A

Project Report

On

CROP RECOMMENDATION SYSTEM

Submitted to

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE AND TECHNOLOGIES
RK VALLEY

in partial fulfilment of the requirement for the award of the Degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE & ENGINEERING

Submitted by

MALLAKUNTA MANOJ KUMAR (R180615)

SABBARAPU HARSHA VARDHAN (R180993)

Under the Guidance of

Ms. P.UDAYASREE, Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

(catering the Educational Needs of Gifted Rural Youth of AP)

R.K Valley, Vempalli(M), Kadapa(Dist) – 516330

2020 - 2024

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

(**A.P.Government Act 18 of 2008)**

RGUKT-RK Valley

Vempalli, Kadapa, Andhrapradesh - 516330.

CERTIFICATE OF PROJECT COMPLETION

This is to certify that I have examined the thesis entitled “**Crop Recommendation System**” submitted by **Sabbarapu Harsha Vardhan(R180993), Mallakunta Manoj Kumar(R180615)** under our guidance and supervision for the partial fulfilment for the degree of Bachelor of Technology in computerScience and Engineering during the academic session July 2023 –Nov 2023 at RGUKT-RKVALLEY.

Project Internal Guide

Ms P Udayasree,
Asst.Prof. in Dept of CSE,
RGUKT-RK Valley.

Head of the Department

Mr. N.Satyanandaram,
Lecturer in Dept of CSE,
RGUKT-RK Valley.

Project External Guide

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

(A.P.Government Act 18 of 2008)

RGUKT-RK Valley

Vempalli, Kadapa, Andhrapradesh-516330.

DECLARATION

We, **Sabbarapu Harsha Vardhan(R180993), Mallakunta Manoj Kumar(R180615)** hereby declare that the project report entitled “**Crop Recommendation System**” done under guidance of **Ms. P.Udayasree** is submitted in partial fulfillment for the degree of Bachelor of Technology in Computer Science and Engineering during the academic session July 2023 – Nov 2023 at RGUKT-RK Valley. I also declare thatthis project is a result of our own effort and has not been copied or imitated from any source. Citations from any websites are mentionedin the references. To the best of my knowledge, the results embodiedin this dissertation work have not been submitted to any university orinstitute for the award of any degree or diploma.

Date : Mallakunta Manoj Kumar-R180615

Place : RK Valley Sabbarapu Harsha Vardhan-R180993

ACKNOWLEDGEMENT

I would like to express my deep sense of gratitude & respect to all those people behind the screen who guided, inspired and helped me crown all my efforts with success. I wish to express my gratitude to **P.Udayasree** for her valuable guidance at all stages of study, advice, constructive suggestions, supportive attitude and continuous encouragement, without which it would not be possible to complete this project.

I would also like to extend our deepest gratitude & reverence to the Director of RGUKT, RK Valley **Prof. A.V.S.S Kumara Swami Gupta** and HOD ofComputer Science and Engineering **Mr. N. Satyanandaram** for their constant support and encouragement.

Last but not least I express my gratitude to my parents for their constant source of encouragement and inspiration for me to keep my morals high.

With Sincere Regards,

Mallakunta Manoj Kumar-R180615,

Sabbarapu Harsha Vardhan-R180993.

ABSTRACT

Crop Recommendation System for agriculture is based on various input parameters.

This proposes a hybrid model for recommending crops to south Indian states by considering various attributes such as soil type, Rainfall, Groundwater level, Temperature, Fertilizers, Pesticides and season.

The recommender model is built as a hybrid model using the classifier machine learning algorithm. Based on the appropriate parameters, the system will recommend the crop. Technology based crop recommendation system for agriculture helps the farmers to increase the crop yield by recommending a suitable crop for their land with the help of geographic and the climatic parameters.

The proposed hybrid recommender model is found to be effective in recommending a suitable crop. Crop yield production value updation has a positive practical significance for guiding agricultural production and for notifying the change in market rate of crop to the farmer. The concept of this paper is to implement the crop selection method so that this method helps in solving many agriculture and farmers problems. This improves our Indian economy by maximizing the yield rate of crop production. Different types of land condition. So the quality of the crops are identified using ranking process. By this process the rate of the low quality and high quality crop is also notified. The usage of ensemble of classifiers paves a path way to make a better decision on predictions due to the usage of multiple classifiers. Further, a ranking process is applied for decision making in order to select the classifiers results. This system is used to predict the cost of crop which is yielded for further.

Index

Content **Page No**

List of Figures

CHAPTER 1

INTRODUCTION

1.1 Motivation

1.2 The world's largest industry, agriculture, is finding it harder and harder to estimate earnings as the global population is expanding at an exponential rate. Over 70% of the people in our nation, India, depend on agriculture. Recently most of the farmers are leaving farming and doing other daily wages work for their livelihood because of the losses they are making the agriculture they can't live with loses. If they know the exact yield they can produce before the harvesting then they can escape from their losses, to do that there are different approaches in the technologies to predict the yield before the harvesting. With the perfect decision making algorithm we can predict the accuracy of each crop and make them cultivate so that they can get profits. With the exact values of the soil content and the factors affecting the agriculture we predict which crop should be grown.

1.3 Objective of Project :

● Data parsing and cleansing technique is applied to make the raw data into processing data.

● The data collected is subject to machine learning system along with run time analysis makes an efficient crop value updation system.

● Ranking technique used in the project helps us to make efficient decisions.

● Creating a web application for user registrations and collection of data.

● The main objective is to obtain a better variety of crops that can be grown over the season. The proposed system would help to minimize the difficulties faced by farmers in choosing a crop and maximize the yield.

● The model predicts the crop yield by studying factors such as rainfall, temperature, area, season, soil type etc.

1.4 Features:

1.Soil and Weather Understanding:
It checks the type of soil and the local weather to know what kinds of crops grow best there.

2.Data-based Advice:
Uses a bunch of information about different crops and the area's history to suggest the best crops to plant.

3.Easy for Farmers:
Shows suggestions in a simple way, maybe on a phone or computer, so farmers can easily understand and use the recommendations**.**

4.Gets Smarter:
Learns from what works and what doesn't, so it can give better advice over time as it learns more about the area and the crops.

CHAPTER 2

REQUIREMENT ANALYSIS

2.1 REQUIREMENT SPECIFICATIONS

2.1.1 HARDWARE CONFIGURATION :

Client Side :

Ram	512 MB
Hard Disk	10 GB
Processor	1.0 GHz

Server Side:

Ram	1 GB
Hard Disk	20 GB
Processor	2.0 GHz

2.1.2 Software requirement:

Front end	1.HTML 2.CSS 3.Numpy 4.Pandas
Back end	Python,Machine Learning
Database Server	Flask
Web Browser	Firefox, Google Chrome, or any compatible
Operating System	Ubuntu, Windows
Software	Ubuntu, Windows

2.1.3 FUNCTIONAL REQUIREMENTS

Data Pre-Processing:

- Cleaning the Raw Data

Training Model:

- Train the Data with Suitable Model

Testing:

- After Training,Test the Model with Test Data

Data Entry:

- Enter N,P,K,Temperature,Humidity,Ph ,Rainfall Values

Prediction:

- Based on Data, Prediction Occurs

Fetching

- Suggesting the Best Crop

2.1.4 NON-FUNCTIONAL REQUIREMENTS

- **Usability Requirement**

  The system shall allow the users to access the system from the system using any web browsers. The system uses a web browsers as an interface. Since all users are familiar with the general usage of a website, no special training is required. The system is user friendly which makes the system easy.

