Mini Project with Seminar

On

Precision Agriculture Using Machine Learning

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

Bachelor of Technology

In

Department of Computer Science and Engineering

By

M.Lakshman Chowdary	**19241A0533**
M.Shashank Reddy	**19241A0528**
K.Sai Anurag	**19241A0521**
H.Saaketh	**19241A0546**

Under the Esteemed guidance of

D.Ushasree

Assistant Professor

Department of Computer Science and Engineering GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

Bachupally, Kukatpally, Hyderabad-500090

Department of Computer Science and Engineering

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

Bachupally, Kukatpally, Hyderabad-500090

CERTIFICATE

This is to certify that the mini project entitled “**Precision Agriculture**” is submitted by **H.Saaketh(19241A0546),M.Shashank Reddy(19241A0528),K.Sai Anurag(19241A0521), M.LakshmanChowdary(19241A0533)** in partial fulfilment of the award of degree in BACHELOR OF TECHNOLOGY in Computer Science and Engineering during academic year 2021-2022.

INTERNAL GUIDE	**HEAD OF THE DEPARTMENT**
D.Ushasree	Dr. K. MADHAVI
Assistant Professor	

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We are happy to express our sense of gratitude to the principal of the college **Dr.J.Praveen**, Gokaraju Rangaraju Institute of engineering and technology, for providing us with facilities required to pursue our project. Our special thank to **Dr.K.Madhavi**, professor and head, Department of Computer Science and Engineering Gokaraju Rangaraju Institute of Engineering and Technology, for providing us with this wonderful opportunity the freedom to use laboratories and the library.

We are extremely grateful to our project guide **D.Ushasree**, Assistant Professor for Department of Computer Science and Engineering, Gokaraju Rangaraju Institute of Engineering and Technology, for her suggestions, encouragement and guidance throughout our project work.

We thank our seniors and all teaching and non-teaching staff of the Department of Computer Science And Engineering for their suggestions and motivation during the course of this work.

With great respect, we would like to thank our parents for their immense support, encouragement and strength.We would also like to thank our classmates who always helped us and gave moral support.

Finally, our sincere thanks to other individuals who have either directly or indirectly helped us at the right time for the development and success of our project.

H.Saaketh (19241A0546)

M.Shashank Reddy (19241A0528)

K.Sai Anurag (19241A0521)

M.Lakshman Chowdary (19241A0533)

DECLARATION

We hereby declare that the industrial mini project entitled "**Precision Agriculture**" is the work done during the period from 17th January 2022 to 14th May 2022 and is submitted in the partial fulfilment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering from Gokaraju Rangaraju Institute of Engineering and Technology (Autonomous under Jawaharlal Nehru Technology University, Hyderabad).The results embodied in this project have not been submitted to any other university or Institution for the award of any degree or diploma.

H.Saaketh (19241A0546)

M.Shashank Reddy (19241A0528)

K.Sai Anurag (19241A0521)

M.Lakshman Chowdary (19241A0533)

ABSTRACT

Agriculture is one of the sectors where most people are dependent on it for their livelihood.Crop production is influenced by a variety of seasonal, economic, and biological patterns, yet unforeseen variations in these patterns result in significant losses for farmers. When appropriate procedures are used on data linked to soil type, temperature, air pressure, humidity, and crop type, these hazards can be mitigated. Crop and weather forecasting, on the other hand, may be forecasted by obtaining helpful insights from agricultural data that help farmers pick which crop to plant for the future year in order to maximise profit. This study provides an overview of the various weather, agricultural yield, and crop cost prediction algorithms.

TABLE OF CONTENTS

CHAPTER-1

INTRODUCTION

Agriculture is the foundation of all economies. It has long been regarded as the primary and the most important culture performed in each place. There are numerous methods for increasing and improving agricultural output and quality. Data mining can be used to forecast crops. Data mining, in general, is the process of examining data from various angles and synthesising it into valuable knowledge. Crop forecasting is a significant agricultural issue. Every farmer wants to know which crop to grow in a given region at that particular time. Crop prediction used to be calculated by studying a farmer's previous experience with a specific crop. It's critical to have accurate knowledge regarding crop yield history while making judgments. Data analytics is the act of analysing data collections in order to derive conclusions about the information they contain, with the use of specialised systems and software becoming more common. Farmers are required to produce more and more crops as the weather changes swiftly from day to day. Given the current circumstances, many of them are unaware of the potential losses and are unaware of the benefits they receive by farming them. To detect and process data, the proposed system uses machine learning and prediction algorithms such as Logistic Regression and Clustering. This in turn will aid predicting the crop.

1.1 Rationale

A farmer has an idea on what crop to grow at a particular field through years of observation and trial and error methods.

Today , the level of knowledge on fields has become difficult because of soil erosion , change in various attributes like Nitrogen(N) , Phosphorus(P) , Potassium(K) , Humidity, Rainfall and Ph levels of soil , the farmers are not getting proper yield of crops than expected.

By using Precision Agriculture , it gives the perfect crop and farmers can easily make quick decisions and can implement to get more yield

1.2 GOAL

The major goal of this project is to determine the best crop to plant by performing the techniques of logistic classification and clustering. This technique is an efficient and a precise way to detect a suitable crop, by using key attributes such as nitrogen, phosphorus, potassium, rainfall, temperature and the pH value.

1.3 EXISTING SYSTEMS

Algorithms like Random Forest and Naive Bayes for precise agriculture have been used in the past.

We used Random Forests (RF), for its ability to predict crop yield responses to different variables at global and regional scales, in comparison with multiple linear regressions (MLR) serving as a standard. When forecasting the extreme ends or reactions beyond the confines of the training data, however, RF may result in a loss of accuracy.

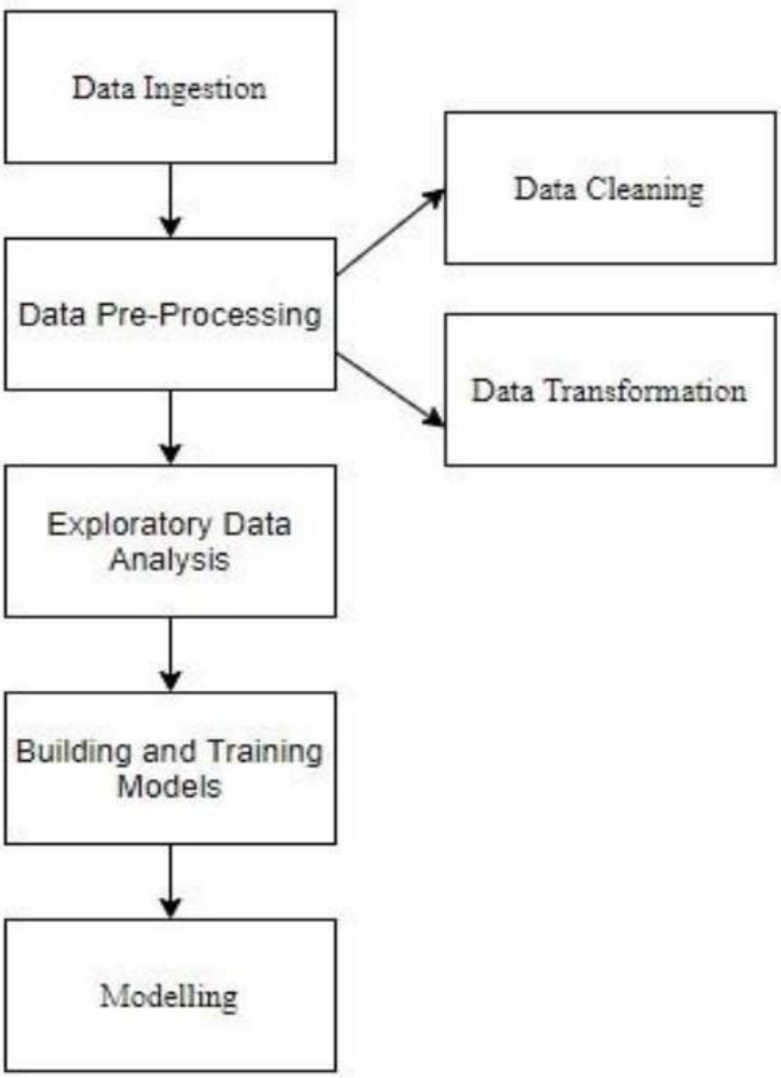
The Naive Bayes method is a supervised learning technique for addressing classification problems that are based on Bayes theorem. The original naive Bayes algorithm has a severe flaw: it generates redundant predictors.

