.88 | 1 | 0.93 | 28 |
| Kidney beans | 1 | 1 | 1 | 14 |
| Lentil | 1 | 1 | 1 | 23 |
| Maize | 1 | 1 | 1 | 21 |
| Mango | 1 | 1 | 1 | 26 |
| Moth beans | 1 | 1 | 1 | 19 |
| Mungbean | 1 | 1 | 1 | 24 |
| Muskmelon | 1 | 1 | 1 | 23 |
| Orange | 1 | 1 | 1 | 29 |
| Papaya | 1 | 1 | 1 | 19 |
| Pigeon peas | 1 | 1 | 1 | 18 |
| Pomegranate | 1 | 1 | 1 | 17 |
| Rice | 1 | 0.75 | 0.86 | 16 |
| Watermelon | 1 | 1 | 1 | 15 |
|  |  |  |  |  |
| **Accuracy** | 0.99 | 440 |  |  |
| **Macro-average** | 0.99 | 0.99 | 0.99 | 440 |
| **Average** | 0.99 | 0.99 | 0.99 | 440 |

### Support Vector Machine (SVM)

from sklearn.SVM import SVC

## To normalize the data with sklearn

from sklearn.preprocessing import MinMaxScaler

## Scaler fit on training data in SVM

Normalization\_SVM = MinMaxScaler().fit(Xtrain)

X\_train\_normalization = Normalization\_SVM.transform(Xtrain)

## Transforming the testing database by using normalization

X\_test\_normalization = Normalization\_SVM.transform(Xtest)

SVM\_norm = SVC(kernel='poly', degree=3, C=1)

SVM\_norm.fit(X\_train\_normalization, Ytrain)

SVM\_predicted\_values = SVM\_norm.predict(X\_test\_normalization)

Normalization\_x = metrics.accuracy\_score(Ytest, SVM\_predicted\_values)

Accuracy.append(Normalization\_x)

model.append('SVM\_norm')

print("SVM's Accuracy is: ", Normalization\_x)

print(classification\_report(Ytest,SVM\_predicted\_values))

**OUTPUT:**

SVM's **Accuracy through Normalization is: 0.9795454545454545**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1-score** | **Support** |
|  |  |  |  |  |
| Apple | 1 | 1 | 1 | 13 |
| Banana | 1 | 1 | 1 | 17 |
| Black gram | 1 | 1 | 1 | 16 |
| Chickpea | 1 | 1 | 1 | 21 |
| Coconut | 1 | 1 | 1 | 21 |
| Coffee | 1 | 0.95 | 0.98 | 22 |
| Cotton | 0.95 | 1 | 0.98 | 20 |
| Grapes | 1 | 1 | 1 | 18 |
| Jute | 0.83 | 0.89 | 0.86 | 28 |
| Kidney beans | 1 | 1 | 1 | 14 |
| Lentil | 1 | 1 | 1 | 23 |
| Maize | 1 | 0.95 | 0.98 | 21 |
| Mango | 1 | 1 | 1 | 26 |
| Moth beans | 1 | 1 | 1 | 19 |
| Mungbean | 1 | 1 | 1 | 24 |
| Muskmelon | 1 | 1 | 1 | 23 |
| Orange | 1 | 1 | 1 | 29 |
| Papaya | 1 | 1 | 1 | 19 |
| Pigeon peas | 1 | 1 | 1 | 18 |
| Pomegranate | 1 | 1 | 1 | 17 |
| Rice | 0.8 | 0.75 | 0.77 | 16 |
| Watermelon | 1 | 1 | 1 | 15 |
|  |  |  |  |  |
| **Accuracy** | 0.98 | 440 |  |  |
| **Macro-Average** | 0.98 | 0.98 | 0.98 | 440 |
| **Average** | 0.98 | 0.98 | 0.98 | 440 |

### Logistic Regression

from sklearn.linear\_model import LogisticRegression

Logistic\_Regre = LogisticRegression(random\_state=2)

Logistic\_Regre.fit(Xtrain,Ytrain)

Logistic\_Regre\_predicted\_values = Logistic\_Regre.predict(Xtest)

Logistic\_Regre\_x = metrics.accuracy\_score(Ytest, Logistic\_Regre\_predicted\_values)

