

ML BASED OPTIMIZATION AGRICULTURE PRODUCTION

Project report submitted in partial fulfillment of the
requirement for the degree of Bachelor of
Technology

In

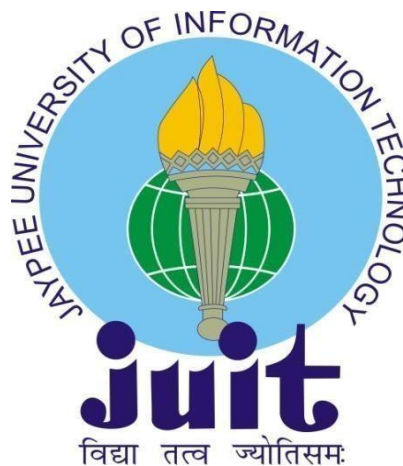
**Computer Science and Engineering/Information
Technology**

By

Bhavya Gupta (191543)

Under the supervision of

Dr. Pankaj Dhiman
(Assistant Professor, CSE)



Department of Computer Science & Engineering
Information Technology

Candidate's Declaration

I hereby declare that the work presented in this report entitled “ **ML based optimization agriculture production**” in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from July 2022 to May 2023 under the supervision of **Dr. Pankaj Dhiman, Assistant Professor (SG)**. I also authenticate that I have carried out the above-mentioned project work under the proficiency stream Cloud Computing. The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Bhavya Gupta, 191543

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Pankaj Dhiman
Assistant Professor
(SG)
Computer Science and Engineering
Dated: 26/04/2023

Plagiarism Certificate

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date:

Type of Document (Tick): PhD Thesis M.Tech Dissertation/ Report B.Tech Project Report Paper

Name: _____ Department: _____ Enrolment No _____

Contact No. _____ E-mail. _____

Name of the Supervisor: _____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): _____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/ revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

[Signature of Student]

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at.....(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor)

Signature of HOD

FOR IRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
Report Generated on	<ul style="list-style-type: none"> • All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String 	Submission ID	Word Counts	
			Character Counts	
			Total Pages Scanned	
			File Size	

Checked by
Name & Signature

Librarian

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.jult@gmail.com

ACKNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to almighty God for His divine blessing makes it possible to complete the project work successfully.

I am grateful and wish my profound indebtedness to Supervisor **Dr. Pankaj Dhiman Assistant Professor (SG)**, Department of CSE Jaypee University of Information Technology, Wakhnaghat. Deep Knowledge & keen interest of my supervisor in the field of “**Cloud Computing**” to carry out this project. This project was made possible by his never-ending patience, academic leadership, constant encouragement, constant and energetic supervision, constructive criticism, insightful counsel, reading numerous subpar versions and fixing them at all levels.

I would like to express my heartiest gratitude to **Dr. Pankaj Dhiman**, Department of CSE, for his kind help to finish my project.

I would also generously welcome each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non- instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

(Student Signature)

Project Group No.: 80

Student Name: Bhavya Gupta

Roll No.: 191543

Table of Content

Title	Page No.
Certificate	(I)
Plagiarism Certificate	(II)
Acknowledgement	(III)
Table of Content	(IV)
List of Abbreviations	(V)
List of Figures	(VI)-(VII)
List of graphs	(VIII)
List of Tables	(IX)
Abstract	(X)-(XI)
Chapter-1 (Introduction)	1-13
Chapter-2 (Literature Survey)	14-18
Chapter-3 (System Design, Analysis/Design/Development/ Algorithm)	19-41
Chapter-4 (Performance Analysis)	42-59
Chapter-5 (Conclusion)	60-62
References	63-64

LIST OF ABBREVIATIONS

ML	Machine Learning
URL	Uniform Resource Locator
DFD	Data Flow Diagram
SWOT	Strength weakness objective threat
UI	User Interface
PH	PH value of the soil
N	Amount of nitrogen
P	Amount of Phosphorus
K	Amount of potassium
API	Application programming interface
JSON	JavaScript Object Notation
URL	Uniform Resource Locator
UI	User Interface

LIST OF FIGURES

Figure No.	Figure Title	Page No.
1	Graphical Overview of project	6
2	DFD LEVEL – 0	11
3	DFD LEVEL – 1	12
4	DFD diagram	12
5	Use case diagram of the system	21
6	System Sequence Diagram	22
7	Implementation	27
8	System Architecture	31
9	Importing the libraries and dataset	34
10	Printing the dataset	34
11	Describing the dataset	34
12	Grouping the dataset according to labels	35
13	Checking for null values	35
14	Applying label encoding	35
15	Scaling the data using Standard Scaler	36
16	Checking that crops that have unusual requirements	36
17	Crops having more and less avg. amount of nitrogen	37
18	Describing the crops according to weather condition	37
19	Splitting the data	39
20	Installing libraries for creating API	41
21	Confusion matrix	48
22	Classification report	49
23	Predicted crop based on input parameter	49
24	Importing base model and saved model	50

25	API function with passing input parameter	50
26	Performance metrics of Random Forest algorithm	52
27	Performance metrics of Naïve bayes algorithm	53
28	Performance metrics of Decision tree algorithm	53
29	Performance metrics of Logistic regression algorithm	54
30	Connecting ngrok with port number	57
31	Python file in spyder notebook with public URL passed	58
32	Predicting which crop would be most suitable	58

LIST OF GRAPHS

Graph No.	Figure Title	Page No.
1	Adoption trends of new agricultural practices vs year	6
2	Graph of actual vs predicted values	40
3	Histogram showing correlation of temperature wrt count of label	46
4	Histogram showing correlation of rainfall wrt count of label	47
5	Histogram showing correlation of humidity wrt count of label	47
6	Graph showing performance metrics of all algorithms	62

LIST OF TABLES

Table Number	Table Title	Page No.
1	Attribute description of agricultural dataset	25
2	SWOT analysis	44
3	Showing Results obtained from all algorithms	59

ABSTRACT

Agriculture plays an important role in the economic sector of our country. With the increase in population of our country day by day, there increases the demand of food, hence the need of production of agriculture. India – being known as the agricultural country, its economy depends over the production of agriculture but its contribution towards the GDP of our country is just 14 percent. Farmers struggle a lot to grow and harvest crops but still fail that is could not receive that much amount of productivity as needed and thinking about the reason why they could not produce the crops with maximum productivity and this might be due to the lack of knowledge on which crops would be best suited to cultivate, cause knowing the best suited crop to grow would aid in maximum utilization of the agricultural land and maximum production of agriculture .Machine learning being the most emerging field in nowadays, helped humans a lot in predicting outcomes of unfavorable conditions like weather predictions, stock price prediction ,or predicting the type of crop to be grown depending on the type of soil, pH value ,humidity, temperature etc. It is a field which helps to predict the outcome based on input parameters and predicts the categorical or numerical outcome as per the scenario of the situation. We know that farmers nowadays don't have any idea on which crop to cultivate and hence due to this they face unpredictable outcomes caused by climatic changes, knowing which crop to grow beforehand will help them to increase the productivity of the crop grown and moreover would be beneficial to make some important decisions .So, we have developed a machine learning using various machine learning algorithms which will predict which crop is best suited to grow based on the climatic factors .Several type of machine learning algorithms like Logistic Regression, Random forest, Decision tree and Naïve Bayes are used to predict the outcome and the highest accuracy to predict the type of crop was obtained by Naïve Bayes. After that we have performed clustering analysis using K-means algorithm and created four clusters. Then we passed an array with input parameters to predict which crop would be suitable to grow based on those conditions.

Secondly, we imported pickle library as we wanted to deploy our machine learning model as public API in Google Collaboratory using ngrok which is used to create proxy URLs and FastAPI and for that we created two python files in Spyder notebook where our main idea is to integrate it with UI and we can build it with tools like react.js. We are loading the saved model before creating the API otherwise when API would be running, it would unnecessarily again and again load the model. The result would be an URL and with that we must mention the endpoint in order to create public URL and since we deploy in localhost so others cannot access it, only we can save it in order to rectify ngrok. These predictions could be helpful for both agricultural sector and farmers as a whole and would contribute in increase of the economy of our country.

CHAPTER-1

INTRODUCTION

1.1 Introduction

Food plays a vital role in our lives and is very important for living. Given that the production of food depends on the productivity of agriculture, it is quite important that the productivity of agriculture must increase. Agriculture plays a vital role in the country's economic sector. With the increase in population of our country day by day, there increases the demand of food, hence the need of production of agriculture.

With the increase in the population of India, there has been an increase in the demand for food and hence the need for increased agricultural production. Farmers struggle a lot to grow and harvest crops but still fail that is could not receive that much amount of productivity as needed and thinking about the reason why they could not produce the crops with maximum productivity, which might be due to the lack of knowledge on which crops would be best suited to cultivate, because knowing the best suited crop to grow would aid in maximum utilization of the agricultural land and maximum production of agriculture.

Old agricultural practices were disregarded when the amount of crops and demand and supply increased in the market; hence, the hardcore labor of farmers never brought success to agriculture, then came machines to sow and plod the field and then harvest the crops which came as a boon for farmers and made the whole farming process bit easy but still could not result much in the amount of productivity of the land and crops that would need to grow; farmers failed in one way or the other to increase and optimize agricultural production and it sooner or later affected the economy of India, hugely being known as the agricultural country, its economy depends on the production of agriculture, but its contribution to the GDP of our country is just 14 percent.

Computer science, bringing new tools, products, and technology in today's lives, brings such technologies to help people develop things for mankind and their ease of life, such as machine learning, which is currently the most emerging field, helping humans a lot in predicting outcomes of unfavourable conditions such as loan predictions, stock price prediction, or predicting the type of crop to be grown depending on the type of soil, pH value, humidity, etc. It is a field that helps predict the outcome based on input parameters and predicts the categorical or numerical outcome as per the scenario of the situation.

India- being known as the agricultural country—its overall economy mainly depends on the production of agriculture, and the land provided to the farmers for the cultivation of crops is not utilized fully because of the failure of knowledge of the crops to grow depending on the climatic conditions. Hence, there is a need to build a machine learning model that could predict which crop to be cultivated based on the climatic conditions. Unfavorable climatic changes pose a great threat to crop cultivation and hence decrease not only the overall economy of the country but also the revenue for farmers working hard in the ground. Crops are affected by unpredictable changes in humidity, soil pH, and rainfall, which destroy the early stages of crops.

The main goal is to maximise the production of food, fibre, or other agricultural goods by achieving the largest crop yield per unit of land possible. This entails maximising several variables, including crop choice, irrigation, pest control, and nutrient management. Optimisation methods, such as data analysis and predictive modelling, offer important insights for sensible agricultural decision-making. Farmers may make better choices about planting schedules, irrigation, fertilisation, and resource allocation by combining data-driven approaches and advanced analytics. By increasing the availability of wholesome, high-quality food, optimising agricultural production helps to increase global food security. It assists in meeting the rising demand for food in a sustainable manner by increasing yields and minimising post-harvest losses.

Utilising resources effectively is a goal of agricultural production optimisation. Some of these resources include energy, fertilisers, and water. It aims to reduce waste and input costs while maintaining or raising crop output by optimising resource allocation and management practises. Improving agricultural output also entails raising farmers' and stakeholders' bottom lines. This entails striking a balance between input costs, market demand, and profitability while preserving sustainability over the long term. Optimisation of agricultural production encourages environmentally friendly farming methods that reduce the harmful effects of conventional agriculture. This includes minimising chemical use, preserving water resources, preventing soil erosion, and fostering biodiversity.

Therefore, it is important to adopt new agricultural technologies. Knowing which crop to grow depending on the climatic conditions would aid in increasing the productivity of agriculture and thus would increase the economy of the country along with increasing the profit margin for farmers and hence avoid soil pollution and land wastage. Machine learning, the most advanced and budding field in recent times, is a part of artificial intelligence that proves to be a game changer in the agricultural sector, and has outgrown itself with big data technologies and high-performance computing to create new opportunities in the multi-disciplinary agrotechnology domain. Therefore, to solve this problem, we built a machine learning model that could recommend which crop would be best suited to grow based on various input parameters such as rainfall, pH value, humidity, and temperature.

For this we have downloaded the dataset from Kaggle and then after cleaning the data, applied various machine learning algorithm on it like Logistic regression which predicts the outcome based of classification problems mostly predicting the binary outcome of the result, three types of logistic regression are binary, ordinal and multinomial, random forest however predicts the outcome on basis of ensemble learning that is it learns the classifier by combining multiple generated decision trees and then predicts the outcome based on the aggregation of those combined trees and is used for solving classification problems.

Whereas naïve bayes being the probabilistic classifier predicts the outcome on the basis of conditional probability theorem known as bayes theorem and is names as naïve bayes cause in it the presence of one feature is independent from the presence of other features and bayes cause it's based on bayes theorem and is used for solving classification problems and decision tree is one of the simplest and easy algorithm to apply and works on classification problem, it is also used to predict the binary outcome of the situation presented.

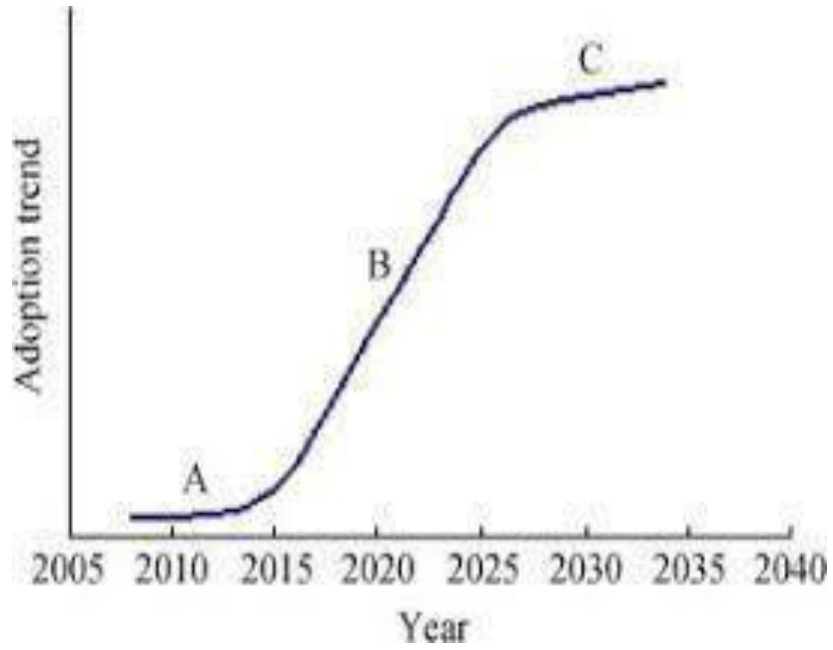
It is like a tree with branches as yes and no outcomes, asks questions, and builds a classifier in such a way that asks and solves such questions and produces the output answer as yes or no. Among all the algorithms used and worked upon, we found that random forest gave the highest accuracy in predicting the outcome (which crop would be best suited to grow). Random forest predicts the outcome on the basis of ensemble learning; that is, it learns the classifier by combining multiple generated decision trees and then predicts the outcome based on the aggregation of those combined trees and is used for solving classification problems, the output of which is the classifier with the highest number selected.