Each method is more accurate than the others, but they are not optimised and do not work smoothly on all computers.

1.4 METHODOLOGY

INSTRUMENTS: The entire system is implemented in the jupyter notebook using the Python programming language.

ARCHITECTURE:



Data Ingestion:

Data ingestion is the process of transferring information from many sources to a storage media where it may be accessed, used, and analysed. A data warehouse, database, or document store is frequently used as the destination. SaaS data, internal apps, databases, spread sheets, and even information scraped from the internet can all be used as sources. Any analytics architecture's data ingestion layer is the foundation. Data consistency and accessibility are essential for downstream reporting and analytics. Different models or architectures might be used to construct a data ingestion layer.

Data Preprocessing:

Data preprocessing is a data mining approach for transforming raw data into a format that is both useful and efficient.

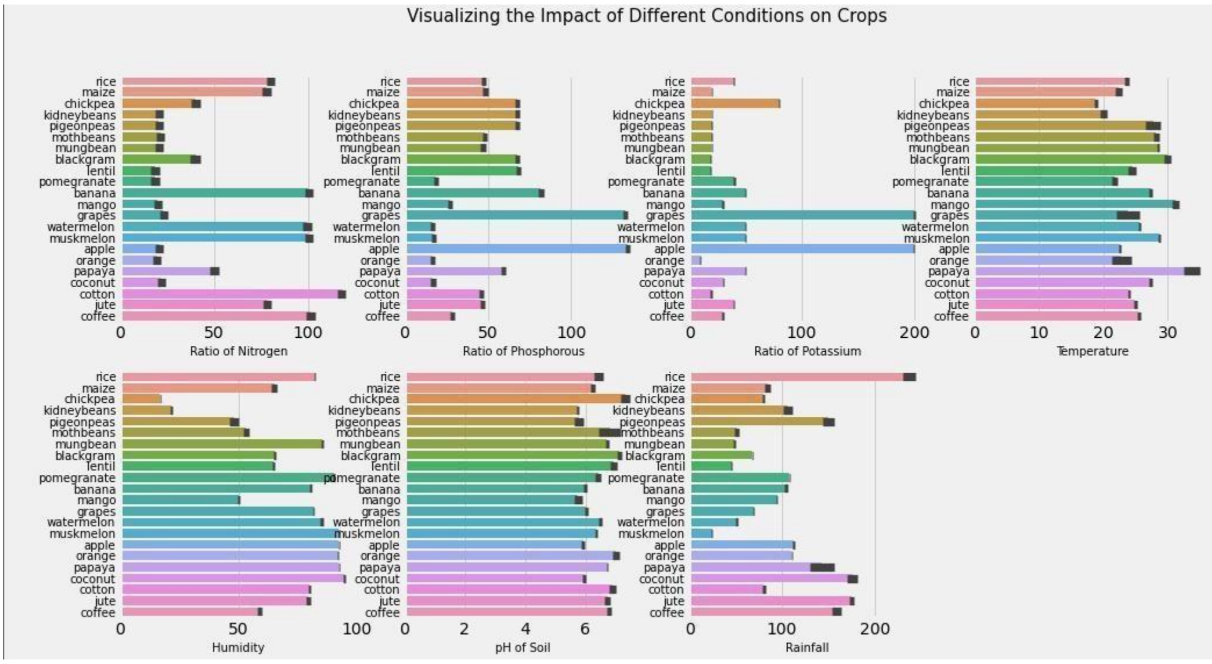
The information presented here goes through two stages:

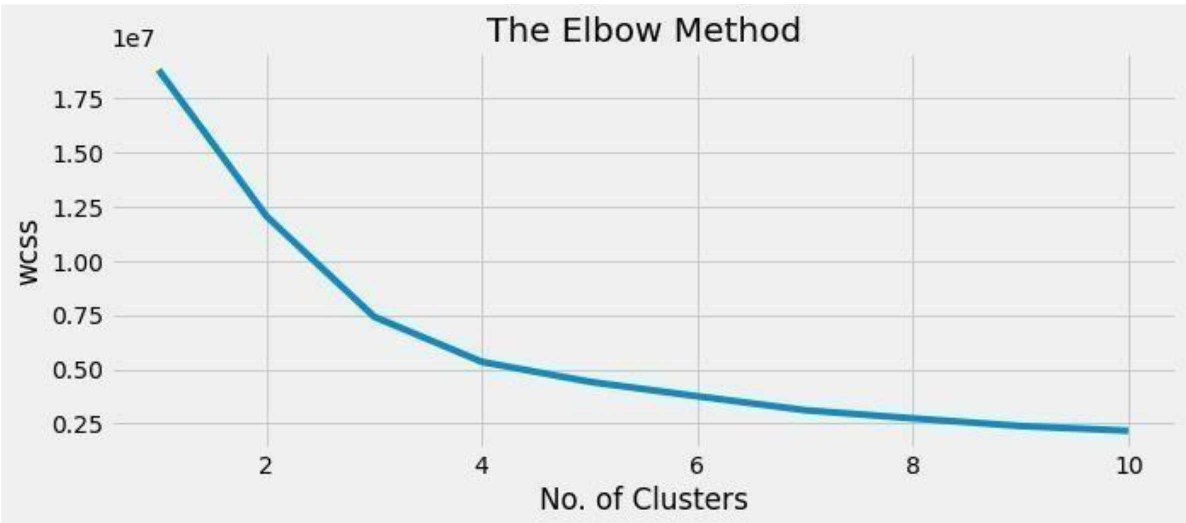
1. Data cleaning: It is critical that data be free of errors and undesirable data. As a result, the data is cleared before proceeding. Missing values, duplicate records, and improper formatting are all checked for and removed during data cleansing.

2. Data Transformation: Data transformation is the mathematical transformation of datasets; data is changed into appropriate formats for data mining. This allows us to better grasp the data by arranging the hundreds of records in a logical sequence. Normalisation, standardisation, and attribute selection are examples of transformations.

Exploratory Data Analysis:

Exploratory data analysis (EDA) is a technique for better understanding datasets using visual features such as scatter plots and bar charts. This helps us to more effectively identify data trends and execute analysis accordingly.


Visualizing the Impact of Different Conditions on Crops
rice
maize
chickpea
kidneybeans
pigeonpeas
mothbeans
mungbean
blackgram
lentil
pomegranate
banana
mango
grapes
watermelon
muskmelon
apple
orange
papaya
coconut
cotton
jute
coffee
Ratio of Nitrogen
Ratio of Phosphorous
Ratio of Potassium
Temperature
Humidity
pH of Soil
Rainfall


The Elbow Method
1e7
WCSS
No. of Clusters

Building and Training models:

Train-Test Split:

In machine learning model development, it is desirable that the trained model performs well on new, unseen data. To simulate new, unseen data, the available data is subjected to data segmentation, which splits it into two.

The first part is a larger data subset typically used as the training set and the second is a smaller subset typically used as the test set.

The training set is then used to build a predictive model, which is then applied to the test set to make predictions. The best model is chosen based on its performance on the test set, and various optimizations can be performed to obtain the best possible model.

Train validation test split:

Different approach for data splitting is to split the data into three parts:

(1) Training set,

(2) Validation set, and

(3) Test set.

Similar to the above, the training set is used to build the predictive model and is evaluated against the validation set. This allows you to make predictions, tune your model, and select the best performing model based on. Result of validation set. As you can see, it does the same for the validation set instead, similar to the test set above. Note that the test set is not involved in model building and preparation. Therefore, the test set can actually act as new hidden data.

Data Modelling:

Data modelling entails building a data model for data that will be kept in a database. Modelling entails training a Machine Learning Algorithm to predict labels from features, adjusting it for business requirements, and verifying it on holdout data.Modeling is the result of a trained Model that can be used to estimate new data points and make predictions. modelling is different from previous processes in the machine learning process, and it uses standardised input so we may change the prediction issue without having to rewrite all of our code. We can produce new label timings, build associated features, and insert them into the model if the business requirements change. Models are developed and then evaluated.

