AGRI-TECH
(Crop and Fertilizer Prediction)

A Report Submitted
In Partial Fulfillment of the Requirements
For the Degree of

MASTER OF COMPUTER APPLICATION

in
Computer Science

By
BHAWANA (21718)
KRISHAN KANT SINGH GAUTAM (21725)
KUNAL ASTAHNA (21727)
MOHD HESHAM ASHRAF ANSARI (21732)

Under the Supervision of
Prof. Nilesh Chandra
Prof. Kriti Chaurasia
Kamla Nehru Institute of Technology

Department Of Computer Science
Dr. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY,
LUCKNOW

May, 2023

DECLARATION

We hereby declare that the work presented by in this report entitled “AGRI-TECH”, was carried by us. We have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or institute.

We have given due to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution.

We have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.

We affirm that no portion of our work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, we shall be fully responsible and answerable.

BHAWANA (21718)

KRISHAN KANT SINGH GAUTAM (21725)

KUNAL ASTANA (21727)

MOHD HESHAM ASHRAF ANSARI (21732)

CERTIFICATE

This is to certify that **Bhawana (21718), Krishan Kant Singh Gautam (21725), Kunal Asthana (21727), Mohd Hesham Ashraf Ansari (21732)** has carried out the project work entitled **“AGRI-TECH (Crop and Fertilizer Prediction)”** for partial fulfilment for the award of **Master of Computer Application** at Kamla Nehru Institute of Technology, Sultanpur affiliated to Dr. A.P.J. Abdul Kalam Technical University, Lucknow. This report is the record of candidates own work carried out by then under our guidance. The Project work is the part of their Master of Computer Application curriculum. Their performance was excellent and we wish them good luck for their future Endeavour.

Prof. Nilesh Chandra	Prof. Kriti Chaurasia	Prof. Awadhesh Kumar
(Project Guide)	(Project Guide)	(Head of Department)

ACKNOWLEDGEMENT

I thank the people who were a part of this project in numerous ways, people who gave their unending support right from the stage the project idea was conceived. The four things that go on to make a successful endeavour are dedication, hard work, patience and correct guidance. I would like to thank our principle who has always been the source of inspiration. I take this opportunity to offer sincere thanks to **Prof. Nilesh Chandra** and **Prof. Kriti Chaurasia** who was very much kind enough to give us an idea and guide us throughout our project work. I am thankful to all teaching staff who shared their experience and gave their suggestions for developing our projects in better way. Last but not least I would like to thank all our friends and family members for their support, and all others who have contributed to the development to the completion to this project directly or indirectly.

ABSTRACT

India being an agriculture country, its economy predominantly depends on agriculture yield growth and AGRO-INDUSTRY products. Data Mining is an emerging research field in crop yield analysis. Yield prediction is a very important issue in agricultural. Any farmer is interested in knowing how much yield he is about to expect. Analyse the various related attributes like location, pH value from which alkalinity of the soil is determined. Along with it, percentage of nutrients like Nitrogen (N), Phosphorous (P), and Potassium (K) Location is used along with the use of third-party applications like APIs for weather and temperature, type of soil, nutrient value of the soil in that region, amount of rainfall in the region, soil composition can be determined. All these attributes of data will be analysed, train the data with various suitable machine learning algorithms for creating a model. The system comes with a model to be precise and accurate in predicting crop yield and deliver the end user with proper recommendations about required fertilizer ratio based on atmospheric and soil parameters of the land which enhance to increase the crop yield and increase farmer revenue.

INDEX

CHAPTER - 1

CHAPTER - 2

CHAPTER - 3

CHAPTER - 4

CHAPTER - 5

CHAPTER - 6

Table of figures

CHAPTER 1

INTRODUCTION

Agriculture is indeed one of the world's largest industries, providing food and raw materials to sustain human life and support economic development. However, the intensive farming practices that have been widely adopted since the 1960s have come with significant environmental costs. While these practices have led to high yields and increased productivity, they have also depleted the soil, contributed to greenhouse gas emissions, caused biodiversity loss, and led to ocean acidification, among other negative impacts.

To address these challenges, innovative solutions are required. One such solution is the development of an application integrated with machine learning that can help farmers grow the best crops in their farms, taking into account the specific demographic conditions of their land. By leveraging data analytics and advanced algorithms, this application can provide farmers with personalized recommendations for the optimal planting and harvesting times, as well as the most suitable fertilizer for a particular crop.

The integration of machine learning in agriculture has the potential to revolutionize the industry, by enabling farmers to make more informed decisions, increase productivity, and reduce the negative impact of farming practices on the environment. The application could help farmers optimize their use of resources, such as water and fertilizers, reducing waste and improving sustainability. This, in turn, can contribute to the production of healthier and more nutritious crops, while also supporting the long-term viability of the farming sector.

1.1 IMPORTANCE

Crop and fertilizer prediction is a very important issue in agricultural. Any farmer is interested in knowing how much crop he is about to expect. In the past, crop & fertilizer prediction was performed by considering farmer's experience on particular field and crop. The crop prediction is a major issue that remains to be solved based on available data. machine learning techniques are the better choice for this purpose. Different machine learning techniques are used and evaluated in agriculture for estimating the future year's crop prediction. This project proposes and implements a system to predict crop predict on the basis of pregiven dataset. This is achieved by applying association rule mining on agriculture data. This model focuses on creation of a prediction model which may be used to future prediction of crop yield. This paper presents a brief analysis of crop yield prediction using machine learning technique based on association rules for the selected region in India. The experimental results shows that the proposed work efficiently predict the crop & fertilizer prediction.

1.2 OBJECTIVE:

1.2.1 Improving Agriculture

Three-quarters of the world’s poorest people get their food and income from farming small plots of land—typically the size of a football field or smaller—and most of them labor under difficult conditions. They grow a diversity of local crops and must deal with unique diseases, pests, and drought, as well as unproductive soil. Their livestock are Partial owners in a local farming cooperative. Reliable markets for their products and good information about pricing are hard to come by.

Agricultural production has always involved the exploitation of resources such as soil, water, and energy. Increasing production to feed a growing world population while at the same time conserving resources for future generations has led to a search for ‘sustainable’ agricultural methods. Farm managers must take a long-term view when making decisions about which technologies to follow and what commodities to produce while still generating sufficient profits in the short run to earn a living. Farm managers must also be aware of possible trends in climatic conditions, and learn how to adapt their production methods accordingly.

1.2.2 Cultivating Crops To Make Farmers Increase Profit

This proposed system helps the farmers to select suitable crop based on season and region of sowing. It will in- turn help the farmers by reducing the loss faced by them and improve net crop yield.

Our Model predicts the suitable crop to yield on the basis of region, soil and weather at real time so it will

1.2.3 Improving Lives

Alongside dramatic increases in crop production over the last 50 years, global food systems have become more dependent on a few major `staple’ crops - just three cereals now provide about 60% of plant-based human energy intake.

Over the last 50 years, many sciences have contributed to the almost 5-fold increase in the production of chicken, 3-fold increase in several cereals (rice, maize, wheat), ocean fish and pig production, and near doubling in the yields of several other crops and animals (Godfray et al., 2010).

While these advances have reduced the proportion of humans suffering under-nutrition, only 15 crops now provide 90% of the world's food energy intake. Four crops (rice, maize, wheat and soybean) dominate global food systems. The three cereals provide the staple diet of more than four billion people and about 60% of plant-based human energy intake (IPES-Food 2016).

1.3 SERVICES

1.3.1 Crop Prediction-:

Crop prediction attributes are defined by multiple factors such as genotype, climate and the interactions between the two. Accurate crop prediction needs a fundamental understanding of the functional relationship between cultivation and interactive factors like the genotype and climate.

This process is done on the basis of listed factors-

- Measurement of NITROGEN
- Measurement of PHOSPHORUS
- Measurement of POTASSIUM
- Ph value
- Rainfall (in mm)
- Region

TYPES OF CROPS-:

- Maize
- Sugarcane
- Cotton
- Tobacco
- Paddy
- Barley
- Wheat
- Millets
- Oilseeds
- Pulses

- Groundnuts

1.3.2 ***Fertiliser Prediction-:***

The prediction of crop yield based on location and proper implementation of algorithms have proved that the higher crop yield can be achieved. From above work I conclude that for soil classification Random Forest is good with accuracy 86.35% compare to Support Vector Machine. For crop yield prediction Support Vector Machine is good with accuracy 99.47% compare to Random Forest algorithm. The work can be extended further to add following functionality. Mobile application can be built to help farmers by uploading image of farms. Crop diseases detection using image processing in which user get pesticides based on disease images. Implement Smart Irrigation System for farms to get higher yield.

The fertilizer prediction process is done on the basis of several listed factor-

- Measurements of Nitrogen
- Measurements of Phosphorous
- Measurements of Potassium
- Measurements of Moisture
- Types of crops want to grow
- Types of soil

TYPES OF SOIL-:

The soil is defined as the organic and inorganic materials on the earth's surface which provide the medium for the growth of the plants. It slowly develops over time and it is composed of various different materials and it varies due to its composition and structure. There are following types of soil:

- Sandy soil
- Clayey soil
- Loamy soil
- Black soil
- Red soil

TYPES OF FERTILISERS-:

- Urea
- DAP
- 14-35-14
- 28-28
- 20-20
- 17-17-17
- 10-26-26

CHAPTER 2

LITERATURE SURVEY

Clearly, a farmer is the best decision maker in the selection and cultivation of crops. Today, however, cultivar prediction is done manually in laboratories, and farmers need the help of experts to determine the most suitable crop/s for a specific piece of land. The experts collect soil samples from a particular portion of land and test them in the laboratory, following which they offer suggestions on the ideal crop/s to be raised. Prediction takes time, and selecting the most suitable crop/s is a complex task in agriculture. Manual prediction has largely failed, owing to climatic changes and environmental factors that affect crop cultivation. Accurate predictions of suitable crops for cultivation improve production levels. Crop prediction attributes are defined by multiple factors such as genotype, climate and the interactions between the two. Accurate crop prediction needs a fundamental understanding of the functional relationship between cultivation and interactive factors like the genotype and climate. Further, it requires both detailed datasets and efficient algorithms to examine these relationships. Justified by these facts, machine learning techniques are used in this study to predict the most suitable crop for a specific stretch of land, and this technique is ideal for considering factors like the soil and environmental conditions. A number of related studies are discussed in this review.

Sanmay Das discussed the pros and cons of the filter and wrapper methods, and implemented a new hybrid feature selection approach using the boosting technique. The experiments were carried out using real-world datasets from the University of California, Irvine (UCI) repository. The results proved that the proposed method is much faster than the wrapper method. Huan Liu and Lei Yu reviewed the existing feature selection algorithm for classification and clustering techniques. Subsequently, an intermediate step on a unifying platform was proposed in their work.

Al Maruf et al. demonstrated the superiority of the gapped k-mer composition and reverse complement features of the k-mer composition over other compositions. The Support Vector Machine (SVM) with the Radial Basis Function (RBF) kernel was used as a classification algorithm. Compared to other approaches, the iRSpot-SF performs considerably better than the Matthews, with a correlation coefficient and sensitivity of 69.41% and 84.57% and it has 84.58% accuracy. Jana Novovicova et al. proposed a feature selection method with no search procedure, and one best suited for multimodal data. Isabelle Guyon and Andre Elisseeff, also briefly discussed a feature selection method based on the filter and wrapper approaches and, in addition, defined feature ranking and multivariate feature selection. Jia-You Hsieh et al. in their study, discussed Rice Blast Disease (RBD). The Recursive Feature Elimination (RFE) algorithm with the Auto-Sklearn was used to select key features impacting RBD. The aim of their work was to build a model as a warning mechanism for RBD.

Ron Kohavi and George H. John compared the wrapper and induction methods without feature subset selection, and proceeded thereafter to compare them to Relief, a filter method with feature subset selection. The strengths and weaknesses of the wrapper approach were discussed, and a series of improved designs shown. Isabelle Guyon et al. implemented a Support Vector Machine (SVM) technique based on the Recursive Feature Elimination (RFE) for gene selection. Of the different methods used to select features, the RFE is a newly-developed method that selects features for small sample classification problems.

Marc Sebban and Richard Nock analysed the filter model with information gain and a statistical test. A hybrid model was implemented using a minimum spanning tree that was replaced by the first nearest neighbour. Lei Yu and Huan Liu proposed a correlation filter method termed the fast correlation-based filter. Their technique was verified by two different classification algorithms in terms of real-world data, with and without feature selection.

