**A Project/Dissertation Report**

on

# AGRICULTURAL PRODUCTION OPTIMIZATION

***Submitted in partial fulfillment of the***
***requirement for the award of the degree of***

## B.Tech in Computer Science Engineering

GALGOTIAS UNIVERSITY
(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

**Under The Supervision of**
**Mr. Ajay Shankar**
**Associate Professor**

Submitted By

Ashutosh Rana (20scse1010802)
Monal Raj Singh (20scse1010821)

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**
**GALGOTIAS UNIVERSITY, GREATER NOIDA, INDIA**
**DECEMBER, 2023**

## SCHOOL OF COMPUTER APPLICATION AND TECHNOLOGY
## GALGOTIAS UNIVERSITY, GREATER NOIDA

# CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled "AGRICULTURAL PRODUCTION OPTIMIZATION" in partial fulfillment of the requirements for the award of the B.Tech submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of September 2023 to December 2023, under the supervision of Mr. Ajay Shankar, Associate Professor, Department of Computer Science and Engineering , of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by us for the award of any other degree of this or any other places.

Ashutosh Rana (20scse1010802)
Monal Raj Singh (20scse1010821)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Mr. Ajay Shankar
Associate Professor

# CERTIFICATE

This is to certify that Project Report entitle"Agricultural Production Optimization" which is submitted by Ashutosh Rana & Monal Raj Singh in partial fulfillment of the requirement for the award of degree B. Tech. in Department of School of Computing Science and Engineering Department of Computer Science and Engineering

Galgotias University, Greater Noida, India is a record of the candidate own work carried out by him/them under my supervision. The matter embodied in this thesis is original and has not been submitted for the award of any other degree

**Signature of Examiner(s)** **Signature of Supervisor(s)**

**Signature of Program Chair** **Signature of Dean**

Date: 5th December, 2023

Place: Greater Noida

# ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe special debt of gratitude to Professor Mr. Ajay Shankar, Department of Computer Science & Engineering, Galgotias University, Greater Noida, India for his constant support and guidance throughout the course of our work. His/Her sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of Professor Head, Department of Computer Science & Engineering, Galgotias University, Greater Noida, India for his full support and assistance during the development of the project.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Name : Ashutosh Rana
Roll No.: 20131010846
Date :05/12/2023

Name : Monal Raj Singh
Roll No.: 20131010862
Date : 05/12/2023

## Table of Contents

# ABSTRACT

In the economic sector agriculture plays a vital role. Day by day the population is increasing on a large scale with this increases the demand of food. The early methods used by farmers are not sufficient enough to full fill today's requirement, thus new methods are invented which in return brings employment for people. Machine learning Technology in agriculture has helped humans a lot such as identifying particular climate for particular crop similarly, it's soil type, pH value and water supply to the crop. The project consists of implementing a new method for different crop at similar time for larger productivity by predicting it accurately. In this project we are Building a Predictive Model so as to suggest the most suitable crops to grow based on the available Climate and Soil conditions.

# List of Figures

# TABLE OF CONTENTS

# Introduction

As we all know that agriculture depends largely on the nature of soil and the climatic conditions and many a times, we face unpredictable changes in climate like, non-seasonal rainfall or heat waves or fluctuations in humidity levels, etc. and all such events cause a great loss to our farmers and farming, because of which they are not able to utilize their agricultural land to it's fullest .So to solve all such problems, I have build a Machine Learning Model by the virtue of which we can help farmers, optimize the agricultural production, because this predictive model will help them understand that for a particular soil & given climatic condition, which crop will be best suitable for the harvest.

There are 7 key factors that I've taken into account which will help us in determining, exactly which crop should be grown and at what period of time, viz. Amount of Nitrogen, Phosphorus and Potassium in soil, Temperature in degree Celsius, Humidity, pH and Rainfall.

Machine learning approaches are essential for us to take care of numerous issues. In this paper, we present machine learning models and structures in detail. Machine learning has numerous applications innumerous perspectives and has incredible advancement potential.

In future, it is predictable that machine learning could set up ideal speculations to clarify its exhibitions. In the meantime, its capacities of unsupervised learning will be improved since there is much information on the planet however it isn't relevant to add names to every one of them. It is additionally anticipated that neural system structures will turn out to be increasingly unpredictable with the goal that they can separate all the more semantically important highlights. In addition, profound learning will consolidate with support adapting better and we can utilize this point of interest to achieve more assignments.

## 1.1 Setup Checklist for Project

**Minimum System Requirements**

- Microsoft Excel
- Microsoft Windows 2010 or above.
- Memory: 4gb of RAM (4gb or more recommended)
- Internet Explorer 10 or higher
- SQL Server 2012 or higher.
- Tableau 10 or higher /SSRS / Power BI/ QlikView

## 1.2 Instructions

- The code modules in the mini project should follow all the coding standards.
- Create a directory by your name in drive **<drive>**. In this directory, create a subdirectory **Project**. Store your Project here.
- You can refer to your course material.
- You may also look up the help provided in the BI docs and documentation provided in respective tools.

## 1.3 Background and Subjects Declarations

During my internship, I got a chance to work in the company called (DELISIS INFORMATICS) as a Data Analyst Intern to know about how a software company uses big data in applications used by various public institutions, so the department which I was working on naturally dealing with massive volume of data that concerns data science.

- **Data science** is an interdisciplinary field that uses scientific methods, processes, algorithms, and systems to extract value from data.

Data scientists combine a range of skills4including statistics, computer science, and business knowledge, machine learning, deep learning 4 to analyze data collected from the web, smartphones, customers, sensors, and other source. These data sets are so voluminous that traditional data processing software just can't manage them, so big data is becoming one of the most important technology trends that has the potential for

dramatically changing the way organizations use information to enhance the customer experience and transform their models. For example,

· **Big Data visualization** is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data. This mass of data is useless unless we analyse it and find the patterns hidden within.

· **Machine Learning** techniques are used to automatically find the valuable underlying patterns within complex data that we would otherwise struggle to discover. We'll classify it supervised and unsupervised learning.

## 1.3 Outline of the Report

**Section II: DATA SCIENCE**

**I.** Definitions and Declarations about Data Science
**II.** Tools and Platforms for Data Scientists
**III.** How Data Science Is Transforming Business?
**IV.** How Data Science Is Conducted?
**V.** Data Science and the Growth of Data
**VI.** Definition of Big Data
**VII.** Characteristics of Big Data
**VIII.** Big Data Visualization
**IX.** Hands-on Big Data Visualization

**Section III: MACHINE LEARNING**

**I.** Definitions and Declarations about Machine Learning

# 1.3.1. DATA SCIENCE

### 1.3.1. Definitions and Declarations about Data Science

Data science is the future of Artificial Intelligence. Therefore, it is very important to understand what is Data Science and how can it add value to your business. Traditionally, the data that we had was mostly structured and small in size, which could be analyzed by using the simple tools like SQL, PostgreSQL, Oracle, etc.

Unlike data in the traditional systems which was mostly structured, today most of the data is unstructured or semi-structured. This data is generated from different sources like financial logs, text files, multimedia forms, sensors and instruments.

Simple tools mentioned above are not capable of processing this huge volume and variety of data. This is why we need more complex and advanced analytical tools and algorithms for processing, analyzing and drawing meaningful insights out of it. Data science reveals trends and produces insights that businesses can use to make better decisions and create more innovative products and services. Data is the bedrock of innovation, but its value comes from the information data scientists can glean from it and then act upon.