The dataset used, which was downloaded from Kaggle, had features such as the amount of nitrogen, phosphorus, potassium, soil pH, temperature, humidity, and rainfall, all of which were used to predict which crop would be the most suitable for growth and would yield maximum productivity.

Second, we passed the input parameters in the form of an array, and according to the values, we predicted which crop would be the most suitable to grow. Second, after building the machine learning model, we deployed our model as a public API using FastAPI and ngrok on Google Colab. Public API is basically the one that can be accessed by anyone over the Internet through the URL that anyone can post input parameters to the API to determine whether a suitable crop exists or not. Ngrok is used to create proxy URLs, and we have imported various libraries such as FastAPI for creating APIs, uvicorn to access the server, pickle to load the saved data

model, pydantic library to basically structure the format in which the data would be posted to our API and import the base model, requests for posting the URLs, and json in order to post the data in the form of a json object. There are two methods in API that is get and post method and we are using this and parameters are URL to which we have to post the data and the json file which we have to post. We created two Python files in a spyder notebook, which is an open platform for Python, and tested the APIs. Our main idea in deploying our machine learning model as an API is to integrate this with the user interface, and we can build it with tools such as react.js or other UI tools. We imported CorsMiddleware, which is cross-origin resource sharing, to allow the domain to use our API; we used an asterisk, which means it would allow all the domains under it; if we do not follow a particular procedure to run, it would give course error when we would integrate with UI.

We also imported the base model from the pydantic library and loaded the saved model even before creating the API; otherwise, each time the API would run, it would unnecessarily load the model again and again. Because our API would only receive JSON data, we need to convert the dictionary from which we extracted the data and stored it in the list to the JSON object using the post method. The result after connecting the ngrok with the port number would be a URL, and with that we must mention the endpoint in order to create public URL, we deployed in localhost so that others cannot access it only we can save in order to rectify ngrok, which basically created a proxy URL for our localhost, so running the code would give us public URL and that port numbers with it are in which our API would be hosted.



Graph 1: Adoption trends of new agricultural practices vs year

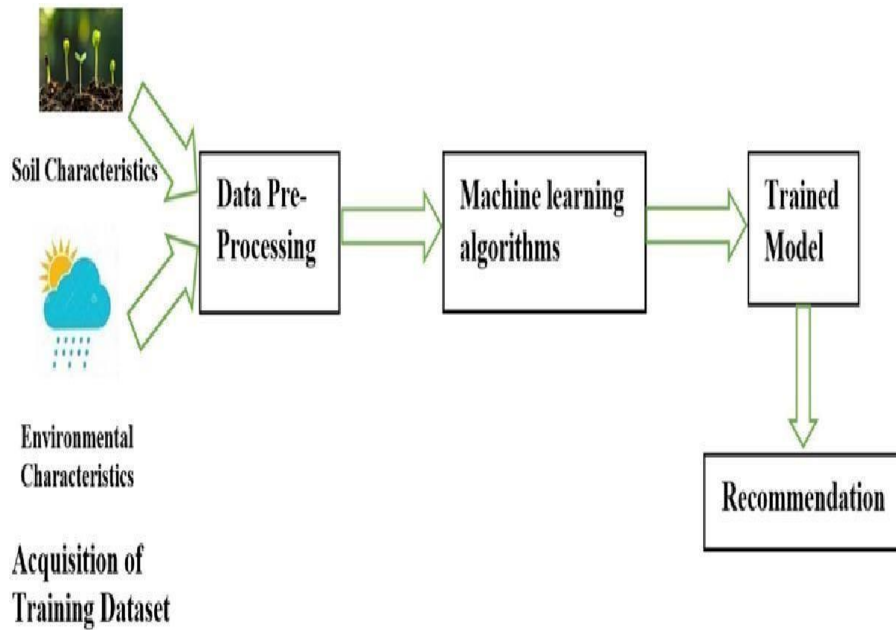


Fig 1 : Graphical Overview of project

1.2 Problem Statement

As we are aware that agricultural production depends largely on the type of soil and the changes in climate, many times, we face unpredictable changes in weather, such as fluctuations in humidity, heat waves, and unpredictable rainfall, which cause a great loss to farmers and, hence, decrease the productivity of agriculture and farmers fail to utilize their land to their fullest.

Farmers struggle a lot to grow and harvest crops but still fail that is could not receive that much amount of productivity as needed and thinking about the reason why they could not produce the crops with maximum productivity and this might be due to the lack of knowledge on which crops would be best suited to cultivate, cause knowing the best suited crop to grow would aid in maximum utilization of the agricultural land and maximum production of agriculture.

The unfavourable climatic changes pose a great threat to the crops cultivating and hence decreases not only the overall economy of country but also a decreased revenue for farmers working hard in the ground. Crops are affected by the unpredictable changes in humidity, pH value of soil and unpredictable rainfalls which destroy the early stages of crops. Therefore, adopting new agricultural technologies is quite important. Therefore, our main motive is to create a machine learning model that can predict which crop would be best suited to grow, depending on various factors such as humidity, rainfall, and temperature.

1.3 Objectives

India is an agricultural country that produces crops in large amounts, and unpredictable climatic conditions pose a great threat to ungrown crops and cultivated land and decrease the economy of the country by decreasing the GDP of India. Machine learning – being the most advanced and budding field in nowadays is a part of artificial intelligence proves to be a game changer in field of agricultural sector.

It has outgrown itself with bigdata technologies and high- performance computing to create new opportunities in the multi-disciplinary agrotechnology domain. The unfavourable climatic changes pose a great threat to the crops cultivating and hence decreases not only the overall economy of country but also a decreased revenue for farmers working hard in the ground. Crops are affected by the unpredictable changes in humidity, pH value of soil and unpredictable rainfalls which destroy the early stages of crops. Therefore, adopting new agricultural technologies is quite important.

The goal of agricultural production optimisation is to reduce the risks brought on by variables including climate change, pests, illnesses, and market swings. Farmers may lower losses and improve the resilience of their agricultural systems by putting risk management principles into practise.

Optimisation methods, such as data analysis and predictive modelling, offer important insights for sensible agricultural decision-making. Farmers may make better choices about planting schedules, irrigation, fertilisation, and resource allocation by combining data-driven approaches and advanced analytics. By increasing the availability of wholesome, high-quality food, optimising agricultural production helps to increase global food security. It assists in meeting the rising demand for food in a sustainable manner by increasing yields and minimising post-harvest losses.

Utilising resources effectively is a goal of agricultural production optimisation. Some of these resources include energy, fertilisers, and water. It aims to reduce waste and input costs while maintaining or raising crop output by optimising resource allocation and management practises. Utilising resources effectively is a goal of agricultural production optimisation. Some of these resources include energy, fertilisers, and water. It aims to reduce waste and input costs while maintaining or raising crop output by optimising resource allocation and management practises.

Improving agricultural output also entails raising farmers' and stakeholders' bottom lines. This entails striking a balance between input costs, market demand, and profitability while preserving sustainability over the long term. Optimisation of agricultural production encourages environmentally friendly farming methods that reduce the harmful effects of conventional agriculture. This includes minimising chemical use, preserving water resources, preventing soil erosion, and fostering biodiversity.

Hence, our main objective is to help farmers and the government by developing a machine learning model that could increase the productivity of agriculture by predicting which crop is best suited to grow depending on various climatic factors and conditions. Moreover, it increases the accuracy of crop yield production by using various machine learning techniques that would increase agricultural production and financial income for farmers by utilizing their agricultural land to its fullest and cultivating the crop that would be best suited to grow.

1.4 Methodology

This is the implementation of an agricultural optimization production model by developing a machine learning model. Machine learning – being the most advanced and budding field in nowadays is a part of artificial intelligence proves to be a game changer in field of agricultural sector, it has outgrown itself with bigdata technologies and high- performance computing to create new opportunities in the multi-disciplinary agrotechnology domain. To build this model, a Jupyter notebook and an Anaconda prompt were installed and set up on the PC. Then installing and setting up python 3.6 followed by downloading all the data files which were downloaded from Kaggle, in the jupyter notebook.

Then building the machine learning model by applying various machine learning algorithms like Random Forest, Decision tree, Naïve bayes and Logistic regression and then graphically visualizing the data. Among all the algorithms used and worked upon, we found that naïve Bayes gave the highest accuracy in predicting the outcome (which crop would be best suited to grow).

Naïve Bayes based on the bayes law predicts the outcome on basis of probability and is part of basically generative learning algorithms that is it learns the classifier by modelling the distribution of input parameters of a class or category, it predicts based on probability of hypothesis with prior knowledge. Secondly, we have passed the input parameters in form of array and according to the values, we have predicted which crop would be the most suitable crop to grow.

After building the machine learning model, we have deployed our model as public API using FastAPI and ngrok in google colab. Public API is basically the one which can be accessed by anyone over the internet through URL that is anyone can post input parameters to API in order to find whether a suitable crop exists or not. After installing the spyder, we created two Python files in the spyder notebook, which is an open platform for Python, and tested the APIs. Our main idea to deploy our machine learning model as API is we have to integrate this with user interface and we can build it with tools like react.js or other UI tools. Since our API would only receive json data, so we need to convert dictionary from which we extracted the data and stored in the list to json object using post method. The result after connecting the ngrok with the port number would be a URL, and with that we have to mention the endpoint in order to create public URL, we deployed in localhost so that others cannot access it only we can save in order to rectify ngrok, which basically created a proxy URL for our localhost, so running the code would give us public URL and that port numbers with it are in which our API would be hosted.

The data flow diagram maps the entire information of any process or system by using symbols such as circles, rectangles, and squares. Data flow between internal processes and external entities is denoted by arrows. Data flow within a system is graphically represented by a data flow diagram (DFD). It provides an example of how data is transferred between various processes, external entities, and data repositories. To comprehend and record the system's data flows, inputs, outputs, and transformations, DFDs are frequently employed.

It is crucial to remember that a DFD's complexity and amount of detail might change based on the goal and extent of the system being modelled. By including extra information, such as additional processes, data stores, and data flows, data flow diagrams can be further enhanced. Data stores can be added to represent the storage of data within the system, and each process can be broken down into lower-level processes. The main purpose is to show and describe the boundaries and scope of any process or system. It describes how the data will enter and leave the system and where that data would be stored, below are different levels of data flow diagram.

The data flow diagram of each level is being drawn below:

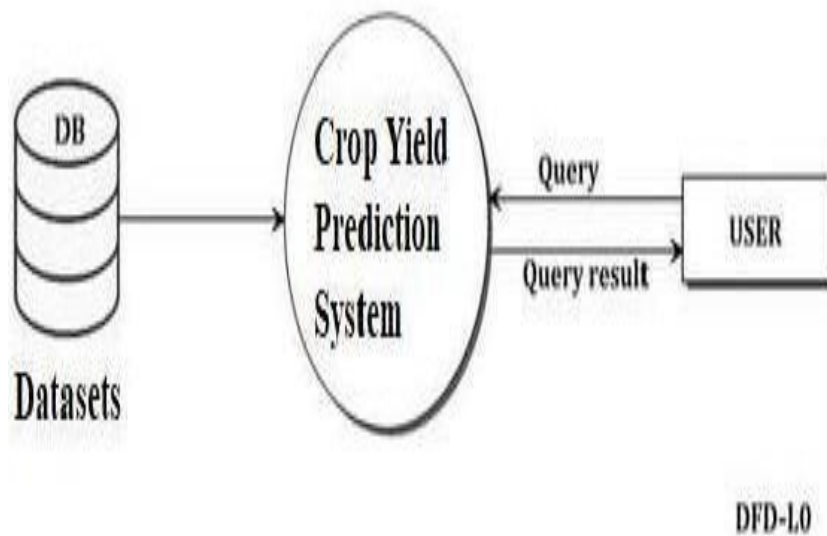


Fig 2 : DFD LEVEL – 0

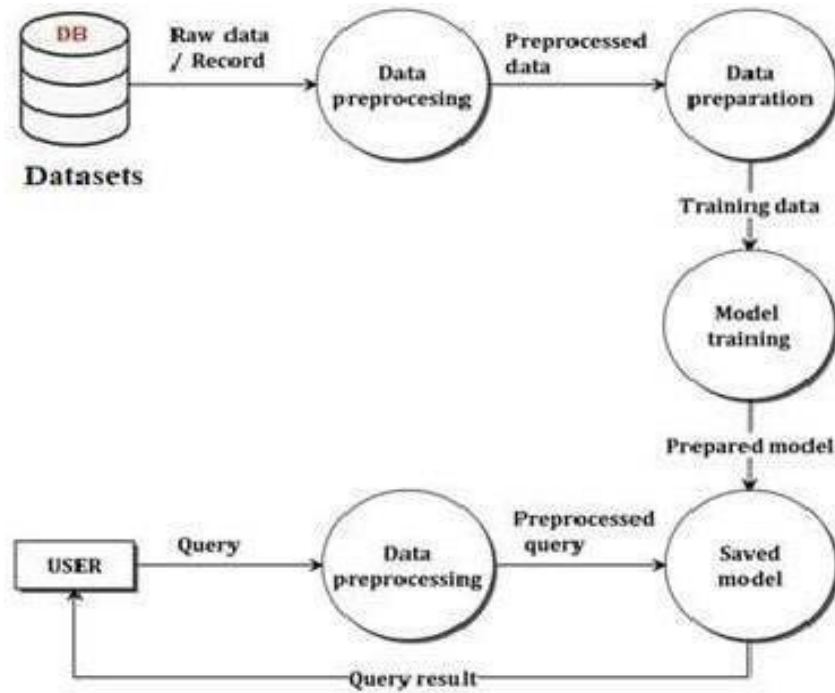


Fig 3: DFD level-1

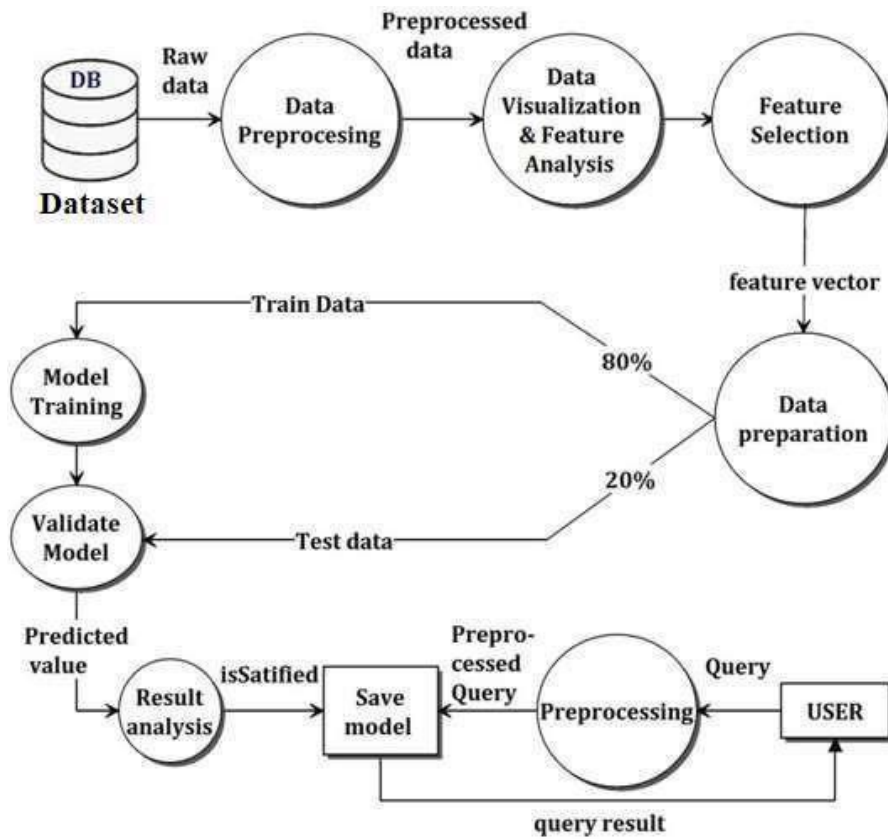


Fig 4 : DFD diagram

1.5 Organization

The rest of the paper is organized as follows: In Chapter 2, we present a literature survey that depicts the various approaches used by the authors to create an agricultural optimization production model. Section 3 highlights the methodology and system development of the project. It represents the various computational, experimental, and mathematical concepts of the project.

In addition, we focused on the software and hardware platforms required to implement the model. In Chapter 4, we present a performance analysis of the project, which specifies the accuracy of the project. In addition, we show the required dataset and related information. Section 5 presents the conclusions of the project and observations of the results. It also provides applications for the project and future scope of the project.

CHAPTER-2

LITERATURE SURVEY

2.1. Literature Survey

This chapter provides a survey of possible approaches to agricultural production optimization. We frequently experience unforeseen weather changes, such as humidity fluctuations, heat waves, and unpredictable rainfall, which cause a great loss to farmers and, as a result, decrease the productivity of agriculture and prevent farmers from making the best use of their land. As we are aware, agricultural production depends greatly on the type of soil and the changes in climate. This survey aids in identifying the shortcomings of several current techniques. The problem with most approaches is that they are not used in real-time, which makes it difficult for end users to use them. The focus is on increasing the productivity of agriculture and utilizing agricultural land to its fullest, which would aid in increasing the overall economy of the country.

As a result, our main goal is to assist farmers and the government by creating a machine learning model that might boost agricultural productivity by foretelling which crop will thrive under certain climatic conditions. Additionally, it improves crop yield accuracy with a variety of machine learning approaches, which would raise agricultural production and financial income for farmers by making the best use of their agricultural area and cultivating the crop that would be most suitable to grow. The survey of nine research papers is given below, which describes how and what methodology they have used, what were the disadvantages caused by that, and what algorithms and results they obtained in predicting which crop would be the most suitable crop to grow depending upon various climatic factors.

The description of the research papers are organized as follows.

[1] M. Suganya, Dayana R, Revathi.R, “Crop Yield Prediction Using Supervised Learning Technique “IJCET [2020]

This research paper was published by the IJCET. The proposed system performs tasks, such as preprocessing, classification, regression, clustering, and visualization. The algorithms used in this study to predict which crop would be the most suitable crop to grow were KNN, logistic regression, SVM, and random forest, among which random forest produced the highest accuracy. One of the disadvantages of this method is that the accuracy of predicting the crop yield using SVM is quite low.

[2] Mrugank Gandhi, Shubham Kothavade, “Agricultural Production Optimization Engine,” IRJMETS [2022]

This research paper was published by the IRJMETS. This study used the KNN algorithm to predict which crop would be the most suitable crop to grow, depending on various climatic factors. The main objective was to increase the accuracy of crop yield production. The dataset they used had only three columns: district, season, and crop names. This study considers the information related to the yield of season, area, soil parameters, and past year and then predicts which are the best beneficial crops that could be grown in that condition. The disadvantage of this study was the use of a single ML algorithm, and thus did not compare its efficiency with other algorithms.

[3] Yogita Masare, Sneha Mahale, Manjusha Kele, Ashvin Upadhyay, Bhushan R. Nanwalkar, “The System for Maximize the Yielding Rate of Crops using Machine Learning Algorithm,” IJERT [2021]

This research paper was published by the IJERT. The proposed system uses naive Bayes classifiers, which predict the outcome based on a probabilistic classifier, for lot yield prediction and fertilizer prediction.

They classified the production yield rate as poor, moderate, and good.

One of the disadvantages of this study was the use of a single algorithm to predict the results; however, a comparative study would yield better results.

[4] Vrushali C. Waikar, Sheetal Y. Thorat, A. A. Ghute, and Priya P. Rajput, Mahesh S. Shinde, "Crop Prediction based on Soil Classification using Machine Learning with Classifier Ensembling," IRJET [2020]

This research paper was published by the IRJET. They designed a web-based system to predict crops based on soil classification and used machine learning algorithms such as SVM, Adaboost, Naïve Bayes, ANN, and bagged tree; the accuracy of SVM was the highest at 98.5 percent. Their system can predict crops based on soil parameters. Their future work was to recommend a location module that could be added according to the crop suggestion means according to the crop that would predict the available and suggested location. One of the disadvantages of this study was an overly perfect model with an accuracy as high as 98 %, but it could suggest a location to grow according to the suggested crop.

[5] Ramesh D, Vishnu B. Vardhan, "Data Mining Techniques and Applications to Agricultural Yield Data," IJARCEE [2013]

This study was published by the IJARCEE. K-means, KNN, ANN, and SVM were used in this study to predict yield production. The dataset considered in this study contained four input variables and one output variable. Their future work is aimed at using more machine-learning algorithms to improve the efficiency of the proposed model. The results obtained showed that the multiple linear regression method gave the highest accuracy of 98%, which was used where one needs to build a relationship between the dependent variable and one or more independent variables, whereas the K-means algorithm, which was applied by forming four clusters of data where rainfall was considered as the key factor, gave 96 percent of accuracy in predicting the yield with the highest accuracy and generality.

[6] Shubham Prabhu, Prem Revandekar, Swami Shirdhankar, "Soil Analysis and Crop Prediction," IJSRST [2020]

This research paper was published by the IJSRST. They analyzed soil samples collected from Maharashtra. The project achieved high efficiency in obtaining live data samples of temperature and soil. The model was built using machine learning algorithms such as naïve Bayes, C4.5, and logistic regression, of which the highest accuracy was achieved by C4.5, which was 85%, whereas the accuracy of logistic and naïve Bayes was as low as 76 percent and 66%, respectively. The analysis of soil in this project has been proposed using Arduino and cloud computing, and their main objective is to predict the type of crop to grow based on the type of soil and weather conditions and the type of precipitation that the region has. One of the disadvantages of this study is that the accuracy of the logistic regression and naïve Bayes obtained by this model is quite low; hence, more machine learning algorithms could be applied to achieve high accuracy.

[7] Zeel Doshi, Rashi Agrawal, Subhash Nadkarni, Prof. Neepa Shah, "Agro Consultant: Intelligent Crop Recommendation System Using Machine Learning Algorithms," ICCUBEA [2018]

This research paper was published at (Fourth International Conference on Computing Communication Control and Automation. In this paper the authors made a system known as Agro Consultant and for that system they used machine learning algorithms like KNN, Decision tree, Random Forest and Neural networks to predict which crop would be the most suitable crop to grow based on parameters like season, geographical location, and soil type. Their agro-consultant system could be divided into two parts, mainly crop suitable predictor and rainfall predictor, on which they achieved 71 percent accuracy by applying a rainfall predictor model and an accuracy of 91 percent by applying a neural network algorithm to their crop suitable predictor system. They also included the map visualization feature and map predictor.

[8] Jain, Sonal, and Ramesh Dharavath. "Machine Learning convergence for weather-based crop selection." In 2020 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS) [2020]

This research paper was published by the SCEECS. In this study, the authors divided the methodology into two phases: seasonal weather prediction and identification of suitable crops. They used recurrent neural networks for weather forecasting and used a random forest algorithm for the classification of crops based on the input parameters. Their dataset consisted of ten crops mainly rice, cotton, maize, sunflower, castor, chili, red gram, green gram, jowar and soybean. They compared their results given by RNN with ANN using the root mean squared method, and their future work was to improve the efficiency of the random forest algorithm by using IOT devices to collect accurate soil and weather data of any particular farm, such as soil parameters, and the ratio of N-P-K could be considered for better accuracy.

[9] Suresh, G., A. Senthil Kumar, S. Lekashri, and R. Manikandan. "Efficient Crop Yield Recommendation System Using Machine Learning For Digital Farming." International Journal of Modern Agriculture 10 [2021]

This research paper was published by the IJMA. In this study, the authors used an SVM to identify a particular crop based on the given input parameters and achieved high productivity and accuracy. They worked on two datasets, mainly a dataset of location and crop data. Based on this, they predicted the most suitable crop based on the nutrient value (value of nitrogen, value of phosphorus, value of potassium, and pH of soil) and also identified the fertilizer type and quantity required for particular crops, such as rice, maize, black gram, carrot, and radish.

CHAPTER-3

SYSTEM DEVELOPMENT

3.1 Non-Functional Requirements

3.1.1 User interface

The user interface must be simple and easy to use and understand, and based on the given inputs, the model must be able to efficiently predict the outcome on which crop would be best suited to grow.

3.1.2 Hardware

No special hardware interface is required for the successful implementation of the system other than a PC.

3.1.3 Software

- Python for training the model

- Chrome browser

- Jupyter notebook and anaconda prompt

- Goggle Collaboratory

- Spyder (Python 3.9)

3.1.4 User Requirements

The user requirements include that the user could comfortably and smoothly navigate through the mobile application and could be able to use all the services easily without runtime lag.

It must be able to use without any problem or error or any unexpected crashes of the system. It also includes the constraints on the user that the user must not provide or leak any false information to other users using the

application and shall not be able to access any other user's profile or make any false changes in that.

3.1.5 Performance

The machine learning model should be able to predict the outcome efficiently and in the most accurate manner and the working of public API which we have created after deploying our machine learning model as public API in google colab using ngrok and FastAPI.

3.1.6 Computational metrics

The training time is dependent on the GPU. GPUs with a higher memory like 4-16 GB are recommended for such applications. Software such as Jupyter notebook is preferred, but applications such as PyCharm and VScode can also be used along with Python libraries such as NumPy, Keras, and Tensorflow. Spyder (Python 3.9) was used to create two Python files to test the API.

For our project, we used Jupyter notebook, which uses a browser to write and execute code in Python language, and is well suited to machine learning, deep learning, data analysis, and education. Experiments were run on a Jupyter notebook with an Intel® Xeon® processor at 2.20 GHz using 12.72 GB RAM coupled with an Nvidia Tesla T4.

3.2 Constraints And Assumptions

3.2.1 Constraints

Language used to build the project and implementation is python because of its in-built libraries. The platform used to build the project is jupyter notebook followed by installation of Anaconda prompt and then we have deployed our machine learning model as public API using ngrok and FastAPI in google colab and for testing of the API, we have created two python files in Spyder notebook.

3.2.2 Assumptions

The user has a basic knowledge and learning of how to use and implement machinelearning algorithms and how to run API and basic knowledge of what are the libraries imported to create an API and how to import them and use them while creating a public API.

3.3 Use case diagram

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will oftenbe accompanied by other types of diagrams as well. They are like the blueprints of the system. The use cases are represented by either circles or ellipses. It depicts what a user would do in proposed undertaken work.