Petr Somol et al. proposed a flexible hybrid sequential floating search algorithm based on the principles of the filter and wrapper methods. The advantage of the proposed method was its flexibility in terms of a trade-off between the quality of the results versus computational time, as well as enabling the wrapper-based feature selection approach to deal with problems of higher dimensionality. Experiments were carried out using the WAVEFORM dataset from the UCI repository and the SPEECH dataset from British Telecom. Salappa et al. analysed the performance and efficiency of an array of feature selection algorithms with classification methods. Their experimental analysis was carried out on 15 datasets from the UCI repository. The results show that most Feature Selection Algorithms (FSAs) significantly reduce data dimensionality without impacting the performance of the resultant models.

Kursa et al.implemented Boruta, an all-relevant feature selection method which gathers every feature that is critical to the outcome in certain circumstances. By contrast, most traditional feature selection algorithms follow a minimally optimal method in which they rely on a small subset of features that yield a minimal error on a selected classifier. Marcano Cedeno et al.

proposed a feature selection method based on sequential forward selection and the feed forward neural network to find the prediction error as a criterion for selection.

Zahra Karimi et al. implemented a feature ranking method using a hybrid filter feature selection scheme for intrusion detection in a standard dataset. The experimental results show that the proposed technique offers higher accuracy than other methods. Surabhi Chouhan et al. proposed a hybrid combination method of applying the Particle Swarm Optimization – Support Vector Machine (PSO-SVM) to select features from a dataset. Assorted benchmark datasets were tested with this technique.

David Heckmann et al. described that the harnessing of natural variability in photosynthesizing ability as a way to improve yields, through a functional phylogenetic analysis for large-scale genetic screening is a laborious task. The potential for leaf reflectance spectroscopy to estimate photosynthetic efficiency specifications in Brassica oleracea and Zea mays, a C3 and a C4 seed, respectively, were analysed, the findings show that phenotyping leaf reflectance is an effective method to enhance the photosynthetic ability of crops.

Maya Gopal and Bhargavi proposed a wrapper feature selection method featuring Boruta that extracts features from a dataset for crop prediction. The technique improves prediction performance and provides effective predictors. In Boruta, the Z score has the most accurate measure, since it takes into consideration the variability of the mean loss of accuracy among trees in a forest.

Aileen Bahl et al. developed a random forest (RF) model with the RFE for improved prediction accuracy. Maya Gopal and Bhargavi analysed the performance of machine learning (ML) algorithms with a variety of feature selection techniques for crop yield prediction. The results showed that the random forest provides higher accuracy than other ML algorithms. Maya Gopal and Bhargavi proposed sequential forward selection, which is a special sequential feature selection process. It is a greedy search algorithm that attempts to find the ‘optimal’ feature subset by iteratively selecting features based on the performance of the classifiers.

CHAPTER-3

TOOLS & TECHNOLOGY/ALGORITHM

3.1 MACHINE LEARNING-:

Machine learning is one of the most advancing fields in computer science. It has important and valuable applications in artificial intelligence which provides our system the ability to automatically learn from their environment and improve the functionality of the system without explicit programming of the system. The main motive behind every machine learning algorithm is to develop such a tool which is automated and can access and train data.

Over the decades, researchers have been trying to better email communication among which email spam detection and filter is one of the major tasks that needs to be paid attention. Many research papers have been published regarding the issue but there are still research gaps that needs to be covered. Spam detection is one of topics of curiosity that can fill those research gaps. Machine learning algorithms use statistical models to classify data. In the case of spam detection, a trained machine learning model must be able to determine whether the sequence of words found in an email are closer to those found in spam emails or safe ones.

There are three types of machine learning:

3.1.1) Supervised learning
3.1.2) Unsupervised learning
3.1.3) Reinforcement learning

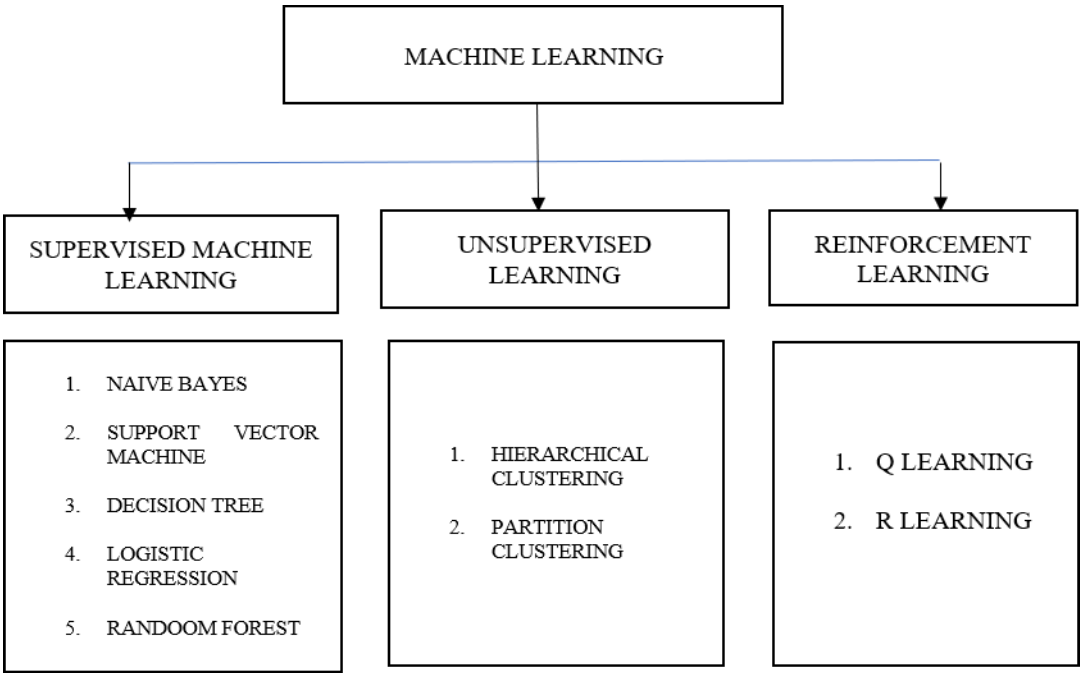


Figure 3.1 Machine learning & types

3.1.1 __Supervised Machine Learning:__

Supervised machine learning is a learning algorithm that needs labelled data. By providing labelled data, our model is trained and predicts future events based on that trained data. In other words, we can say that supervised machine learning models analyse the existing labelled dataset and on basis of that generate a method to make predictions for upcoming future event. This type of learning is used to solve various kinds of problems like face recognition, spam classification, object classification advertisement popularity etc. Supervised machine learning has various types including:

- Naïve Bayes Classifier
- Support Vector Machine
- Random Forest
- Decision Tree
- Logistic Regression

3.1.2 __Unsupervised Machine Learning:__

Unsupervised machine learning are the algorithms in which training dataset is not labelled. Unsupervised machine learning algorithms generate solution for hidden structures inferring a feature from labelled dataset. Unsupervised learning classify data by making clusters of the data based on the features of data available. Clustering is one of the main features of unsupervised machine learning algorithms and it is of two types:

- Hierarchical Clustering
- Partition Clustering

3.1.3. __Reinforcement Learning:__

Reinforcement learning is a type of machine learning algorithm which works on taking rewards from the environment. It is somewhat same as supervised machine learning but it follows the process of taking suitable actions to make or get maximum reward from the environment in a given situation. The main difference between supervised learning algorithm and reinforcement learning is that supervised learning algorithm works on labelled dataset while reinforcement learning is that supervised learning with no correct labelled dataset. Here, the agent decides what step should be taken to do to perform the required task. The agent learns from its experience in its environment when no labelled dataset is given.

3.2 __ALGORITHM:__ -

a) __Random Forest Algorithm__

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning which is a process of combining multiple classifiers As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

The below diagram explains the working of the Random Forest algorithm:

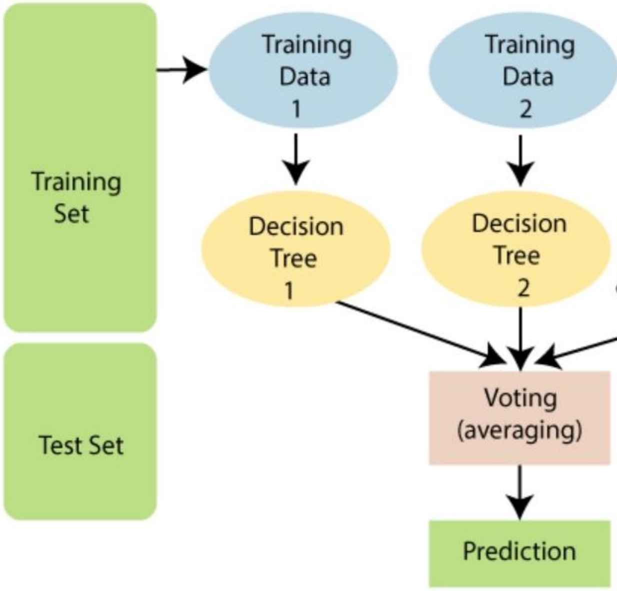


Figure 3.2 Random Forest algorithm

Before understanding the working of the random forest, we must look into the ensemble technique. Ensemble simply means combining multiple models. Thus, a collection of models is used to make predictions rather than an individual model.

***<u>Ensemble uses two types of methods</u>*:**

- ***<u>Bagging-:</u>***

Bagging, also known as ***Bootstrap Aggregation*** is the ensemble technique used by random forest. Bagging chooses a random sample from the data set. Hence each model is generated from the samples (Bootstrap Samples) provided by the Original Data with replacement known as ***row sampling***. This step of row sampling with replacement is called ***bootstrap***. Now each model is trained independently which generates results. The final output is based on majority voting after combining the results of all models. This step which involves combining all the results and generating output based on majority voting is known as ***aggregation***.

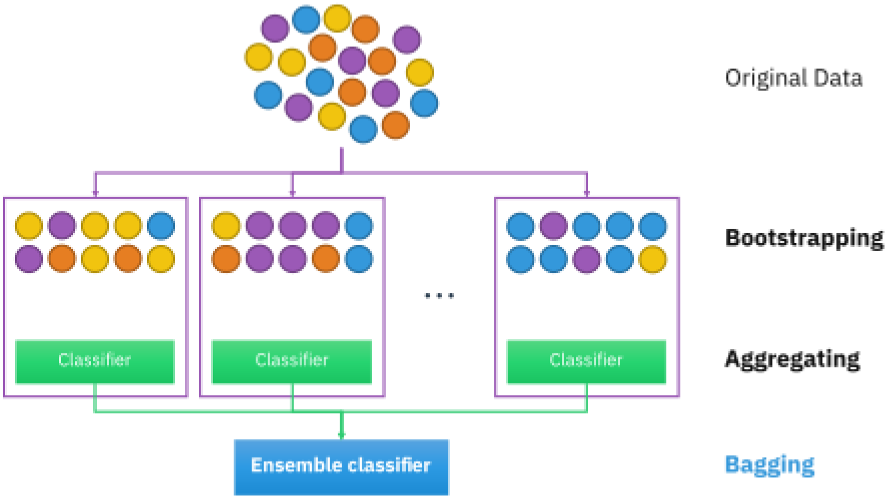


Figure 3.3 Bagging

- ***<u>Boosting–:</u>***

It combines weak learners into strong learners by creating sequential models such that the final model has the highest accuracy. For example, ADA BOOST, XG BOOST.

Steps involved in random forest algorithm:

<u>Step 1</u>: In Random Forest n number of random records are taken from the data set having k number of records.

<u>Step 2</u>: Individual decision trees are constructed for each sample.

<u>Step 3</u>: Each decision tree will generate an output.

<u>Step 4</u>: Final output is considered based on ***Majority Voting or Averaging*** for Classification and regression respectively.