- **Availability Requirement**

  The system is available 100% for the user and is used 24 hrs a day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week.

- **Efficiency Requirement**

  Mean Time to Repair (MTTR) - Even if the system fails, the system will be recovered back up within an hour or less.

- **Accuracy**

  The system should accurately provide real time information taking into consideration various concurrency issues. The system shall provide 100% access reliability.

- **Reliability Requirement**

  The system has to be 100% reliable due to the importance of data and the damages that can be caused by incorrect or incomplete dat

2.2 Technologies Used

2.2.1 Numpy:

NumPy helps with big number crunching. It's great at handling many numbers at once, doing math quickly, and organizing these numbers in a way that's easy to work with

2.2.2 Python

Python is a computer programming language often used to build websitesand software, automate tasks, and conduct data analysis. Python is a general-purpose language, meaning it can be used to create a variety of different programsand isn't specialized for any specific problems.

Features:

Easy To Learn and Readable Language. Python is extremely easy to learn

- Interpreted Language.
- Dynamically Typed Language
- Open Source and Free
- Large Standard Library
- High-Level Language
- Object Oriented Programming Language
- Large Community Support

2.2.3 Pandas

Pandas is a library in Python that helps you manage and play with tables of data. It's all about making it easy to do things with lots of rows and columns of information.

2.2.4 Flask

Flask is like a toolbox for creating web applications using Python. It helps in handling web requests, managing data, and creating web pages.

2.2.5 Scikit-learn

Scikit-learn is like a toolbox full of tools for machine learning. It provides ready-to-use algorithms and functions for various machine learning tasks, making it easier to build predictive models and perform data analysis.

2.2.6 HTML

HTML is like the skeleton of a webpage. It uses tags to define different parts, such as headings, paragraphs, images, links, etc. Think of these tags as building blocks that tell the browser how to structure the content.

2.2.7 Machine Learning:

Machine Learning (ML) is a branch of artificial intelligence (AI) that focuses on enabling computers to learn from data and make predictions or decisions without being explicitly programmed for each task. Here's a breakdown of key aspects:

Applications of Machine Learning:
Image and Speech Recognition
Recommendation Systems.
Natural Language Processing (NLP):
Finance:
Autonomous Vehicles:

Features:
Learning from Examples: Instead of telling the computer exactly what to do, we show it lots of examples so it can learn and make its own decisions.

Improving with Experience: Just like you get better at something by practicing, these systems get better when they see more examples or data.

Automating Tasks: Once they've learned, they can do things automatically, like recognizing pictures, understanding speech, or making predictions without us telling them exactly how.

Making Predictions and Decisions: They're great at guessing things or making choices based on what they've learned from the examples.

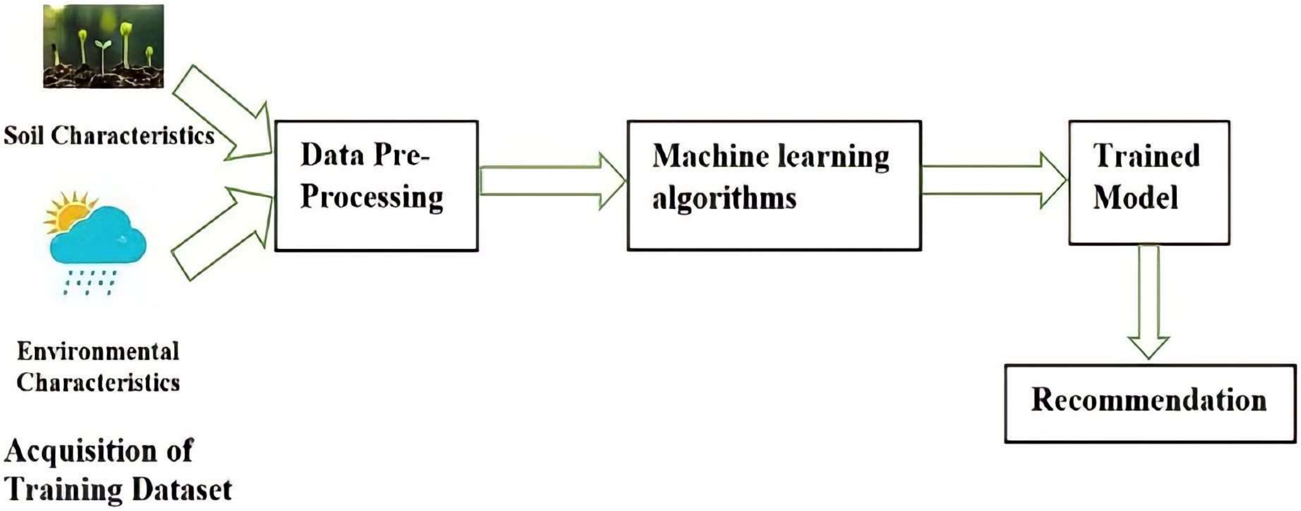
Adapting to New Situations: Even if they see something new, they can use what they've learned before to figure it out**.**

CHAPTER 3

SYSTEM ARCHITECTURE

3.1 Context Diagram

A context diagram is a visual representation that shows the system you're focusing on as a single entity, surrounded by its external interfaces. It helps to illustrate the interactions between the system and its environment without diving into the internal details. Context diagrams are commonly used in software development, systems analysis, and business process modeling to provide a high-level overview of a system's interactions with external entities.



Soil Characteristics
Environmental Characteristics
Acquisition of Training Dataset
Data Pre-Processing
Machine learning algorithms
Trained Model
Recommendation

Figure 3.1.1: Context Diagram

3.2 Use case Diagram

A use case diagram is a type of diagram used in software development and systems engineering to visualize the interactions between users (actors) and a system's functionalities (use cases).

It provides a high-level view of how users interact with the system and the specific tasks the system can perform in response.

In a use case diagram, actors are represented as stick figures, and use cases are depicted as ovals. Arrows are used to show the communication between actors and use cases, indicating which functionalities the actors can access. It helps stakeholders understand the overall behavior and requirements of the system from a user's perspective. Use case diagrams are valuable tools for requirements analysis, project planning, and communication between project teams and stakeholders.