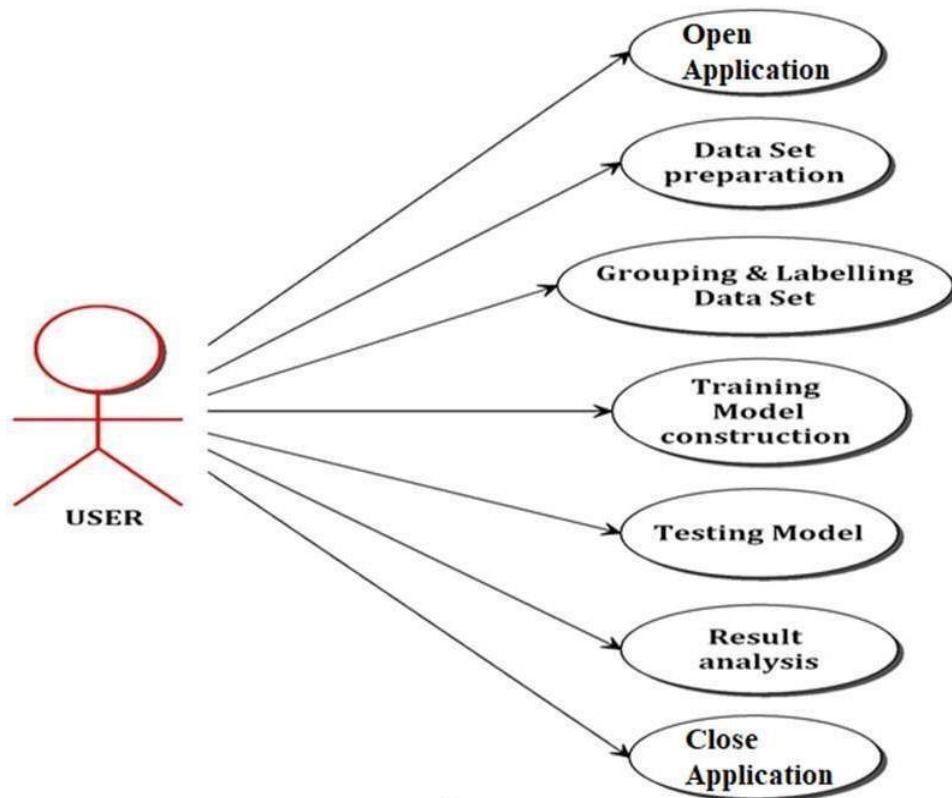


Fig 5: Use case diagram of the system

3.4 Sequence Diagram

A sequence diagram is known as UML diagram which illustrates the order of messages between objects in an interaction. It basically describes the communication happened between the user and model being built. Sequence diagrams are helpful for visualizing a system's dynamic behavior, comprehending the sequence of interactions between components, and seeing possible problems or design bottlenecks. They are frequently employed in the creation of software, notably in the study and design of objects. The sequence of interactions between the user and the plugin are shown in the figure below:

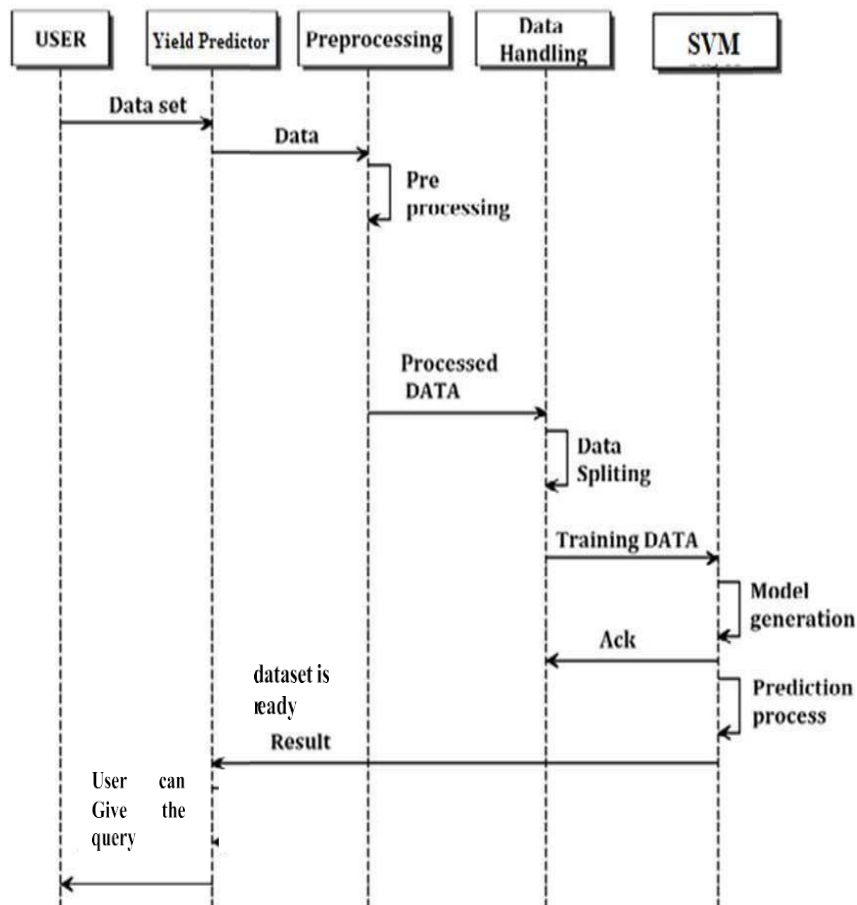


Fig 6 : System Sequence Diagram

3.5 System Development

System development is the process of defining, identifying, testing, and implementing a program or any software built. It is like how the activities or steps have proceeded in the project. Here is a summary of the typical steps in creating a machine learning model:

- **Problem Definition:** Clearly state the issue that you hope to use machine learning to resolve. Determine the aims and targets as well as the information at hand for instruction and assessment.
- **Gathering:** Gather pertinent data for the machine learning model's training by gathering it. To develop meaningful input features, this may entail gathering data from diverse sources, cleaning and preparing the data, addressing missing values, and doing feature engineering.
- **Data Splitting:** Create training, validation, and testing datasets from the collected data. The validation set aids in tuning hyperparameters and assessing model performance throughout development, while the testing set is utilized for the model's final evaluation after training.
- **Model selection:** Decide which machine learning model or method best addresses the current issue. Think about elements like the type of data, the nature of the issue (classification, regression, etc.), and the resources accessible.
- **Model training:** Apply the training dataset to the chosen model. By modifying its internal parameters, the model discovers patterns and relationships in the data. In this procedure, the loss function of the model is minimised using optimisation techniques like gradient descent.
- **Model testing:** After the model's performance on the validation set has been determined to be satisfactory, assess it on the distinct testing dataset. This offers an objective evaluation of the model's generalisation and forecasting skills.

- **Deployment:** Deploying the model as a public API on google collab using ngrok also includes the machine learning model that has been trained in a real-world setting. To do this, the model must be packaged and an interface must be made so that it can receive input data and output predictions. Scalability, monitoring, and performance optimization are possible deployment factors.
- **Monitoring and Upkeep:** Continue to keep an eye on the model's performance and review it as necessary. As new data become available or the problem domain shifts, update the model as necessary. The model is maintained regularly to ensure accuracy and applicability.

3.6 Data Set Features

Dataset has been taken from Kaggle named as agriculture optimization dataset. The unique traits or characteristics of the data points within a dataset are referred to as dataset features. Depending on the situation, these features may alternatively be referred to as variables, dimensions, or columns. Each feature represents a particular piece of data that characterizes a certain aspect of the data.

Features are essential when using machine learning to train and create predictive models. They supply the data the model needs as input to make predictions or discover trends. The performance and generalizability of the model are strongly influenced by the choice and caliber of the features. Attributes description of Agricultural Dataset consists of a total of 7 attributes with one target attribute which is used to predict which crop would be most suitable to grow depending upon various climatic conditions.

Table no 1:- Attribute description of agricultural dataset

S.No	Attribute	Code
1	Amount of Nitrogen	N
2	Amount of Phosphorus	P
3	Amount of Potassium	K
4	Temperature	temperature
5	Rainfall	rainfall
6	pH	pH
7	Humidity	humidity

Brief description of attributes:

- **Amount of Nitrogen:** This attribute defines the amount of nitrogen present in the soil.
- **Amount of Phosphorus:** This attribute defines the amount of phosphorus present in the soil.
- **Amount of Potassium:** This attribute defines the amount of potassium present in the soil.
- **Temperature:** This attribute defines the temperature of the atmosphere.
- **Rainfall:** This attribute defines the rainfall in the area.
- **pH:** This attribute defines the pH of the soil.
- **Humidity:** This attribute defines the humidity of the atmosphere.

3.6.1 Implementation

Step by Step approach for the implementation of machine learning model:

Step1: After setting up Jupyter notebook we must download the agricultural dataset from Kaggle.

Step 2: Importing Python libraries like NumPy, Pandas, Sklearn, Matplotlib along with the dataset in jupyter notebook.

Step 3: Check the data set for null values or missing values and follow the steps of pre- processing the data.

Step 4: Graphically visualize the data by preparing suitable graphs for better interpretation of the data.

Step 5: Splitting the dataset into testing and training data and then applying suitable machine learning algorithms.

Step 6: Find the performance metrics from each of the applied machine learning algorithm and seeing which is the best algorithm with the highest accuracy.

Step 7: After building the machine learning model, we must install all the libraries like uvicorn, pickle, FastAPI, json, pydantic, pyngrok, requests etc, in a different Collab file to create a public API.

Step 8: Then created an API by passing all the domains [*] in CorsMiddleware and input the base model with all the input parameters and their type, we loaded the saved model of the dataset and created a post method function with endpoint and connected ngrok with the port number to create a public API.

Step 9: Similarly, we created two python files in spyder notebook, one like colab file with the base model imported from undefined importing request and json library and then passing the URL as the response.

Step 10: Create another python file with input parameters and function to create an API, after which on running the colab file and copying the public URL into the spyder notebook in the URL section, we could obtain the result in the console section along with whether the code is running and is accepted or not.

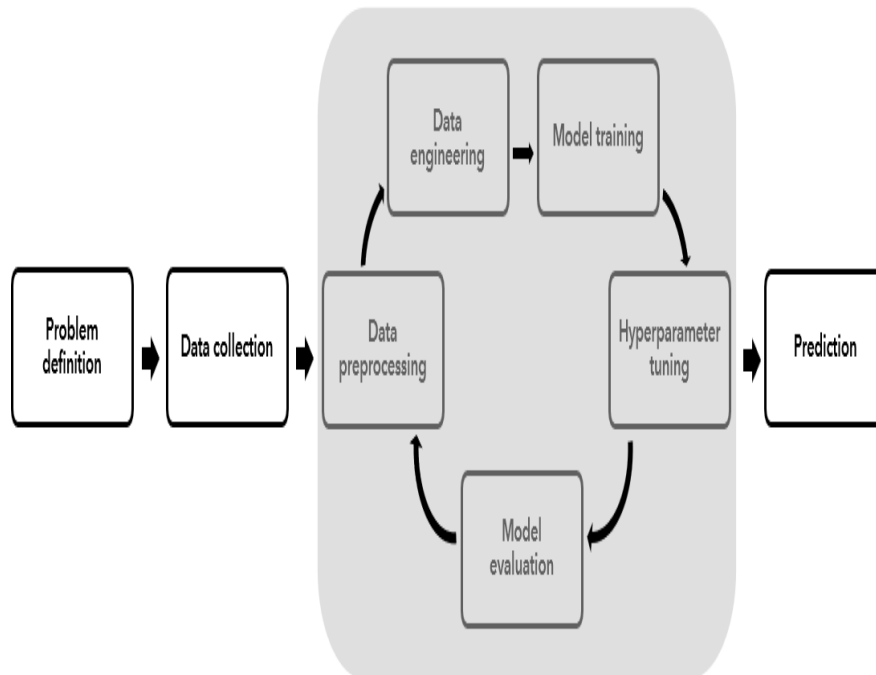


Fig 7 : Implementation

3.7 Prototype across the modules

This section details the input and output to each system module.

- **Preprocessing:**

Preprocessing is a crucial stage in machine learning that entails converting raw data into a format appropriate for training and enhancing the functionality of machine learning models. For preprocessing of our machine learning model, we check for noisy data and missing values and then split the dataset in two parts mainly training and testing data and then train the dataset using various machine learning algorithms to predict the outcome.

- **Standardization of Data:**

It is a technique which is most performed on machine learning algorithms after pre-processing of data, it is used to scale the data in a particular range so that no input parameter or variable has range other than other variables or parameters.

The standard score of a sample x can be determined by the formula:

$$z = (x - \mu) / \sigma$$

where μ is the mean of the training samples and σ is the standard deviation of the training samples.

- **Training:**

Using the preprocessor output files, this module is trained on four machine learning algorithms that is decision tree, naïve bayes, random forest and logistic regression and found out that random forest produced the maximum accuracy among all.

- **Classification:**

Using the feature vector from the feature extraction module, this module generates an output indicating which crop would be the most suitable crop to be grown. The algorithms applied are logistic regression, decision tree, random forest and naïve bayes. Based on the type of problem they answer

and the learning strategy they use, algorithms for machine learning may be divided into several categories. Here are a few standard categories for machine learning models:

1): Models for supervised learning: Models that learn under supervision use labelled training data in which each data point contains input features and a target or label that corresponds to it.

2): Models for unsupervised learning: These models use data that has not been labelled and for which there are no predetermined target labels. Unsupervised learning models look for structures, correlations, or trends in the data.

3): Models of Reinforcement Learning: In reinforcement learning, an agent discovers how to interact with the environment to increase rewards or decrease punishments. The environment reacts to the agent's behavior by giving it rewards or punishments.

- **API Creation:**

A combination of software development expertise, business requirements knowledge, and consideration for security and usability are needed while developing an API.

To deploy our machine learning model as a public API in google colab, we installed various libraries in google colab like pydantic, ngrok, requests, json, pickle, uvicorn etc and along with that we created two python files in spyder notebook, our main idea was to integrate with the user interface and then we could build it with tools like react.js. The result was a local public URL with endpoints and port number in which our API would be hosted locally so that others could not access it, only we could save in order to rectify ngrok.

3.8 Development details

Python 3.9 is required to implement the machine learning project and the agricultural dataset to be worked upon was downloaded from Kaggle. The dataset used which was downloaded from Kaggle had features such as amount of nitrogen, amount of phosphorus, amount of potassium, pH value of soil, temperature, humidity, and rainfall.

These were all used to predict which crop would be the most suitable to grow and would yield maximum productivity. Anaconda prompt and jupyter notebook were installed and set up on the pc to implement the model in jupyter. After that we deployed the machine learning as public API using ngrok and FastAPI in google colab, for which we firstly installed spyder (Python 3.9) which is an open platform for python, then we created two python files for creating an API and connecting it with the port number using ngrok tunnel, for which we installed various libraries like pickle, uvicorn, requests, json, pydantic, FastAPI etc. In Spyder notebook, we have basically tested our API. Our main idea was to integrate with the user interface so we could build it with tools like react.js.

The result was a local public URL with endpoints and port number in which our API would be hosted locally so others could not access it, we could only save in order to rectify ngrok and since our API would only receive json data, converted dictionary to json data using post method which was extracted from dictionary to store in list.

3.8.1 System Architecture

It is basically a visual diagram that depicts the physical implementation of components of the project made. Using Python scikit-learn, and many other libraries, we have applied four machine learning algorithms named mainly – Logistic regression, random forest, decision tree and naïve bayes. We had applied these algorithms on to their cultural dataset which was downloaded from Kaggle which had columns and then the was imported into jupyter notebook for further implementation. The dataset was divided into training and testing data.

Data is preprocessed that is checking for any missing or null values and then filling them in by their means, mode etc. Follow further additional steps to preprocess the data and visualize the dataset graphically by preparing some meaningful graphs for better interpretation and understanding of the data.