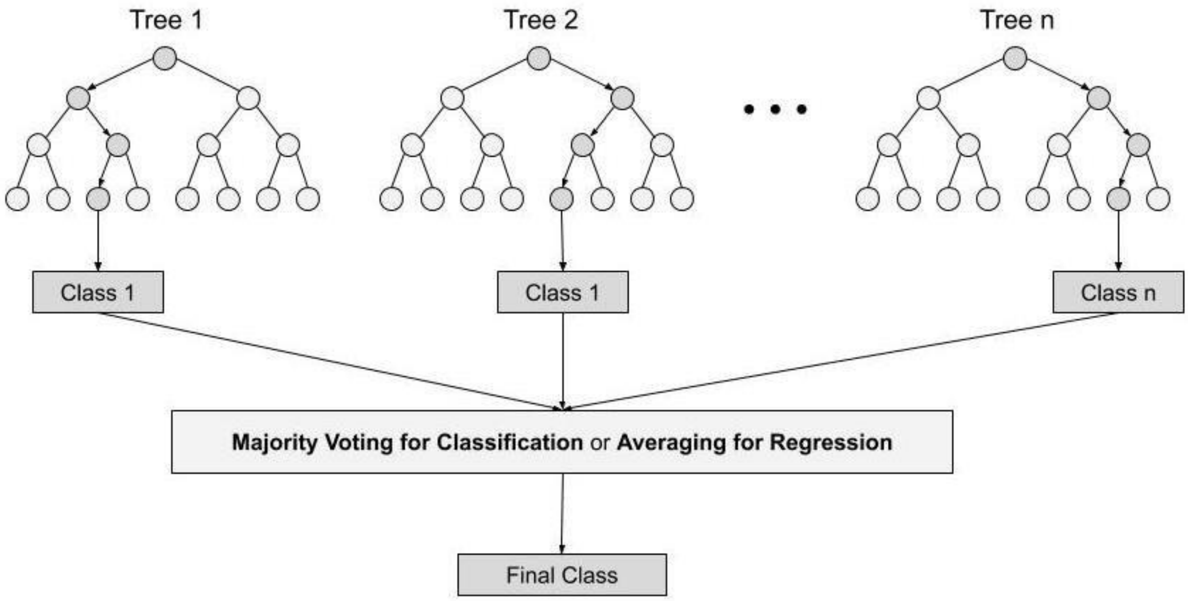

Tree 1
Tree 2
Tree n
Class 1
Class 1
Class n
Majority Voting for Classification or Averaging for Regression
Final Class

Figure 3.4 steps involved in random forest

B) <u>Column Transfer</u>

The developers of the library might have realised that people use LabelEncoding and OneHotEncoding very frequently. So, they decided to come up with a new library called the ColumnTransformer, which will basically combine LabelEncoding and OneHotEncoding into just one line of code. And the result is exactly the same.

Column Transformer is a sciket-learn class used to create and apply separate transformers for numerical and categorical data. To create transformers, we need to specify the transformer object and pass the list of transformations inside a tuple along with the column on which you want to apply the transformation.

One-hot Encoding is a type of vector representation in which *all of the elements* in a vector are 0, except for one, which has 1 as its value, where 1 represents a Boolean specifying a category of the element.

There also exists a similar implementation called *One-Cold Encoding*, where all of the elements in a vector are 1, except for one, which has 0 as its value.

For instance, [0, 0, 0, 1, 0] and [1 ,0, 0, 0, 0] could be some examples of one-hot vectors. A similar technique to this one, also used to represent data, would be dummy variables in statistics.

This is very different from other encoding schemes, which all allow multiple bits to have 1 as its value. Below is a table that compares the representation of numbers from 0 to 7 in binary, Gray code, and one-hot:

Decimal	**Binary**	**Gray code**	**One-Hot**
0	000	000	0000000
1	001	001	0000001
2	010	011	0000010
3	011	010	0000100
4	100	110	0001000
5	101	111	0010000
6	110	101	0100000
7	111	100	1000000

3.3 TOOLS AND TECHNOLOGY USED

a) Flask

Flask is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it.

It provides libraries to build lightweight web applications in python. Flask is considered as a micro framework. Before learning Flask, you must have the basic knowledge of Python concepts.

Flask is based on Werkzeuge WSGI toolkit and Jinja2 template engine. Both are Pocco projects.

It basically allows creating web applications in a Pythonic syntax and concepts. With Flask, we can use Python libraries and tools in our web applications. Using Flask, we can set up a webserver to load up some basic HTML templates along with Jinja2 templating syntax.

b) HTML

The Hypertext Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading style sheets (CSS) and scripting languages such as JavaScript. Web browser receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

c) CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file which reduces complexity and repetition in the structural content as well as enabling the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

Separation of formatting and content also makes it possible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device. The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/CSS is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents. In addition to HTML, other markup languages support the use of CSS including XHTML, plain XML, SVG, and XUL.

d) JavaScript

JavaScript is a text-based programming language used both on the client-side and server-side that allows you to make web pages interactive. Where HTML and CSS are languages that give structure and style to web pages, JavaScript gives web pages interactive elements that engage a user. Common examples of JavaScript that you might use every day include the search box on Amazon, a news recap video embedded on The New York Times, or refreshing your Twitter feed.

Incorporating JavaScript improves the user experience of the web page by converting it from a static page into an interactive one. To recap, JavaScript adds behaviour to web Page.

e) Kaggle

Kaggle is an online community platform for data scientists and machine learning enthusiasts. Kaggle allows users to collaborate with other users, find and publish datasets, use GPU integrated notebooks, and compete with other data scientists to solve data science challenges. The aim of this online platform (founded in 2010 by Anthony Goldbloom and Jeremy Howard and acquired by Google in 2017) is to help professionals and learners reach their goals in their data science journey with the powerful tools and resources it provides. As of today (2021), there are over 8 million registered users on Kaggle.

f) PYTHON

f.1) Pandas

Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data. The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008.

Pandas allows us to analyze big data and make conclusions based on statistical theories. Pandas can clean messy data sets, and make them readable and relevant. Relevant data is very important in data science.

Pandas gives you answers about the data. Like:

- Is there a correlation between two or more columns?
- What is average value?
- Max value?
- Min value?

f.2) Numpy

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more. NumPy stands for Numerical Python.

NumPy is a Python library used for working with arrays.

It also has functions for working in domain of linear algebra, Fourier transform, and matrices.

f.3) SCIKIT-Learn

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon **NumPy, SciPy** and **Matplotlib**.

it provides simple and efficient tools for data mining and data analysis. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means, etc. it is open source, commercially usable – BSD license.

f.4) Pickle

Python pickle module is used for serializing and de-serializing python object structures. The process to converts any kind of python objects (list, dict, etc.) into byte streams (0s and 1s) is called pickling or serialization or flattening or marshalling. We can convert the byte stream (generated through pickling) back into python objects by a process called as unpickling.

In real world scenario, the use pickling and unpickling are widespread as they allow us to easily transfer data from one server/system to another and then store it in a file or database.

Any object in Python can be pickled so that it can be saved on disk. The idea is that this character stream contains all the information necessary to reconstruct the object in another python script.

g) GITHUB

GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.

This tutorial teaches you GitHub essentials like repositories, branches, commits, and pull requests. You'll create your own Hello World repository and learn GitHub's pull request workflow, a popular way to create and review code.

- Create and use a repository
- Start and manage a new branch
- Make changes to a file and push them to GitHub as commits
- Open and merge a pull request

h) GOOGLECOLAB

Collaboratory, or “Collab” for short, is a product from Google Research. Collab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. More technically, Collab is a hosted

Jupyter notebook service that requires no setup to use, while providing access free of charge to computing resources including GPUs.

i) WEATHER API

A weather API is an Application Programming Interface that allows weather data to be queried from scripts and code. Good weather APIs provide both historical weather data and forecast data via an easy-to-use, well-defined programming interface. The best APIs have dozens of weather measures, near-real-time current conditions reporting, and decades of worldwide historical weather reports. Ideally both historical and forecast look-ups would be combined into the same API entry point with the addition of an ultra-long-range forecast based on climate statistics.

A weather API is ideally suited for use cases that need large volumes of weather data or need to access weather data in an automated way.

j) RENDER

Render template is a Flask function from the flask. templating package. render template is used to generate output from a template file based on the Jinja2 engine that is found in the application's templates folder. Note that render template is typically imported directly from the flask package instead of from flask.

A template is rendered with specific data to produce a final document. Flask uses the Jinja template library to render templates. In your application, you will use templates to render HTML which will display in the user's browser. In Flask, Jinja is configured to auto escape any data that is rendered in HTML templates.

3.4 SOFTWARE RQUIREMENT

- PYTHON 3.10 version
- GoogleColab
- VS-Code
- SCI-KET learn, PANDA, NUMPY

3.5 HARDWARE REQUIREMENT

- Processor Pentium IV and motherboard
- RAM 256 mb or above
- HARD DISK 4GB or above
- Printers for reports
- Keyboard, mouse
- Monitor, CPU

CHAPTER-4

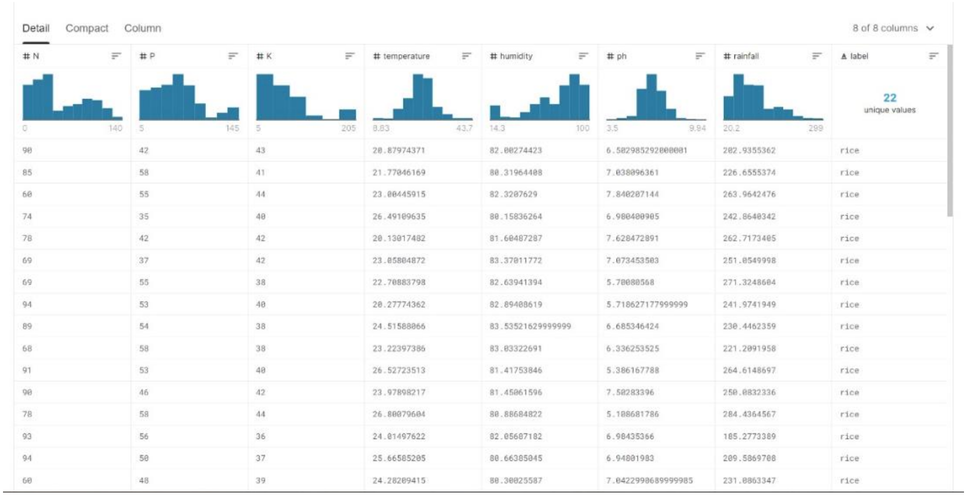
ANALYSES OF DATASET

4.1 DATASETS USED FOR CROP PREDICTION-:

Precision agriculture is in trend nowadays. It helps the farmers to get informed decision about the farming strategy. Here, I present you a dataset which would allow the users to build a predictive model to recommend the most suitable crops to grow in a particular farm based on various parameters.

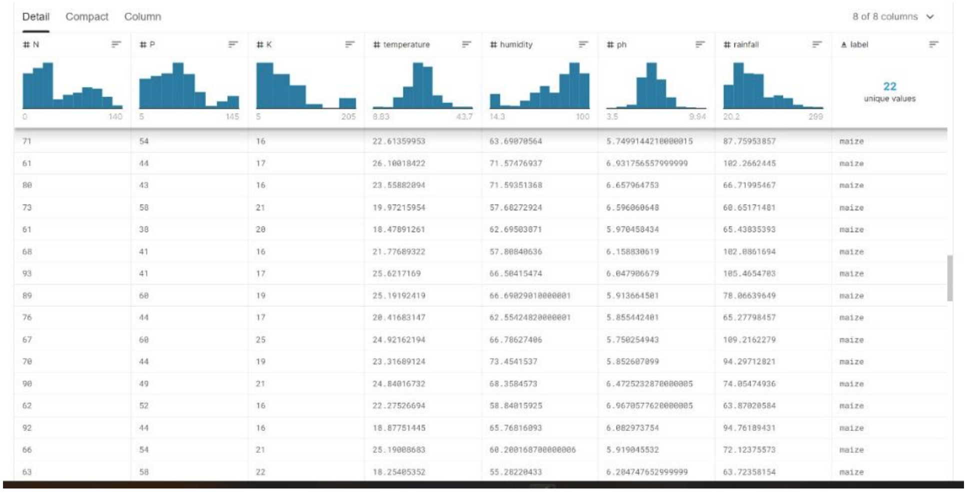
Data fields

- N - ratio of Nitrogen content in soil
- P - ratio of Phosphorous content in soil
- K - ratio of Potassium content in soil
- temperature - temperature in degree Celsius
- humidity - relative humidity in %
- ph. - ph. value of the soil
- rainfall - rainfall in mm



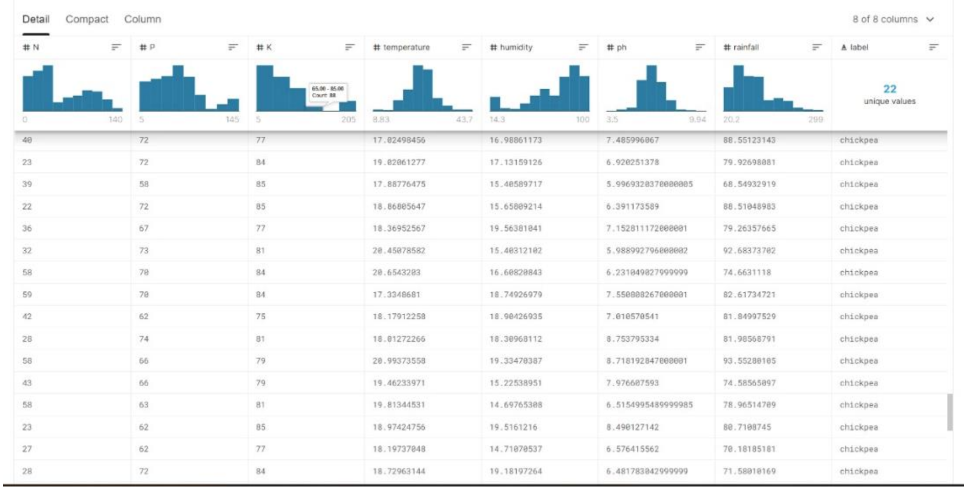
# N	# P	# K	# temperature	# humidity	# ph	# rainfall	label
90	42	43	20.87974371	82.00274423	6.502985292000001	202.9355362	rice
85	58	41	21.77046169	80.31964408	7.038096361	226.6555374	rice
60	55	44	23.00445915	82.3207629	7.840207144	263.9642476	rice
74	35	40	26.49109635	80.15836264	6.980400905	242.8640342	rice
78	42	42	20.13017482	81.60487287	7.628472891	262.7173405	rice
69	37	42	23.05804872	83.37011772	7.073453503	251.0549998	rice
69	55	38	22.70883798	82.63941394	5.70080568	271.3248604	rice
94	53	40	20.27774362	82.89408619	5.718627177999999	241.9741949	rice
89	54	38	24.51588066	83.53521629999999	6.685346424	230.4462359	rice
68	58	38	23.22397386	83.03322691	6.336253525	221.2091958	rice
91	53	40	26.52723513	81.41753846	5.386167788	264.6148697	rice
90	46	42	23.97898217	81.45061596	7.50283396	250.0832336	rice
78	58	44	26.80079604	80.88684822	5.108681786	284.4364567	rice
93	56	36	24.01497622	82.05687182	6.98435366	185.2773389	rice
94	50	37	25.66585205	80.66385045	6.94801983	209.5869708	rice
60	48	39	24.28209415	80.30025587	7.0422990689999985	231.0863347	rice