## 1.3.2. Tools and Platforms for Data Scientists

### 1.3.2.1. Data Collection Tools

Collecting quality data that can be transformed into rich analysis is the starting point every data strategy. The right data collection tools can reduce errors and duplicates, ensure greater accuracy, and preserve the integrity of data coming from all sources.
·GoSpotCheck, IBM Datacap, Mozenda, Octoparse, etc.

### 1.3.2.2. Data Analysis Tools

Finding meaning in and extracting value from your data is the core of all data analysis

tools that enable you to easily understand and derive real meaning from your data help you male right business decisions that impact revenue and competitiveness.

· Alteryx, Domino Data Lab, KNIME Analytics Platform, etc.

#### 1.3.2.3. Data Warehousing Tools

Data warehouses function as repositories for data that's been combined and integrated from multiple, disparate sources and then standardized for ease of use.

· Amazon RedShift, Google BigQuery, Microsoft Azure, MySQL, etc.

#### 1.3.2.4. Data Visualization Tools

Visual analytics tools identify patterns and trends in your data and help end users Understand and digest complex concepts.

· Google Fusion Tables, Microsoft Power BI, SAS, etc.

#### 1.3.2.5. Data Scientists Home: Kaggle

Kaggle is an online community of data scientists and machine learners, owned by Google LLC. Kaggle allows users to find and publish data sets, explore and build models in a web-based data-science environment work with other data scientists and machine learning engineers and enter competitions to solve data science challenges.

### 1.3.3. How Data Science Is Transforming Business?

Organizations are using data science teams to turn data into a competitive advantage by refining products and services. For example, companies analyze data collected from call centers to identify customers who are likely to churn, so marketing can take action to retain them. Logistics companies analyze traffic patterns companies analyze medical test data and reported symptoms to help doctors diagnose diseases earlier and treat them more effectively., weather conditions, and other factors to improve delivery speeds and reduce costs.

### 1.3.4 How Data Science is Conducted?

The process of analyzing and acting upon data is iterative rather than linear, but this is how the work typically flows for a data modelling.

project:

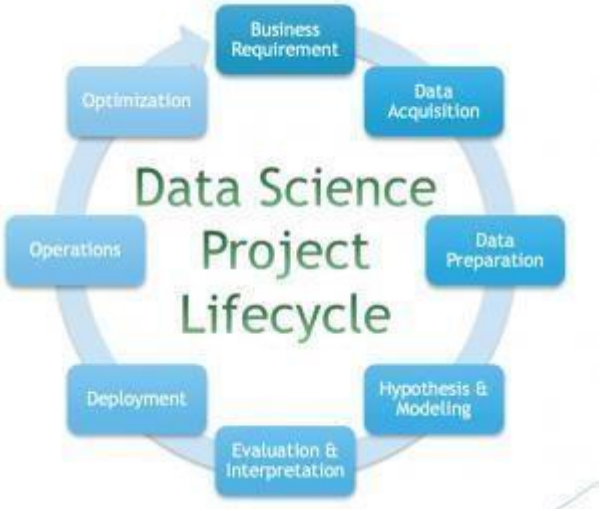



Figure 1.3.4 Data Science Project Lifecycle

### 1.3.5. Data Science and the Growth of Data

As modern technology has enabled the creation and storage of increasing amounts of information, the volume of data has soared. It's estimated that 90 percent of the data in the world was created in the last two years. For 10 example, Facebook users upload 10 million photos every hour. The number of connected devices in the world4the Internet of Things (IoT)4is projected to grow to more than 75 billion by 2025. The wealth of data being collected and stored by these technologies can bring transformative benefits to organizations societies around the world but only if we can interpret it. That's where data science comes in.

## 1.3.6. Big Data and Big Data Visualization

### 1.3.6. Big Data Definition and Characteristics

Big data is becoming one of the most important technology trends that has the potential for dramatically changing the way organizations use information to enhance the customer experience and transform their models. These data sets are so voluminous that traditional data processing software just can't manage them. But these massive volumes of data can be used to address business problems you wouldn't have been able to tackle before.

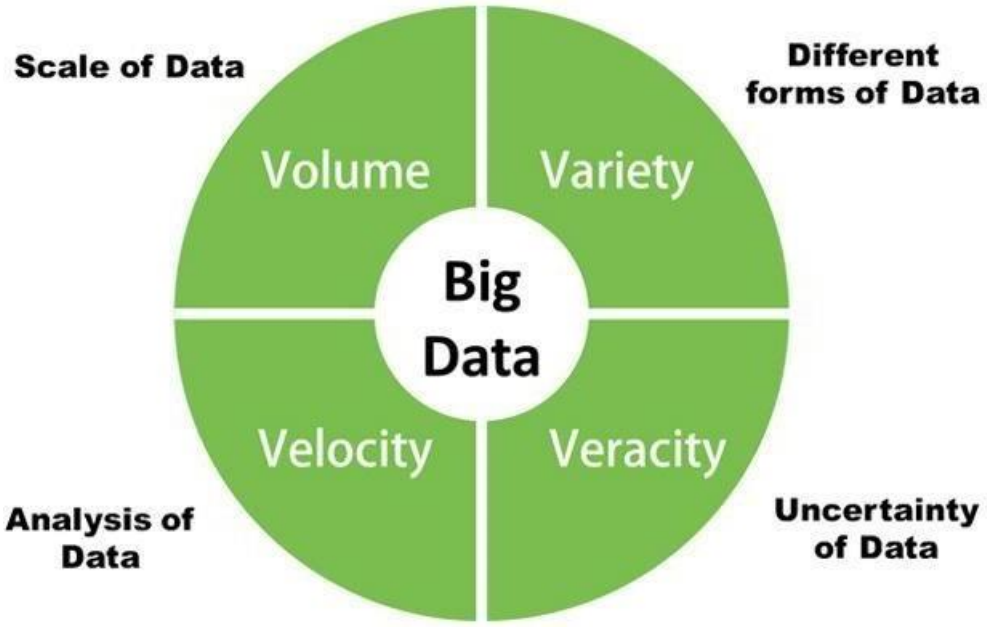



Figure 1.3.6 Characteristics of Big Data

### 1.3.8. Big Data Visualization

Because of the way the human brain processes information, using charts or graphs to visualize large amounts of complex data is easier than poring over spreadsheets or reports. Data visualization is a quick, easy way to convey concepts in a universal manner 3 and you can experiment with different scenarios by making slight adjustments.

Data visualization can also:

· Identify areas that need attention or improvement.

· Clarify which factors influence customer behavior.

· Help you understand which products to place where.

· Predict sales volumes.

### 1.3.9. Hands-on Big Data Visualization

In this section, I'll show you how to investigate a dataset from the very start step by step before projects part.