1.5 CONTRIBUTION OF THE PROJECT

Innovativeness :

Crop cultivation used to be done by farmers with on-the-ground experience. However, crop yields have begun to suffer as a result of climate change. As a result, farmers are unable to select the appropriate crop based on soil and environmental parameters, and the process of manually predicting the appropriate crop for a land has frequently failed. Crop prediction accuracy leads to higher crop production. In the field of crop prediction, this is where machine learning comes into play. Crop forecast is influenced by soil, regional, and climatic factors. Choosing acceptable qualities for the right crop is an integral aspect of feature selection techniques' prediction.

Usefulness :

- It is used to predict suitable crop
- Logistic Regression algorithm used in this model is very efficient and can further be used in many other applications

1.6 REPORT ORGANISATION

chapter 1 : We provide Introduction which includes Rationale, Goal, Existing Systems, Methodology and Contribution of the project.

chapter 2 : Here, we discuss the objective, Problem Statement, Functional Requirements, Non Functional Requirements, Software/Hardware details and Architecture of the project.

chapter 3 : In this section, We look at the construction of the project i.e the modules and libraries that were used .Also, the accuracy of code is discussed and finally, the outputs that we have got.

chapter 4 : We complete the report with a conclusion and future scope of the project.

CHAPTER-2

2.1 OBJECTIVE

The objective of this project is to determine the best crop to plant by performing the techniques of logistic classification and clustering. This technique is an efficient and a precise way to detect a suitable crop, by using key attributes such as nitrogen, phosphorus, potassium, rainfall, temperature and the pH value.

2.2 PROBLEM STATEMENT

Farmers have numerous challenges, one of which is determining the best crop to plant, depending on various conditions. Well developed countries such as the USA have extensively incorporated technology in their Green Revolutions and as a direct outcome of this, they are able to procure large yields with relatively less crop failures. On the other hand, India, as a developing country, is still yet to use technology in its vast area of agriculture so that the farmers output good yields and are less prone to crop failures. Our problem statement is predicting the best crops to grow based on various conditions, which is directly trying to implement our vision to incorporate technology in this field.

2.3 FUNCTIONAL REQUIREMENTS

The Functional Requirements describes how the system works in the model.All these requirements should be implemented in the system.the basic functional requirements of our project are as follows:

- We need a Dataset containing some attributes like Nitrogen(N),Phosphorus(p),Potassium(K),Humidity,Temperature,Ph level,Rainfall.
- Performing K-means clustering for dividing into groups and Logistic Regression for prediction of a categorical dependent using a set of independent variables.
- Predicting the crop according to the given input.

2.4 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements define systems performance and the system's operation capabilities and constraints that enhance its capabilities.

The Requirement of our project is:

- Availability Requirement - The program is easy to use and is available anytime.
- Accuracy - This program uses K-means clustering , Logistic Regression methods has more accuracy than random forest , naives bayes method .
- Performance Requirement - This program can be executed in 10 seconds with much faster execution for powerful devices.

2.5 SOFTWARE DETAILS

The Operating System on which our project runs is Windows 10 and the language we have used to code is Python (version 3.8.4) because of its object oriented features and simple, English-like coding syntax.

The distribution we used is anaconda distribution because It is open source and has a free Integrated Development Environment (IDE) written in Python and for Python. It is commonly used for scientific calculations.

We selected anaconda as our choice because it is powerful, and has a wide collection of packages that are seamlessly integrated into our program. It is compatible with most devices and is not heavy and resource consuming on a mid-range device as well.

2.6 HARDWARE DETAILS

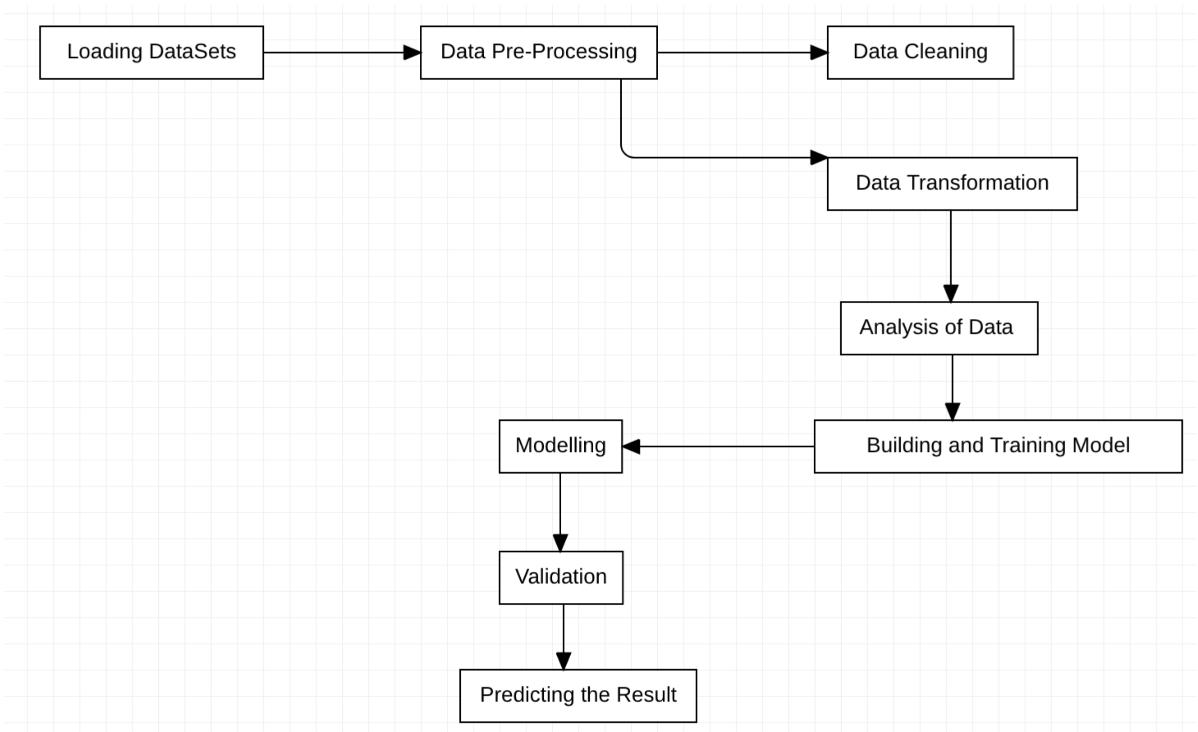
The Hardware Requirements for our program to run smoothly are:-

1. Processor- Intel core i5 9th gen
2. RAM- 8 GB
3. Speed- 2.1 Ghz
4. Hard Disk- 512 GB SSD

Better hardware can vastly improve the performance of the program.

2.7 ARCHITECTURE

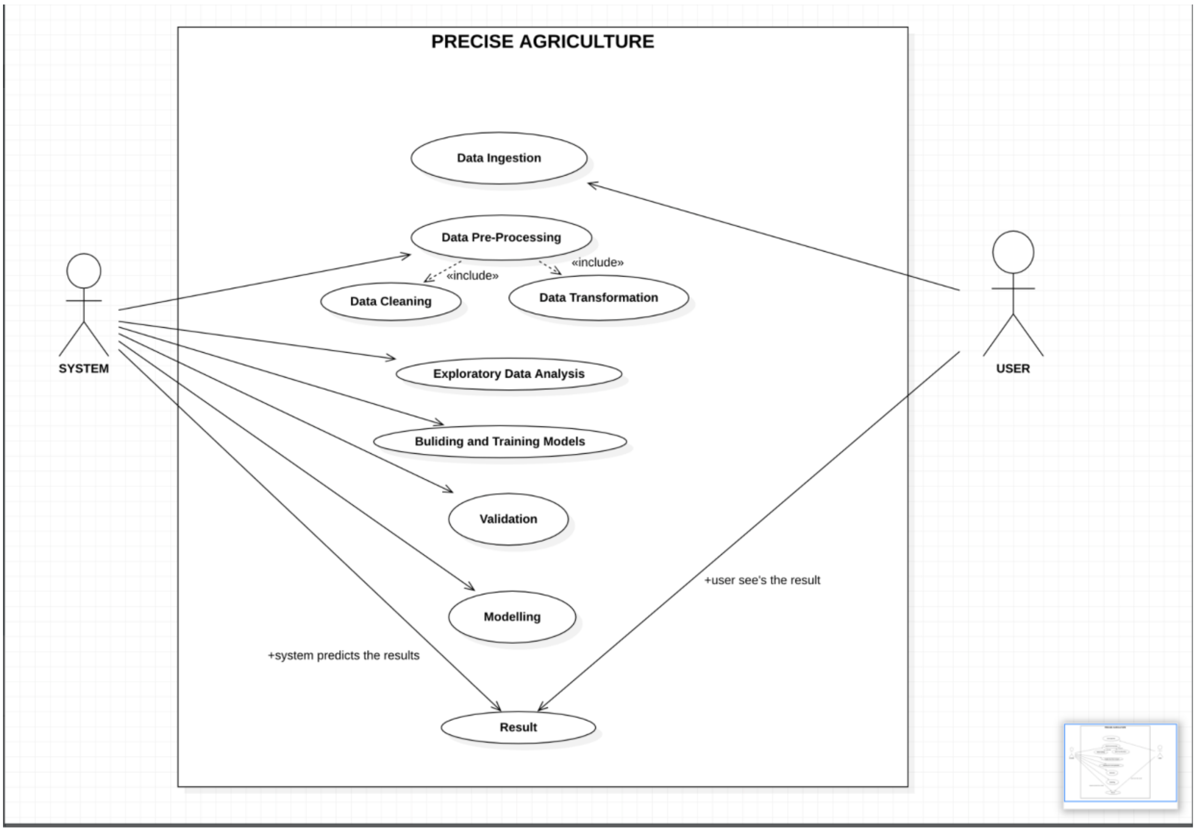
An architecture of a system defines the working model of a system. It provides an overview of a system in the form of UML(unified modelling language) diagrams . We can understand the behaviour and modelling of the system through the architecture of a system.