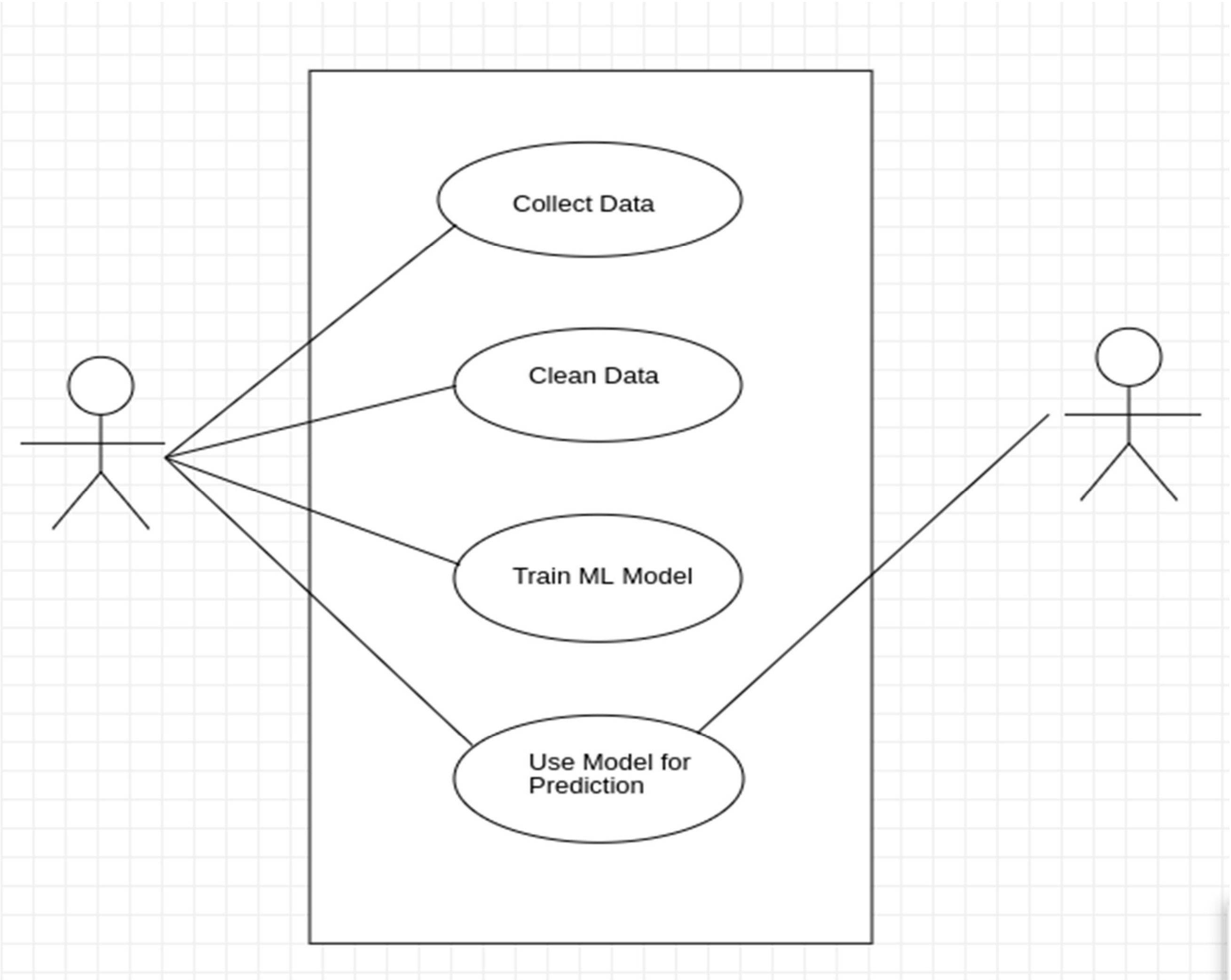


Figure 3.2.1 : Use case Diagram

3.3 Class Diagram

Class diagrams are **the blueprints of your system or subsystem**. You can use class diagrams to model the objects that make up the system, to display the relationships between the objects, and to describe what those objects do and the services that they provide. Class diagrams are useful in many stages of system design.