Accuracy.append(Logistic\_Regre\_x)

model.append('Logistic Regression)

print("Logistic Regression's Accuracy is: ", Logistic\_Regre\_x)

print(classification\_report(Ytest, Logistic\_Regre\_predicted\_values))

**OUTPUT:**

Logistic Regression's **Accuracy is: 0.9522727272727273**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1-score** | **Support** |
|  |  |  |  |  |
| Apple | 1 | 1 | 1 | 13 |
| Banana | 1 | 1 | 1 | 17 |
| Blackgram | 0.86 | 0.75 | 0.8 | 16 |
| Chickpea | 1 | 1 | 1 | 21 |
| Coconut | 1 | 1 | 1 | 21 |
| Coffee | 1 | 1 | 1 | 22 |
| Cotton | 0.86 | 0.9 | 0.88 | 20 |
| Grapes | 1 | 1 | 1 | 18 |
| Jute | 0.84 | 0.93 | 0.88 | 28 |
| Kidney beans | 1 | 1 | 1 | 14 |
| Lentil | 0.88 | 1 | 0.94 | 23 |
| Maize | 0.9 | 0.86 | 0.88 | 21 |
| Mango | 0.96 | 1 | 0.98 | 26 |
| Moth beans | 0.84 | 0.84 | 0.84 | 19 |
| Mungbean | 1 | 0.96 | 0.98 | 24 |
| Muskmelon | 1 | 1 | 1 | 23 |
| Orange | 1 | 1 | 1 | 29 |
| Papaya | 1 | 0.95 | 0.97 | 19 |
| Pigeon peas | 1 | 1 | 1 | 18 |
| Pomegranate | 1 | 1 | 1 | 17 |
| Rice | 0.85 | 0.69 | 0.76 | 16 |
| Watermelon | 1 | 1 | 1 | 15 |
|  |  |  |  |  |
| **Accuracy** | 0.95 | 440 |  |  |
| **Macro-Average** | 0.95 | 0.95 | 0.95 | 440 |
| **Weighted average** | 0.95 | 0.95 | 0.95 | 440 |

### Random Forest

from sklearn.ensemble import RandomForestClassifier

Random\_For = RandomForestClassifier(n\_estimators=20, random\_state=0)

Random\_For.fit(Xtrain,Ytrain)

Random\_For\_predicted\_values = Random\_For.predict(Xtest)

Random\_For\_x = metrics.accuracy\_score(Ytest, Random\_For\_predicted\_values)

Accuracy.append(Random\_For\_x)

model.append(' Random\_For')

print("RF's Accuracy is: ", x)

print(classification\_report(Ytest, Random\_For\_predicted\_values))

**OUTPUT:**

Random\_Forest's **Accuracy is: 0.9931818181818182**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1-score** | **Support** |
|  |  |  |  |  |
| Apple | 1 | 1 | 1 | 13 |
| Banana | 1 | 1 | 1 | 17 |
| Black gram | 0.94 | 1 | 0.97 | 16 |
| Chickpea | 1 | 1 | 1 | 21 |
| Coconut | 1 | 1 | 1 | 21 |
| Coffee | 1 | 1 | 1 | 22 |
| Cotton | 1 | 1 | 1 | 20 |
| Grapes | 1 | 1 | 1 | 18 |
| Jute | 0.9 | 1 | 0.95 | 28 |
| Kidney beans | 1 | 1 | 1 | 14 |
| Lentil | 1 | 1 | 1 | 23 |
| Maize | 1 | 1 | 1 | 21 |
| Mango | 1 | 1 | 1 | 26 |
| Moth beans | 1 | 0.95 | 0.97 | 19 |
| Mungbean | 1 | 1 | 1 | 24 |
| Muskmelon | 1 | 1 | 1 | 23 |
| Orange | 1 | 1 | 1 | 29 |
| Papaya | 1 | 1 | 1 | 19 |
| Pigeon peas | 1 | 1 | 1 | 18 |
| Pomegranate | 1 | 1 | 1 | 17 |
| Rice | 1 | 0.81 | 0.9 | 16 |
| Watermelon | 1 | 1 | 1 | 15 |
|  |  |  |  |  |
| **Accuracy** | 0.99 | 440 |  |  |
| **Macro Average** | 0.99 | 0.99 | 0.99 | 440 |
| **Weighted Average** | 0.99 | 0.99 | 0.99 | 440 |