After that, find the performance metrics for each of the following applied machine learning algorithms and select the best one with highest accuracy to predict which crop would be best suited to grow depending upon the climatic conditions. The further more work is to deploy our created machine learning model as public API using ngrok and FastAPI in google colab, for which for which we firstly installed spyder (Python 3.9) which is an open platform for python, then we created two python files for creating an API and connecting it with the port number using ngrok tunnel, for which we installed various libraries like pickle, uvicorn, requests, json, pydantic, FastAPI etc. For the Spyder notebook, we have basically tested our API. We have also imported base model from pydantic library and loaded saved model even before creating the API since otherwise each time when it would be running, it would unnecessarily again and again load the model.

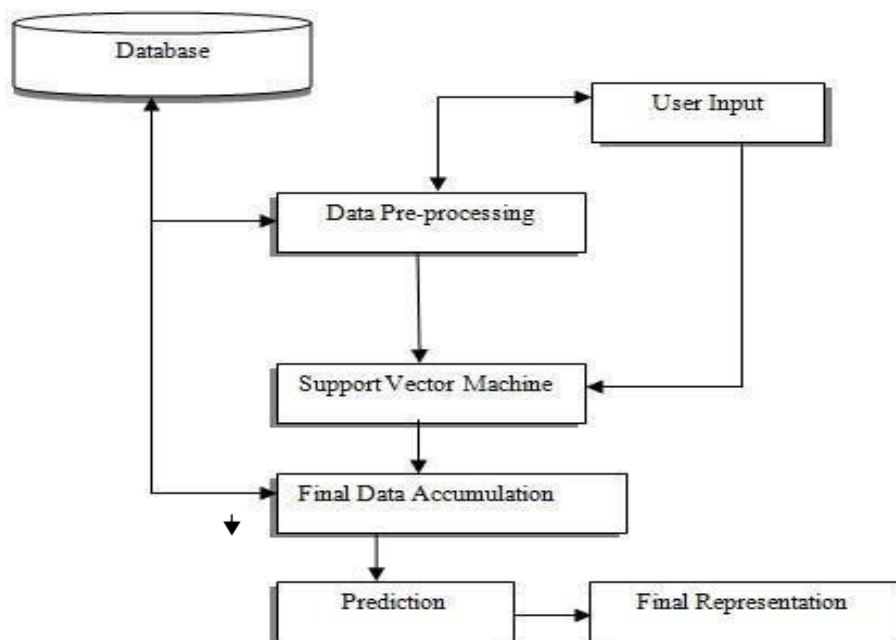


Fig 8: System Architecture

3.8.2 Class diagram

The class diagram clearly depicts the functions of various modules in the computer. It also depicts the interaction of the system's modules, providing a clear idea for implementation. The class diagram gives more light on the blueprints of the computer.

We can use class diagrams to model the objects that make up the computer, to display the relationships between the objects, and to describe what those objects do and the services that they provide. Each class type is represented as a rectangle with three compartments for the class name, attributes, and operations. Objects are represented as ovals that contain class names inside class name compartments. The flow is shown with the help of arrow.

3.9 Module Design

3.9.1 Preprocessing

The agricultural dataset to be used in the machine learning research was downloaded from Kaggle and requires Python 3.9 to be implemented. The dataset that was used and downloaded from Kaggle included features including soil pH, temperature, humidity, and rainfall as well as amounts of nitrogen, phosphorus, and potassium. The dataset downloaded from the Kaggle has all in all 7 features (input parameters) to predict which crop would be the most suitable to grow. The first step in building a model is preprocessing and it is the most important part of the machine learning model since we not always the get the clean and formatted data to work upon hence we need to clear the data and do the formatting that is remove outliers, noise data, and fill up missing values in the data.

It is very important to perform cause if we do not pre-process the data it would result in poor results that are low accuracy and would lead to issues such as overfitting and underfitting and hence it is quite important to be performed. It involves steps such as : getting the dataset at first ,which we have downloaded from the Kaggle named agriculture production dataset having 7 features like rainfall, temperature, humidity, amount of nitrogen ,

amount of phosphorus , amount of potassium and ph. value of soil and the outcome to be predicted which crop would be the most suitable to grow depending on these factors, next step is importing the libraries such as pandas, matplotlib, scikit learn in the jupyter notebook or any platform where machine learning model would be built.

Then comes importing the dataset -here the dataset is agricultural dataset with 7 input parameters ,then comes finding the missing values in the data and then filling up the missing values with either mean or mode ,encoding the data if needed by label encoder and then splitting the dataset in two parts namely testing and training data ,ratio of both could be anything but the most feasible is 80-20 ratio to be worked upon, then comes feature scaling which is important step, where we scale the value of the data in same range or scale so that no other value is far different from one or other value. By normalizing the data, each characteristic is guaranteed to have a comparable scale and distribution. When comparing characteristics with various units or ranges or when employing distance-based methods, normalization is extremely crucial.

To work with machine learning algorithms, categorical variables must be encoded into a number format. The encoding of labels assigns distinct integer amounts to every category, whereas one-hot encoding creates binary columns for each category. Scaling numerical features to a comparable range may prove advantageous in some circumstances. Scaling makes guarantee that a learning method is not dominated by data with bigger values. Standardization and min-max scaling are popular scaling techniques. All the steps are quite important for data to be precise and produce high accuracy. Evaluation of a machine learning model's effectiveness and efficiency in creating predictions or classifications is a common component of performance analysis. Depending on the dataset and the needs of the machine learning issue, the preprocessing procedures may change. Understanding the qualities of the data is crucial in order to select the proper preprocessing methods.

OPTIMIZATION FOR AGRICULTURE PRODUCTION

```
#: #Importing Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from ipywidgets import interact
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
import scipy.stats as stats
from sklearn.metrics import plot_confusion_matrix ,confusion_matrix, precision_score, recall_score, f1_score

#: #Importing Dataset
df = pd.read_csv("data.csv")
```

Fig 9 : Importing the libraries and dataset

```
#: #Printing first 5 observations of dataset
df.head()
```

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

Fig 10: Printing the dataset

```
#: #Description of the dataset
df.describe()
```

	Amount of Nitrogen	Amount of Phosphorus	Amount of Potassium	temperature	humidity	ph	rainfall
count	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000	2200.000000
mean	50.551818	53.362727	48.149091	25.616244	71.481779	6.469480	103.463655
std	36.917334	32.985883	50.647931	5.063749	22.263812	0.773938	54.958389
min	0.000000	5.000000	5.000000	8.825675	14.258040	3.504752	20.211267
25%	21.000000	28.000000	20.000000	22.769375	60.261953	5.971693	64.551686
50%	37.000000	51.000000	32.000000	25.598693	80.473146	6.425045	94.867624
75%	84.250000	68.000000	49.000000	28.561654	89.948771	6.923643	124.267508
max	140.000000	145.000000	205.000000	43.675493	99.981876	9.935091	298.560117

Fig 11: Describing the dataset

```
#Grouping the parameters according to mean values of Label
df.groupby(['label']).mean()
```

label	Amount of Nitrogen	Amount of Phosphorus	Amount of Potassium	temperature	humidity	ph	rainfall
apple	20.80	134.22	199.89	22.630942	92.333383	5.929663	112.654779
banana	100.23	82.01	50.05	27.376798	80.358123	5.983893	104.626980
blackgram	40.02	67.47	19.24	29.973340	65.118426	7.133952	67.884151
chickpea	40.09	67.79	79.92	18.872847	16.860439	7.336957	80.058977
coconut	21.98	16.93	30.59	27.409892	94.844272	5.976562	175.686646
coffee	101.20	28.74	29.94	25.540477	58.869846	6.790308	158.066295
cotton	117.77	46.24	19.56	23.988958	79.843474	6.912675	80.398043
grapes	23.18	132.53	200.11	23.849575	81.875228	6.025937	69.611829
jute	78.40	46.86	39.99	24.958376	79.639864	6.732778	174.792798
kidneybeans	20.75	67.54	20.05	20.115085	21.605357	5.749411	105.919778
lentil	18.77	68.36	19.41	24.509052	64.804785	6.927932	45.680454
maize	77.76	48.44	19.79	22.389204	65.092249	6.245190	84.766988
mango	20.07	27.18	29.92	31.208770	50.156573	5.766373	94.704515

Fig 12 : Grouping the dataset according to labels

```
#Checking the missing values
df.isnull().sum()
```

```
Amount of Nitrogen      0
Amount of Phosphorus    0
Amount of Potassium     0
temperature             0
humidity                0
ph                      0
rainfall                0
label                   0
dtype: int64
```

Fig 13 : Checking for null values

```
from sklearn.preprocessing import LabelEncoder
labelencoder_y=LabelEncoder()
y_train=labelencoder_y.fit_transform(y_train)
```

```
y_train
array([ 0,  8,  8, ...,  2, 10, 16])
```

```
labelencoder_y=LabelEncoder()
y_test=labelencoder_y.fit_transform(y_test)
```

```
y_test
```

Fig 14 : Applying label encoding

```
from sklearn.preprocessing import StandardScaler
ss=StandardScaler()
x_train=ss.fit_transform(x_train)
x_test=ss.fit_transform(x_test)
```

```
print(x_train)
```

```
[[-0.64463185  2.00512125  2.96466347 ...  0.86301256 -0.4785779
  0.23692732]
 [ 0.81334894 -0.52889759 -0.2035231 ...  0.7381018 -0.18240887
  0.84590226]
 [ 1.05634574  0.1649409 -0.26219322 ...  0.82317986  0.82415091
  1.29625399]
 ...
 [-0.42863469  0.31577536 -0.65332736 ... -0.36751109  0.03555503
 -0.61777686]
 [-0.32063612  0.34594225 -0.4968737 ... -0.11470255  1.59909782
 -1.13413007]
 [-0.99562722 -0.95123406 -0.7706676 ...  0.880759  0.09231399
  0.1609325 ]]
```

Fig 15 : Scaling the data using Standard scaler

```
+ Code + Text Connect ▾
#Checking that crops those have unusual requirements
print("Some Interesting Patterns")
print(".....")
print("Crops that require very High Ratio of Nitrogen Content in Soil:", data[data['N'] > 120]['label'].unique())
print("Crops that require very High Ratio of Phosphorous Content in Soil:", data[data['P'] > 100]['label'].unique())
print("Crops that require very High Ratio of Potassium Content in Soil:", data[data['K'] > 200]['label'].unique())
print("Crops that require very High Rainfall:", data[data['rainfall'] > 200]['label'].unique())
print("Crops that require very Low Temperature:", data[data['temperature'] < 10]['label'].unique())
print("Crops that require very High Temperature:", data[data['temperature'] > 40]['label'].unique())
print("Crops that require very Low Humidity:", data[data['humidity'] < 20]['label'].unique())
print("Crops that require very Low pH:", data[data['ph'] < 4]['label'].unique())
print("Crops that require very High pH:", data[data['ph'] > 9]['label'].unique())

Some Interesting Patterns
.....
Crops that require very High Ratio of Nitrogen Content in Soil: ['cotton']
Crops that require very High Ratio of Phosphorous Content in Soil: ['grapes' 'apple']
Crops that require very High Ratio of Potassium Content in Soil: ['grapes' 'apple']
Crops that require very High Rainfall: ['rice' 'papaya' 'coconut']
Crops that require very Low Temperature: ['grapes']
Crops that require very High Temperature: ['grapes' 'papaya']
Crops that require very Low Humidity: ['chickpea' 'kidneybeans']
Crops that require very High pH: ['cotton']
```

Fig 16: Checking that crops that have unusual requirements


```
@interact
def summary(val=['Amount of Nitrogen','Amount of Phosphorus','Amount of Potassium','temperature',
print("crops which need more than average",val,'\n')
print(df[df[val]>df[val].mean()]['label'].unique())
print(".....")
print("crops which need less than avg",val,'\n')
print(df[df[val]<df[val].mean()]['label'].unique())
```

val ▼

crops which need more than average Amount of Nitrogen

```
['rice' 'maize' 'chickpea' 'blackgram' 'banana' 'watermelon' 'muskmelon'
'papaya' 'cotton' 'jute' 'coffee']
.....
crops which need less than avg Amount of Nitrogen
```

```
['chickpea' 'kidneybeans' 'pigeonpeas' 'mothbeans' 'mungbean' 'blackgram'
'lentil' 'pomegranate' 'mango' 'grapes' 'apple' 'orange' 'papaya'
'coconut']
```

Fig 17 : Crops having more and less avg. amount of nitrogen

```
print("Summer Crops")
print(df[(df['temperature'] > 30) & (df['humidity'] > 50)]['label'].unique())
print(".....")
print("Winter Crops")
print(df[(df['temperature'] < 20) & (df['humidity'] > 30)]['label'].unique())
print(".....")
print("Monsoon Crops")
print(df[(df['rainfall'] > 200) & (df['humidity'] > 30)]['label'].unique())
```

```
Summer Crops
['pigeonpeas' 'mothbeans' 'blackgram' 'mango' 'grapes' 'orange' 'papaya']
.....
Winter Crops
['maize' 'pigeonpeas' 'lentil' 'pomegranate' 'grapes' 'orange']
.....
Monsoon Crops
['rice' 'papaya' 'coconut']
```

Fig 18 : Describing the crops according to weather condition

3.9.2 Splitting the dataset

The agricultural dataset downloaded from Kaggle has 7 input parameters and predicts which crop would be the most suitable crop to harvest depending on the climatic and soil conditions. Splitting the dataset is a very important part of training or developing a machine learning model where the data is split into two parts that is training data and testing data and it can be in any ratio as per choice but the most appropriate ratio that is always taken into consideration is a 80-20 split which means 80 percent of data is for training and 20 percent of the data is for testing. The training set is used to fit the model and train the model using all the preprocessing steps which start with cleaning the data to encoding the data and testing the data set is solely used for making predictions and calculating the accuracy. The usual splits consist of:

1): Training Set:

- a): The machine learning model is trained using the training set.
- b): It makes up the majority of the dataset, usually between 60 and 80 percent of the entire data.
- c): To produce predictions, the model gathers relationships and patterns from this data.