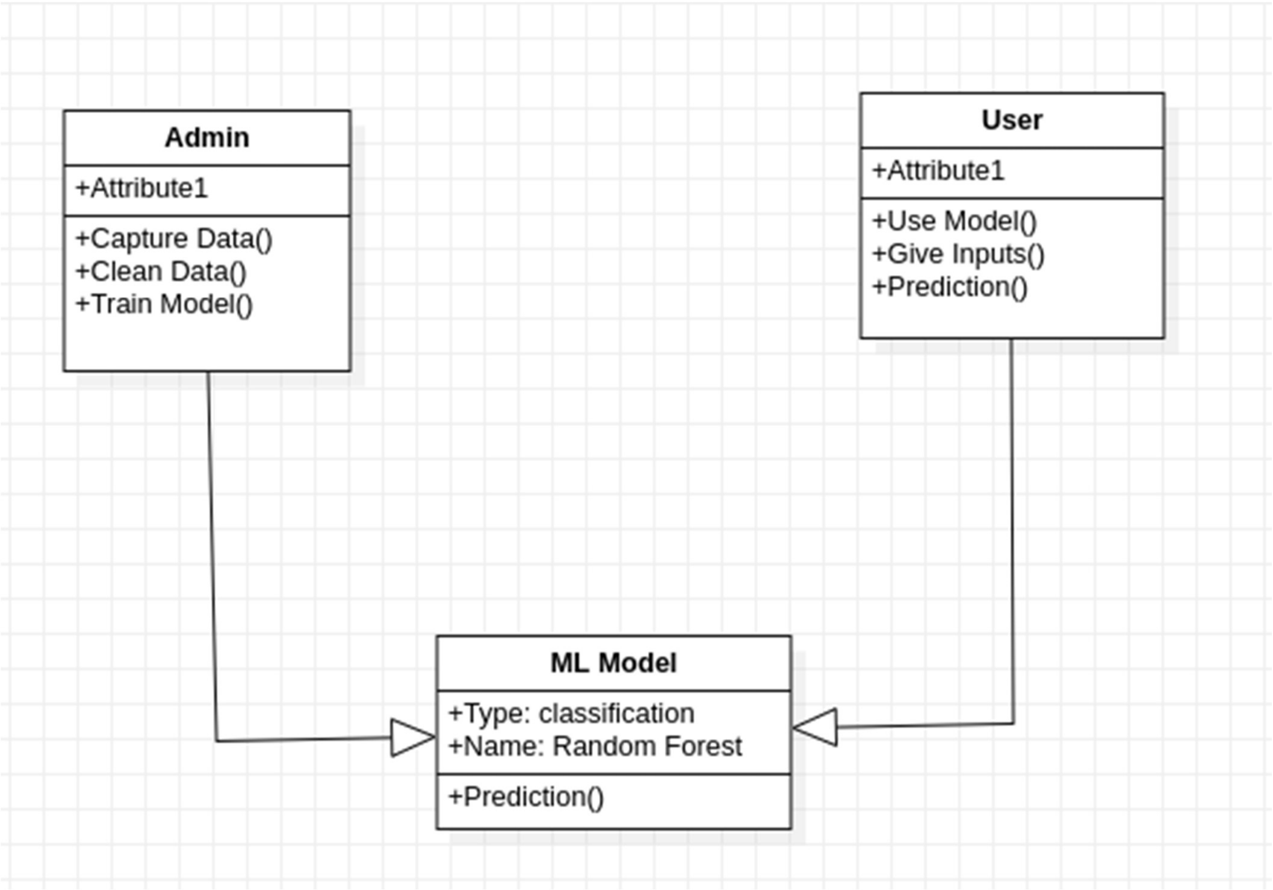


Figure 3.3.1 : Class Diagram

CHAPTER 4

SOFTWARE ENVIRONMENT

4.1 PyCharm

PyCharm is the most popular IDE used for Python scripting language. This chapter will give you an introduction to PyCharm and explains its features.

PyCharm offers some of the best features to its users and developers in the following aspects −

- Code completion and inspection
- Advanced debugging
- Support for web programming and frameworks such as Django and Flask

4.1.1 Features of PyCharm

- Besides, a developer will find PyCharm comfortable to work with because of the features mentioned below −
- PyCharm enables smoother code completion whether it is for built in or for an external package.
- You can set a breakpoint, pause in the debugger and can see the SQL representation of the user expression for SQL Language code.
- When coding in Python, queries are normal for a developer. You can check the last commit easily in PyCharm as it has the blue sections that can define the difference between the last commit and the current one.
- You can run .py files outside PyCharm Editor as well marking it as code coverage details elsewhere in the project tree, in the summary section etc.
- All the installed packages are displayed with proper visual representation. This includes list of installed packages and the ability to search and add new packages.
- Refactoring is the process of renaming one or more files at a time and PyCharm includes various shortcuts for a smooth refactoring process.

CHAPTER 5

IMPLEMENTATION

5.1 : Model View Controller :

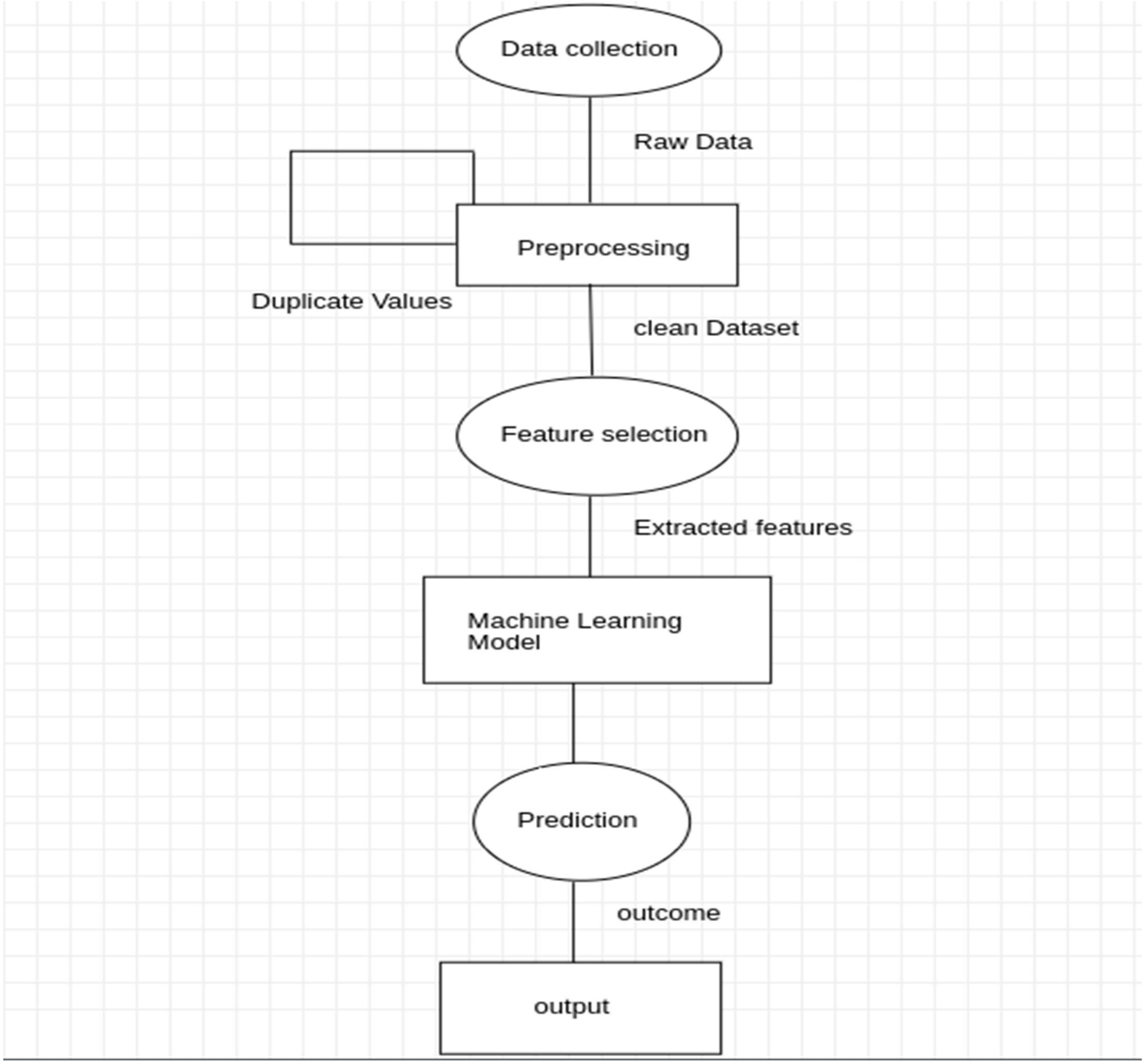


Figure 5.1.1 Model View Controller

5.2 : Project Planning:

For a successful software project, the following steps can be followed:

- Select a project
  - Identifying project's aims and objectives
  - Understanding requirements and specification
  - Methods of analysis, design and implementation
  - Testing techniques
  - Documentation.

- Project milestones and deliverables
- Budget allocation
  - Exceeding limits within control
- Project Estimates
  - Cost
  - Time
  - Size of code
  - Duration
- Resource Allocation
  - Hardware
  - Software
  - Previous relevant project information
- Risk Management
  - Risk avoidance
  - Risk detection

5.3 Detailed Design of Implementation

This phase of the systems development life cycle refines hardware and software specifications, establishes programming plans, trains users and implements extensive testing procedures, to evaluate design and operating specifications and/or provide the basis for further modification.