2.8 PROCESS DESIGN

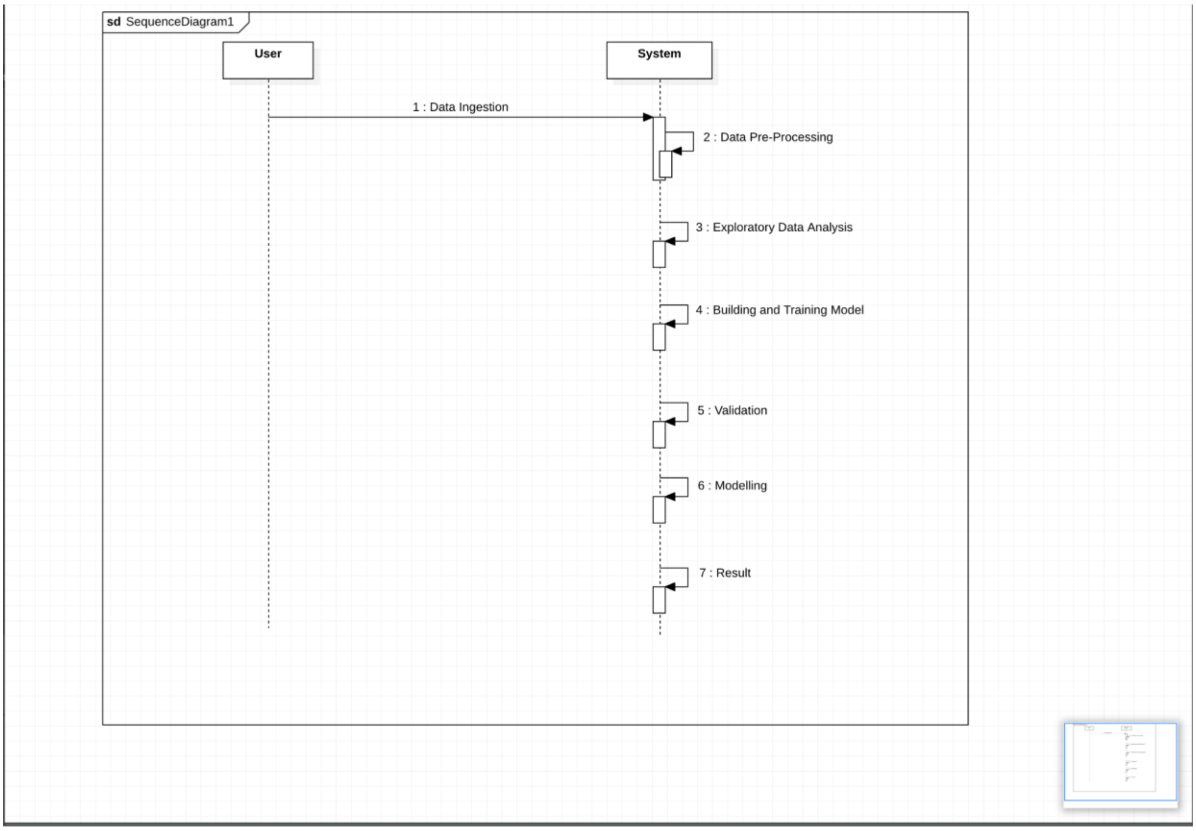
2.8.1 USE CASE DIAGRAM

Use case diagrams represent the overall scenario of the system. A scenario is nothing but a sequence of steps describing an interaction between a user and a system. The use case diagrams are drawn for exposing the functionalities of the system.



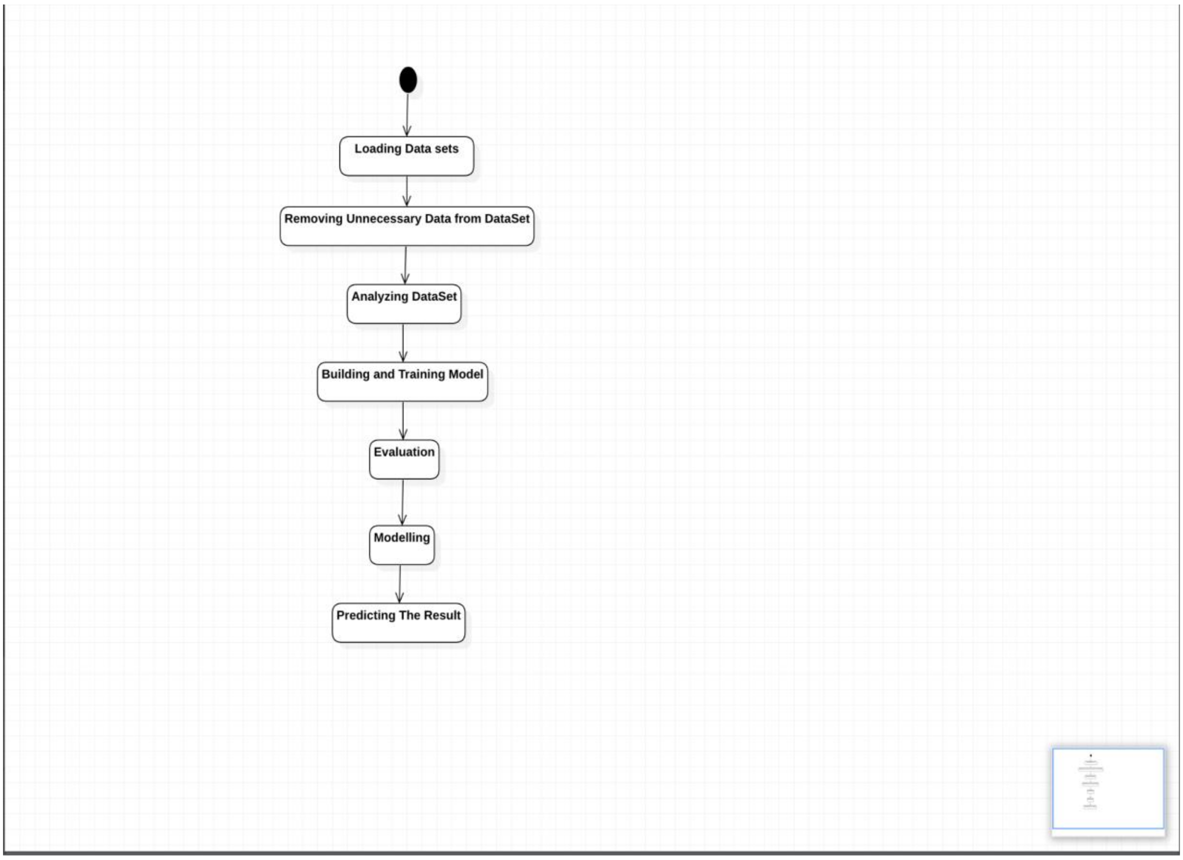
2.8.2 SEQUENCE DIAGRAM

In the sequence diagram how the object interacts with the other object is shown. There are sequences of events that are represented.
It is a period situated perspective on the association between items to achieve a conduct objective of the framework



2.8.3 ACTIVITY Diagram

The activity diagram is a graphical representation for representing the flow of interaction within specific scenarios. It is similar to a flowchart in which various activities that can be performed in the system are represented.