## 1. Import Required Libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
from ipywidgets import interact
```

Figure 1.3.9. How to import required libraries for visualization?

In this report I'll use python programming language and for visualization python has matplotlib, seaborn, pyplot, etc. libraries. Here in the above code segment we imported the required libraries for visualization. Numpy is imported for linear algebra operation, Pandasfor file operations, Seaborn and Matplotlib for visualization operations and others for structural requirements.

## 2. Reading the csv Files From Directory

```
df = pd.read_csv(r'C:\Users\mehul\OneDrive\Desktop\data.csv')
```

```
df
```

Figure 1.3.10. read_csv() method usage to read file from true directory.

We may have one or more datasets in separate files so they must be read by using pandas library one by one. Here the method is read csv used for file reading. And the kill variable is holding the dataframe after reading this file.

## 3. Understanding The Data

```
In [34]: df.head()
```

Out[34]:

| | 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 |

Figure 1.3.11. Examining the first 5 rows of sample dataset with head()

After reading the file we must understand the data, understanding the column names and datasets story is import for exploratory data analysis ( EDA ) .By using head() method we can see the first five rows of the dataset, and by using tail() we can see the last five columns.

## 4. Data Preparation and Cleaning

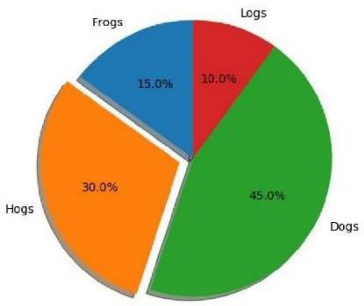
```
In [12]: df.isnull().sum()