5.3.1 Technical Design

This activity builds upon specifications produced during new system design, adding detailed technical specifications and documentation.

5.3.2 Test Specifications and Planning

This activity prepares detailed test specifications for individual modules and programs, job streams, subsystems, and for the system as a whole.

5.3.3 Programming and Testing

This activity encompasses actual development, writing, and testing of program units or modules.

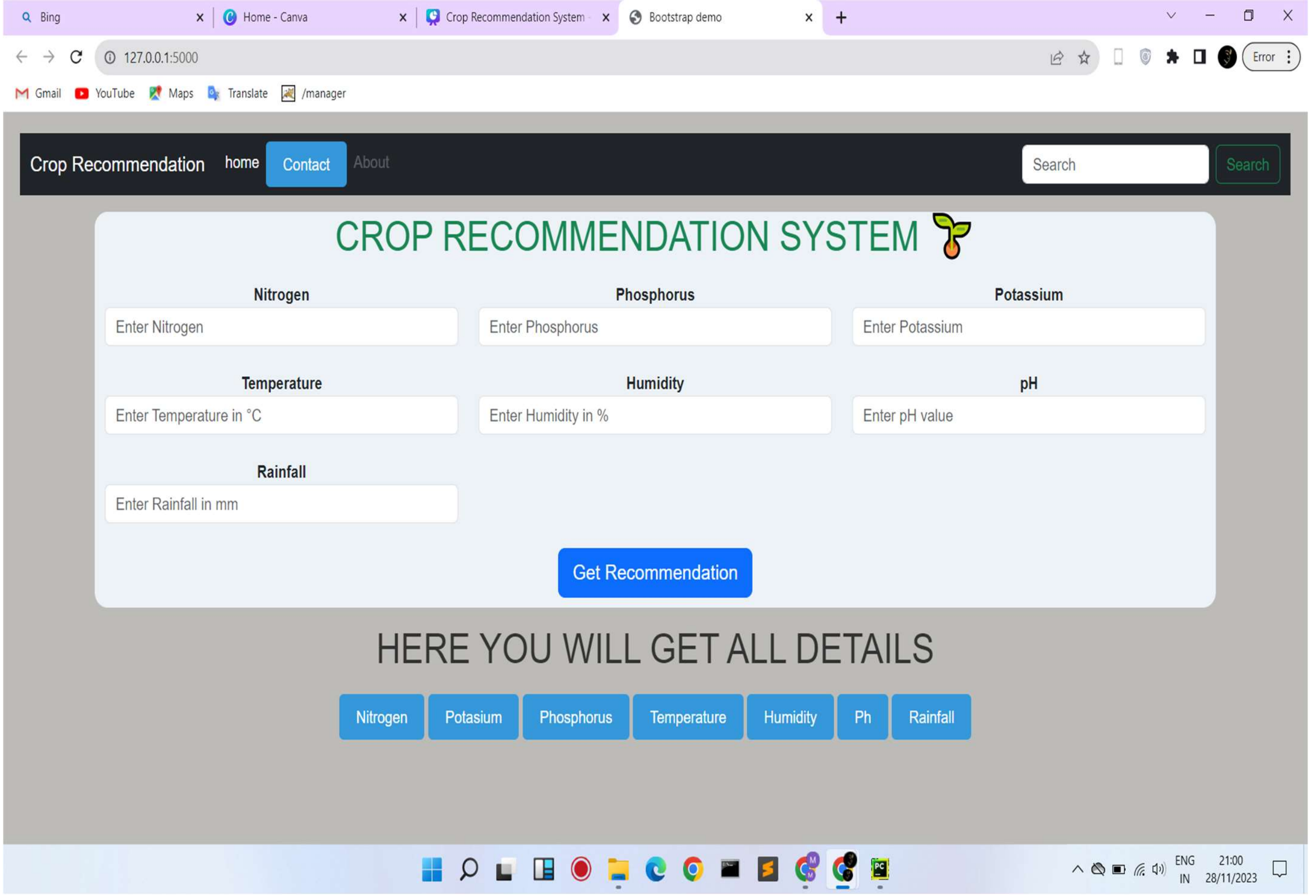
5.3.4 User Training

This activity encompasses writing user procedure manuals, preparation of user training materials, conducting training programs, and testing procedures.

5.3.5 Acceptance Test

A final procedural review to demonstrate a system and secure user approval before a system becomes operational.