CHAPTER-3

3.1 IMPLEMENTATION

The implementation of this project in Python programming language this is because python is an effective, powerful and user friendly language. It is an open-source language. Its plan reasoning emphasises code coherence with the utilisation of critical indentation. Python bolsters modules and bundles, which energises code reusability and program modularity. Python gives expanded efficiency.

Applications of Python:

Data science,

Web development and

Computer graphics

Libraries used:

IMPORTING LIBRARIES

```
In [1]:
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report
```

NumPy: It is a library that consists of multi - dimensional (2d) array objects and a set of approaches for manipulating them. NumPy can be used to conduct logical and mathematical operations on arrays.

Pandas : Pandas is a data cleaning and analysis tool that is widely used in Data Science and Machine Learning.

Here, Pandas are used to read the following dataset

Matplotlib: It's a cross-platform library that creates 2D charts from array data. It has an object-oriented API for embedding plots in Python GUI toolkits like PyQt and WxPython Tkinter.

We use Matplotlib to plot graphs for better visualisation of clustering

Seaborn: Seaborn is a matplotlib-based data visualisation library. It helps in creating statistical graphs easily.

Scikit-learn: Classification, regression and clustering are some of the useful tools in the sklearn package. Cross-validation, feature extraction, supervised learning methods, and unsupervised learning algorithms are all features of scikit learn.

sklean.clusters.KMeans: It is used to combine the attributes of similar data points together. It scales well to big/large no.of values and has been used in a wide range of application fields.

sklearn.model selection.train: split the arrays into training and testing subsets

Sklearn.linear.model.LogisticRegression: is a classification technique that is used to predict the categorical dependent variable

sklearn.metrics.classification_report : A classification report consists of precision, recall, fi-score and support which gives us the accuracy of the model.

Loading DataSet :

We read the csv file by using 'pd.read_csv' command and then, check if there are any null values present in them by using 'is.null()'

READING DATASET

In [46]:
```
# data = pd.read_csv('data.csv')
data = pd.read_csv('data1.csv')
```

In [47]:
```
data.head()
```

	N	P	K	temperature	humidity	ph	rainfall	label	date	Unnamed: 9
0	90	42	43	20.879744	82.002744	6.502985	202.935536	rice	10/1/2015	NaN
1	85	58	41	21.770462	80.319644	7.038096	226.655537	rice	10/2/2015	NaN
2	60	55	44	23.004459	82.320763	7.840207	263.964248	rice	10/3/2015	NaN
3	74	35	40	26.491096	80.158363	6.980401	242.864034	rice	10/4/2015	NaN
4	78	42	42	20.130175	81.604873	7.628473	262.717340	rice	10/5/2015	NaN

DESCRIPTION FOR EACH OF THE COLUMNS IN DATASET

In [24]:
```
data.isnull()
```

	N	P	K	temperature	humidity	ph	rainfall	label
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...
2195	False	False	False	False	False	False	False	False
2196	False	False	False	False	False	False	False	False
2197	False	False	False	False	False	False	False	False
2198	False	False	False	False	False	False	False	False
2199	False	False	False	False	False	False	False	False

2200 rows × 8 columns

Descriptive Statistics :

DESCRIPTIVE STATISTICS

```
In [6]: print("Average Ratio of N : {0:.2f}".format(data['N'].mean()))
        print("Average Ratio of P: {0:.2f}".format(data['P'].mean()))
        print("Average Ratio of K: {0:.2f}".format(data['K'].mean()))
        print("Average Tempature : {0:.2f}".format(data['temperature'].mean()))
        print("Average Humidity : {0:.2f}".format(data['humidity'].mean()))
        print("Average PH value : {0:.2f}".format(data['ph'].mean()))
        print("Average Rainfall : {0:.2f}".format(data['rainfall'].mean()))
```

```
Average Ratio of N : 50.55
Average Ratio of P: 53.36
Average Ratio of K: 48.15
Average Tempature : 25.62
Average Humidity : 71.48
Average PH value : 6.47
Average Rainfall : 103.46
```

```
In [45]: print("Winter Crops")
         print(data[(data['temperature']<20.1)&(data['humidity']>30.1)]['label'].unique())
         print("\n")
         print("Rainy Crops")

         print(data[(data['rainfall']>=200.9)&(data['humidity']>=30.1)]['label'].unique())
         print("\n")
         print('Summer Crops')
         print(data[(data['temperature']>=30.1)&(data['humidity']>=60.1)]['label'].unique())
```

```
Winter Crops
['rice' 'maize' 'pigeonpeas' 'lentil' 'pomegranate' 'grapes' 'orange']


Rainy Crops
['rice' 'papaya' 'coconut']


Summer Crops
['pigeonpeas' 'mothbeans' 'blackgram' 'grapes' 'orange' 'papaya']
```

Here, we get a descriptive statistics and see different crops that grow in different seasons

K-means Implementation :

```
In [9]:
#K Means algorithm to perform Clustering analysis
from sklearn.cluster import KMeans
k_means = KMeans(max_iter = 300,n_clusters = 4, n_init = 10,init = 'k-means++')
ymeans = k_means.fit_predict(x)

n = data['label']
ymeans = pd.DataFrame(ymeans)
m = pd.concat([ymeans,n], axis = 1)
m = m.rename(columns = {0: 'cluster'})

#Clusters of each Crops
print("K_Means Cluster Analysis\n")
print("Crops in First Cluster: ", m[m['cluster'] == 0]['label'].unique())
print("\t")
print("Crops in Second Cluster:", m[m['cluster'] == 1]['label'].unique())
print("\t")
print("Crops in Third Cluster:", m[m['cluster'] == 2]['label'].unique())
print("\t")
print("Crops in Forth Cluster:", m[m['cluster'] == 3]['label'].unique())
```

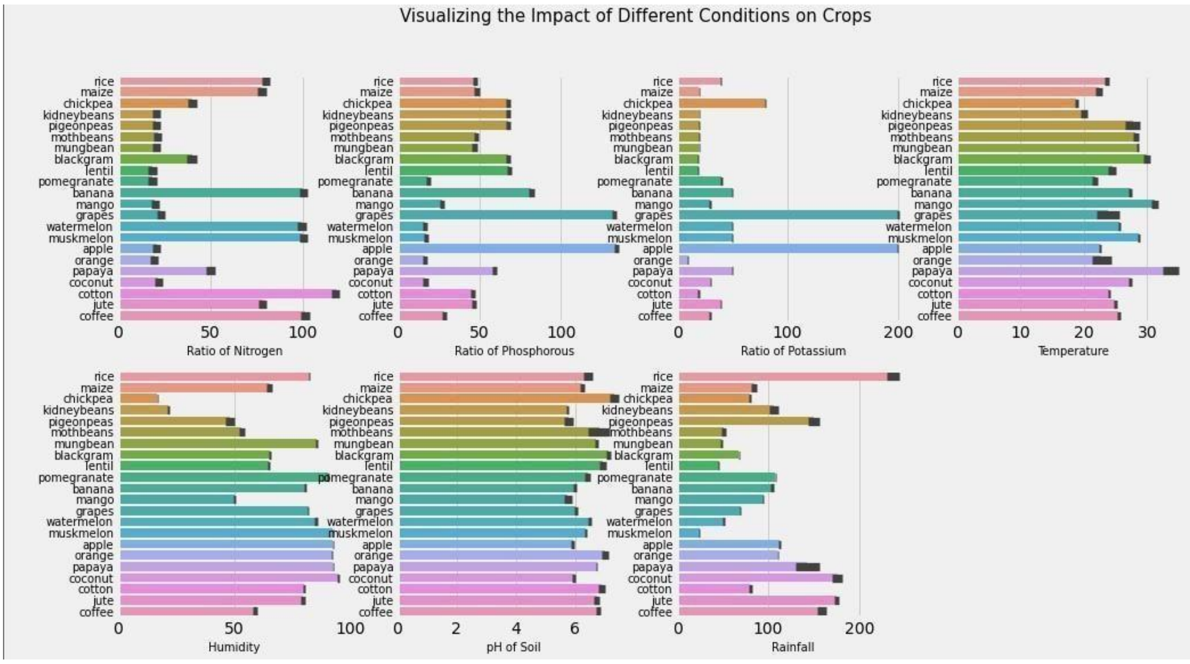
```
K_Means Cluster Analysis

Crops in First Cluster:  ['maize' 'banana' 'watermelon' 'muskmelon' 'papaya' 'cotton' 'coffee']

Crops in Second Cluster: ['rice' 'pigeonpeas' 'papaya' 'coconut' 'jute' 'coffee']

Crops in Third Cluster: ['maize' 'chickpea' 'kidneybeans' 'pigeonpeas' 'mothbeans' 'mungbean'
 'blackgram' 'lentil' 'pomegranate' 'mango' 'orange' 'papaya' 'coconut']

Crops in Forth Cluster: ['grapes' 'apple']
```

Now, we use k-means .Then, use Matplotlib and Seaborn for visualization of the clustering.