Out[12]: N              0
         P              0
         K              0
         temperature    0
         humidity       0
         ph             0
         rainfall       0
         label          0
         dtype: int64
```

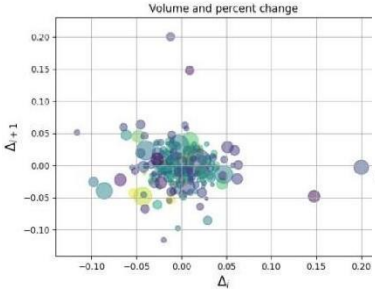
Figure 2.6. Checking for Null values we use isnull() method.

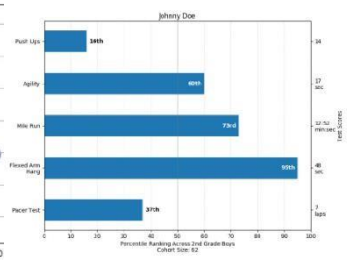
Data preparation and cleaning part is also important to meaningful exploratory data analysis. For example, here we use null. We can also use dropna (inplace=True) command we drop the null values permanently.

## 5. Visualization Tools

We have a lot of visualization tools in different libraries but just to demonstrate we'll use matplotlib for understanding types of charts. Some of the types we use in visualization part are like the following.



# 1.4 Machine Learning

### 1.4 What Is Machine Learning?

In the past 30 years there has been an explosion of data. This **mass of data is unless. we analyse it** and find the patterns hidden within. Machine Learning techniques are used to automatically find the valuable underlying patterns within complex data that we would otherwise struggle to discover. The hidden patterns and knowledge about a problem can be used to predict future events and perform all kinds of complex decision making.

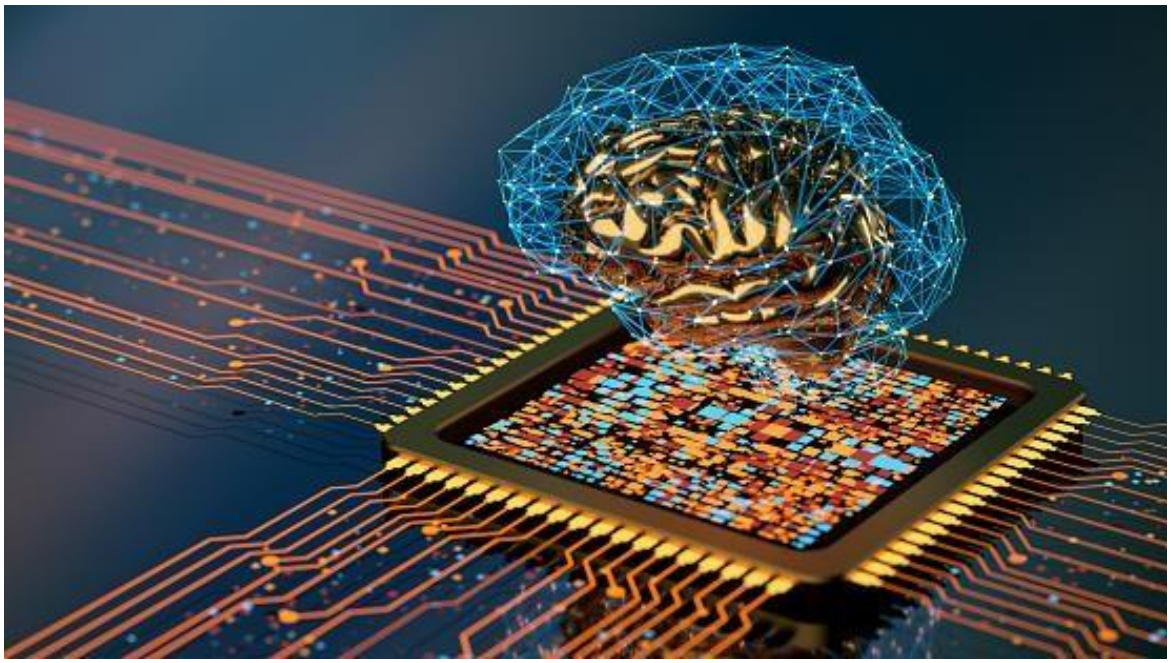


Figure 1.4 Machine Learning

## 1.4.1 Two Approaches To Machine Learning

### 1.4.1 Supervised Machine Learning

Supervised machine learning algorithms are the most commonly used. With this model, a data scientist acts as a guide and teaches the algorithm what conclusions it should make. Just as a child learns to identify fruits by memorizing them in a picture book, in supervised learning, the algorithm is trained by a dataset that is already labeled and has a predefined output. Examples of supervised machine learning include algorithms such as linear and logistic regression, multiclass classification, and support vector machines.

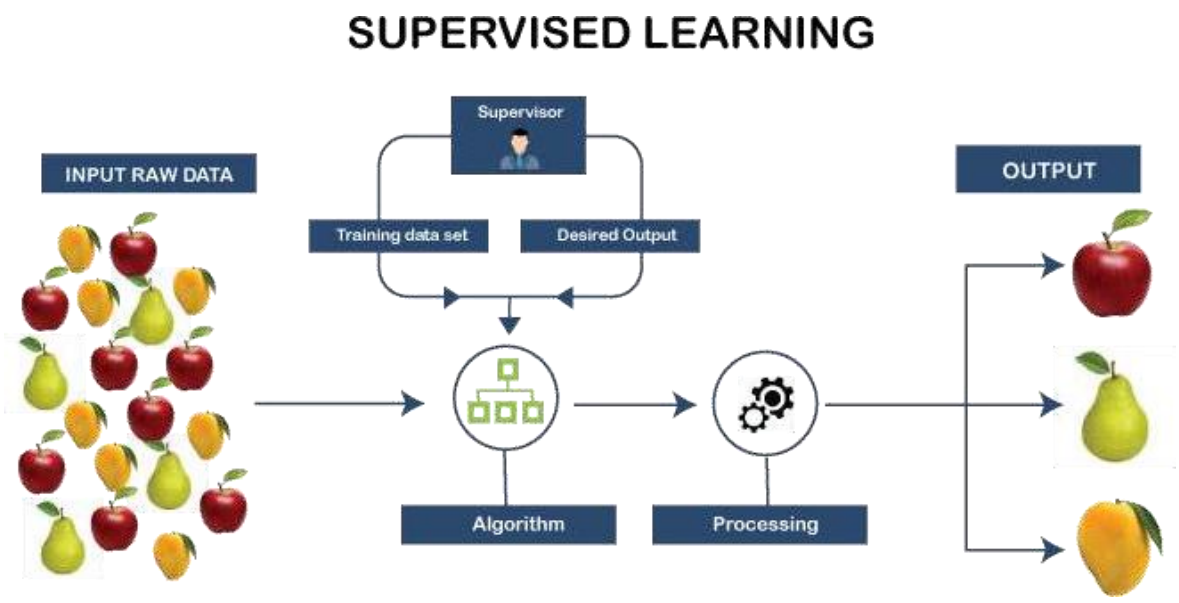



Figure 1.4.1 Supervised Learning

### 1.4.2 Unsupervised Machine Learning

Unsupervised machine learning uses a more independent approach, in which a computer learns to identify complex processes and patterns without a human providing close, constant guidance. Unsupervised machine learning involves training based on data that does not have labels or a specific, defined output. To continue the childhood teaching analogy, unsupervised machine learning is akin to a child learning to identify fruit by observing colors and patterns, rather than memorizing the names with a teacher's help.

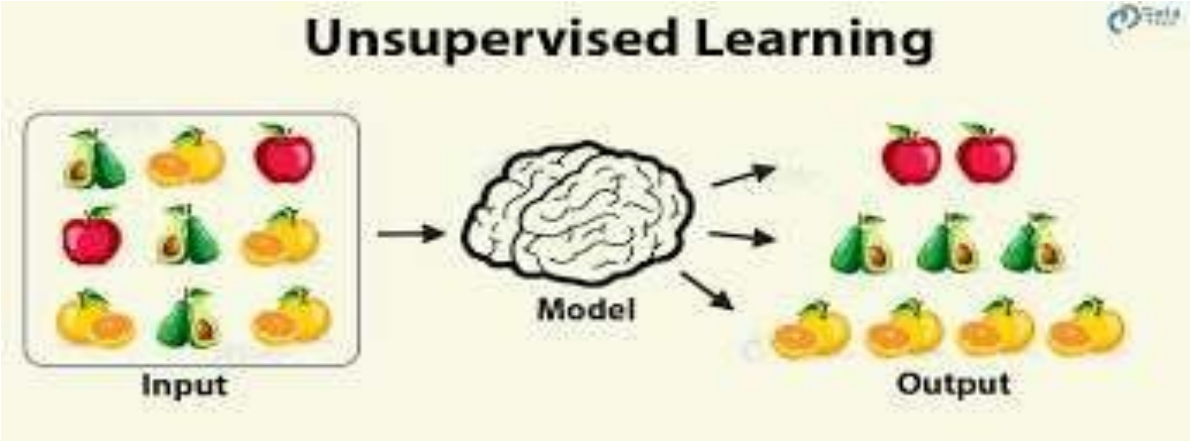



Figure 1.4.2 Unsupervised Learning

# 2 Literature Survey

There are several relevant literatures topic of Agriculture production optimization engine. The investigation has been done on crop yield prediction using Machine Learning approaches with special emphasis on Palm oil yield prediction. First and foremost, the agriculture production is important to Human Beings and as well as to Country Economics and the World, As our interest is in the determinants or building Agriculture production optimization engine, we work with Crops data and the requirement conditions for the Crop to grow.

A Comprehensive review of crop yield prediction using Machine Learning approaches with special emphasis on Agriculture production optimization engine prediction. In this project Mamunur Rashid, Bifta Sama Bari, Yusri Yusup, Mohamad Anvar Kamaruddin and Nuzhat Khan have collectively worked on predicting on Agriculture production. The Machine learning frameworks used offers a clear insight into the process by accessing vast sets of data and interpreting the obtained information. The model implementation of wide ranges of features like the difference in crop, location and intensity has also been observed. The selection of the features relies on the dataset's accessibility and the research objective.