Figure 4.1 dataset for crop prediction 1



# N	# P	# K	# temperature	# humidity	# ph	# rainfall	label
71	54	16	22.61359953	63.69070564	5.7499144210000015	87.75953857	maize
61	44	17	26.10018422	71.57476937	6.931756557999999	102.2662445	maize
80	43	16	23.55882094	71.59351368	6.657964753	66.71995467	maize
73	58	21	19.97215954	57.68272924	6.596060648	60.65171481	maize
61	38	20	18.47891261	62.69503071	5.970458434	65.43835393	maize
68	41	16	21.77689322	57.80840636	6.158830619	102.0861694	maize
93	41	17	25.6217169	66.50415474	6.047906679	105.4654703	maize
89	60	19	25.19192419	66.69029010000001	5.913664501	78.06639649	maize
76	44	17	20.41683147	62.55424820000001	5.855442401	65.27798457	maize
67	60	25	24.92162194	66.78627406	5.750254943	109.2162279	maize
70	44	19	23.31689124	73.4541537	5.852607099	94.29712821	maize
90	49	21	24.84016732	68.3584573	6.4725232870000005	74.05474936	maize
62	52	16	22.27526694	58.84015925	6.9670577620000005	63.87020584	maize
92	44	16	18.87751445	65.76816093	6.082973754	94.76189431	maize
66	54	21	25.19008683	60.200168700000006	5.919045532	72.12375573	maize
63	58	22	18.25405352	55.28220433	6.204747652999999	63.72358154	maize

Figure 4.2 dataset for crop prediction 2

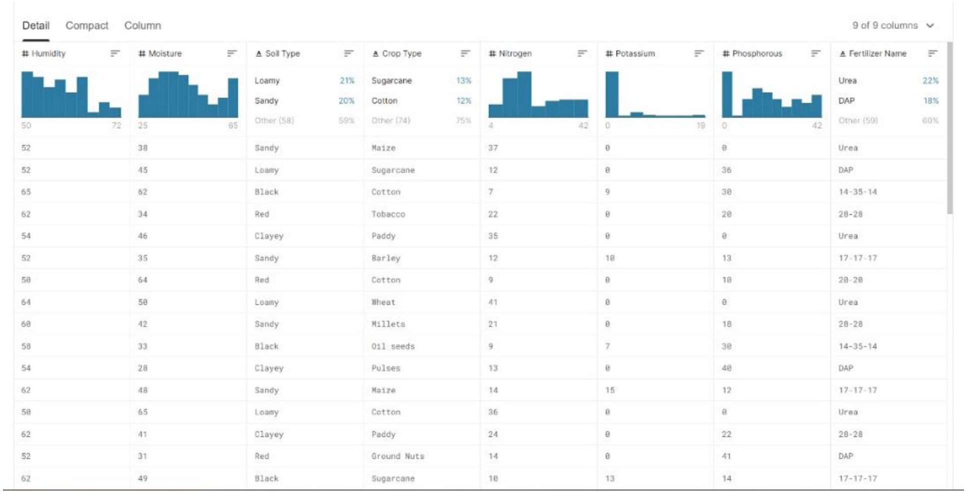


# N	# P	# K	# temperature	# humidity	# ph	# rainfall	label
40	72	77	17.02498456	16.98861173	7.485996067	88.55123143	chickpea
23	72	84	19.02061277	17.13159126	6.920251378	79.92698081	chickpea
39	58	85	17.88776475	15.40589717	5.9969320370000005	68.54932919	chickpea
22	72	85	18.86805647	15.65809214	6.391173589	88.51048983	chickpea
36	67	77	18.36952567	19.56381041	7.152811172000001	79.26357665	chickpea
32	73	81	20.45078582	15.40312102	5.988992796000002	92.68373702	chickpea
58	70	84	20.6543283	16.60820843	6.231049827999999	74.6631118	chickpea
59	70	84	17.3348681	18.74926979	7.550808267000001	82.61734721	chickpea
42	62	75	18.17912258	18.90426935	7.010570541	81.84997529	chickpea
28	74	81	18.01272266	18.30968112	8.753795334	81.98568791	chickpea
58	66	79	20.99373558	19.33470387	8.718192847000001	93.55280105	chickpea
43	66	79	19.46233971	15.22538951	7.976607593	74.58565097	chickpea
58	63	81	19.81344531	14.69765308	6.5154995489999985	78.96514709	chickpea
23	62	85	18.97424756	19.5161216	8.490127142	80.7108745	chickpea
27	62	77	18.19737048	14.71070537	6.576415562	70.18185181	chickpea
28	72	84	18.72963144	19.18197264	6.481783842999999	71.58010169	chickpea

Figure 4.3 dataset for crop prediction 3

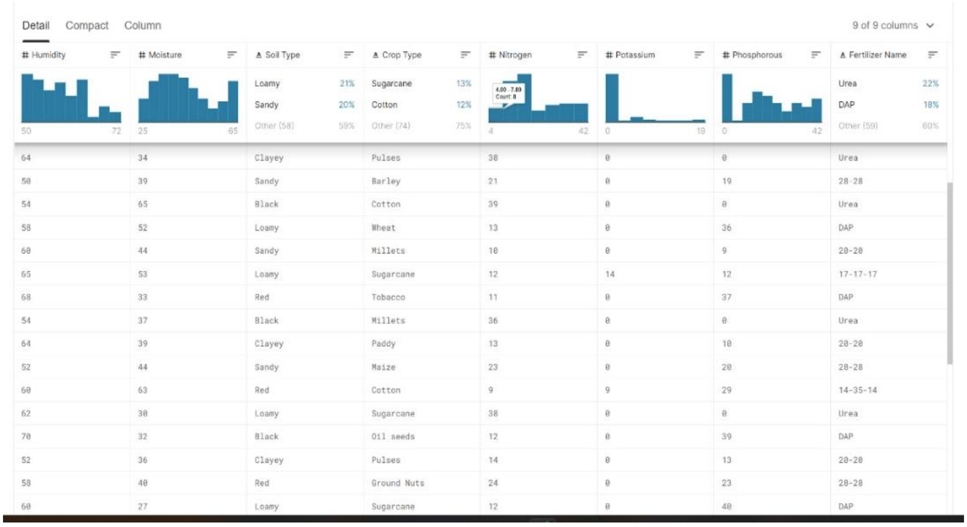
4.2 DATASET USED FOR FERTILISER PREDICTION-:

Here is the data of various fertilizers information. This data was by researching various websites and sources. You can use this data for further prediction and analysis.



# Humidity	# Moisture	Soil Type	Crop Type	# Nitrogen	# Potassium	# Phosphorous	Fertilizer Name
50 – 72	25 – 65	Loamy 21% Sandy 20% Other (58) 59%	Sugarcane 13% Cotton 12% Other (74) 75%	4 – 42	0 – 19	0 – 42	Urea 22% DAP 18% Other (59) 60%
52	38	Sandy	Maize	37	0	0	Urea
52	45	Loamy	Sugarcane	12	0	36	DAP
65	62	Black	Cotton	7	9	30	14-35-14
62	34	Red	Tobacco	22	0	20	28-28
54	46	Clayey	Paddy	35	0	0	Urea
52	35	Sandy	Barley	12	10	13	17-17-17
50	64	Red	Cotton	9	0	10	20-20
64	50	Loamy	Wheat	41	0	0	Urea
60	42	Sandy	Millets	21	0	18	28-28
58	33	Black	Oil seeds	9	7	30	14-35-14
54	28	Clayey	Pulses	13	0	40	DAP
62	48	Sandy	Maize	14	15	12	17-17-17
50	65	Loamy	Cotton	36	0	0	Urea
62	41	Clayey	Paddy	24	0	22	28-28
52	31	Red	Ground Nuts	14	0	41	DAP
62	49	Black	Sugarcane	10	13	14	17-17-17

Figure 4.4 dataset for fertilizer prediction 1



# Humidity	# Moisture	Soil Type	Crop Type	# Nitrogen	# Potassium	# Phosphorous	Fertilizer Name
50 – 72	25 – 65	Loamy 21% Sandy 20% Other (58) 59%	Sugarcane 13% Cotton 12% Other (74) 75%	4 – 42	0 – 19	0 – 42	Urea 22% DAP 18% Other (59) 60%
64	34	Clayey	Pulses	38	0	0	Urea
50	39	Sandy	Barley	21	0	19	28-28
54	65	Black	Cotton	39	0	0	Urea
58	52	Loamy	Wheat	13	0	36	DAP
60	44	Sandy	Millets	10	0	9	20-20
65	53	Loamy	Sugarcane	12	14	12	17-17-17
68	33	Red	Tobacco	11	0	37	DAP
54	37	Black	Millets	36	0	0	Urea
64	39	Clayey	Paddy	13	0	10	20-20
52	44	Sandy	Maize	23	0	20	28-28
60	63	Red	Cotton	9	9	29	14-35-14
62	30	Loamy	Sugarcane	38	0	0	Urea
70	32	Black	Oil seeds	12	0	39	DAP
52	36	Clayey	Pulses	14	0	13	20-20
50	40	Red	Ground Nuts	24	0	23	28-28
60	27	Loamy	Sugarcane	12	0	40	DAP

Figure 4.5 Dataset For Fertiliser Prediction 2

4.3 MEASURING ACCURACY -

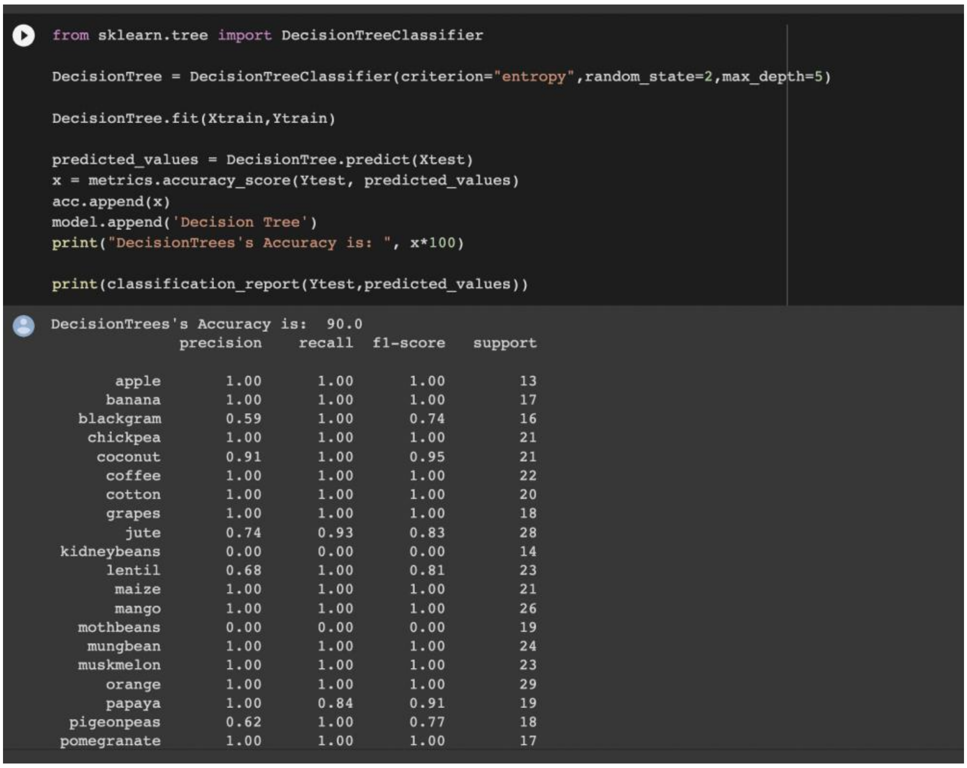
To get the maximum accuracy of this model to analyse the datasets, we performed different listed model and check the accuracy, after that which model generate the maximum accuracy we implement in this model -