Visual Representation :

In [10]:

```
### Data Visualizations
plt.subplot(3,5,1)
plt.xlabel('Nitrogen',fontsize=5)
sns.barplot(data['N'],data['label'])
plt.ylabel(' ')
plt.subplot(3, 5, 2)
plt.xlabel(' Phosphorous',fontsize=5)
sns.barplot(data['P'], data['label'])
plt.ylabel(' ')
plt.subplot(3, 4, 3)
plt.xlabel('Potassium',fontsize=5)
sns.barplot(data['K'], data['label'])
plt.ylabel(' ')
plt.subplot(3, 4, 4)
plt.xlabel('Temperature',fontsize=5)
sns.barplot(data['temperature'], data['label'])
plt.ylabel(' ')
plt.subplot(3, 4, 5)
plt.xlabel('Humidity',fontsize=5)
sns.barplot(data['humidity'], data['label'])
plt.ylabel(' ')
plt.subplot(3, 4, 6)
plt.xlabel('pH value', fontsize = 5)
sns.barplot(data['ph'], data['label'])
plt.ylabel(' ')
plt.subplot(3, 4, 7)
plt.xlabel('Rainfall',fontsize=5)
sns.barplot(data['rainfall'], data['label'])
plt.ylabel(' ')
plt.suptitle('Different Conditions on Crops',fontsize=15)
plt.show()
```



Predictive Modelling :

Now, let's split the Dataset for Predictive Modelling.

Then, by using “train_test_split” training and testing of subsets is done for validation of results.

Finally, we create a Predictive Model

```
In [11]: y=data['label']
         x=data.drop(['label'],axis = 1)
```

```
In [12]: from sklearn.model_selection import train_test_split
         train_x, test_x, train_y, test_y = train_test_split(x, y, test_size = 0.2, random_state = 0)

         print("Shape of X Train:", train_x.shape)
         print("Shape of X Test:", test_x.shape)
         print("Shape of Y Train:", train_y.shape)
         print("Shape of Y Test:", test_y.shape)
```

```
Shape of X Train: (1760, 7)
Shape of X Test: (440, 7)
Shape of Y Train: (1760,)
Shape of Y Test: (440,)
```

```
In [13]: from sklearn.linear_model import LogisticRegression

         model = LogisticRegression()
         model.fit(train_x, train_y)
         pred_y = model.predict(test_x)
```

Classification Report :

Here is the classification report for using logistic regression technique

In [27]:

```
# Lets evaluate the Model Performance
from sklearn.metrics import classification_report

# Lets print the Classification Report also
cr = classification_report(y_test, y_pred)
print(cr)
```

	precision	recall	f1-score	support
apple	1.00	1.00	1.00	18
banana	1.00	1.00	1.00	18
blackgram	0.86	0.82	0.84	22
chickpea	1.00	1.00	1.00	23
coconut	1.00	1.00	1.00	15
coffee	1.00	1.00	1.00	17
cotton	0.89	1.00	0.94	16
grapes	1.00	1.00	1.00	18
jute	0.84	1.00	0.91	21
kidneybeans	1.00	1.00	1.00	20
lentil	0.94	0.94	0.94	17
maize	0.94	0.89	0.91	18
mango	1.00	1.00	1.00	21
mothbeans	0.88	0.92	0.90	25
mungbean	1.00	1.00	1.00	17
muskmelon	1.00	1.00	1.00	23
orange	1.00	1.00	1.00	23
papaya	1.00	0.95	0.98	21
pigeonpeas	1.00	1.00	1.00	22
pomegranate	1.00	1.00	1.00	23
rice	1.00	0.84	0.91	25
watermelon	1.00	1.00	1.00	17
accuracy			0.97	440
macro avg	0.97	0.97	0.97	440
weighted avg	0.97	0.97	0.97	440

Final Output:

Finally, by giving the values of N, P ,K, Temperature, Ph, Humidity, Rainfall we get the suitable crop for those climatic conditions.

In [17]:

```
prediction =model.predict((np.array([[90,40,40,20,80,7,200]])))
print("The recommended Crop for Climatic Condition :", prediction)
```

```
The recommended Crop for Climatic Condition : ['rice']
```

In [18]:

```
prediction =model.predict((np.array([[50,10,30,30,40,40,100]])))
print("The recomended Crop for Given Climatic Condition :", prediction)
```

```
The recomended Crop for Given Climatic Condition : ['coffee']
```

CHAPTER - 4

4.1 Conclusion

To conclude this documentation, we have discussed what our project is, its rationale and the goal we achieve with this project. We applied specific data analysis techniques to find out a suitable crop by using existing data. Logistic regression is used to extract important information from agriculture recording. The analytical process began with data ingestion followed by data cleansing and processing, missing value and research .It concludes by analysing the data, and finally modelling and evaluation. The highest accuracy in the public test set . Higher accuracy evaluation of machine learning method by mutual verification calculation verification, precision, recall, F1 score. The proposed system takes into account relevant data on nitrogen, phosphorus, potassium, temperature, rainfall, and highest yield crops over the past year. It can be cultivated under appropriate environmental conditions. Lists all possible crops that are cultivated and it helps farmers make decisions about the culture that is cultivated. The system also takes into account historical data generation. This helps farmers gain insights into the requirements of the various crops that are adequate to grow on the given plot of land.

4.2 Scope

The applications of this project are at present limited, but with proper research, great results are expected. There are limitations with this project, such as taking a larger amount of data to determine suitable crops than the conventional methods. But there is scope for improvement. With better algorithms, there could be a significant decrease in evaluation time.

Further, this project can also be extended by creating an application, to be compatible with mobiles, so that the users can determine their own crop without relying on others.

References :

https://scholar.google.co.in/scholar?q=random+forest+based+on+crop+prediction&hl=en&as_sdt=0&as_vis=1&oi=scholart - random forest existing system

https://ieeexplore.ieee.org/abstract/document/9214190 - NAIVE BAYES

https://publications.waset.org/9997276/modified-naive-bayes-based-prediction-modeling-for-crop-yield-prediction -

[1]. A Survey on Crop Prediction using Machine Learning Approach

[2]. Predicting Yield of the Crop Using Machine Learning Algorithm

[3]. Crop Price Forecasting System Using Supervised Machine Learning Algorithms

[4]. www.kaggle.com for datasets

[5]. https://en.wikipedia.org/wiki/Agriculture

[6]. https://en.wikipedia.org/wiki/Data_analysis

[7]. JeetendraShenoy, YogeshPingle, "IOT in agriculture", 2016 IEEE.

[8]. M.R. Bendre, R.C. Thool, V.R.Thool, "Big Data in Precision agriculture",Sept,2015 NGCT.

[9]. Monali Paul, Santosh K. Vishwakarma, Ashok Verma, "Analysis of Soil Behavior and Prediction of Crop Yield using Data Mining approach"

[10]. Abdullah Na, William Isaac, ShashankVarshney, Ekram Khan, "An IoT Based System for Remote Monitoring of Soil

Characteristics", 2016 International Conference of Information Technology.

[11]. Dr.N.Suma, Sandra Rhea Samson, S.Saranya, G.Shanmugapriya, R.Subhashri, "IOT Based Smart Agriculture Monitoring System",

Feb 2017 IJRITCC.

[12]. N.Heemageetha, "A survey on Application of Data Mining Techniques to Analyse the soil for agricultural purpose", 2016IEEE.