# 3 Problem Statement

## 3.1 Objective

Development of Agriculture Production Optimization Engine

## 3.2 Abstract of the project

Build a Predictive Model so as to suggest the most suitable crops to grow based on the available Climate and Soil conditions.

## 3.2 Technology and Tools used:

- Python(libraries-Numpy,Pandas,Matplotlib,ipwidgets)
- Power BI

**Tools:**

- Jupyter Notebook
- Excel
- Machine Learning

# 4. PROGRAMMING LANGUAGE USED

**Python 3.8**

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. The version of python used in this project is 3.8 64-bit. A lot many packages were used too for the completion of the said project.


Figure 4.1 Python

# 5 TECHNOLOGY AND TOOLS

## 5.1 Jupiter Notebook

The Jupiter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupiter Notebook is maintained by the people at Project Jupyter.

Jupiter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupiter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupiter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.


Figure 5.1 Jupyter

## 5.3 Pandas

Pandas is an open-source library that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. This library is built on top of the NumPy library. Pandas is fast and it has high performance & productivity for users.

Pandas is one of the tools in Machine Learning which is used for data cleaning and analysis. It has features which are used for exploring, cleaning, transforming and visualizing from data.


Figure 5.3 Pandas

### 5.4 NumPy

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It is open-source software. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

### 5.5 Sklearn

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

Before we start using scikit-learn latest release, we require the following −

- Python (>=3.5)
- NumPy (>= 1.11.0)
- Scipy (>= 0.17.0)li
- Joblib (>= 0.11)
- Matplotlib (>= 1.5.1) is required for Sklearn plotting capabilities.
- Pandas (>= 0.18.0) is required for some of the scikit-learn examples using data structure and analysis.


Figure 5.5 Sklearn

## 5.6 Matplotlib

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002.

One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

Figure 5.6 Matplotlib


## 5.7 Seaborn

Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures.

Seaborn helps you explore and understand your data. Its plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots. Its dataset-oriented, declarative API lets you focus on what the different elements of your plots mean, rather than on the details of how to draw them.

Important Features of Seaborn

Seaborn is built on top of Python's core visualization library Matplotlib. It is meant to serve as a complement, and not a replacement. However, Seaborn comes with some very important features. Let us see a few of them here. The features help in −

- Built in themes for styling matplotlib graphics
- Visualizing univariate and bivariate data
- Fitting in and visualizing linear regression model
- Seaborn works well with NumPy and Pandas data structures
- It comes with built in themes for styling Matplotlib graphics


Figure 5.7 Seaborn

# 6 Implementation

## 6.1 Summary of the functionality to be built

The participants need to develop the Predictive model which will help farmers the

know which crops to grow based on climatic and soil Conditions.

## 6.2 Guidelines on the functionality to be built

**Project flow:**

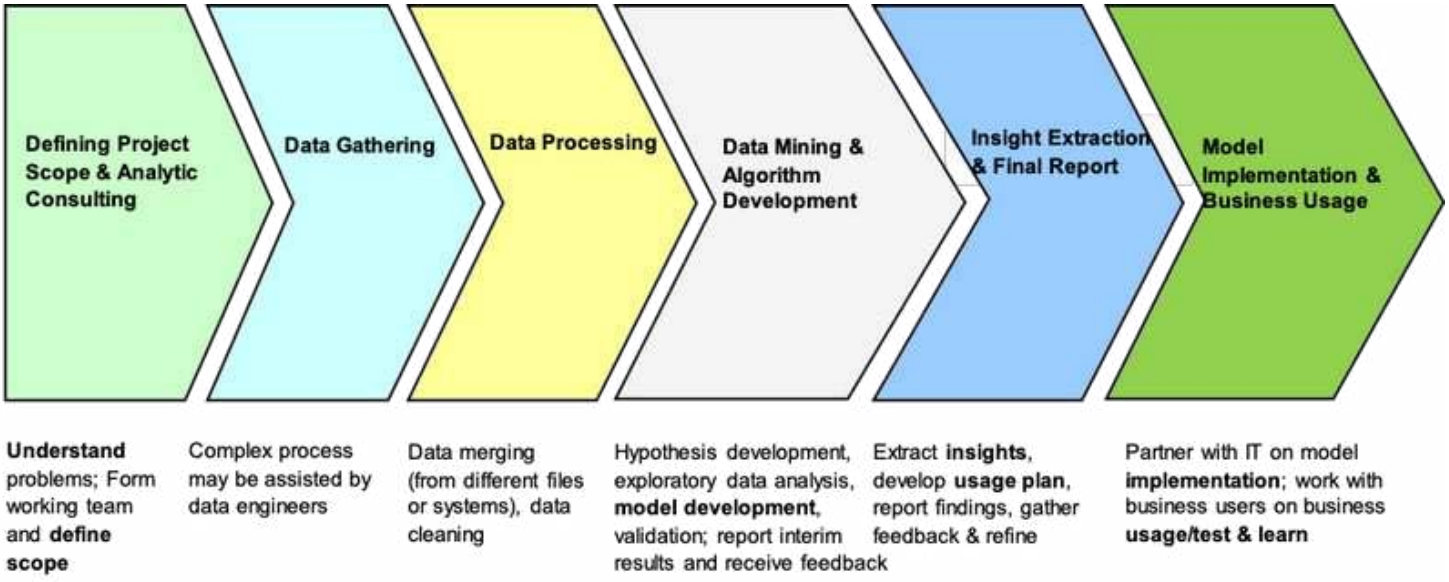



*Fig 6.2: Workflow of Project*

**Distribution for Agricultural Conditions:**

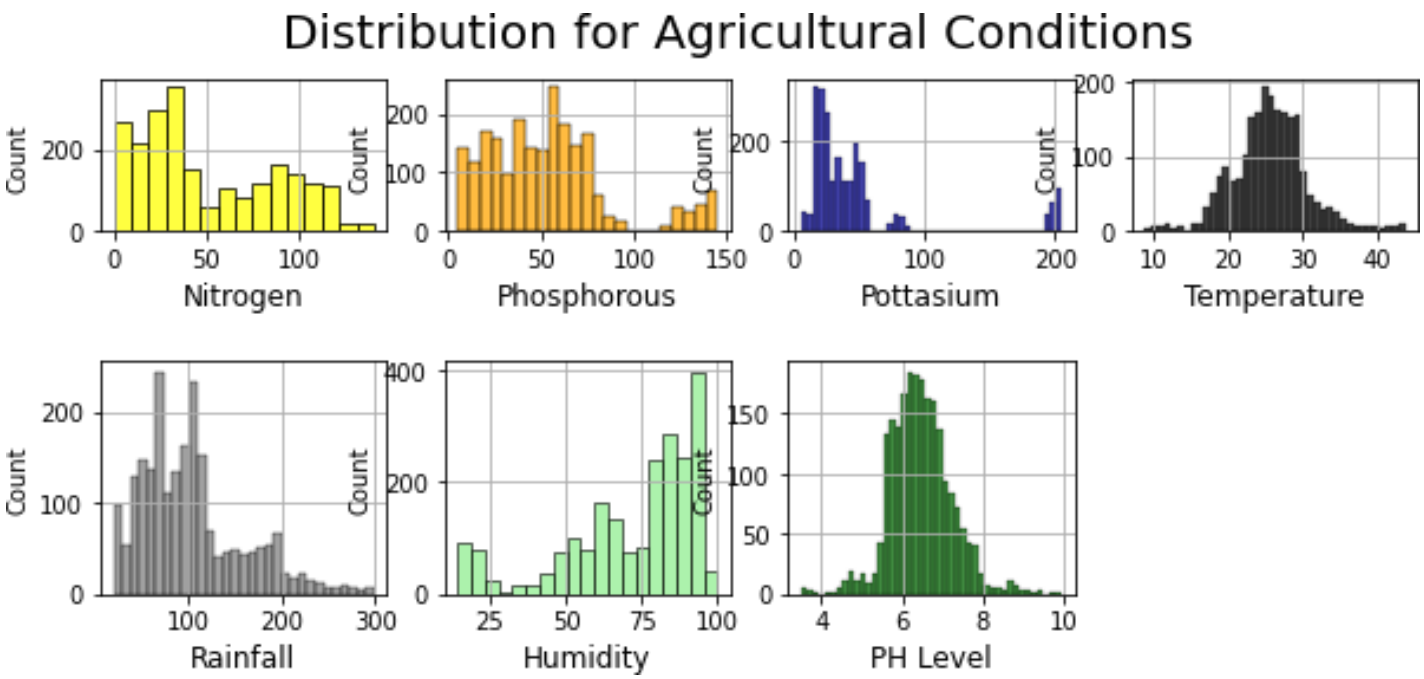



*Fig 6.3: Distribution for Agricultural Conditions*

## CODE:

### Importing the Modules

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
from ipywidgets import interact
```