2): Testing Set:

- a): The trained model's final performance and generalizability are assessed using the testing set.
 - b): It offers an objective evaluation of the model's performance using hypothetical data.
 - c): It shouldn't be utilised for hyperparameter tuning or model construction.
- Usually 10–20% of the total data make up the testing set. By splitting the data, one can minimize the effects of data discrepancies and is for a better understanding of the characteristics of the model

```
#Creating training and testing sets for results validation
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 0)
print("The Shape Of x train:", x_train.shape)
print("The Shape Of x test:", x_test.shape)
print("The Shape Of y train:", y_train.shape)
print("The Shape Of y test:", y_test.shape)
```

The Shape Of x train: (1760, 7)

The Shape Of x test: (440, 7)

The Shape Of y train: (1760,)

The Shape Of y test: (440,)

Fig 19: Splitting of the data

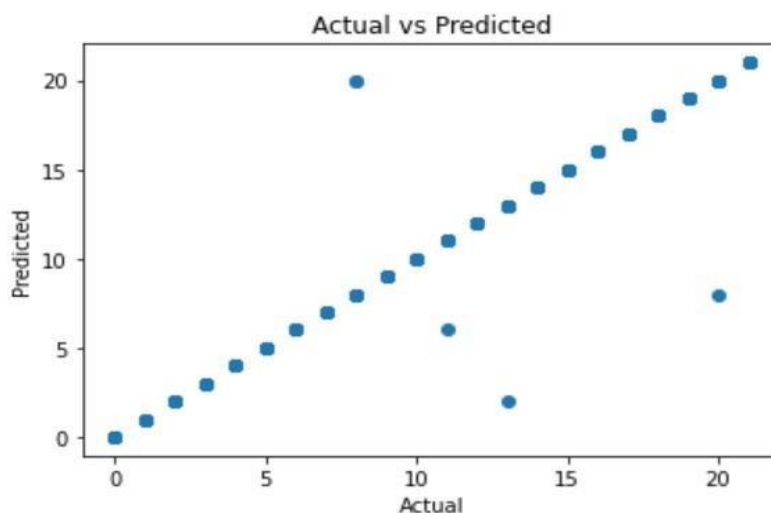
3.7.3 Training

Training the model includes applying machine learning algorithms to the training data to train and then check each algorithm's performance metrics to help determine which algorithm is best and would yield maximum results and would be most suitable to predict the crop to grow. We applied various machine learning algorithms on it like Logistic regression which predicts the outcome based on classification problems mostly predicting the binary outcome of the result, three types of logistic regression are binary, ordinal and multinomial, random forest however predicts the outcome on basis of ensemble learning that is it learns the classifier by combining multiple generated decision trees and then predicts the outcome based on the aggregation of those combined trees and is used for solving classification problems.

Naïve bayes being the probabilistic classifier predicts the outcome on the basis of conditional probability theorem known as bayes theorem and is names as naïve bayes cause in it the presence of one feature is independent from the presence of other features and bayes cause it's based on bayes theorem and is used for solving classification problems and decision tree is one of the simplest and easy algorithm to apply and works on classification problem, it is also used to predict the binary outcome of the situation presented, it is like a tree with branches as yes and no outcomes and asks questions and builds up classifier in such way which asks and solves such questions and produces the output answer as a yes or no. Among all the algorithms used and worked upon, we found that random forest gave the highest accuracy to predict the outcome (which crop would be best suited to grow).

```
import matplotlib.pyplot as plt
plt.scatter(y_test,y_pred)
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.title('Actual vs Predicted')
```

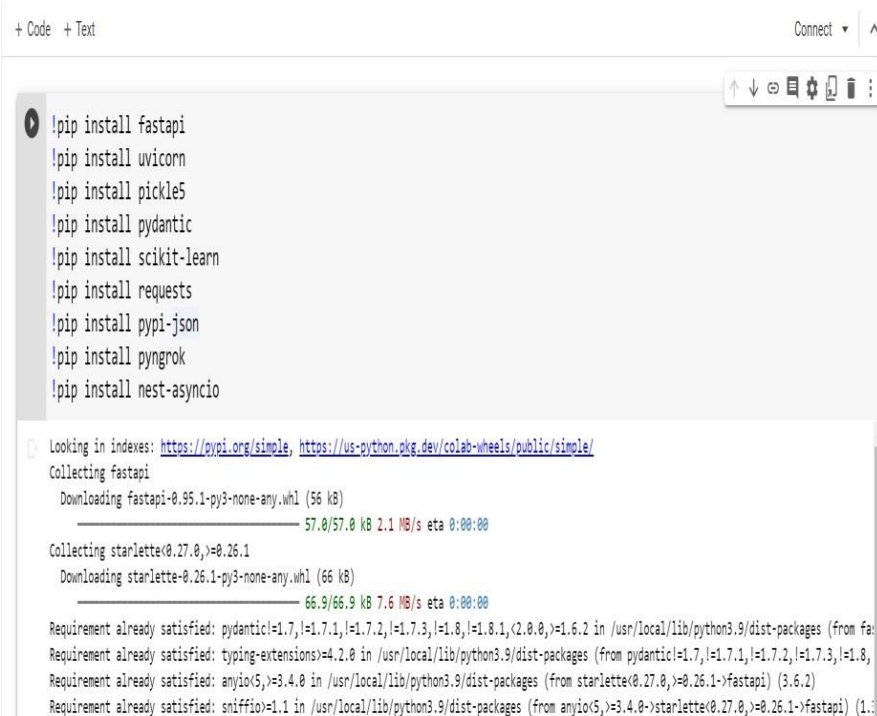
```
Text(0.5, 1.0, 'Actual vs Predicted')
```



Graph 2 : Graph of actual vs predicted values

3.9.3 Installing libraries and their uses

- **FastAPI:** This is used for creating APIs.
- **Uvicorn:** This is used to access the server.
- **Pickle:** This is used to load the saved model.
- **Pydantic:** This is used to import the base model (purpose to set up the format in which the data will be posted in our API).
- **JSON:** This is used to convert json object to dictionary.
- **Requests:** This is used to get our post values to our API.
- **Ngrok:** This is used to create proxy URLs.
- **CORSMiddleware:** This means cross origin resource sharing and is imported because if we do not follow a particular procedure to run, it could give course error when integrating with UI.
- **Nest-asyncio:** Asyncio is basically used as a foundation for multiple Python asynchronous frameworks.



```
+ Code + Text Connect ▾ ^
```

```
!pip install fastapi
!pip install uvicorn
!pip install pickle5
!pip install pydantic
!pip install scikit-learn
!pip install requests
!pip install pypi-json
!pip install pyngrok
!pip install nest-asyncio
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting fastapi
  Downloading fastapi-0.95.1-py3-none-any.whl (56 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 57.0/57.0 kB 2.1 MB/s eta 0:00:00
Collecting starlette[0.27.0]>=0.26.1
  Downloading starlette-0.26.1-py3-none-any.whl (66 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 66.9/66.9 kB 7.6 MB/s eta 0:00:00
Requirement already satisfied: pydantic[1.7,1=1.7.1,1=1.7.2,1=1.7.3,1=1.8,1=1.8.1,<2.0.0,>=1.6.2 in /usr/local/lib/python3.9/dist-packages (from fa
Requirement already satisfied: typing-extensions>=4.2.0 in /usr/local/lib/python3.9/dist-packages (from pydantic[1.7,1=1.7.1,1=1.7.2,1=1.7.3,1=1.8,
Requirement already satisfied: anyio<5,>=3.4.0 in /usr/local/lib/python3.9/dist-packages (from starlette[0.27.0,>=0.26.1->fastapi) (3.6.2)
Requirement already satisfied: sniffio=1.1 in /usr/local/lib/python3.9/dist-packages (from anyio<5,>=3.4.0->starlette[0.27.0,>=0.26.1->fastapi) (1.3
```

Fig 20 : Installing libraries for creating API

CHAPTER-4

PERFORMANCE ANALYSIS

4.1. Introduction

Performance analysis is the way of analysis of the project based on factors like accuracy, F1 score, confusion matrix, cross validation etc. obtained by various machine learning algorithms applied in the training model to train the model to predict the outcome. Evaluation of a machine learning model's effectiveness and efficiency in creating predictions or classifications is a common component of performance analysis. Depending on the problem domain, the make-up of the dataset, and the analysis's goals, several metrics and methods may be employed. For this goal, several essential metrics and methods are frequently employed. Some of the algorithms applied were logistic regression, random forest, naïve bayes and decision tree where logistic regression predicts the outcome based on classification problems mostly predicting the binary outcome of the result, three types of logistic regression are binary, ordinal, and multinomial.

Random forest however predicts the outcome based on ensemble learning that is it learns the classifier by combining multiple generated decision trees and then predicts the outcome based on the aggregation of those combined trees and is used for solving classification problems. Naïve bayes being the probabilistic classifier predicts the outcome because of conditional probability theorem known as bayes theorem and its names as naïve bayes cause in it the presence of one feature is independent from the presence of other features and bayes because it is based on bayes theorem and is used for solving classification problems. Decision tree is one of the simplest and easiest algorithms to apply and works on classification problem, it is also used to predict the binary outcome of the situation presented, it is like a tree with

branches as yes and no outcomes and asks questions and builds up classifier in such a way which asks and solves such questions and produces the output answer as a yes or no.

Among all the algorithms used and worked upon, we found that random forest gave the highest accuracy to predict the outcome (which crop would be best suited to grow). Random forest predicts the outcome based on ensemble learning that is it learns the classifier by combining multiple generated decision trees and then predicts the outcome based on the aggregation of those combined trees and is used to solve classification problems and the output of this is the classifier that is highest times selected.

It helps to know which algorithm predicts the best outcome and which algorithm has the highest accuracy and is best in a particular scenario to predict which crop would be the most suitable crop to grow and would increase productivity. It is very important to know as it is the only important part of the whole project to analyze and look at which algorithm would work best and could be used further to predict the result and whose predicted result must be considered.

The SWOT analysis, on the other hand, is a tool for examining the overall strengths, weakness, opportunity, and threat of the project made. It helps identify the person what all advantages and disadvantages the project holds and then implement on the project further. An organization's internal strengths and weaknesses, as well as external opportunities and threats, can be determined with the aid of a SWOT analysis, a strategic planning framework. It offers a methodical way to assess the circumstances at hand and come to wise conclusions. It is a strategic planning technique which provides assessment tools to identify the core strengths, weakness, opportunity, and threats of the computer.

4.1. SWOT Analysis

Table 2: SWOT analysis

STRENGTH	WEAKNESS
<ul style="list-style-type: none">▫ Increases the productivity of agriculture.▫ Quickly predicts which crop would be most suitable to grow.▫ Can also predict with difficult input parameters.	<ul style="list-style-type: none">▫ Cannot predict the amount of land required to harvest the crops.▫ Python is used to build the model which already has inbuilt libraries.
OPPORTUNITY	THREAT
<ul style="list-style-type: none">▫ Would increase the economy of the country along with the overall income of farmers working hard to grow the crops.▫ The increased productivity in agriculture would lead to less wastage of agricultural land and hence less soil pollution.	<ul style="list-style-type: none">▫ If in case the predicted crop would not work or the unfavorable weather at unfavorable time would destroy the harvested crops or could not increase the productivity.

4.2. PESTLE Analysis

4.3.1 Political

Politics rarely has any control over this enterprise. It would affect at every sector of the field.

4.3.2 Economical

Since the overall GDP and economy depends on the agriculture, the project would benefit economically a lot by increasing the overall economy of country by increasing the productivity of agriculture.

4.3.3 Social

Socially, it would be quite beneficial for the people of the country, since basic need for living is food which would automatically be affected by the increase in the productivity of agriculture and would help people a lot, since need and demand would day by day increase with the increase in population day by day.

4.3.4 Technological

Technically, it would be a boon for researchers and data scientist to develop such a model which would be a profitable not only for the country but for every individual living.

4.3.5 Legal

Legally, it would lay no grounds to be proven as disadvantage for the country or its people, it would directly or indirectly would affect in the increase of the economy as well as the overall income for farmers.

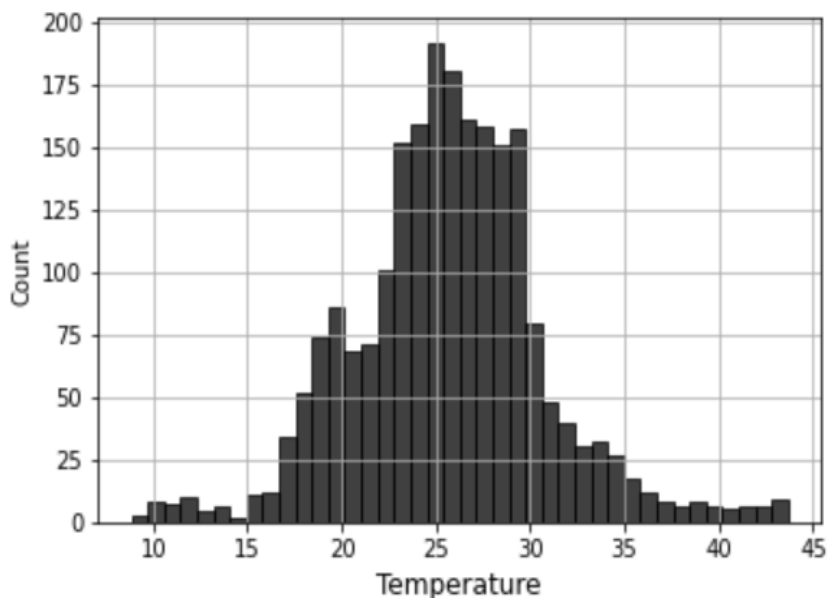
4.3.6 Environmental

The project would be environmentally friendly as it would decrease soil pollution by preventing the agricultural land wastage caused due to crops destroyed due to unfavorable conditions.