5.4 Snapshots Crop Recommendation System:



CROP RECOMMENDATION SYSTEM

Nitrogen: Enter Nitrogen

Phosphorus: Enter Phosphorus

Potassium: Enter Potassium

Temperature: Enter Temperature in °C

Humidity: Enter Humidity in %

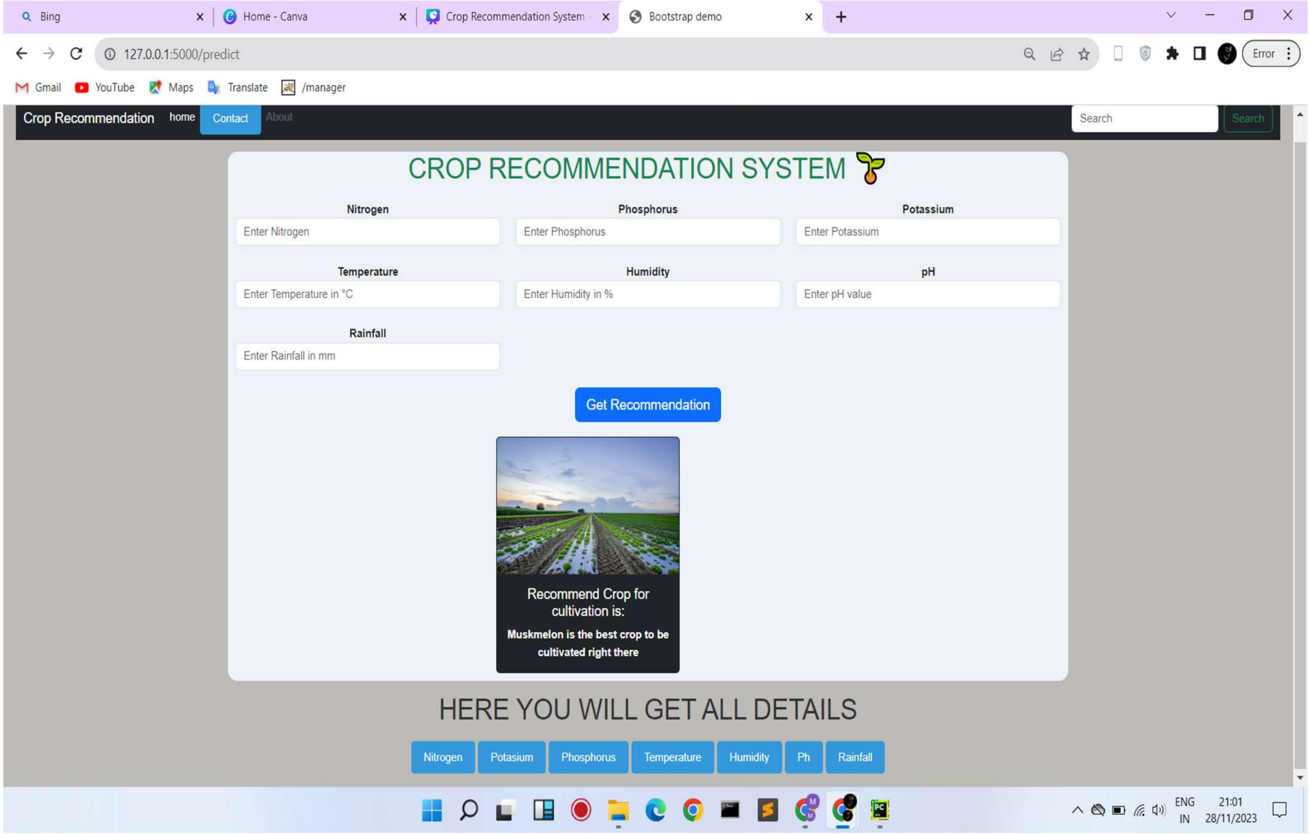
pH: Enter pH value

Rainfall: Enter Rainfall in mm

Get Recommendation

HERE YOU WILL GET ALL DETAILS

Nitrogen | Potasium | Phosphorus | Temperature | Humidity | Ph | Rainfall



CROP RECOMMENDATION SYSTEM

Nitrogen: Enter Nitrogen

Phosphorus: Enter Phosphorus

Potassium: Enter Potassium

Temperature: Enter Temperature in °C

Humidity: Enter Humidity in %

pH: Enter pH value

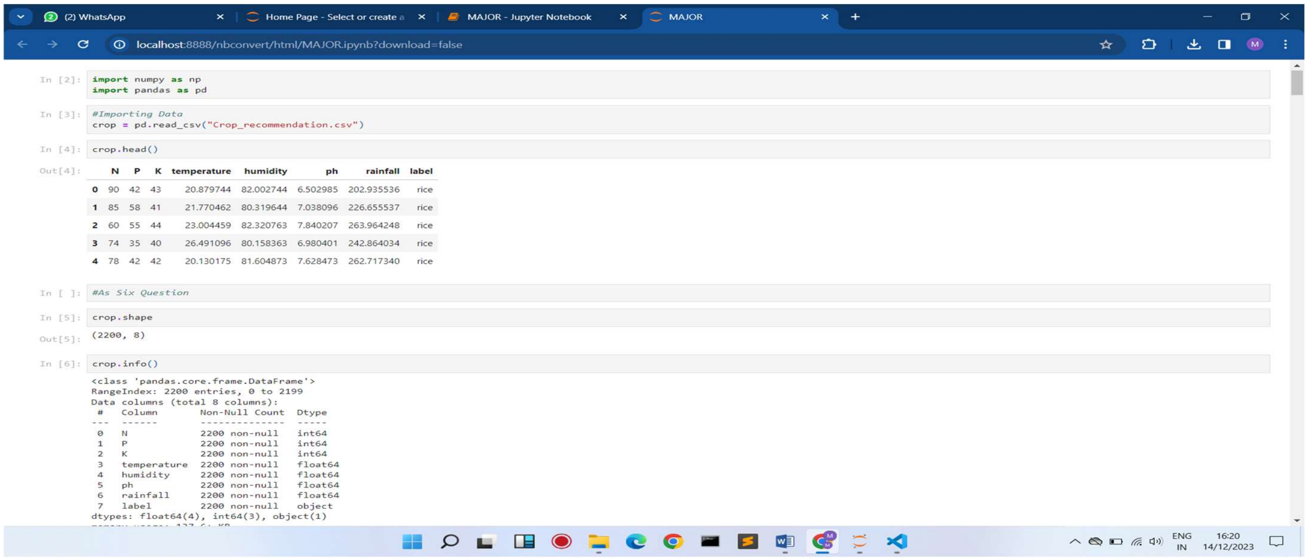
Rainfall: Enter Rainfall in mm

Recommend Crop for cultivation is:

Muskmelon is the best crop to be cultivated right there

HERE YOU WILL GET ALL DETAILS

Chapter 6
Code Implementation

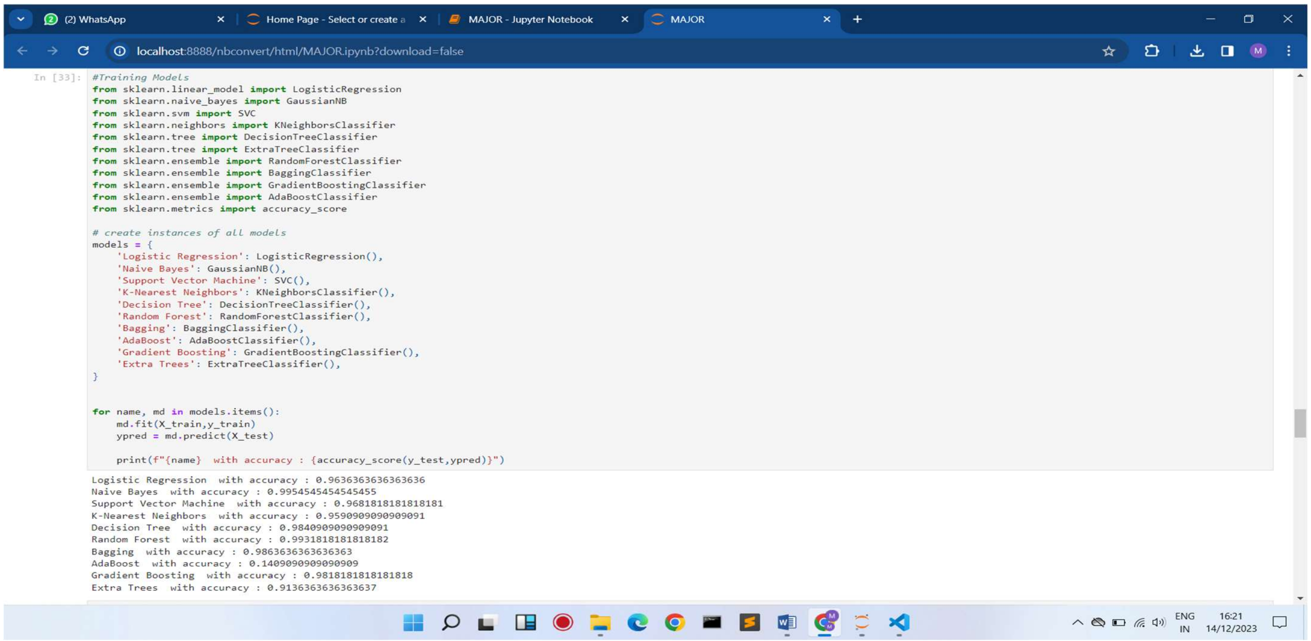

(2) WhatsApp
Home Page - Select or create a
MAJOR - Jupyter Notebook
MAJOR
localhost:8888/nbconvert/html/MAJOR.ipynb?download=false
ENG
IN
16:20
14/12/2023

```
In [2]: import numpy as np
        import pandas as pd

In [3]: #Importing Data
        crop = pd.read_csv("Crop_recommendation.csv")

In [4]: crop.head()
```