1. Using decision tree
2. Using gaussian naïve model
3. Using support vector machine
4. Using logistic regression

5. Using random forest algorithm

i. Accuracy measured by decision tree

Decision tree measured the accuracy of the model using the functions- Entropy, Random_state and maximum depth. The accuracy provided by the decision tree model is 0.9.



```
from sklearn.tree import DecisionTreeClassifier

DecisionTree = DecisionTreeClassifier(criterion="entropy",random_state=2,max_depth=5)

DecisionTree.fit(Xtrain,Ytrain)

predicted_values = DecisionTree.predict(Xtest)
x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('Decision Tree')
print("DecisionTrees's Accuracy is: ", x*100)

print(classification_report(Ytest,predicted_values))
```

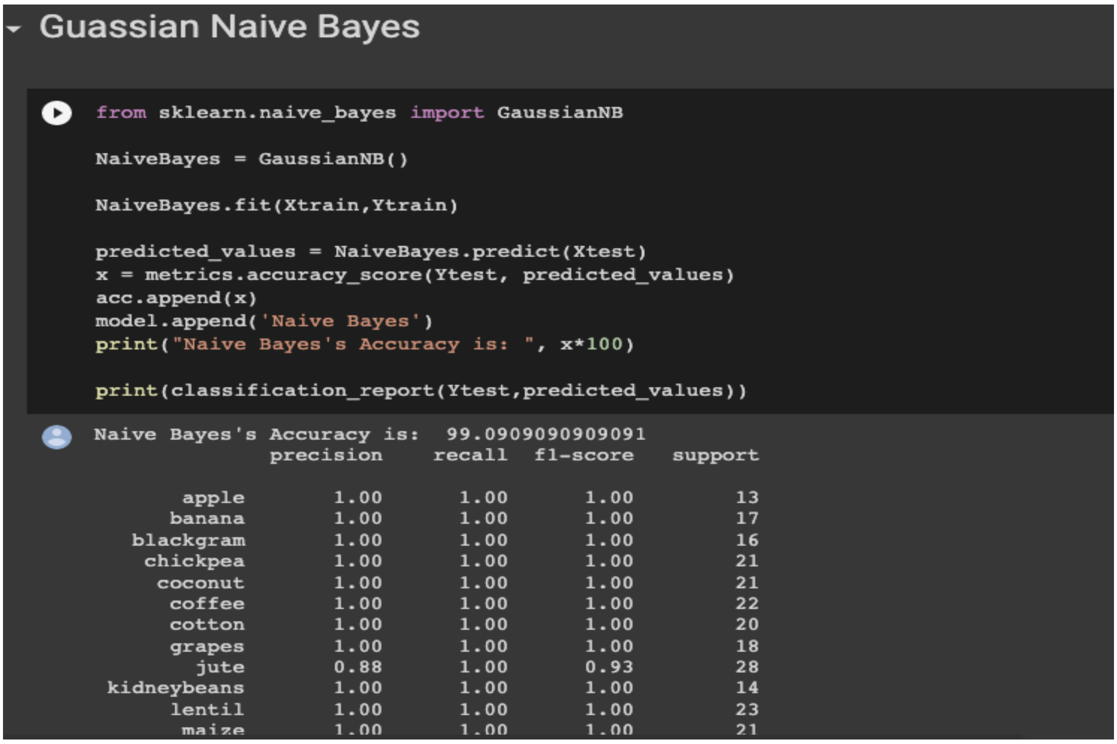
```
DecisionTrees's Accuracy is:  90.0
              precision    recall  f1-score   support

       apple       1.00      1.00      1.00        13
      banana       1.00      1.00      1.00        17
   blackgram       0.59      1.00      0.74        16
    chickpea       1.00      1.00      1.00        21
     coconut       0.91      1.00      0.95        21
      coffee       1.00      1.00      1.00        22
      cotton       1.00      1.00      1.00        20
      grapes       1.00      1.00      1.00        18
        jute       0.74      0.93      0.83        28
 kidneybeans       0.00      0.00      0.00        14
      lentil       0.68      1.00      0.81        23
       maize       1.00      1.00      1.00        21
       mango       1.00      1.00      1.00        26
   mothbeans       0.00      0.00      0.00        19
    mungbean       1.00      1.00      1.00        24
   muskmelon       1.00      1.00      1.00        23
      orange       1.00      1.00      1.00        29
      papaya       1.00      0.84      0.91        19
  pigeonpeas       0.62      1.00      0.77        18
 pomegranate       1.00      1.00      1.00        17
```

Figure 4.6 Decision Tree Accuracy

ii. Accuracy measured by gaussian naïve model

This algorithm used two factors to measure the accuracy of the model- xtrain(input of training data) and ytrain(output of training data). After performing output of testing dataset, it measured the accuracy of the model which is 0.99090090091.



Guassian Naive Bayes

```
from sklearn.naive_bayes import GaussianNB

NaiveBayes = GaussianNB()

NaiveBayes.fit(Xtrain,Ytrain)

predicted_values = NaiveBayes.predict(Xtest)
x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('Naive Bayes')
print("Naive Bayes's Accuracy is: ", x*100)

print(classification_report(Ytest,predicted_values))
```

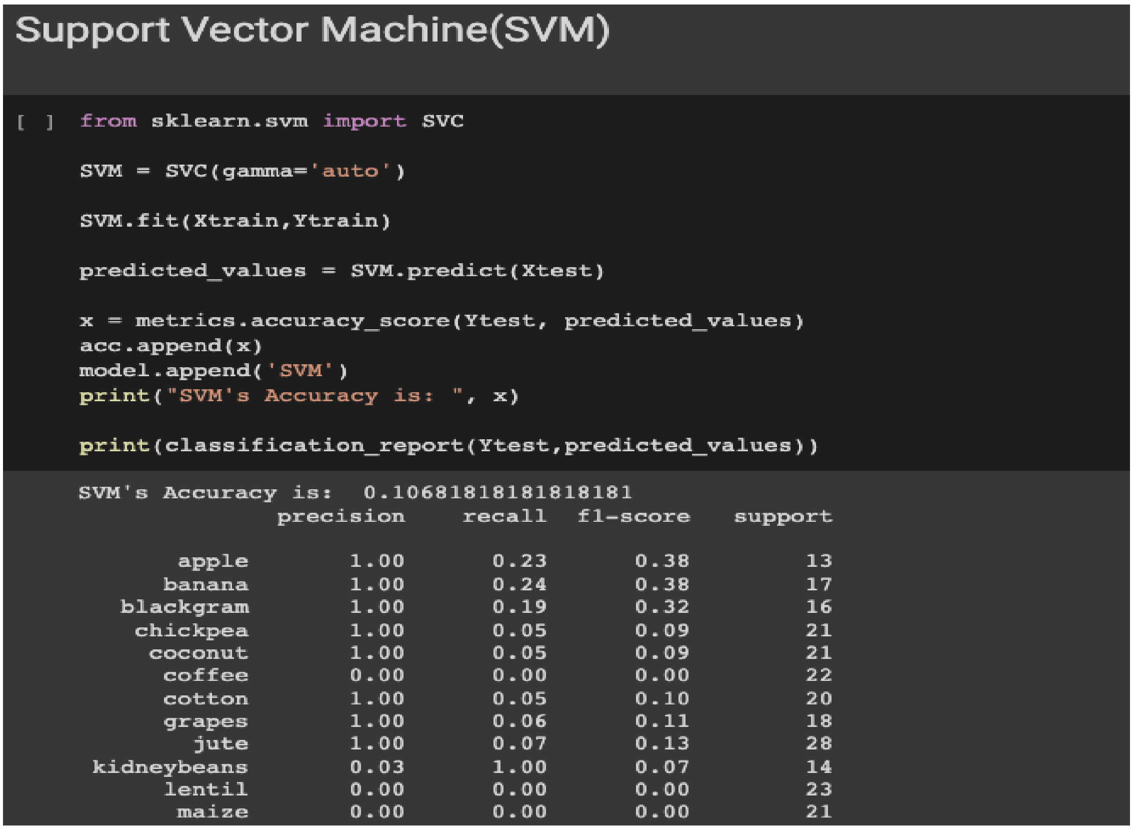
```
Naive Bayes's Accuracy is:  99.0909090909091
              precision    recall  f1-score   support

       apple       1.00      1.00      1.00        13
      banana       1.00      1.00      1.00        17
   blackgram       1.00      1.00      1.00        16
    chickpea       1.00      1.00      1.00        21
     coconut       1.00      1.00      1.00        21
      coffee       1.00      1.00      1.00        22
      cotton       1.00      1.00      1.00        20
      grapes       1.00      1.00      1.00        18
        jute       0.88      1.00      0.93        28
 kidneybeans       1.00      1.00      1.00        14
      lentil       1.00      1.00      1.00        23
       maize       1.00      1.00      1.00        21
```

Figure 4.7 Gaussian Naïve Accuracy

iii. Accuracy measured by Support Vector Machine

As gaussian naïve model this algorithm also uses two factors to measure the accuracy of the model- xtrain(input of training data) and ytrain(output of training data). It performs tests on input of testing data and output of testing data to measure the accuracy which is 0.1068181818.



Support Vector Machine(SVM)

```
[ ]  from sklearn.svm import SVC

     SVM = SVC(gamma='auto')

     SVM.fit(Xtrain,Ytrain)

     predicted_values = SVM.predict(Xtest)

     x = metrics.accuracy_score(Ytest, predicted_values)
     acc.append(x)
     model.append('SVM')
     print("SVM's Accuracy is: ", x)

     print(classification_report(Ytest,predicted_values))
```

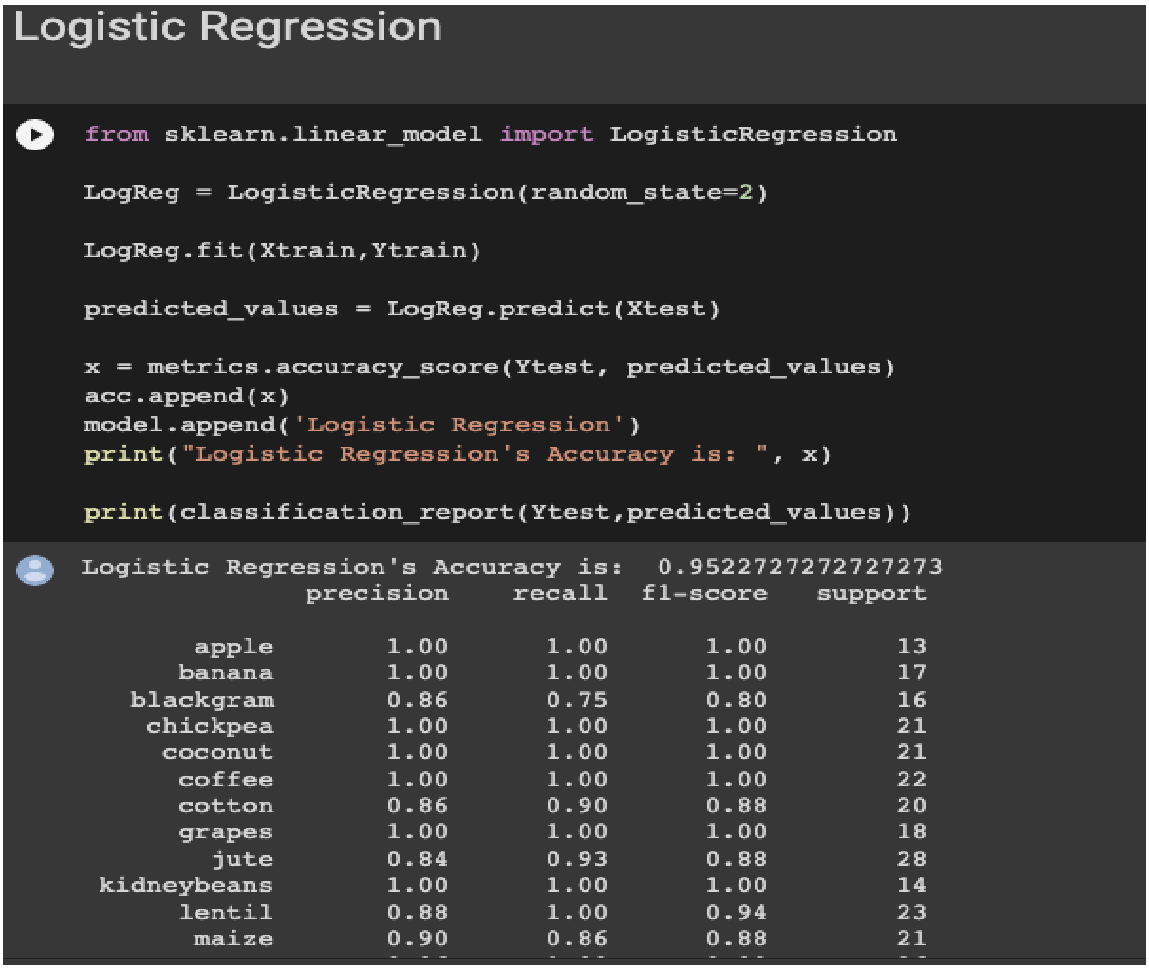
```
SVM's Accuracy is:  0.10681818181818181
              precision    recall  f1-score   support

       apple       1.00      0.23      0.38        13
      banana       1.00      0.24      0.38        17
   blackgram       1.00      0.19      0.32        16
    chickpea       1.00      0.05      0.09        21
     coconut       1.00      0.05      0.09        21
      coffee       0.00      0.00      0.00        22
      cotton       1.00      0.05      0.10        20
      grapes       1.00      0.06      0.11        18
        jute       1.00      0.07      0.13        28
 kidneybeans       0.03      1.00      0.07        14
      lentil       0.00      0.00      0.00        23
       maize       0.00      0.00      0.00        21
```

Figure 4.8 Support Vector Machine Accuracy

iv. Accuracy measured by logistic regression

Logistic regression uses a function called random_state which performs a ytest on xtrain and ytrain. It provides an accuracy of 0.952272727272.