### Reading the file from desktop

```
In [2]: df = pd.read_csv(r'C:\Users\mehul\OneDrive\Desktop\data.csv')
```

```
In [3]: df
```

Out[3]:

| | 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 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 2195 | 107 | 34 | 32 | 26.774637 | 66.413269 | 6.780064 | 177.774507 | coffee |
| 2196 | 99 | 15 | 27 | 27.417112 | 56.636362 | 6.086922 | 127.924610 | coffee |
| 2197 | 118 | 33 | 30 | 24.131797 | 67.225123 | 6.362608 | 173.322839 | coffee |
| 2198 | 117 | 32 | 34 | 26.272418 | 52.127394 | 6.758793 | 127.175293 | coffee |
| 2199 | 104 | 18 | 30 | 23.603016 | 60.396475 | 6.779833 | 140.937041 | coffee |

2200 rows × 8 columns

**Exploratory Data Analysis ( EDA ):**

Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and
graphical representations.

**EDA explained using sample Data set:**

```
In [4]: df.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 [5]: df.tail()
```

Out[5]:

| | N | P | K | temperature | humidity | ph | rainfall | label |
|---|---|---|---|---|---|---|---|---|
| 2195 | 107 | 34 | 32 | 26.774637 | 66.413269 | 6.780064 | 177.774507 | coffee |
| 2196 | 99 | 15 | 27 | 27.417112 | 56.636362 | 6.086922 | 127.924610 | coffee |
| 2197 | 118 | 33 | 30 | 24.131797 | 67.225123 | 6.362608 | 173.322839 | coffee |
| 2198 | 117 | 32 | 34 | 26.272418 | 52.127394 | 6.758793 | 127.175293 | coffee |
| 2199 | 104 | 18 | 30 | 23.603016 | 60.396475 | 6.779833 | 140.937041 | coffee |

Original data is separated by delimiter < ; < in given data set. To take a closer look at the data took help of <.head()=function of pandas library which returns first five observations of the data set. Similarly <.tail()=returns last five observations of the data set. We found out the total number of rows and columns in the data set using <.shape=.

```
In [6]: rows , columns =df.shape

In [7]: rows
Out[7]: 2200

In [8]: columns

Out[8]: 8
```

## Checking type of DataFrame:

```
In [9]: type(df)
Out[9]: pandas.core.frame.DataFrame

In [10]: df.columns
Out[10]: Index(['N', 'P', 'K', 'temperature', 'humidity', 'ph', 'rainfall', 'label'], dtype='object')
```

```
In [11]: df.info() #in info you will get to know the data type

<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)
memory usage: 137.6+ KB
```

Data has only float and integer values. No variable column has null/missing values. The describe() function in pandas is very handy in getting various summary statistics. This function returns the count, mean, standard deviation, minimum and maximum values and the quantiles of the data.

## Checking Missing or Null Values:

```
In [12]: df.isnull().sum()

Out[12]: N              0
         P              0
         K              0
         temperature    0
         humidity       0
         ph             0
         rainfall       0
         label          0
         dtype: int64
```

## Describe:

```
In [13]: df.describe()
```

Out[13]:

| | N | P | K | 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 |

## Label (Type):

```
In [14]: df['label'].value_counts()

Out[14]: rice          100
         maize         100
         jute          100
         cotton        100
         coconut       100
         papaya        100
         orange        100
         apple         100
         muskmelon     100
         watermelon    100
         grapes        100
         mango         100
         banana        100
         pomegranate   100
         lentil        100
         blackgram     100
         mungbean      100
         mothbeans     100
         pigeonpeas    100
         kidneybeans   100
         chickpea      100
         coffee        100
         Name: label, dtype: int64
```

## Summary of Crops:

### Summary of Crops

```
In [15]: print('Average Ratio of Nitrogen in the Soil: {0:.2f}'.format(df['N'].mean()))
         print('Average Ratio of Phosphorous in the Soil: {0:.2f}'.format(df['P'].mean()))
         print('Average Ratio of Potassium in the Soil: {0:.2f}'.format(df['K'].mean()))
         print('Average Ratio of Temperature in Celisus: {0:.2f}'.format(df['temperature'].mean()))
         print('Average Ratio of Humidity in the Soil: {0:.2f}'.format(df['humidity'].mean()))
         print('Average Ratio of Ph in the Soil: {0:.2f}'.format(df['ph'].mean()))
         print('Average Ratio of Rainfall in the Soil: {0:.2f}'.format(df['rainfall'].mean()))

         Average Ratio of Nitrogen in the Soil: 50.55
         Average Ratio of Phosphorous in the Soil: 53.36
         Average Ratio of Potassium in the Soil: 48.15
         Average Ratio of Temperature in Celisus: 25.62
         Average Ratio of Humidity in the Soil: 71.48
         Average Ratio of Ph in the Soil: 6.47
         Average Ratio of Rainfall in the Soil: 103.46
```

## Checking(Minimum, Average, Maximum):

In [16]:
```
@interact
def summary(Crops =list(df['label'].value_counts().index)):
    x = df[df["label"] == Crops]
    print("--------------------------------------------------")
    print("Statistics for Nitrogen")
    print('Minimum Nitrogen Required:',x['N'].min())
    print('Average Nitrogen Required:',x['N'].mean())
    print('Maximum Nitrogen Required:',x['N'].max())
    print("--------------------------------------------------")
    print('Statistics for Phosphorous')
    print('Minimum Phosphorous Required:',x['P'].min())
    print('Average Phosphorous Required:',x['P'].mean())
    print('Maximum Phosphorous Required:',x['P'].max())
    print("--------------------------------------------------")
    print('Statistics for Potassium')
    print('Minimum Potassium Required:',x['K'].min())
    print('Average Potassium Required:',x['K'].mean())
    print('Maximum Potassium Required:',x['K'].max())
    print("--------------------------------------------------")
    print("Statistics for Temperature")
    print("Minimum Temperature Required :{0:.2f}".format(x['temperature'].min()))
    print("Average Temperature Required :{0:.2f}".format(x['temperature'].mean()))
    print("Maximum Temperature Required :{0:.2f}".format(x['temperature'].max()))
    print("--------------------------------------------------")
    print("Statistics for Humidity")
    print("Minimum humidity Required :{0:.2f}".format(x['humidity'].min()))
    print("Average humidity Required :{0:.2f}".format(x['humidity'].mean()))
    print("Maximum humidity Required :{0:.2f}".format(x['humidity'].max()))
    print("--------------------------------------------------")
    print("Statistics for PH")
    print("Minimum PH Required :{0:.2f}".format(x['ph'].min()))
    print("Average PH Required :{0:.2f}".format(x['ph'].mean()))
    print("Maximum PH Required :{0:.2f}".format(x['ph'].max()))
    print("--------------------------------------------------")
    print("Statistics for RainFall")
    print("Minimum Rainfall Required :{0:.2f}".format(x['rainfall'].min()))
    print("Average Rainfall Required :{0:.2f}".format(x['rainfall'].mean()))
    print("Maximum Rainfall Required :{0:.2f}".format(x['rainfall'].max()))
```

## Compare the Average Requirement for each crops with average:

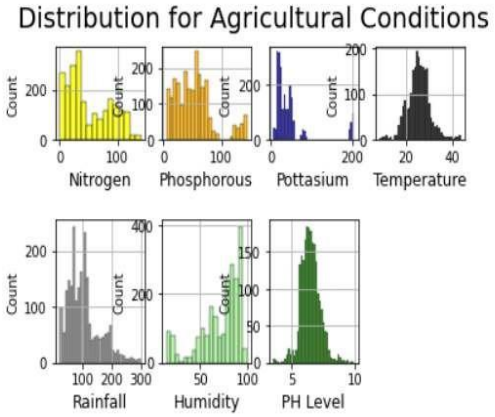
### Compare the Average Requirement for each crops with average Conditions