### XGBoost

import xgboost as xgb1

X\_G\_B = xgb1.XGBClassifier()

X\_G\_B.fit(Xtrain,Ytrain)

X\_G\_B\_predicted\_values = X\_G\_B.predict(Xtest)

X\_G\_B\_x = metrics.accuracy\_score(Ytest, X\_G\_B\_predicted\_values)

Accuracy.append(x)

model.append('XGBoost')

print("XGBoost's Accuracy is: ", X\_G\_B\_x)

print(classification\_report(Ytest, X\_G\_B\_predicted\_values))

**OUTPUT:**

XGBoost's **Accuracy is: 0.990909090909091**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1-score** | **Support** |
|  |  |  |  |  |
| Apple | 1 | 1 | 1 | 13 |
| Banana | 1 | 1 | 1 | 17 |
| Blackgram | 1 | 1 | 1 | 16 |
| Chickpea | 1 | 1 | 1 | 21 |
| Coconut | 1 | 1 | 1 | 21 |
| Coffee | 0.96 | 1 | 0.98 | 22 |
| Cotton | 1 | 1 | 1 | 20 |
| Grapes | 1 | 1 | 1 | 18 |
| Jute | 1 | 0.93 | 0.96 | 28 |
| Kidney beans | 1 | 1 | 1 | 14 |
| Lentil | 0.96 | 1 | 0.98 | 23 |
| Maize | 1 | 1 | 1 | 21 |
| Mango | 1 | 1 | 1 | 26 |
| Moth beans | 1 | 0.95 | 0.97 | 19 |
| Mungbean | 1 | 1 | 1 | 24 |
| Muskmelon | 1 | 1 | 1 | 23 |
| Orange | 1 | 1 | 1 | 29 |
| Papaya | 1 | 1 | 1 | 19 |
| Pigeon peas | 1 | 1 | 1 | 18 |
| Pomegranate | 1 | 1 | 1 | 17 |
| Rice | 0.94 | 1 | 0.97 | 16 |
| Watermelon | 1 | 1 | 1 | 15 |
|  |  |  |  |  |
| **Accuracy** | 0.99 | 440 |  |  |
| **Macro Average** | 0.99 | 0.99 | 0.99 | 440 |
| **Weighted Average** | 0.99 | 0.99 | 0.99 | 440 |

#### **Cross-validation score (XGBoost)**

XGB\_scoring **=** cross\_val\_score(X\_G\_B,features, target, cv**=**5)

XGB\_scoring

**OUTPUT:**

array([0.990909090909091, 0.9919190909091, 0.99119119090909, 0.991911909090])

## Accuracy Comparison

plt**.**figure(figsize**=**[10,5],dpi **=** 100)

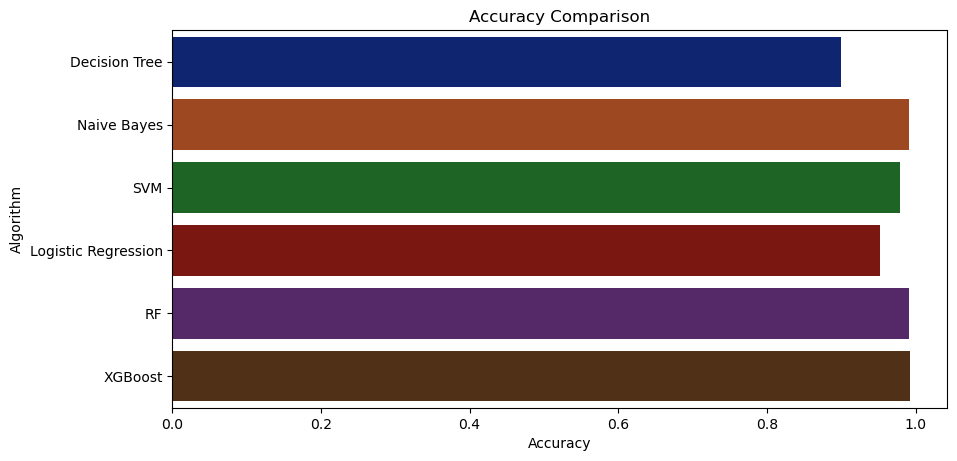
plt**.**title('Accuracy Comparison')

plt**.**xlabel('Accuracy')

plt**.**ylabel('Algorithm')

sns**.**barplot(x **=** Accuracy, y **=** model,palette**=**'dark')