Logistic Regression

```
from sklearn.linear_model import LogisticRegression

LogReg = LogisticRegression(random_state=2)

LogReg.fit(Xtrain,Ytrain)

predicted_values = LogReg.predict(Xtest)

x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('Logistic Regression')
print("Logistic Regression's Accuracy is: ", x)

print(classification_report(Ytest,predicted_values))
```

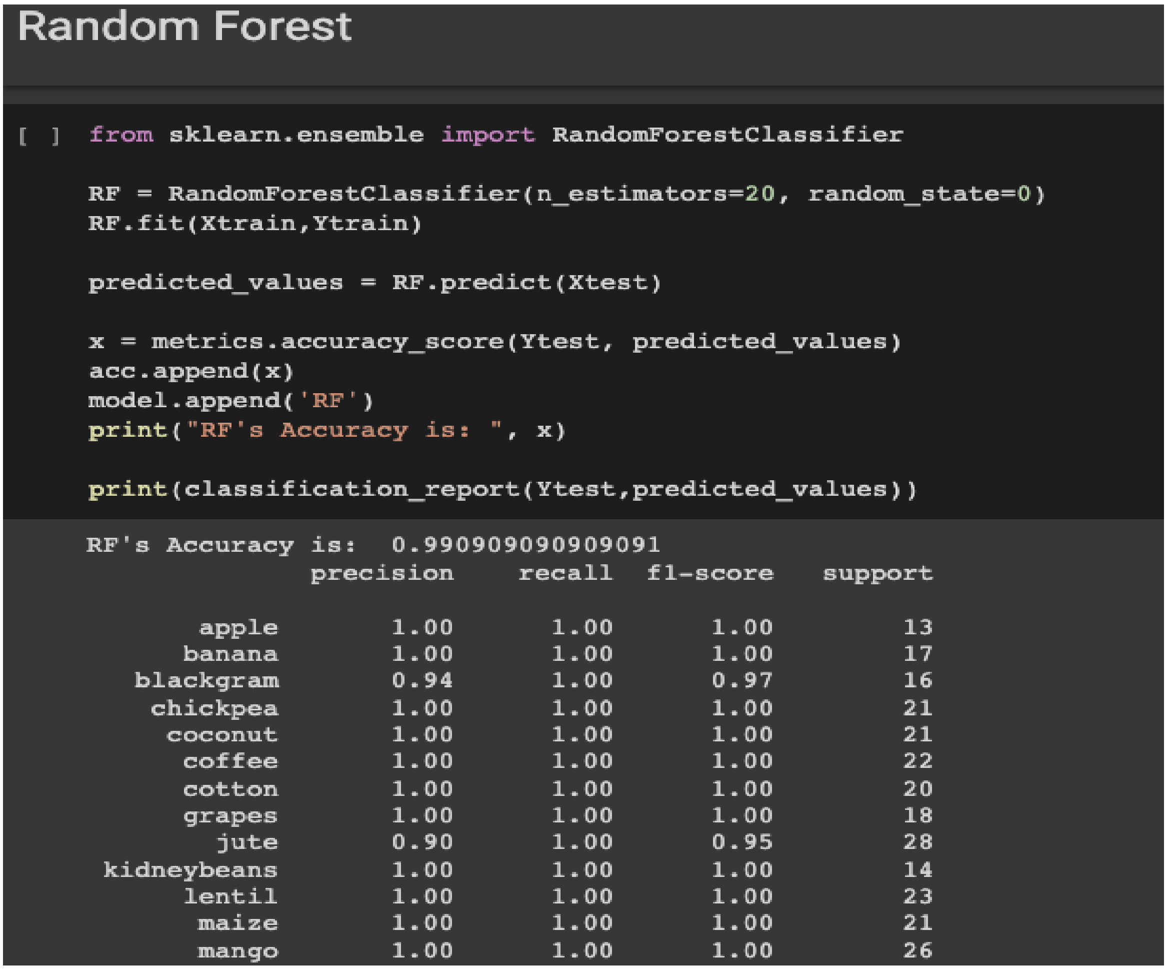
```
Logistic Regression's Accuracy is:  0.9522727272727273
              precision    recall  f1-score   support

       apple       1.00      1.00      1.00        13
      banana       1.00      1.00      1.00        17
   blackgram       0.86      0.75      0.80        16
    chickpea       1.00      1.00      1.00        21
     coconut       1.00      1.00      1.00        21
      coffee       1.00      1.00      1.00        22
      cotton       0.86      0.90      0.88        20
      grapes       1.00      1.00      1.00        18
        jute       0.84      0.93      0.88        28
 kidneybeans       1.00      1.00      1.00        14
      lentil       0.88      1.00      0.94        23
       maize       0.90      0.86      0.88        21
```

Figure 4.9 Logistic Regression Accuracy

v. Accuracy Measured By Random Forest Model

Random forest uses two factors- n_estimator and random_state to perform ytest on xtrain and ytrain. It provides an accuracy of 0.990909090909 01.



Random Forest

```
[ ]  from sklearn.ensemble import RandomForestClassifier

     RF = RandomForestClassifier(n_estimators=20, random_state=0)
     RF.fit(Xtrain,Ytrain)

     predicted_values = RF.predict(Xtest)

     x = metrics.accuracy_score(Ytest, predicted_values)
     acc.append(x)
     model.append('RF')
     print("RF's Accuracy is: ", x)

     print(classification_report(Ytest,predicted_values))
```

```
RF's Accuracy is:  0.990909090909091
              precision    recall  f1-score   support

       apple       1.00      1.00      1.00        13
      banana       1.00      1.00      1.00        17
   blackgram       0.94      1.00      0.97        16
    chickpea       1.00      1.00      1.00        21
     coconut       1.00      1.00      1.00        21
      coffee       1.00      1.00      1.00        22
      cotton       1.00      1.00      1.00        20
      grapes       1.00      1.00      1.00        18
        jute       0.90      1.00      0.95        28
 kidneybeans       1.00      1.00      1.00        14
      lentil       1.00      1.00      1.00        23
       maize       1.00      1.00      1.00        21
       mango       1.00      1.00      1.00        26
```

Figure 4.10 Random Forest accuracy

CHAPTER-5

IMPLEMENTATION/RESULT

5.1 FLOW CHART

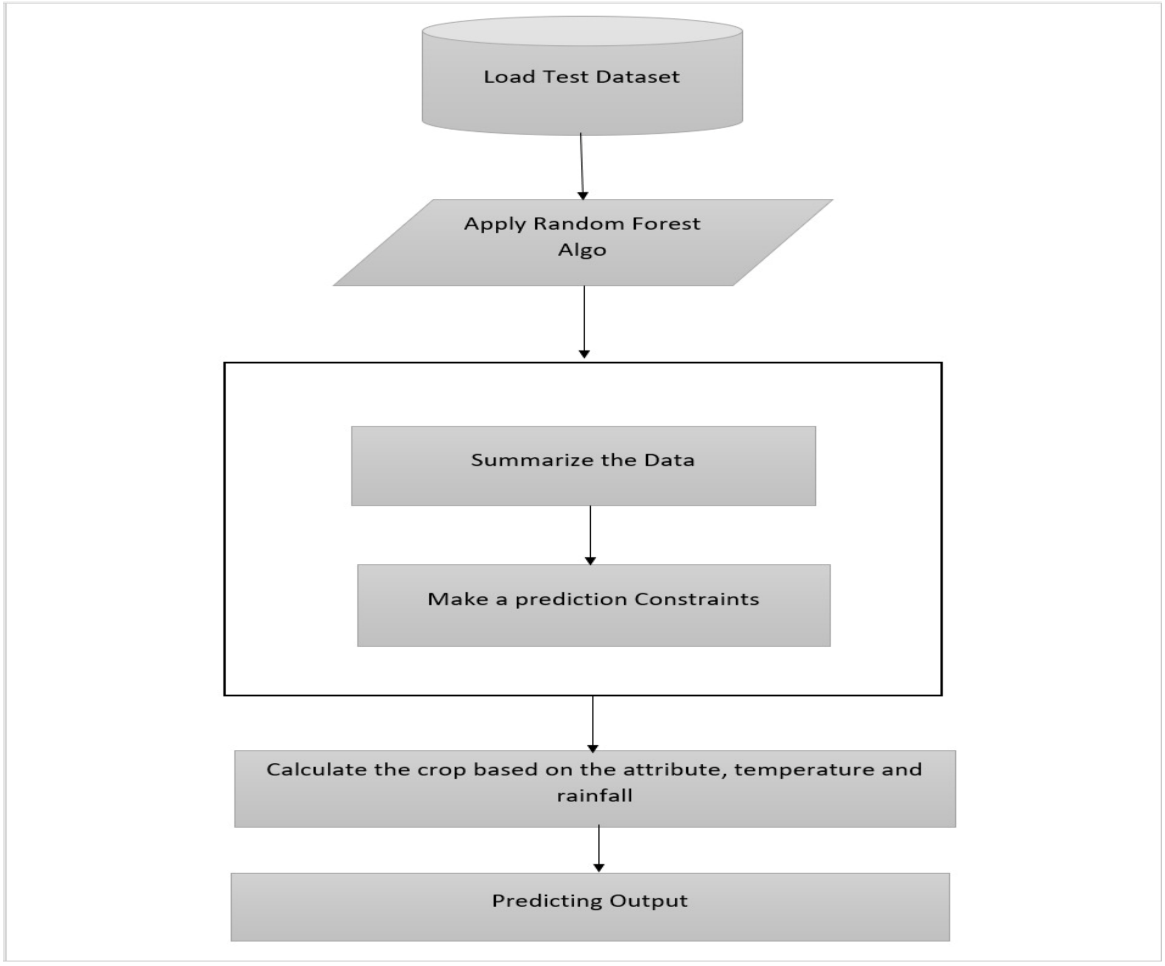

Load Test Dataset
Apply Random Forest Algo
Summarize the Data
Make a prediction Constraints
Calculate the crop based on the attribute, temperature and rainfall
Predicting Output