In [18]:
```
@interact
def compare(conditions =['N','P','K','temperature','humidity','ph','rainfall']):
    print("Average value for",conditions,"is {0:.2f}".format(df[conditions].mean()))
    print("--------------------------------------------------")
    print("Rice : {0:.2f}".format(df[(df['label'] =='rice')][conditions].mean()))
    print("maize : {0:.2f}".format(df[(df['label'] =='maize')][conditions].mean()))
    print("jute : {0:.2f}".format(df[(df['label'] =='jute')][conditions].mean()))
    print("cotton : {0:.2f}".format(df[(df['label'] =='cotton')][conditions].mean()))
    print("papaya : {0:.2f}".format(df[(df['label'] =='papaya')][conditions].mean()))
    print("orange : {0:.2f}".format(df[(df['label'] =='orange')][conditions].mean()))
    print("apple : {0:.2f}".format(df[(df['label'] =='apple')][conditions].mean()))
    print("muskmelon : {0:.2f}".format(df[(df['label'] =='muskmelon')][conditions].mean()))
    print("watermelon : {0:.2f}".format(df[(df['label'] =='watermelon')][conditions].mean()))
    print("grapes : {0:.2f}".format(df[(df['label'] =='grapes')][conditions].mean()))
    print("mango : {0:.2f}".format(df[(df['label'] =='mango')][conditions].mean()))
    print("banana  : {0:.2f}".format(df[(df['label'] =='banana ')][conditions].mean()))
    print("pomegranate : {0:.2f}".format(df[(df['label'] =='pomegranate')][conditions].mean()))
    print("lentil : {0:.2f}".format(df[(df['label'] =='lentil')][conditions].mean()))
    print("blackgram : {0:.2f}".format(df[(df['label'] =='blackgram')][conditions].mean()))
    print("mungbean  : {0:.2f}".format(df[(df['label'] =='mungbean')][conditions].mean()))
    print("mothbeans : {0:.2f}".format(df[(df['label'] =='mothbeans ')][conditions].mean()))
    print("pigeonpeas : {0:.2f}".format(df[(df['label'] =='pigeonpeas')][conditions].mean()))
    print("kidneybeans : {0:.2f}".format(df[(df['label'] =='kidneybeans')][conditions].mean()))
    print("chickpea : {0:.2f}".format(df[(df['label'] =='chickpea')][conditions].mean()))
    print("coffee : {0:.2f}".format(df[(df['label'] =='coffee')][conditions].mean()))
```

```
interactive(children=(Dropdown(description='conditions', options=('N', 'P', 'K', 'temperature', 'humidity', 'p…
```

## Checking distribution for each crop:

In [19]:
```
#Checking distributiion for each crop

plt.subplot(3,4,1)
sns.histplot(df['N'], color="yellow")
plt.xlabel('Nitrogen', fontsize = 12)
plt.grid()

plt.subplot(3,4,2)
sns.histplot(df['P'], color="orange")
plt.xlabel('Phosphorous', fontsize = 12)
plt.grid()

plt.subplot(3,4,3)
sns.histplot(df['K'], color="darkblue")
plt.xlabel('Pottasium', fontsize = 12)
plt.grid()

plt.subplot(3,4,4)
sns.histplot(df['temperature'], color="black")
plt.xlabel('Temperature', fontsize = 12)
plt.grid()

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

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

plt.subplot(2,4,7)
sns.histplot(df['ph'], color="darkgreen")
plt.xlabel('PH Level', fontsize = 12)
plt.grid()

plt.suptitle('Distribution for Agricultural Conditions', fontsize = 20)
plt.show()
```



## Checking which crop grow in Summer, winter and rain season:

### which crop can grow in summer ,winter and rain season

In [21]:
```
print("Summer Season")
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("Rainy Crops")
print(df[(df['rainfall'] > 200) & (df['humidity'] > 30)]['label'].unique())
```

```
Summer Season
['pigeonpeas' 'mothbeans' 'blackgram' 'mango' 'grapes' 'orange' 'papaya']
-----------------------------------------------------
Winter Crops
['maize' 'pigeonpeas' 'lentil' 'pomegranate' 'grapes' 'orange']
-----------------------------------------------------
Rainy Crops
['rice' 'papaya' 'coconut']
```

## Checking which crop require above or below average condition:

### we are checking which crop require above or below average condition

In [20]:
```
@interact
def compare(conditions =['N','P','K','temperature','humidity','ph','rainfall']):
    print("Crops which require greater than average",conditions,'\n')
    print(df[df[conditions] > df[conditions].mean()]['label'].unique())
    print("------------------------------------------------------------------------------------------")
    print("Crops require Lesser then average",conditions,'\n')
    print(df[df[conditions] <= df[conditions].mean()]['label'].unique())
```

```
interactive(children=(Dropdown(description='conditions', options=('N', 'P', 'K', 'temperature', 'humidity', 'p…
```

### print("Patterns")

```
print("----------------------------------------------") print("Crops which require very High ratio of Nitrogen Content in soil:",df[df['N'] > 120]['label'].unique()) print("Crops which require very High ratio of Phosphorous Content in soil:",df[df['P'] > 100]['label'].unique()) print("Crops which require very High ratio of Potassium Content in soil:",df[df['K'] > 200]['label'].unique()) print("Crops which require very High Rainfall:",df[df['rainfall'] > 200]['label'].unique()) print("Crops which require very Low Temperature:",df[df['temperature'] < 10]['label'].unique()) print("Crops which require very High Temperature:",df[df['temperature'] > 40]['label'].unique()) print("Crops which require very Low Humidity:",df[df['humidity'] < 20]['label'].unique()) print("Crops which require very low PH:",df[df['ph'] < 4]['label'].unique()) print("Crops which require very High PH :",df[df['ph'] > 9]['label'].unique())
```

## Clustering Analysis:

In [22]:
```
#using clustering analyics
#clustering alanyics is a technique used to classficy the data points into realittive groups called cluster
#using k means clustering
```

In [23]:
```
#lets import the  warnings library so that we can avoid warnings

from sklearn.cluster import KMeans
import warnings
warnings.filterwarnings('ignore')

#lets select the spending score,and annal income from the data
x =  df.loc[:,['N','P','K','temperature','ph','humidity','rainfall']].values

#let's check the shape of x
print(x.shape)

#lets convert this data into a dataframe
x_data = pd.DataFrame(x)
x_data.head()
```

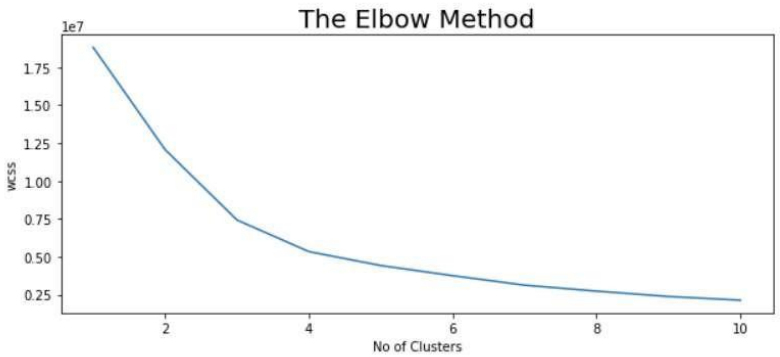
```
(2200, 7)
```

Out[23]:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| 0 | 90.0 | 42.0 | 43.0 | 20.879744 | 6.502985 | 82.002744 | 202.935536 |
| 1 | 85.0 | 58.0 | 41.0 | 21.770462 | 7.038096 | 80.319644 | 226.655537 |
| 2 | 60.0 | 55.0 | 44.0 | 23.004459 | 7.840207 | 82.320763 | 263.964248 |
| 3 | 74.0 | 35.0 | 40.0 | 26.491096 | 6.980401 | 80.158363 | 242.864034 |
| 4 | 78.0 | 42.0 | 42.0 | 20.130175 | 7.628473 | 81.604873 | 262.717340 |

## Determine the optimize number of cluster within the dataset:

In [25]:
```
#Determine the optimum number of cluster within the dataset

plt.rcParams['figure.figsize'] =(10,4)

wcss =[]
for i in range(1,11):
    km = KMeans(n_clusters = i ,init ='k-means++',max_iter = 300,n_init = 10,random_state = 0)
    km.fit(x)
    wcss.append(km.inertia_)

#Ploting the result
plt.plot(range(1,11),wcss)
plt.title('The Elbow Method',fontsize = 20)
plt.xlabel('No of Clusters')
plt.ylabel("wcss")
plt.show()
```