**OUTPUT:**



## Accuracy differentiates between all the methods:

accuracy\_models\_algo **=** dict(zip(model, Accuracy))

for k, v in accuracy\_models\_algo**.**items():

print (k, '-->', v)

**OUTPUT:**

Decision\_Tree Accuracy--> 0.9

Naïve\_Bayes Accuracy --> 0.990909090909091

SVM Normalization Accuracy--> 0.9795454545454545

Logistic\_Regression Accuracy--> 0.9522727272727273

**RF Accuracy--> 0.9931818181818182**

XGBoost Accuracy--> 0.990909090909091

# Result

I am checking various Machine Learning models in a small dataset and the Random Forest best fits this model. I checked through **R- Square** Method it shows the Score **“0.9931818181818182”** accuracy**.**

## Making a prediction

Data\_predict **=** np**.**array([[105,19, 34, 21.6036, 63.3, 6.7, 140.91]])

prediction\_result **=** RF**.**predict(Data\_predict)

print(prediction\_result)

**OUTPUT:**

['coffee']

Data\_predict**=** np**.**array([[83, 45, 60, 28, 70.3, 7.0, 150.9]])

Prediction\_result **=** RF**.**predict(Data\_predict)

print(prediction\_result)

**OUTPUT:**

['jute']

# Conclusion

A study of three distinct kinds of supervised machine-learning models (SVM, Decision Tree, and Random-Forest) is performed to discover the crop that will grow the most effective on a given piece of land. We determined that the crop forecast dataset had the greatest accuracy using Random-Forest-Classifier (RF) in both the Entropy and Inequality standards, with 99.32%. In comparison, SVM has the best error (97.95%) of the three, whereas the Decision Tree Classifier's performance is equivalent to Random Forest Classifier and SVM. When the brings valuable was examined, the Decision Tree Gini criterion produced a superior accuracy of 90 percentage points when contrasted to Decision Tree.

The Branch Density criteria of Future data from the fields might be collected to get a full image of the soil and add more machine-learning and deep-learning algorithms, such as Artificial-Neural-Network or Convolution Neural Networks, to categorize more sorts of crops.

The Random Forest Approach is the greatest fit for this project since it makes precise forecasts. The accuracy is tested using the R-Square met, which provides a value between 0 and 1, with 1 being the finest model and 0 being the worst.