Figure 5.1 flow chart

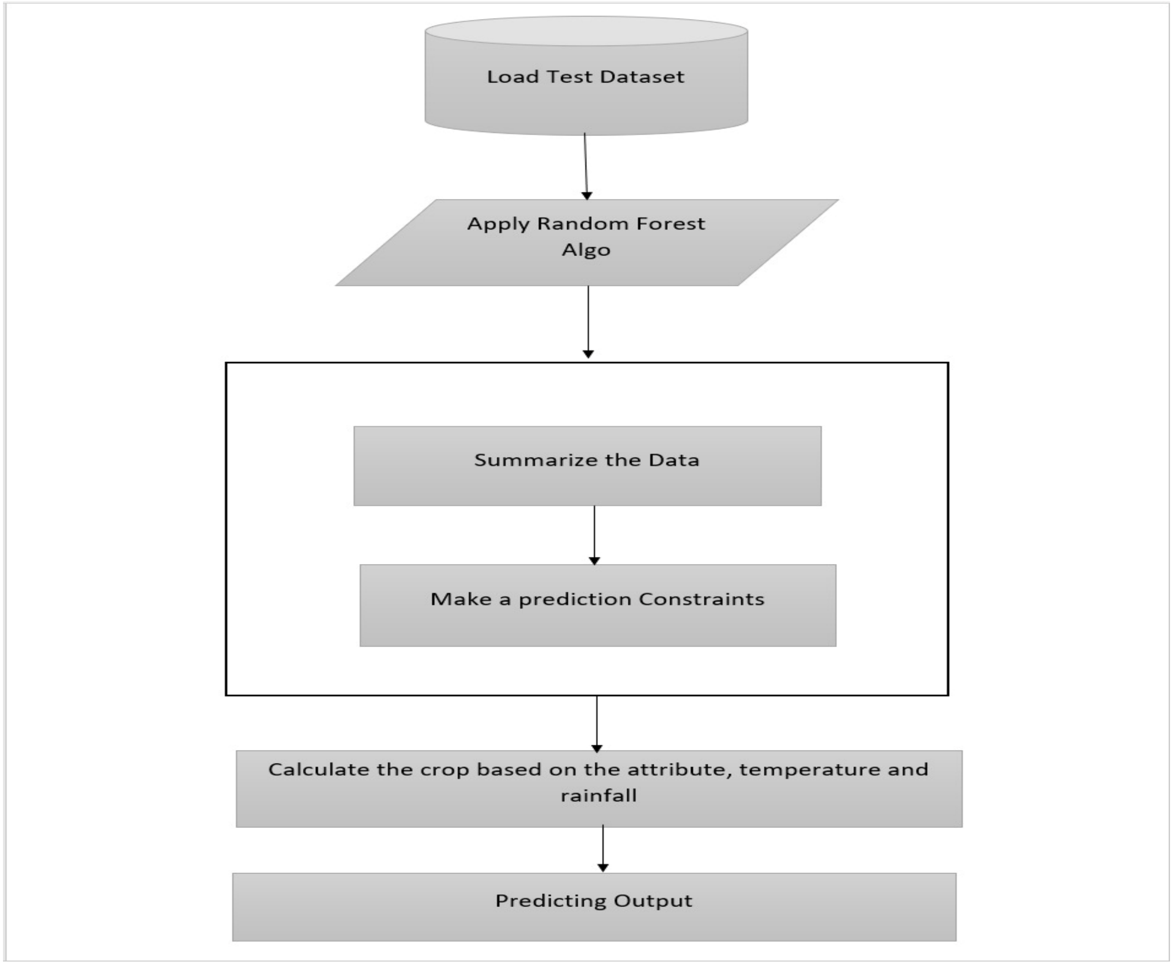
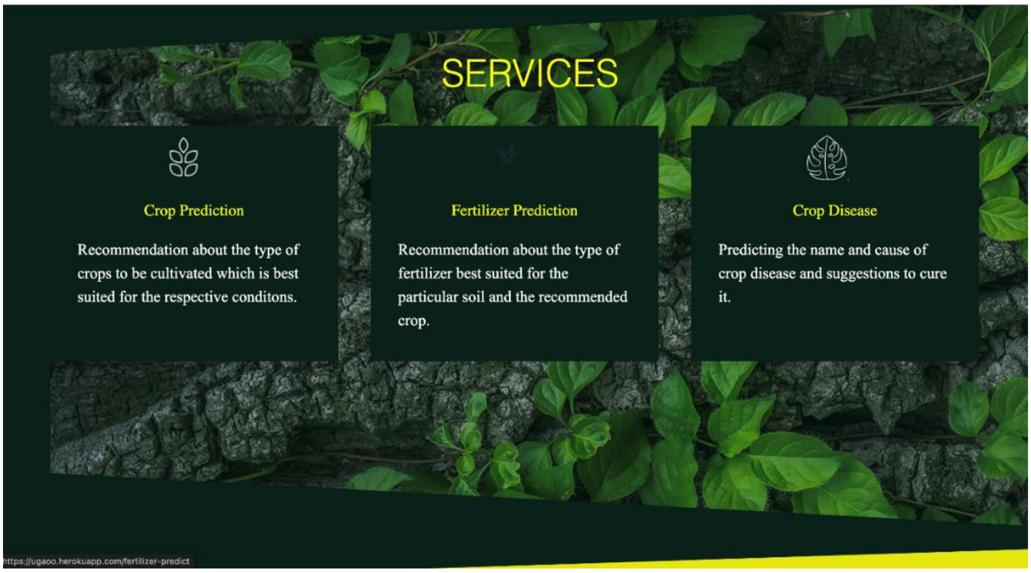

Load Test Dataset
Apply Random Forest Algo
Summarize the Data
Make a prediction Constraints
Calculate the crop based on the attribute, temperature and rainfall
Predicting Output

Figure 5.2 flow chart 2

5.2 <u>SCREENSHOTS:</u>

5.2.1 <u>Services/Home screen-:</u>



SERVICES

Crop Prediction

Recommendation about the type of crops to be cultivated which is best suited for the respective conditons.

Fertilizer Prediction

Recommendation about the type of fertilizer best suited for the particular soil and the recommended crop.

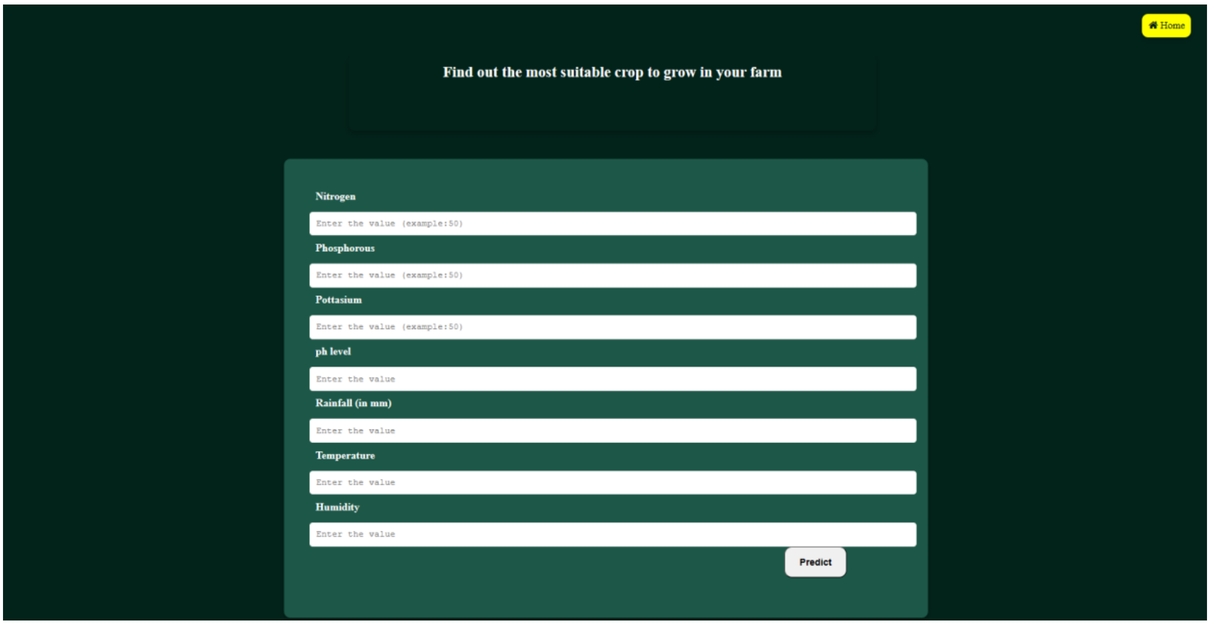
Crop Disease

Predicting the name and cause of crop disease and suggestions to cure it.

ttps://ugaoo.herokuapp.com/fertilizer-predict

Figure 5.3 Home screen

5.2.2 *Crop Prediction-:*



Find out the most suitable crop to grow in your farm

Nitrogen

Enter the value (example:50)

Phosphorous

Enter the value (example:50)

Pottasium

Enter the value (example:50)

ph level

Enter the value

Rainfall (in mm)

Enter the value

Temperature

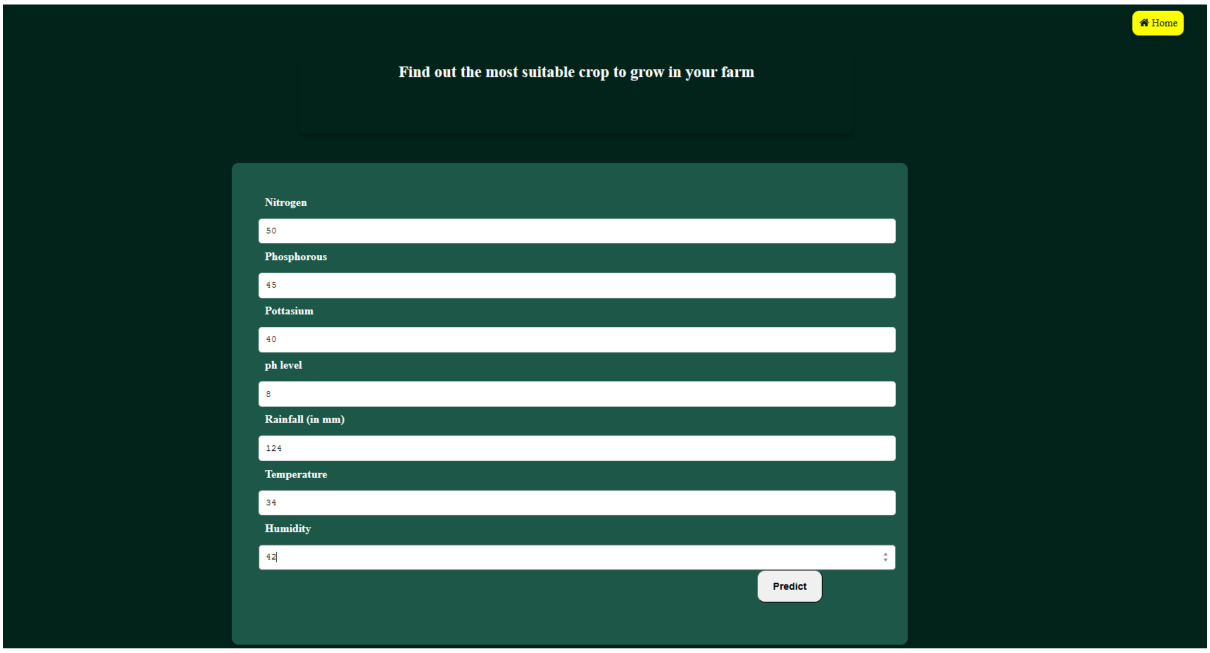
Enter the value

Humidity

Enter the value

Figure 5.4 crop prediction

5.2.3 *Input For Crop Prediction-:*



Find out the most suitable crop to grow in your farm

Nitrogen

50

Phosphorous

45

Pottasium

40

ph level

8

Rainfall (in mm)