Appendix :

IMPORTING LIBRARIES

```
In [1]:
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report
```

READING DATASET

In [46]:
```
# data = pd.read_csv('data.csv')
data = pd.read_csv('data1.csv')
```

In [47]:
```
data.head()
```

	N	P	K	temperature	humidity	ph	rainfall	label	date	Unnamed: 9
0	90	42	43	20.879744	82.002744	6.502985	202.935536	rice	10/1/2015	NaN
1	85	58	41	21.770462	80.319644	7.038096	226.655537	rice	10/2/2015	NaN
2	60	55	44	23.004459	82.320763	7.840207	263.964248	rice	10/3/2015	NaN
3	74	35	40	26.491096	80.158363	6.980401	242.864034	rice	10/4/2015	NaN
4	78	42	42	20.130175	81.604873	7.628473	262.717340	rice	10/5/2015	NaN

DESCRIPTION FOR EACH OF THE COLUMNS IN DATASET

In [24]:
```
data.isnull()
```

	N	P	K	temperature	humidity	ph	rainfall	label
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...
2195	False	False	False	False	False	False	False	False
2196	False	False	False	False	False	False	False	False
2197	False	False	False	False	False	False	False	False
2198	False	False	False	False	False	False	False	False
2199	False	False	False	False	False	False	False	False

2200 rows × 8 columns

DESCRIPTIVE STATISTICS

In [6]:

```
print("Average Ratio of N : {0:.2f}".format(data['N'].mean()))
print("Average Ratio of P: {0:.2f}".format(data['P'].mean()))
print("Average Ratio of K: {0:.2f}".format(data['K'].mean()))
print("Average Tempature : {0:.2f}".format(data['temperature'].mean()))
print("Average Humidity : {0:.2f}".format(data['humidity'].mean()))
print("Average PH value : {0:.2f}".format(data['ph'].mean()))
print("Average Rainfall : {0:.2f}".format(data['rainfall'].mean()))
```

```
Average Ratio of N : 50.55
Average Ratio of P: 53.36
Average Ratio of K: 48.15
Average Tempature : 25.62
Average Humidity : 71.48
Average PH value : 6.47
Average Rainfall : 103.46
```

In [45]:

```
print("Winter Crops")
print(data[(data['temperature']<20.1)&(data['humidity']>30.1)]['label'].unique())
print("\n")
print("Rainy Crops")

print(data[(data['rainfall']>=200.9)&(data['humidity']>=30.1)]['label'].unique())
print("\n")
print('Summer Crops')
print(data[(data['temperature']>=30.1)&(data['humidity']>=60.1)]['label'].unique())
```

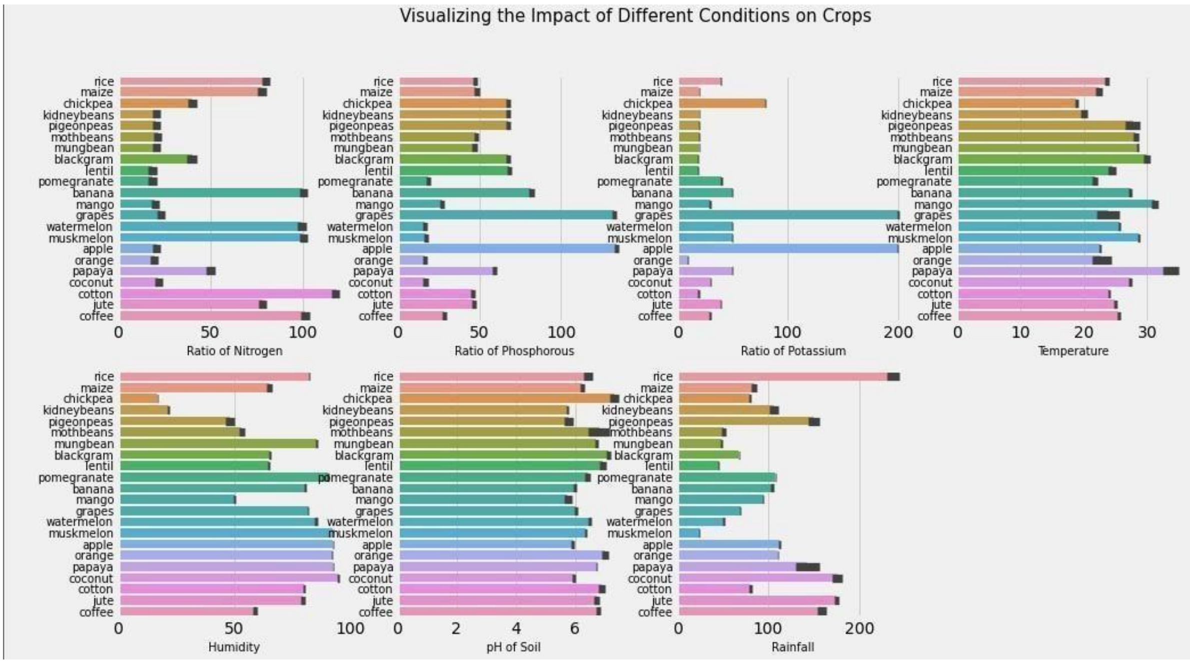
```
Winter Crops
['rice' 'maize' 'pigeonpeas' 'lentil' 'pomegranate' 'grapes' 'orange']


Rainy Crops
['rice' 'papaya' 'coconut']


Summer Crops
['pigeonpeas' 'mothbeans' 'blackgram' 'grapes' 'orange' 'papaya']
```

In [9]:

```
#K Means algorithm to perform Clustering analysis
from sklearn.cluster import KMeans
k_means = KMeans(max_iter = 300,n_clusters = 4, n_init = 10,init = 'k-means++')
ymeans = k_means.fit_predict(x)

n = data['label']
ymeans = pd.DataFrame(ymeans)
m = pd.concat([ymeans,n], axis = 1)
m = m.rename(columns = {0: 'cluster'})

#Clusters of each Crops
print("K_Means Cluster Analysis\n")
print("Crops in First Cluster: ", m[m['cluster'] == 0]['label'].unique())
print("\t")
print("Crops in Second Cluster:", m[m['cluster'] == 1]['label'].unique())
print("\t")
print("Crops in Third Cluster:", m[m['cluster'] == 2]['label'].unique())
print("\t")
print("Crops in Forth Cluster:", m[m['cluster'] == 3]['label'].unique())
```

```
K_Means Cluster Analysis

Crops in First Cluster:  ['maize' 'banana' 'watermelon' 'muskmelon' 'papaya' 'cotton' 'coffee']

Crops in Second Cluster: ['rice' 'pigeonpeas' 'papaya' 'coconut' 'jute' 'coffee']

Crops in Third Cluster: ['maize' 'chickpea' 'kidneybeans' 'pigeonpeas' 'mothbeans' 'mungbean'
 'blackgram' 'lentil' 'pomegranate' 'mango' 'orange' 'papaya' 'coconut']

Crops in Forth Cluster: ['grapes' 'apple']
```

```
In [10]:
### Data Visualizations
plt.subplot(3,5,1)
plt.xlabel('Nitrogen',fontsize=5)
sns.barplot(data['N'],data['label'])
plt.ylabel(' ')
plt.subplot(3, 5, 2)
plt.xlabel(' Phosphorous',fontsize=5)
sns.barplot(data['P'], data['label'])
plt.ylabel(' ')
plt.subplot(3, 4, 3)
plt.xlabel('Potassium',fontsize=5)
sns.barplot(data['K'], data['label'])
plt.ylabel(' ')
plt.subplot(3, 4, 4)
plt.xlabel('Temperature',fontsize=5)
sns.barplot(data['temperature'], data['label'])
plt.ylabel(' ')
plt.subplot(3, 4, 5)
plt.xlabel('Humidity',fontsize=5)
sns.barplot(data['humidity'], data['label'])
plt.ylabel(' ')
plt.subplot(3, 4, 6)
plt.xlabel('pH value', fontsize = 5)
sns.barplot(data['ph'], data['label'])
plt.ylabel(' ')
plt.subplot(3, 4, 7)
plt.xlabel('Rainfall',fontsize=5)
sns.barplot(data['rainfall'], data['label'])
plt.ylabel(' ')
plt.suptitle('Different Conditions on Crops',fontsize=15)
plt.show()
```