# REFERENCES

1. Anastasiya Kolesnikova, Chi-Hwa Song, Won Don Lee, ”Applying UChooBoost algorithm in Precision Agriculture”. ACM International Conference on Advances in Computing, Communication, and Control, Mumbai, India, January 2009
2. Ashwani Kumar Kushwaha, Swetabhattachrya “crop yield prediction using the agro algorithm in Hadoop”.
3. Bharath S, Yeshwanth S, Yashas B L and Vidyaranya R Javalagi 2020 Comparative Analysis of Machine Learning Algorithms in The Study of Crop and Crop yield Prediction International Journal of Engineering Research & Technology (IJERT) NCETESFT – 2020 vol 8 Issue 14.
4. Brouwer and M. Heibloem, Irrigation Water Management: Irrigation Water Needs, manual 6 Reading, ITALY: Food and Agriculture Organization of the United Nations, 1987. [Online] Available: [http://www.fao.org/3/s2022e/s2022e07.htm#TopOfPage.](http://www.fao.org/3/s2022e/s2022e07.htm#TopOfPage)
5. CH. Vishnu Vardhanchowdary, Dr.K.Venkataramana, Tomato Crop Yield Prediction using ID3, March 2018, IJIRT Volume 4 Issue 10 pp,663-62.
6. Crop Selection Method to Maximize Crop Yield Rate using Machine Learning Technique Rakesh Kumar, M.P. Singh , Prabhat Kumar and J.P. Singh, Dept. of CSE NIT Patna, India
7. Dakar S and Rode S V 2014 Agricultural crop yield prediction using artificial neural network approach International Journal of Innovative Research in Electrical, Electronics, Instrumentation, and Control Engineering vol 2 Issue 1 pp 683-6.
8. de Carbon, "PRECISION AGRICULTURE: ITS BENEFITS AND LIMITATIONS", carrhure, 2019 [Online]. Available: https://[www.carrhure.com/precision-](http://www.carrhure.com/precision-) agriculture-benefits-limitations/ [Accessed: Feb.25, 2020].
9. Department of Agriculture Sri Lanka (2015), ‘Crop Suitability Recommendation for Grama Niladhari Divisions in Sri Lanka’, Natural Resources Management Center, Peradeniya.
10. G. Chauhan and A. Chaudhary, ”Crop Recommendation System using Machine Learning Algorithms,” 2021 10th International Conference on System Model- ing & Advancement in Research Trends (SMART), 2021, pp. 109-112, doi: 10.1109/SMART52563.2021.9676210.
11. G. Chauhan and A. Chaudhary, ”Crop Recommendation System using Machine Learning Algorithms,” 2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART), 2021, pp. 109-112, doi: 10.1109/SMART52563.2021.9676210.
12. Geetha V, Punitha A, Abarna M, Akshaya M, Illakiya S and Janani A P 2020 An effective crop prediction using random forest algorithm 2020 International Conference on System, Computation, Automation, and Networking (ICSCAN) pp 1-5 DOI: 10.1109/ICSCAN49426.2020.9262311.
13. Gulati P and Jha S K 2020 Efficient crop yield prediction in India using machine learning techniques International Journal of Engineering Research & Technology (IJERT) ENCADEMS – 2020 vol 8 Issue 10.
14. Gupta A, Nagda D, Nikhare P, Sandbhor A, 2021, Smart crop prediction using IoT and machine learning International Journal of Engineering Research & Technology (IJERT) NTASU – 2020 vol 9 Issue 3.
15. Hassan SM, Maji AK, Jasin´ski M, Leonowicz Z, Jasin´ska E. Identification of Plant-Leaf Diseases Using CNN and Transfer-Learning Approach. Electronics. 2021; 10(12):1388
16. Himani Sharma, Sunil Kumar “A Survey on Decision Tree Algorithms of Classification in Data Mining”.
17. J. Yang, X. Yang and J. Zhang, ”A Parallel Multi- Class Classification Support Vector Machine Based on Sequential Minimal Optimization,” First International Multi-Symposiums on Computer and Computational Sci- ences (IMSCCS’06), 2006, pp. 443-446, doi: 10.1109/IM- SCCS.2006.20.
18. Kalimuthu M, Vaishnavi P and Kishore M 2020 Crop prediction using machine learning 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT) pp 926-32 doi: 10.1109/ICSSIT48917.2020.9214190.
19. Kulkarni, P., Karwande, A., Kolhe, T., Kamble, S., Joshi, A. and Wyawahare, M., 2021. Plant Disease Detection Using Image Processing and Machine Learn- ing. arXiv preprint arXiv:2106.10698.
20. Kulkarni, P., Karwande, A., Kolhe, T., Kamble, S., Joshi, A. and Wyawahare, M., 2021. Plant Disease Detection Using Image Processing and Machine Learning. arXiv preprint arXiv:2106.10698.
21. Lakshmi.N, Priya.M, Sahana Shetty, and Manjunath C. R, Crop Recommendation System for Precision Agriculture, vol. 6 Reading, IND: International Journal for Research in Applied Science & Engineering Technology, 2018. [Online] Available: [www.ijraset.com.Ding,](http://www.ijraset.com.Ding/) W. and Marchionini, G. 1997 A Study on Video Browsing Strategies. Technical Report. University of Maryland at College Park.