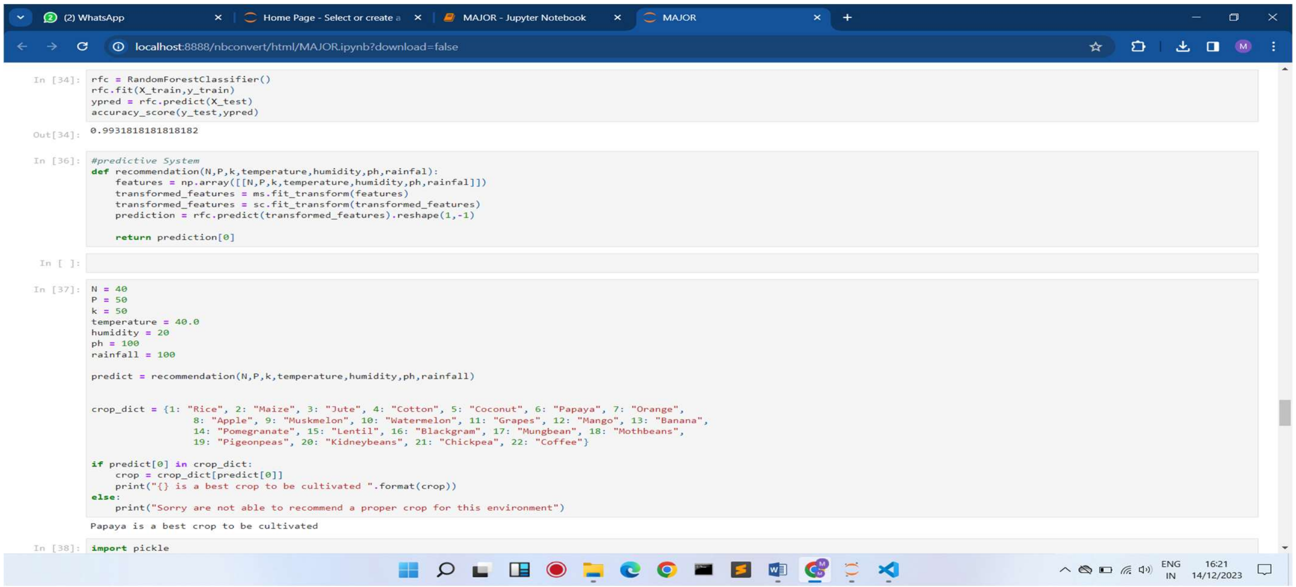
Out[4]:

	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

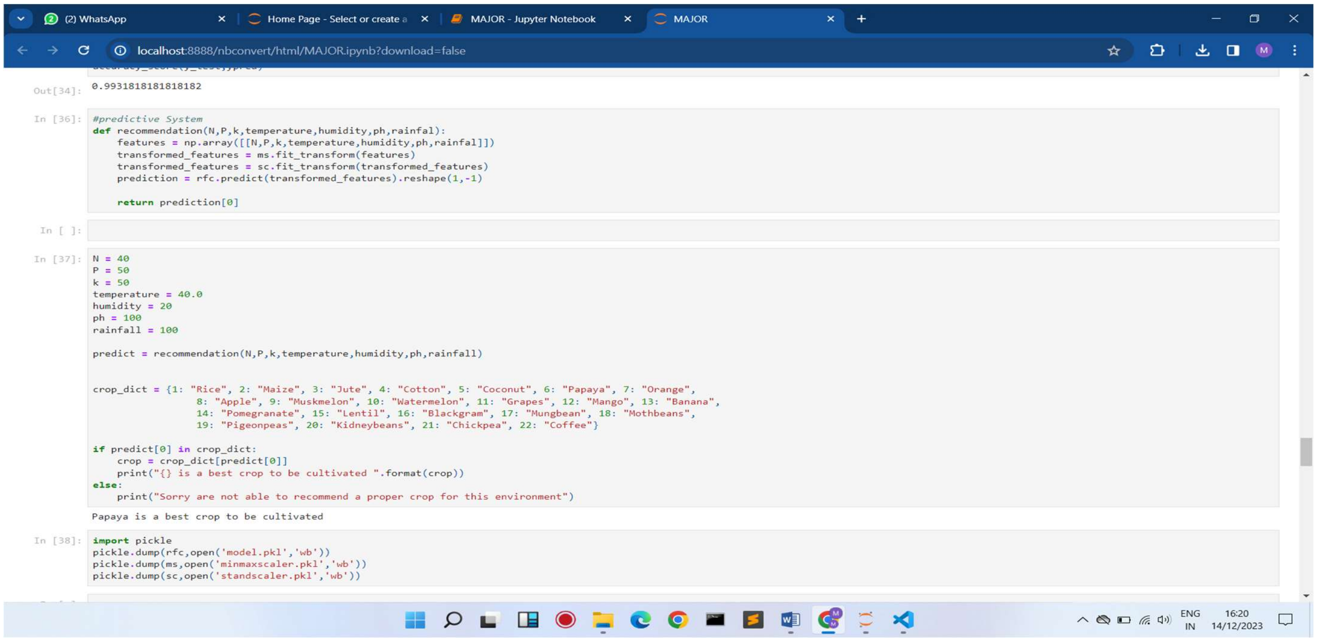
```
In [ ]: #As Six Question

In [5]: crop.shape

Out[5]: (2200, 8)

In [6]: crop.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2200 entries, 0 to 2199
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   N            2200 non-null   int64  
 1   P            2200 non-null   int64  
 2   K            2200 non-null   int64  
 3   temperature  2200 non-null   float64
 4   humidity     2200 non-null   float64
 5   ph           2200 non-null   float64
 6   rainfall     2200 non-null   float64
 7   label        2200 non-null   object 
dtypes: float64(4), int64(3), object(1)
```