In [11]:
```python
y=data['label']
x=data.drop(['label'],axis = 1)
```

In [12]:
```python
from sklearn.model_selection import train_test_split
train_x, test_x, train_y, test_y = train_test_split(x, y, test_size = 0.2, random_state = 0)

print("Shape of X Train:", train_x.shape)
print("Shape of X Test:", test_x.shape)
print("Shape of Y Train:", train_y.shape)
print("Shape of Y Test:", test_y.shape)
```

```
Shape of X Train: (1760, 7)
Shape of X Test: (440, 7)
Shape of Y Train: (1760,)
Shape of Y Test: (440,)
```

In [13]:
```python
from sklearn.linear_model import LogisticRegression

model = LogisticRegression()
model.fit(train_x, train_y)
pred_y = model.predict(test_x)
```

In [27]:
```python
# Lets evaluate the Model Performance
from sklearn.metrics import classification_report

# Lets print the Classification Report also
cr = classification_report(y_test, y_pred)
print(cr)
```

	precision	recall	f1-score	support
apple	1.00	1.00	1.00	18
banana	1.00	1.00	1.00	18
blackgram	0.86	0.82	0.84	22
chickpea	1.00	1.00	1.00	23
coconut	1.00	1.00	1.00	15
coffee	1.00	1.00	1.00	17
cotton	0.89	1.00	0.94	16
grapes	1.00	1.00	1.00	18
jute	0.84	1.00	0.91	21
kidneybeans	1.00	1.00	1.00	20
lentil	0.94	0.94	0.94	17
maize	0.94	0.89	0.91	18
mango	1.00	1.00	1.00	21
mothbeans	0.88	0.92	0.90	25
mungbean	1.00	1.00	1.00	17
muskmelon	1.00	1.00	1.00	23
orange	1.00	1.00	1.00	23
papaya	1.00	0.95	0.98	21
pigeonpeas	1.00	1.00	1.00	22
pomegranate	1.00	1.00	1.00	23
rice	1.00	0.84	0.91	25
watermelon	1.00	1.00	1.00	17
accuracy			0.97	440
macro avg	0.97	0.97	0.97	440
weighted avg	0.97	0.97	0.97	440

In [17]:
```python
prediction =model.predict((np.array([[90,40,40,20,80,7,200]])))
print("The recommended Crop for Climatic Condition :", prediction)
```

```
The recommended Crop for Climatic Condition : ['rice']
```

In [18]:
```python
prediction =model.predict((np.array([[50,10,30,30,40,40,100]])))
print("The recomended Crop for Given Climatic Condition :", prediction)
```

```
The recomended Crop for Given Climatic Condition : ['coffee']
```

BATCH A4

ORIGINALITY REPORT

23%	21%	4%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.ijcrt.org Internet Source	12%
2	innovate.mygov.in Internet Source	3%
3	www.science.gov Internet Source	2%
4	Submitted to Birla Institute of Technology and Science Pilani Student Paper	1%
5	Submitted to Gitam University Student Paper	1%
6	Submitted to Sheffield Hallam University Student Paper	1%
7	www.coursehero.com Internet Source	1%
8	trap.ncirl.ie Internet Source	1%
9	www.geothermal-energy.org Internet Source	1%

10	123dok.com Internet Source	1%
11	Submitted to UNITEC Institute of Technology Student Paper	<1%
12	"International Conference on Innovative Computing and Communications", Springer Science and Business Media LLC, 2022 Publication	<1%
13	air.ashesi.edu.gh Internet Source	<1%
14	dspace.uiu.ac.bd Internet Source	<1%
15	hps.vi4io.org Internet Source	<1%
16	www.ijert.org Internet Source	<1%
17	Maria Anastasiadi, Fady Mohareb, Sally P. Redfern, Mark Berry, Monique S. J. Simmonds, Leon A. Terry. "Biochemical Profile of Heritage and Modern Apple Cultivars and Application of Machine Learning Methods To Predict Usage, Age, and Harvest Season", Journal of Agricultural and Food Chemistry, 2017 Publication	<1%

Code :

IMPORTING LIBRARIES

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.cluster import KMeans

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import classification_report

# data = pd.read_csv('data.csv')

data = pd.read_csv('data1.csv')
```

DESCRIPTION FOR EACH OF THE COLUMNS IN DATASET

```
data.head()

data.isnull()
```

DESCRIPTIVE STATISTICS

```
print("Average Ratio of N : {0:.2f}".format(data['N'].mean()))

print("Average Ratio of P: {0:.2f}".format(data['P'].mean()))

print("Average Ratio of K: {0:.2f}".format(data['K'].mean()))

print("Average Temperature : {0:.2f}".format(data['temperature'].mean()))

print("Average Humidity : {0:.2f}".format(data['humidity'].mean()))

print("Average PH value : {0:.2f}".format(data['ph'].mean()))

print("Average Rainfall : {0:.2f}".format(data['rainfall'].mean()))
```

print("Winter Crops")

print(data[(data['temperature']<20.1)&(data['humidity']>30.1)]['label'].unique())

print("\n")

print("Rainy Crops")

print(data[(data['rainfall']>=200.9)&(data['humidity']>=30.1)]['label'].unique())

print("\n")

print('Summer Crops')

print(data[(data['temperature']>=30.1)&(data['humidity']>=60.1)]['label'].unique())

****CLUSTERING CROPS**

x = data.loc[:, ['N','P','K','temperature','ph','humidity','rainfall']].values

data_x = pd.DataFrame(x)

#K Means algorithm to perform Clustering analysis

from sklearn.cluster import KMeans

k_means = KMeans(max_iter = 300,n_clusters = 4, n_init = 10,init = 'k-means++')

ymeans = k_means.fit_predict(x)

n = data['label']

ymeans = pd.DataFrame(ymeans)

m = pd.concat([ymeans,n], axis = 1)

m = m.rename(columns = {0: 'cluster'})

print("K_Means Cluster Analysis\n")

print("Crops in First Cluster: ", m[m['cluster'] == 0]['label'].unique())

print("\t")

print("Crops in Second Cluster:", m[m['cluster'] == 1]['label'].unique())

print("\t")

```
print("Crops in Third Cluster:", m[m['cluster'] == 2]['label'].unique())

print("\t")

print("Crops in Forth Cluster:", m[m['cluster'] == 3]['label'].unique())
```

Data Visualizations

```
plt.subplot(3,5,1)

plt.xlabel('Nitrogen',fontsize=5)

sns.barplot(data['N'],data['label'])

plt.ylabel(' ')

plt.subplot(3, 5, 2)

plt.xlabel(' Phosphorous',fontsize=5)

sns.barplot(data['P'], data['label'])

plt.ylabel(' ')

plt.subplot(3, 4, 3)

plt.xlabel('Potassium',fontsize=5)

sns.barplot(data['K'], data['label'])

plt.ylabel(' ')

plt.subplot(3, 4, 4)

plt.xlabel('Temperature',fontsize=5)

sns.barplot(data['temperature'], data['label'])

plt.ylabel(' ')

plt.subplot(3, 4, 5)

plt.xlabel('Humidity',fontsize=5)

sns.barplot(data['humidity'], data['label'])

plt.ylabel(' ')

plt.subplot(3, 4, 6)

plt.xlabel('pH value', fontsize = 5)
```

sns.barplot(data['ph'], data['label'])

plt.ylabel(' ')

plt.subplot(3, 4, 7)

plt.xlabel('Rainfall',fontsize=5)

sns.barplot(data['rainfall'], data['label'])

plt.ylabel(' ')

plt.suptitle('Different Conditions',fontsize=15)

plt.show()

#Predictive Modelling

y=data['label']

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

from sklearn.model_selection import train_test_split

train_x, test_x, train_y, test_y = train_test_split(x, y, test_size = 0.2)

print("Shape of X Train:", train_x.shape)

print("Shape of X Test:", test_x.shape)

print("Shape of Y Train:", train_y.shape)

print("Shape of Y Test:", test_y.shape)

from sklearn.linear_model import LogisticRegression

model = LogisticRegression()

model.fit(train_x, train_y)

pred_y = model.predict(test_x)

#Model Performance

from sklearn.metrics import classification_report

cr = classification_report(test_y, pred_y)

print(cr)

Using Logistic Regression To Generate Output

```
prediction =model.predict((np.array([[90,40,40,20,80,7,200]])))

print("The recommended Crop for Climatic Condition :", prediction)

prediction =model.predict((np.array([[50,10,30,30,40,40,100]])))

print("The recomended Crop for Given Climatic Condition :", prediction)
```