22. M. Suganya., Dayana R and Revathi.R, Crop Yield Prediction Using Supervised Learning Techniques, International Journal of Computer Engineering and Technology, 11(2), 2020, pp. 9-20
23. Mahendra N, Vishwakarma D, Nischitha K, Ashwini and Manjuraju M. R 2020 Crop prediction using machine learning approaches, International Journal of Engineering Research & Technology (IJERT) vol 9 Issue 8 (August 2020).
24. Marie ionnotti (2020), ‘Outdoors & Gardning’, The Spruce Web Site.
25. Medar R, Rajpurohit V S and Shweta S 2019 Crop yield prediction using machine learning techniques IEEE 5th International Conference for Convergence in Technology (I2CT) pp 1-5 doi: 10.1109/I2CT45611.2019.9033611.
26. N. Gandhi, L. J. Armstrong, O. Petkar and A. K. Tripathy, Rice crop yield prediction in India using support vector machines 2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE), KhonKaen, 2016, pp. 1-5
27. N. P. Sastra and D. M. Wiharta, ―Environmental monitoring as an IoT application in building smart campus of UniversitasUdayana,ǁ in Proc. Int. Conf. Smart Green Technol. Elect. Inf. Syst. (ICSGTEIS), Oct. 2016, pp. 85–88.
28. Nishant P S, Venkat P S, Avinash B L and Jabber B 2020 Crop yield prediction based on Indian agriculture using machine learning 2020 International Conference for Emerging Technology (INCET) pp 1-4 doi: 10.1109/INCET49848.2020.9154036.
29. P. Revathi, R. Revathi, Dr.M.Hemalatha, ”Compara- tive Study of Knowl- edge in Crop Diseases Using Machine Learning Techniques”. Inter- national Journal of Computer Science and Information Technologies (IJCSIT), Vol. 2 (5), 2180-2182, 2011
30. Pande S M, Ramesh P K, Anmol A, Aishwaraya B R, Rohilla K and Shaurya K 2021 Crop recommender system using machine learning approach 2021 5th International Conference on Computing Methodologies and Communication (ICCMC) pp 1066-71 doi: 10.1109/ICCMC51019.2021.9418351.
31. Pavan Patil, Virendra Panpatil, Prof. Shrikant Kokate “Crop Prediction System using Machine Learning Algorithms”.
32. Pijush Samui, Venkata Ravibabu Mandla, Arun Krishna and Tarun Teja “Prediction of Rainfall Using Support Vector Machine and Relevance Vector Machine”.
33. Prof. D.S. Zingade ,Omkar Buchade ,Nilesh Mehta ,Shubham Ghodekar ,Chandan Mehta “Crop Prediction System using Machine Learning”.
34. R. Sujatha and P. Isakki, A study on crop yield forecasting using classiﬁcation techniques 2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16), Kovilpatti, 2016, pp. 1-4.
35. Rahul Katarya, Ashutosh Raturi, Abhinav Mehndiratta, Abhinav Thapper “Impact of Machine Learning Techniques in Precision Agriculture”.
36. Remi Schmaltz, “What is Precision Agriculture”, April 2017. [Online]. Available:https://agfundernews.com/what-is-precision-agriculture.html [Accessed Feb.23, 2020].
37. Rohith Gandhi (2018), ‘Support Vector Machine — Introduction to Machine Learning Algorithms’, Toward Data Science Web Site.
38. Samvelyan, R. Shaptala and G. Kyselov, ”Ex- ploratory data analysis of Kyiv city petitions,” 2020 IEEE 2nd International Conference on System Analysis & Intelligent Computing (SAIC), 2020, pp. 1-4, doi: 10.1109/SAIC51296.2020.9239185.
39. Sellam V, and Poovammal E 2016 Prediction of crop yield using regression analysis Indian Journal of Science and Technology vol 9(38) pp 1-5.
40. Sharada Prasanna Mohanty, David Hughes, Marcel Salathe, 2016, Using Deep Learning for Image-Based Plant Disease Detection. arXiv preprint arXiv:2106.10698.
41. Sharada Prasanna Mohanty, David Hughes, Marcel Salathe, 2016, Using Deep Learning for Image-Based Plant Disease Detection. arXiv preprint arXiv:2106.10698.
42. Simon Tavasoli (202), ‘10 Machine Learning Algorithms You Need to Know’, Simplilearn Solutions.
43. Sunil Ray (2015), ‘6 Easy Steps to Learn Naive Bayes Algorithm with codes in Python and R’, Analytics Vidhya
44. Suresh A, Ganesh P and Ramalatha M 2018 Prediction of major crop yields of Tamilnadu using K-means and Modified KNN 2018 3rd International Conference on Communication and Electronics Systems (ICCES) pp 88-93 doi: 10.1109/CESYS.2018.8723956.
45. WolkWriter (2019), ‘Periodically measure environmental conditions and send them to WolkAbout IoT Platform to monitor the environment remotely.’, ARDUINO PROJECT HUB.