4.3. Correlation of Independent variables with dependent variables

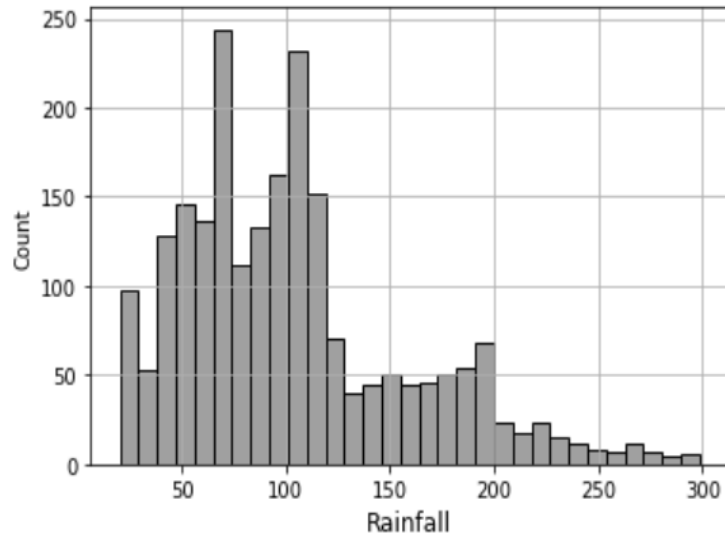
The dataset used which was downloaded from Kaggle had features such as amount of nitrogen, amount of phosphorus, amount of potassium, ph value of soil, temperature, humidity, and rainfall and these all were used to predict which crop would be the best suitable to grow and would yield maximum productivity and the algorithms used to predict the outcome that is which crop would be most suitable to grow are logistic regression, random forest, naïve bayes and decision tree.

```
#plt.subplot(2,4,4)
sns.histplot(df['temperature'], color="black")
plt.xlabel('Temperature', fontsize = 12)
plt.grid()
```



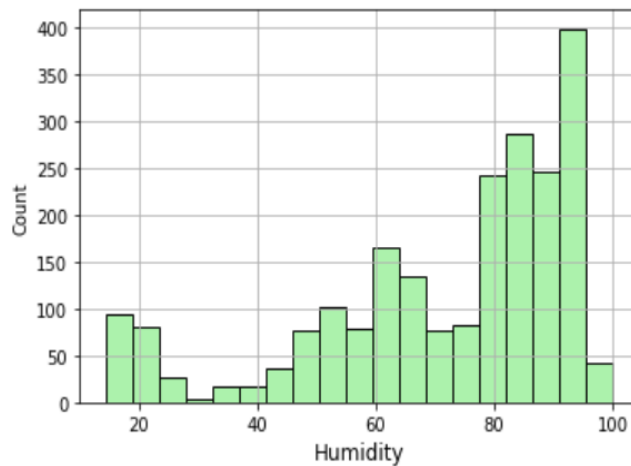
Graph 3 : Histogram showing correlation of temperature wrt count of label

```
#plt.subplot(2,4,5)
sns.histplot(df['rainfall'], color="grey")
plt.xlabel('Rainfall', fontsize = 12)
plt.grid()
```



Graph 4 : Histogram showing correlation of rainfall wrt count of label

```
#plt.subplot(2,4,6)
sns.histplot(df['humidity'], color="lightgreen")
plt.xlabel('Humidity', fontsize = 12)
plt.grid()
```



Graph 5 : Histogram showing correlation of humidity wrt count of label

4.5 Predictive model

We created a predictive model using logistic regression and generated confusion matrix and classification report and then according to the provided input parameters passed as an array, we predicted which crop would be the most suitable crop to grow.



Fig 21: Confusion matrix

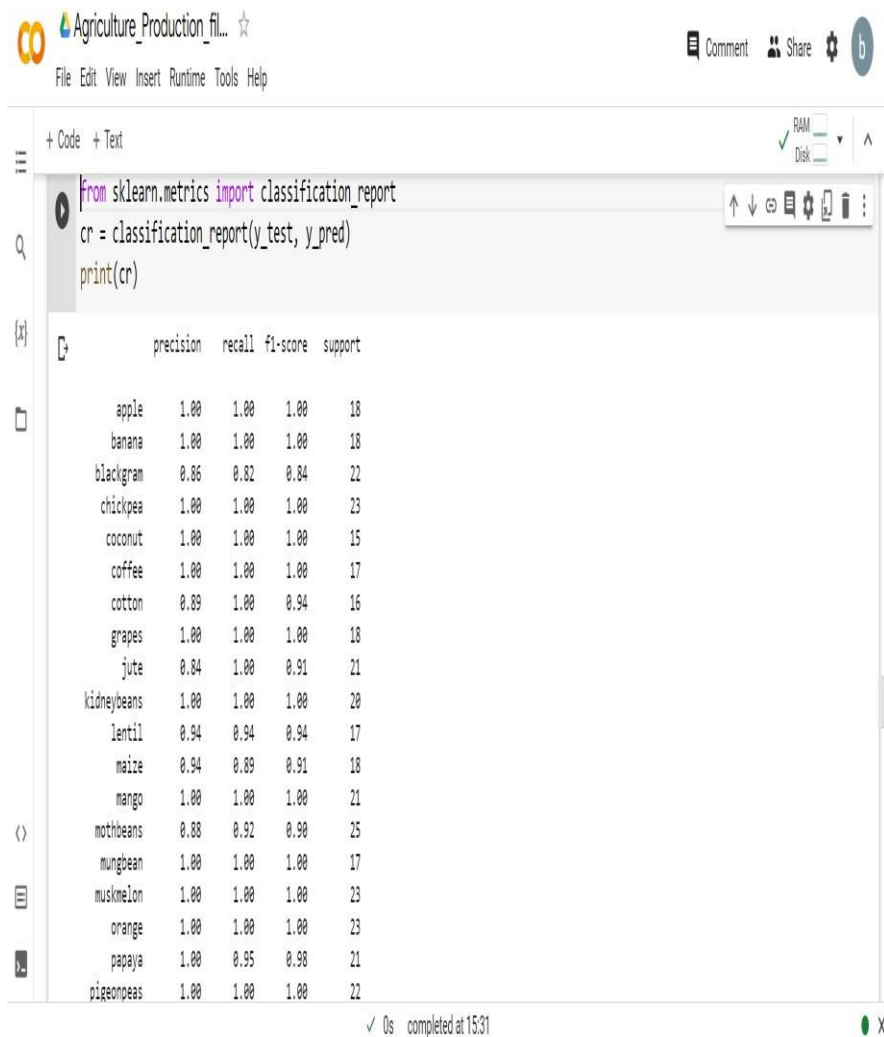


Fig 22 : Classification report

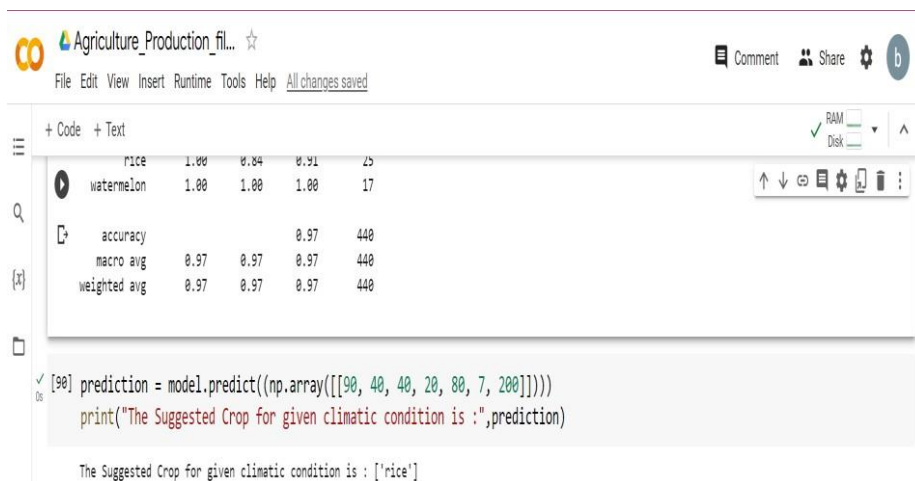


Fig 23: Predicted crop based on input parameters

```

+ Code + Text
allow_headers=["*"],
[4] )

class model_input(BaseModel):
    N : int
    P : int
    K : int
    temperature : float
    humidity : float
    ph : float
    rainfall : float

[6] agriculture_model=pickle.load(open('data.sav', 'rb'))

```

Fig 24: Importing base model and saved model

```

#loading the saved model
agriculture_model=pickle.load(open('data.sav', 'rb'))

@app.post('/agriculture_production')
def agriculture_prod(input_parameters : model_input):

    input_data = input_parameters.json()
    input_dictionary = json.loads(input_data)

    N = input_dictionary['N']
    K = input_dictionary['K']
    P = input_dictionary['P']
    temperature = input_dictionary['temperature']
    rainfall = input_dictionary['rainfall']
    humidity = input_dictionary['humidity']
    ph = input_dictionary['ph']

    input_list = [N,K,P,temperature,rainfall,humidity,ph]

    prediction = agriculture_model.predict(input_list)

    if (prediction[0] == 0):
        return 'There is no suitable/available crop to grow'
    else:
        return 'There is a suitable crop to grow for sure'

```

Fig 25: API Function with passing input parameters

The algorithms applied on the machine learning model were naïve bayes, decision tree, random forest, and logistic regression. The performance of every algorithm is:

Performance-Random Forest

```
from sklearn.ensemble import RandomForestClassifier
clf = RandomForestClassifier(n_estimators = 100)
clf.fit(x_train, y_train)
y_pred = clf.predict(x_test)
print("The accuracy of Random Forest is: ", metrics.accuracy_score(y_test, y_pred)*100)
print('The precision using Random Forest is:', metrics.precision_score(y_pred, y_test, pos_label=1))
print('The f-score using Random Forest is:', metrics.f1_score(y_pred, y_test, pos_label='pos'))
print('The recall using Random Forest is:', metrics.recall_score(y_pred, y_test, pos_label='pos'))
```

The accuracy of Random Forest is: 98.86363636363636

The precision using Random Forest is: 98.98556998556998

The f-score using Random Forest is: 98.91384625475536

The recall using Random Forest is: 98.86363636363636

Fig 26 : Performance metrics of Random forest algorithm

Performance-Naive Bayes

```
from sklearn.naive_bayes import GaussianNB
NBClassifier = GaussianNB()
NBClassifier.fit(x_train,y_train)
y_pred=NBClassifier.predict(x_test)
print('The accuracy of Naive Bayes is: ',metrics.accuracy_score(y_pred,y_test)*100)
print('The precision using Naive Bayes is:',metrics.precision_score(y_pred,y_test,pos_
print('The f-score using Naive Bayes is:',metrics.f1_score(y_pred,y_test,pos_label='p
print('The recall using Naive Bayes is:',metrics.recall_score(y_pred,y_test,pos_label:
```

```
The accuracy of Naive Bayes is: 99.0909090909091
The precision using Naive Bayes is: 99.20202020202021
The f-score using Naive Bayes is: 99.11374880530725
The recall using Naive Bayes is: 99.0909090909091
```

Fig 27: Performance metrics of Naïve bayes algorithm

Performance-Decision Tree Classifier

```
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
DTClassifier = DecisionTreeClassifier(criterion='entropy',random_state=0)
DTClassifier.fit(x_train,y_train)
y_pred=DTClassifier.predict(x_test)
print('The accuracy of decision tree is: ',metrics.accuracy_score(y_pred,y
print('The precision using decision tree is:',metrics.precision_score(y_pr
print('The f-score using decision tree is:',metrics.f1_score(y_pred,y_test
print('The recall using decision tree is:',metrics.recall_score(y_pred,y_t
```

```
The accuracy of decision tree is: 98.18181818181819
The precision using decision tree is: 98.23436041083102
The f-score using decision tree is: 98.189174442057
The recall using decision tree is: 98.18181818181819
```

Fig 28: Performance metrics of Decision tree algorithm

Performance-Logistic Regression Classifier

```
#Creating a Predictive Model
from sklearn.linear_model import LogisticRegression
from sklearn import metrics
model = LogisticRegression()
model.fit(x_train, y_train)
y_pred = model.predict(x_test)
print('The accuracy of Logistic Regression is: ',metrics.accuracy_score(y_pred,y_test)*100)
print('The precision using Logistic Regression is:',metrics.precision_score(y_pred,y_test,po
print('The f-score using Logistic Regression is:',metrics.f1_score(y_pred,y_test,pos_label='
print('The recall using Logistic Regression is:',metrics.recall_score(y_pred,y_test,pos_labe
```

The accuracy of Logistic Regression is: 96.5909090909091

The precision using Logistic Regression is: 96.97281447014066

The f-score using Logistic Regression is: 96.79464548664818

The recall using Logistic Regression is: 96.5909090909091

Fig 29: Performance metrics of Logistic regression algorithm

4.6 Cross Validation Score

Cross Validation Score is a methodological error which is learning to evaluate on same set of data and parameters of a prediction function. It is a model that repeats the labels of the sample which it has just seen so that it can result to score well but it cannot able to make any predictions about data that has not yet been seen. This can lead to the cause of overfitting. This issue can be resolved by making some of the further portion of the dataset a "validation set": in which training is conducted on the training set, which is followed to evaluate on the validation set, and when it appears that the experiment has been successful, we need to finally evaluate on the test set. However, by dividing the dataset into three sets, we are just cutting down the number of observations that can be used to train the model, and the outcomes can vary because they may be in randomized pair of sets. Cross-validation is an approach that can be used to resolve this issue. When we are doing cross validation, the validation set is no longer required, but a test set should still be kept for the final assessment. There is a known strategy, known as K-fold Cross Validation, which divides the training set into smaller sets. Every single one of the k folds is done as follows: The portion which remains of the data is utilized to the building model which it has been trained using training data from k number of folds.

4.7 F1 Score

evaluation of a test's accuracy is the F1 score. In order to calculate the test's score, precision and recall are considered. Precision is used to calculate the dividing in the number of correctly positive or negative results by the total number of positive or negative results which returns to the classifier, and the recall is calculated by dividing the no. of correct positive or negative results based on the total number of significant samples that are being classified as positive. The F1 score is the average of precision and recall, with a maximum value of 1 (perfect precision and recall) and a minimum value of 0. We can also say that the F1 score is the weighted average of precision and recall, in which the best number being 1 and the worst number is 0. Precision and recall are contributing equally to find the percentage.

4.8 Recall

It is the performance measure used in pattern recognition and classification problems. It is used to tell which algorithm is highly accurate and best to predict the outcome. It is the ability of the model to depict all the possible correct cases in the data. It is basically the ratio of positive cases that have been classified positively to total positive cases. It measures the ability of the model to predict positive outcomes.

Recall = True Positive/True Positive + False Negative that is : $\text{Recall} = \frac{TP}{TP+FN}$

It is independent of the number of the negative samples and if the model has classified all the samples correctly as positive then recall = 1.

4.9 Precision

It is known how close the two objects are to each other and that is how close the results are to each other. It is independent of accuracy. It is a performance measure used in pattern recognition and classification problems. It is referred to as the ratio of all correctly classified positive samples (True Positive) to the total number of classified positive samples (either correctly or incorrectly).