(2) WhatsApp
Home Page - Select or create a
MAJOR - Jupyter Notebook
MAJOR
localhost:8888/nbconvert/html/MAJOR.ipynb?download=false
ENG
IN
16:21
14/12/2023

```
In [33]: #Training Models
         from sklearn.linear_model import LogisticRegression
         from sklearn.naive_bayes import GaussianNB
         from sklearn.svm import SVC
         from sklearn.neighbors import KNeighborsClassifier
         from sklearn.tree import DecisionTreeClassifier
         from sklearn.tree import ExtraTreeClassifier
         from sklearn.ensemble import RandomForestClassifier
         from sklearn.ensemble import BaggingClassifier
         from sklearn.ensemble import GradientBoostingClassifier
         from sklearn.ensemble import AdaBoostClassifier
         from sklearn.metrics import accuracy_score

         # create instances of all models
         models = {
             'Logistic Regression': LogisticRegression(),
             'Naive Bayes': GaussianNB(),
             'Support Vector Machine': SVC(),
             'K-Nearest Neighbors': KNeighborsClassifier(),
             'Decision Tree': DecisionTreeClassifier(),
             'Random Forest': RandomForestClassifier(),
             'Bagging': BaggingClassifier(),
             'AdaBoost': AdaBoostClassifier(),
             'Gradient Boosting': GradientBoostingClassifier(),
             'Extra Trees': ExtraTreeClassifier(),
         }


         for name, md in models.items():
             md.fit(X_train,y_train)
             ypred = md.predict(X_test)

             print(f"{name}  with accuracy : {accuracy_score(y_test,ypred)}")

         Logistic Regression  with accuracy : 0.9636363636363636
         Naive Bayes  with accuracy : 0.9954545454545455
         Support Vector Machine  with accuracy : 0.9681818181818181
         K-Nearest Neighbors  with accuracy : 0.9590909090909091
         Decision Tree  with accuracy : 0.9840909090909091
         Random Forest  with accuracy : 0.9931818181818182
         Bagging  with accuracy : 0.9863636363636363
         AdaBoost  with accuracy : 0.1409090909090909
         Gradient Boosting  with accuracy : 0.9818181818181818
         Extra Trees  with accuracy : 0.9136363636363637
```


(2) WhatsApp
Home Page - Select or create a
MAJOR - Jupyter Notebook
MAJOR
localhost:8888/nbconvert/html/MAJOR.ipynb?download=false
ENG IN
16:21 14/12/2023

```
In [34]: rfc = RandomForestClassifier()
         rfc.fit(X_train,y_train)
         ypred = rfc.predict(X_test)
         accuracy_score(y_test,ypred)

Out[34]: 0.9931818181818182

In [36]: #predictive System
         def recommendation(N,P,k,temperature,humidity,ph,rainfal):
             features = np.array([[N,P,k,temperature,humidity,ph,rainfal]])
             transformed_features = ms.fit_transform(features)
             transformed_features = sc.fit_transform(transformed_features)
             prediction = rfc.predict(transformed_features).reshape(1,-1)

             return prediction[0]

In [ ]:

In [37]: N = 40
         P = 50
         k = 50
         temperature = 40.0
         humidity = 20
         ph = 100
         rainfall = 100

         predict = recommendation(N,P,k,temperature,humidity,ph,rainfall)


         crop_dict = {1: "Rice", 2: "Maize", 3: "Jute", 4: "Cotton", 5: "Coconut", 6: "Papaya", 7: "Orange",
                          8: "Apple", 9: "Muskmelon", 10: "Watermelon", 11: "Grapes", 12: "Mango", 13: "Banana",
                          14: "Pomegranate", 15: "Lentil", 16: "Blackgram", 17: "Mungbean", 18: "Mothbeans",
                          19: "Pigeonpeas", 20: "Kidneybeans", 21: "Chickpea", 22: "Coffee"}

         if predict[0] in crop_dict:
             crop = crop_dict[predict[0]]
             print("{} is a best crop to be cultivated ".format(crop))
         else:
             print("Sorry are not able to recommend a proper crop for this environment")

         Papaya is a best crop to be cultivated

In [38]: import pickle
```


(2) WhatsApp
Home Page - Select or create a
MAJOR - Jupyter Notebook
MAJOR
localhost:8888/nbconvert/html/MAJOR.ipynb?download=false
ENG IN
16:20 14/12/2023

```
Out[34]: 0.9931818181818182

In [36]: #predictive System
         def recommendation(N,P,k,temperature,humidity,ph,rainfal):
             features = np.array([[N,P,k,temperature,humidity,ph,rainfal]])
             transformed_features = ms.fit_transform(features)
             transformed_features = sc.fit_transform(transformed_features)
             prediction = rfc.predict(transformed_features).reshape(1,-1)

             return prediction[0]

In [ ]:

In [37]: N = 40
         P = 50
         k = 50
         temperature = 40.0
         humidity = 20
         ph = 100
         rainfall = 100

         predict = recommendation(N,P,k,temperature,humidity,ph,rainfall)


         crop_dict = {1: "Rice", 2: "Maize", 3: "Jute", 4: "Cotton", 5: "Coconut", 6: "Papaya", 7: "Orange",
                          8: "Apple", 9: "Muskmelon", 10: "Watermelon", 11: "Grapes", 12: "Mango", 13: "Banana",
                          14: "Pomegranate", 15: "Lentil", 16: "Blackgram", 17: "Mungbean", 18: "Mothbeans",
                          19: "Pigeonpeas", 20: "Kidneybeans", 21: "Chickpea", 22: "Coffee"}

         if predict[0] in crop_dict:
             crop = crop_dict[predict[0]]
             print("{} is a best crop to be cultivated ".format(crop))
         else:
             print("Sorry are not able to recommend a proper crop for this environment")

         Papaya is a best crop to be cultivated

In [38]: import pickle
         pickle.dump(rfc,open('model.pkl','wb'))
         pickle.dump(ms,open('minmaxscaler.pkl','wb'))
         pickle.dump(sc,open('standscaler.pkl','wb'))
```

CHAPTER 7

CONCLUSION & FUTURE SCOPE

7.1 Conclusion Of Project:

Our project helps farmers choose the best crops for their fields. By looking at things like soil, weather, and location, we give personalized suggestions to increase their chances of a successful harvest.

- Easy Suggestions: Farmers can easily use our system by entering their location and getting instant crop recommendations.
- Making Farming Easier: We aim to make farming less guesswork and more informed decision-making.
- Improvements Ahead: We're planning to make our suggestions even better by adding more local info and expanding the range of crops and areas we cover.

This project is all about using technology to support farmers, making farming smarter and more successful.

7.2 Future Scope :-

Future Plans

- Better Info: We'll add more local details and feedback from farmers to improve our suggestions.
- Smarter Predictions: We aim to make our predictions even more accurate by using smarter technology.
- Handy App: Creating a phone app could help farmers get recommendations anywhere, anytime.
- Working Together: We want to involve farmers to make sure our system meets their needs.
- Helping More Farmers: We're planning to cover more areas and suggest more types of crops to help more farmers.

These plans can make our crop suggestions even more useful and easier to use for farmers

CHAPTER 8

REFERENCES

1.https://stackoverflow.com/

2.https://chat.openai.com/

3.https://wikipedia.org

4. https://google.com/