124

Temperature

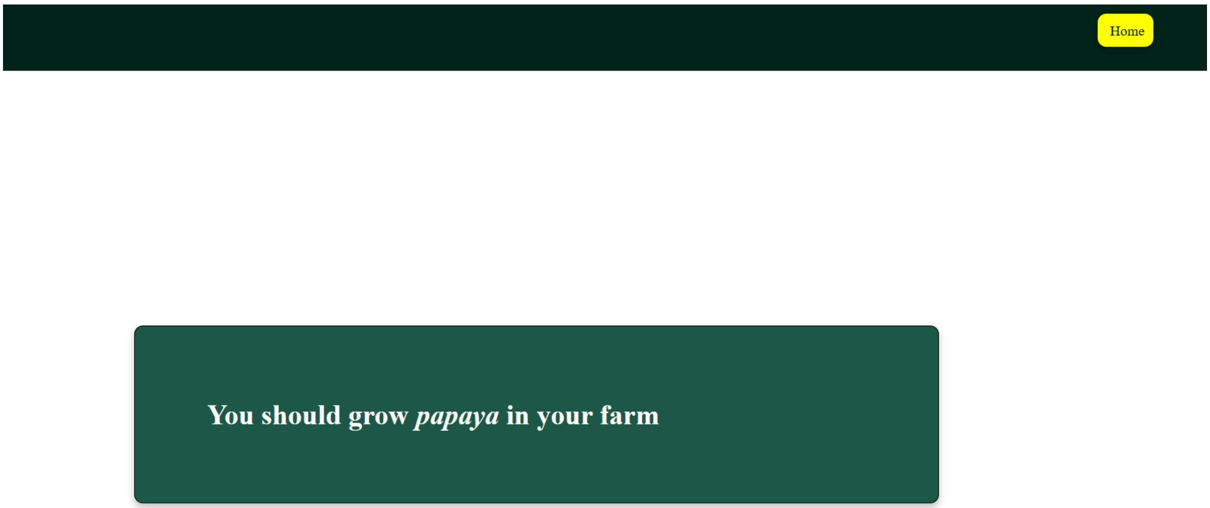
34

Humidity

42

Figure 5.5 Input for crop prediction

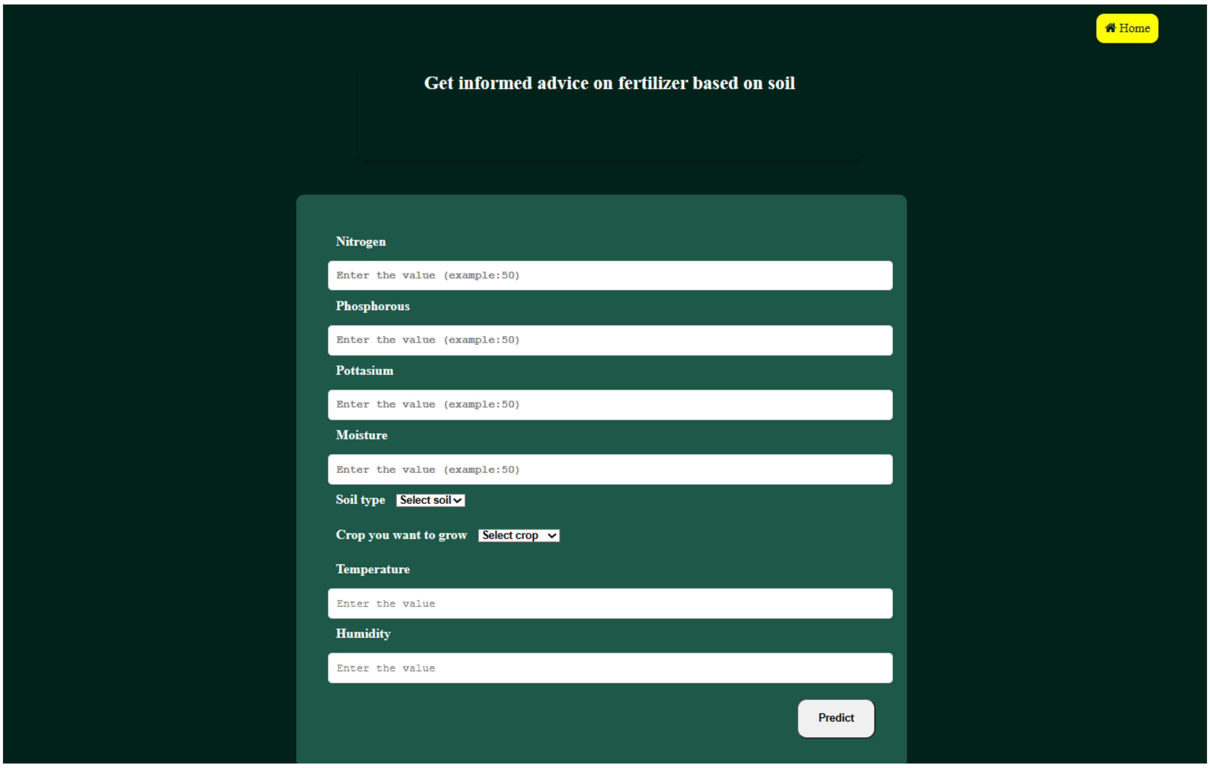
5.2.4 Crop Prdiction Output-:



You should grow *papaya* in your farm

Figure 5.6 Crop prediction output

5.2.5 Fertiliser Prediction-:



Get informed advice on fertilizer based on soil

Nitrogen

Enter the value (example:50)

Phosphorous

Enter the value (example:50)

Pottasium

Enter the value (example:50)

Moisture

Enter the value (example:50)

Soil type Select soil

Crop you want to grow Select crop

Temperature

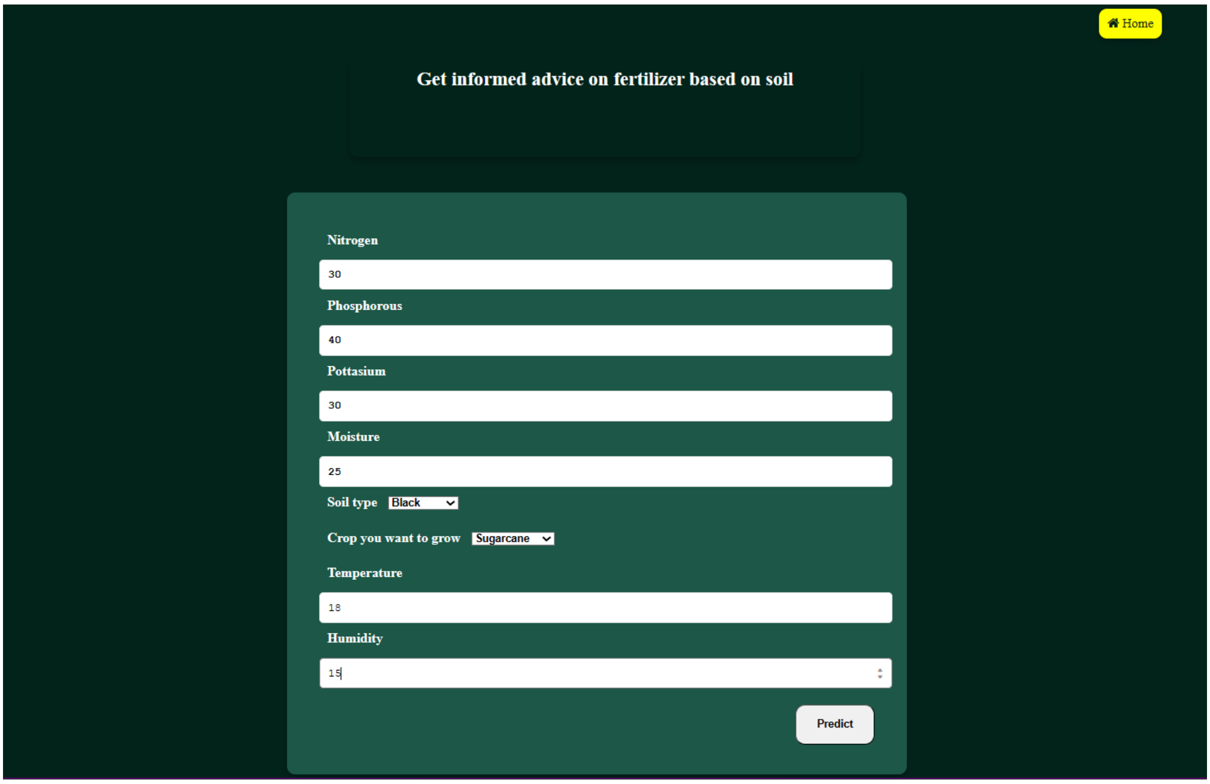
Enter the value

Humidity

Enter the value

Figure 5.7 fertiliser prediction

5.2.6 *Input For Fertiliser Prediction-:*



Get informed advice on fertilizer based on soil

Nitrogen: 30

Phosphorous: 40

Pottasium: 30

Moisture: 25

Soil type: Black

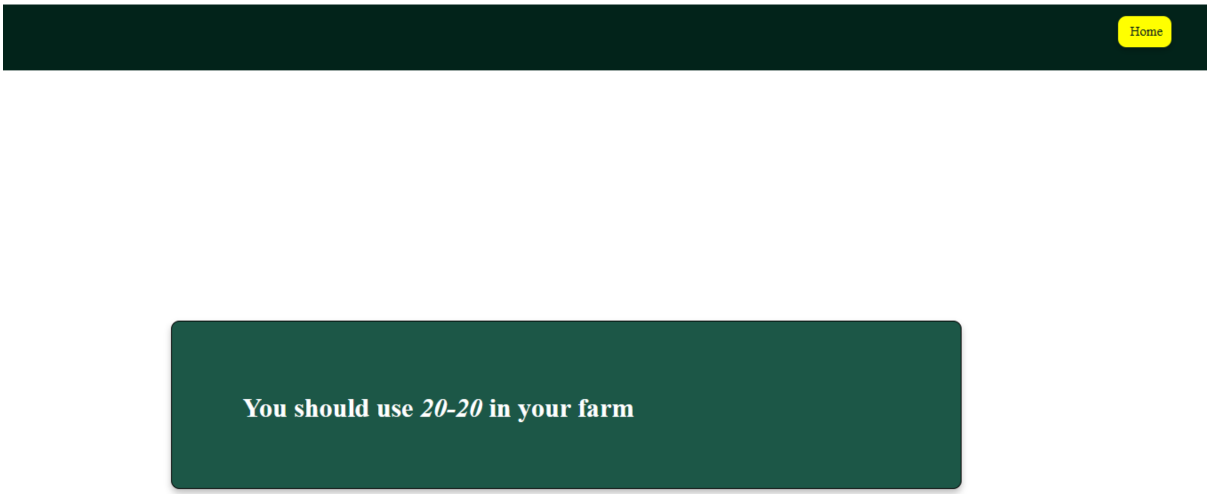
Crop you want to grow: Sugarcane

Temperature: 18

Humidity: 15

Figure 5.8 Input for fertiliser prediction

5.2.7 *Fertiliser Prediction Output-:*



You should use *20-20* in your farm

Figure 5.9 Fertiliser prediction output

CHAPTER 6

CONCLUSION

Based on the climatic input parameters the present study provided the demonstration of the potential use of data mining techniques in predicting the crop yield based. The developed webpage is user friendly and the accuracy of predictions are above 75 per cent in all the crops and districts selected in the study indicating higher accuracy of prediction. By providing climatic data of that place the user-friendly web page developed for predicting crop yield can be used by any user their choice of crop.

Weather aberrations can cause physical damage to crops. With help of this project, we can predict in certain environmental conduction which crop should be taken. From the graph of % of production we can determine sowing and harvesting period of particular crop in given temperature and rainfall. This data will continue to enhance farmer efficiency by further enabling them to monitor each plot of land and determine the precise input needed for their crops.

6.1 FUTURE SCOPE:

As our project progresses, we are committed to continually adapting to new challenges and opportunities, with the ultimate goal of achieving long-term success and sustainability. We have mainly two future scopes of our project:

6.1.1 Language Translator-:

Language barrier is the most common problem in our country. Indian farmers are not very well educated so they face a common problem of language as most of the platforms use English as a general language. In the future we are going to provide a language translator in our model which will provide both options for English as well as Hindi language so that language barrier is no longer a problem for our farmers and they can use our services by themselves without any outer help.

6.1.2 Disease Prediction-:

Crop diseases are a major threat to food security, but their rapid identification remains difficult in many parts of the world due to the lack of the necessary infrastructure. The combination of increasing global smartphone penetration and recent advances in computer vision made possible by deep learning has paved the way for smartphone-assisted disease diagnosis. Using a public dataset of sample images of diseased and healthy plant leaves collected under controlled conditions, we will train a deep convolutional neural network to identify several crop species and every possible disease. The trained model will achieve a well predicted outcome on a held-out test set, demonstrating the feasibility of this approach. Overall, the approach of training deep learning models on increasingly large and publicly available image datasets presents a clear path toward smartphone-assisted crop disease diagnosis on a massive global scale.

REFERENCES

1. S. Veenadhari, Bharat Misra and CD Singh, "Machine learning approach for forecasting crop yield based on climatic parameters", International Conference on Computer Communication and Informatics (ICCCI), 2019, 2014.
2. S. Das. Filters, wrappers and a boosting-based hybrid for feature selection. In International conference on machine learning 1 (2001), pp. 74–81.
3. .F. Balducci, D. Impedovo, and G. Pirlo, Machine learning applications on agricultural datasets for smart farm enhancement, Machines 6 (2018), pp. 38–59. doi:https://doi.org/10.3390/machines6030038. [Crossref], [Web of Science ®], [Google Scholar].
4. Girish L, Gangadhar S, Bharath T R, Balaji K S, Abhishek K T "Crop Yield and Rainfall Prediction in Tumakuru District using Machine Learning.
5. P. Priya, U. Muthaiah and M. Balamurugan, "Predicting yield of the crop using machine learning algorithm", International Journal of Engineering Science Research Technology (IJESRT), 2018.
6. GoogleColab- used to check the accuracy of the model. We checked 4 algorithms and we implemented the most accurate one.
7. JavaTpoint – we studied the working strategies and implementing process of Random Forest algorithm on javatpoint.
8. CROPPROPHET -www.cropprophet.com
9. HARVESTIFY - harvestify.herokuapp.com
10. Frontliners article (for plant disease prediction)- www.frontiersin.org/articles/10.3389/fpls.2016.01419/full
11. W. Edwards, P. Duffy, in Encyclopedia of Agriculture and Food Systems, 2014 - www.sciencedirect.com/topics/earth-and-planetary-sciences/agricultural-production
12. Kaggle (https://www.kaggle.com) - It is used for crop recommendation and fertilizers recommendations.
    - For crop recommendation CV file - '/content/Crop_recommendation.csv'
    - For fertilizer recommendation CV file - "/content/Fertilizer Prediction.csv"