Precision = True Positive/True Positive + False Positive that is : $\text{Precision} = \frac{TP}{TP+FP}$

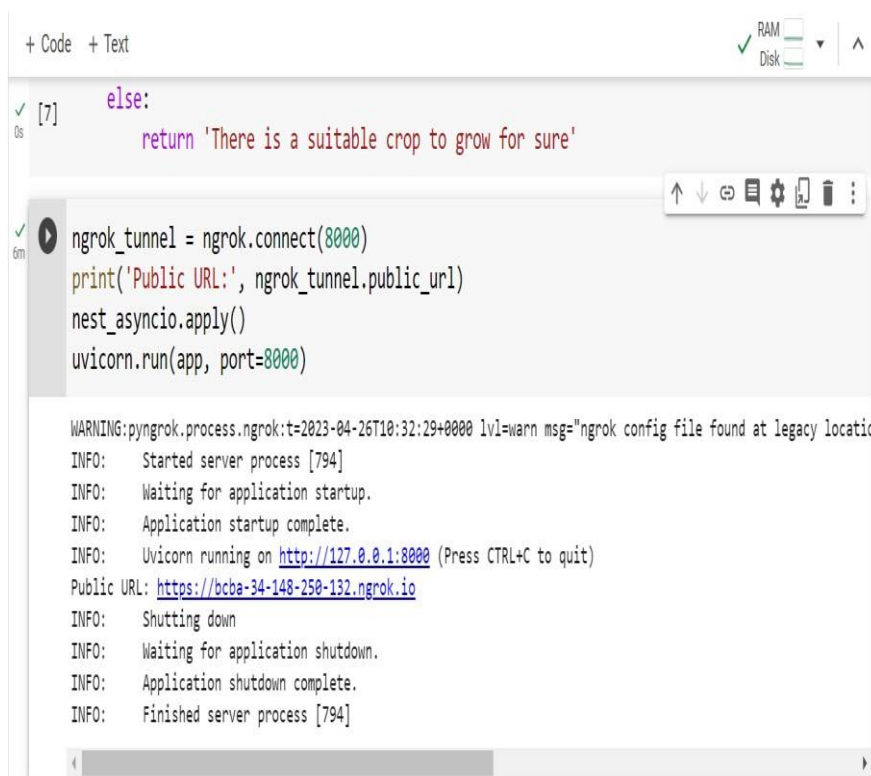
Precision helps us to basically visualize the reality or how reliable the machine learning model is to predict the model as positive.

4.10 Results

The algorithms used to predict which crop would be the most suitable to grow depending on various climatic conditions are logistic regression, random forest, naïve bayes and decision tree and the highest accuracy is found to be of random forest to predict the outcome. To find the result, every classifier has been trained using the training set and the outcome is being predicted using the testing set. The accuracy score of each classifier has been used to assess its performance. Accuracy measure of random forest was 98.5 percent, logistic regression was 95 percent, decision tree was 98.1 percent and naïve bayes was the highest that is 99 percent.

After building the machine learning model, we have deployed it as a public API using FastAPI and ngrok in google colab, the public API is basically what anyone can access on the internet through URL, anyone can post input parameters to the API in order to find out whether a suitable crop exists or not. The result will be a URL and with that we have mentioned the endpoint in order to create a public URL.

After connecting the ngrok with the port number would be an URL and with that we have to mention the endpoint in order to create public URL, we deployed in localhost so others cannot access it only we can save in order to rectify ngrok which basically created proxy URL for our localhost, so running the code would give us public URL and that port number with it is in which our API would be hosted and since our API would only receive json data so we need to convert dictionary which was used for storing list to json data by post method.



```
+ Code + Text RAM  Disk   
[7] else:  
    return 'There is a suitable crop to grow for sure'  
  
ngrok_tunnel = ngrok.connect(8000)  
print('Public URL:', ngrok_tunnel.public_url)  
nest_asyncio.apply()  
uvicorn.run(app, port=8000)  
  
WARNING:pyngrok.process.ngrok:t=2023-04-26T10:32:29+0000 lvl=warn msg="ngrok config file found at legacy locatio  
INFO: Started server process [794]  
INFO: Waiting for application startup.  
INFO: Application startup complete.  
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)  
Public URL: https://bcba-34-148-250-132.ngrok.io  
INFO: Shutting down  
INFO: Waiting for application shutdown.  
INFO: Application shutdown complete.  
INFO: Finished server process [794]
```

Fig 30: Connecting ngrok with port number

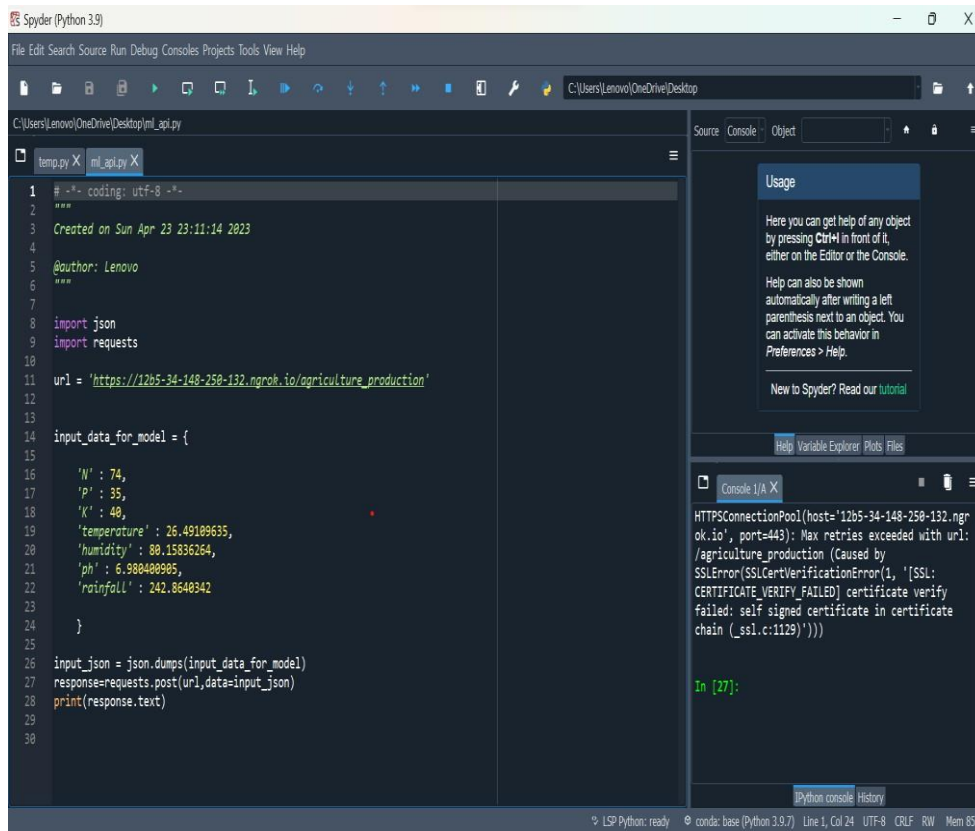


Fig 31: Python file in spyder notebook with public URL passed



Fig 32: Predicting which crop would be most suitable

Table 3: Showing Results obtained from all algorithms

AI	Logistic Regression	Decision Tree	Naïve Bayes	Random Forest
Accuracy	96.59	98.18	99.09	98.86
Precision	96.97	98.23	99.20	98.98
F-score	96.79	98.18	99.11	98.91
Recall	96.59	98.18	99.09	98.86

CHAPTER-5

CONCLUSIONS

5.1 Conclusion

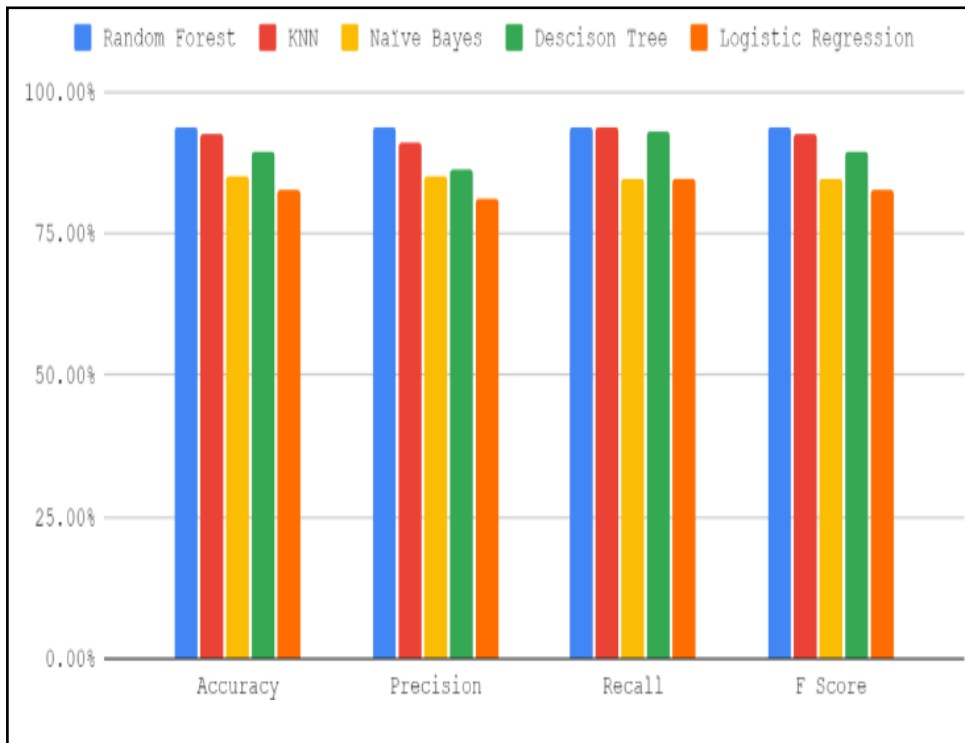
The backbone of our country is agriculture and being an agricultural country, it affects the overall economy of the country. Combining the agricultural sector of our country with information technology and its learning would result in a great boon in the agricultural sector. Machine learning-being the most advanced and budding field nowadays is part of artificial intelligence proves to be a game changer in the field of the agricultural sector, it has outgrown itself with bigdata technologies and high-performance computing to create new opportunities in the multi-disciplinary agrotechnology domain. The accuracy and representativeness of the training data are critical factors in machine learning models. The effectiveness of the model could be hampered by biased, insufficient, or inaccurate training data that is not representative of the target population. The model can only learn from patterns seen in the training distribution; it may find it difficult to generalise successfully to data that it hasn't seen before. To solve the basic problem of agricultural production and maximize the productivity which gets reduced due to lack of knowledge of farmers which crop to harvest and unfavorable and unpredictable weather changes and soil conditions pose a great threat to the crops grown and could also increase the soil pollution.

Thus we are building a machine learning model which could be of a great help not only for farmers but for country too ,it could affect in the overall increase of the economy, would increase the income of the farmers too and would be beneficial environmentally too since would help utilize the land properly and reduce soil pollution by predicting beforehand which crop would be the most suitable crop to grow depending on the various input parameters such as amount of nitrogen, amount of phosphorus,

amount of potassium, ph. value of soil, rainfall, temperature and humidity levels.

The algorithms used for this proposed computer are logistic regression, random forest, decision tree and naïve bayes and among all of them we found that random forest gave the highest accuracy to predict the outcome (which crop would be best suited to grow). Random forest predicts the outcome based on ensemble learning that is it learns the classifier by combining multiple generated decision trees and then predicts the outcome based on the aggregation of those combined trees and is used to solve classification problems and the output of this is the classifier that is highest times selected. The public API created after deploying the machine learning model in google colab using ngrok as a final result gave a URL and with that we have to mention the endpoint and that public URL with the port in which our API would be hosted locally since when we deploy in localhost, then others cannot access it, only we can save in order to rectify ngrok.

Then the prediction function created in which we passed the input parameters as array would predict which crop would be the most suitable crop to grow based on those parameters, the python files in spyder notebook were created to test the API. The various libraries imported in order to create the API were pickle to load the saved model, then FastAPI to create an API, pydantic library to import the base model which would have our input parameters, json to convert json object to dictionary and json.dumps to convert dictionary to json object, requests for posting the URLs, uvicorn to access the server, CorsMiddleware which means cross origin resource sharing to avoid course error when we would integrate with UI, if we don't follow a particular procedure to run. This prediction would help farmers to increase the productivity, growth and life of the plants/crops and would reduce labor and soil pollution. It would be helpful for people in all sectors whether government, farmers, or people of our country.



Graph 6 : Graph showing performance metrics of all algorithm

5.2 Future Scope

Future work would be further more analysis on the data and would be completely devoted in further more improving the efficiency of the project by trying more algorithms to work on. It can also be used to predict the pesticide ratio or the ratio of N-P-K to predict better based on atmospheric and soil conditions given by the farmers to grow the crop on land. We could also consider the income factor for farmers which they would gain. For future work, more parameters could also be considered like wind speed etc. in dataset.

REFERENCES

[A] Journals

- [1] M. Suganya, Dayana R, Revathi.R, “Crop Yield Prediction Using Supervised Learning Techniques, ” IJCET [2020]

- [2] Mrugank Gandhi, Shubham Kothavade, “Agricultural Production Optimization Engine,” IRJMETS [2022]

- [3] Yogita Masare, Sneha Mahale, Manjusha Kele, Ashvin Upadhyay, Bhushan R. Nanwalkar, “The System for Maximize the Yielding Rate of Crops using Machine Learning Algorithm,” IJERT [2021]

- [4] Vrushali C. Waikar, Sheetal Y. Thorat, Ashlesha A. Ghute, Priya P. Rajput, Mahesh S. Shinde, “Crop Prediction based on Soil Classification using Machine Learning with Classifier Ensembling,” IRJET [2020]

- [5] Ramesh D, Vishnu B. Vardhan, “Data Mining Techniques and Applications to Agricultural Yield Data,” IJARCEE [2013]

- [6] Shubham Prabhu, Prem Revandekar, Swami Shirdhankar, “Soil Analysis and Crop Prediction,” IJSRST [2020]

- [7] Zeel Doshi, Rashi Agrawal, Subhash Nadkarni, Prof. Neepa Shah, "Agro Consultant: Intelligent Crop Recommendation System Using Machine Learning Algorithms," Fourth International Conference on Computing Communication Control and Automation (ICCUBEA) [2018]

[8] Jain, Sonal, and Dharavath Ramesh. "Machine Learning convergence for weather-based crop selection." IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS) [2020].

[9] Suresh, G., A. Senthil Kumar, S. Lekashri, and R. Manikandan. "Efficient Crop Yield Recommendation System Using Machine Learning For Digital Farming." International Journal of Modern Agriculture 10 [2021]

[B] Datasets

Soumya Sourav, "Optimizing Agricultural Production." (Aug, 2020).

Kaggle.<https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-dataset>