## Implement the k means algorithm to perform clustering analysis:

No of Clusters

In [26]:
```
#implement the k mean algorithm to perform clustering analysis
km = KMeans(n_clusters=4,init='k-means++',max_iter=300,n_init=10,random_state=0)
y_means =km.fit_predict(x)

#Lets find out the Results
a = df['label']
y_means = pd.DataFrame(y_means)
z = pd.concat([y_means,a],axis = 1)
z = z.rename(columns ={0:'cluster'})
#Lets check the cluster of each Crops
print('Lets check the result After Applying the k means clustering Analyics \n')
print('Crops in first cluster:',z[z['cluster']==0]['label'].unique())
print('-----------------------------------------------------------')
print('Crops in second cluster:',z[z['cluster']==1]['label'].unique())
print('-----------------------------------------------------------')
print('Crops in Third cluster:',z[z['cluster']==2]['label'].unique())
print('-----------------------------------------------------------')
print('Crops in Fourth cluster:',z[z['cluster']==3]['label'].unique())
```

```
Lets check the result After Applying the k means clustering Analyics

Crops in first cluster: ['maize' 'chickpea' 'kidneybeans' 'pigeonpeas' 'mothbeans' 'mungbean'
 'blackgram' 'lentil' 'pomegranate' 'mango' 'orange' 'papaya' 'coconut']
-----------------------------------------------------------
Crops in second cluster: ['maize' 'banana' 'watermelon' 'muskmelon' 'papaya' 'cotton' 'coffee']
-----------------------------------------------------------
Crops in Third cluster: ['grapes' 'apple']
-----------------------------------------------------------
Crops in Fourth cluster: ['rice' 'pigeonpeas' 'papaya' 'coconut' 'jute' 'coffee']
```

## Building Predictive model:

### buliding preadictive model

In [27]:
```
# Split the Dataset for predicatice Modeling

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

print("shape of x: ",x.shape)
print("shape of y: ",y.shape)
```

```
shape of x:  (2200, 7)
shape of y:  (2200,)
```

In [28]:
```
#Creating training and testing sets for validitons of results
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,)
```

In [29]:
```
# lets create a Prediactive model

from sklearn.linear_model import LogisticRegression

model = LogisticRegression()
model.fit(x_train,y_train)
y_pred = model.predict(x_test)
```

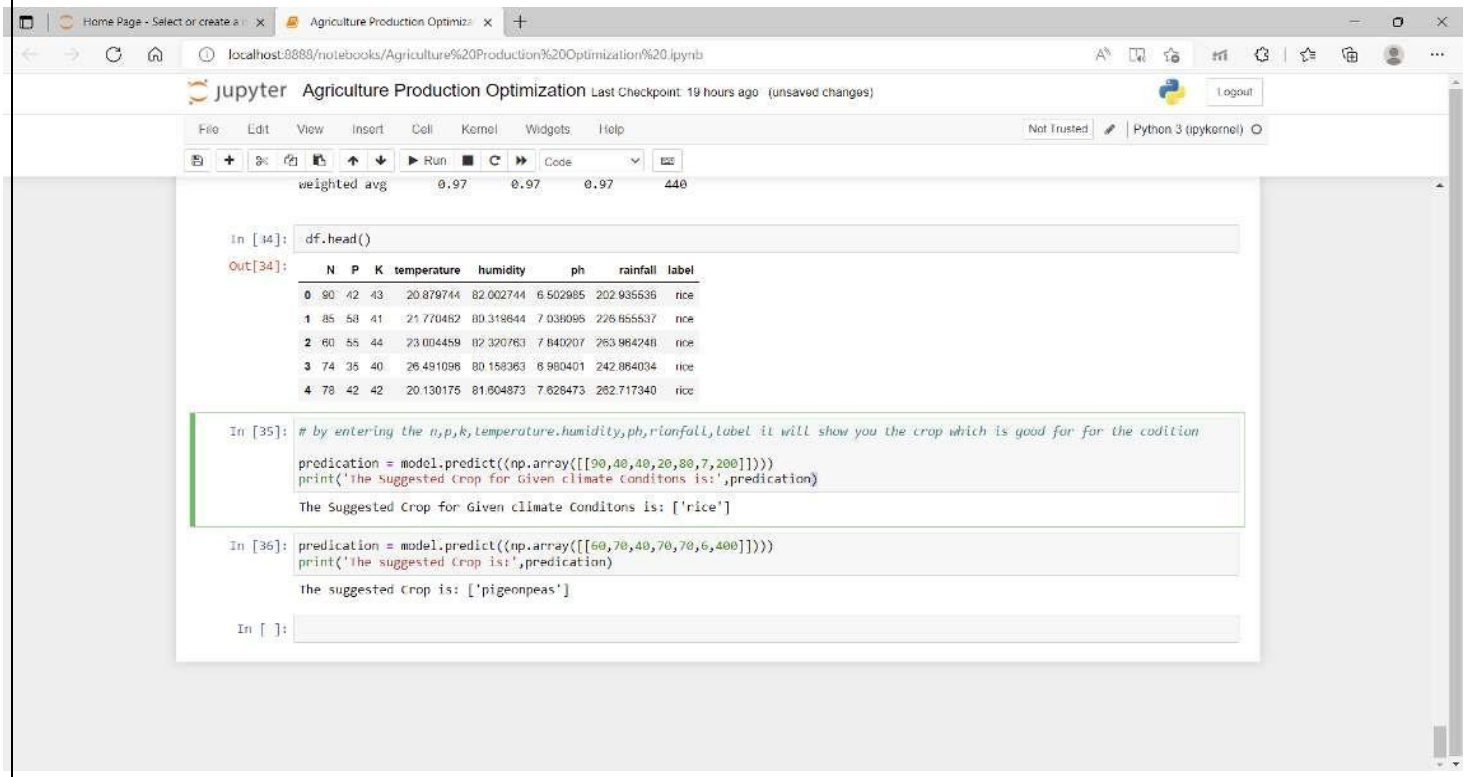
In [30]:
```
#Evaluating the model performance
from sklearn.metrics import confusion_matrix

#Printing the Confusing Matrix
plt.rcParams['figure.figsize'] = (10,10)
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm,annot = True, cmap = 'Wistia')
plt.title('Confusion Matrix For Logistic Regression', fontsize = 15)
plt.show()
```

## Model Performance:

In [31]:
```
#lets evaluate the model performance
from sklearn.metrics import classification_report
#lets priot 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
```

## Output:


Home Page - Select or create a
Agriculture Production Optimiz
localhost:8888/notebooks/Agriculture%20Production%20Optimization%20.ipynb
jupyter Agriculture Production Optimization Last Checkpoint: 19 hours ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
weighted avg 0.97 0.97 0.97 440
In [34]: df.head()
Out[34]:
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 [35]: # by entering the n,p,k,temperature,humidity,ph,rianfall,label it will show you the crop which is good for for the codition
predication = model.predict((np.array([[90,40,40,20,80,7,200]])))
print('The Suggested Crop for Given climate Conditons is:',predication)
The Suggested Crop for Given climate Conditons is: ['rice']
In [36]: predication = model.predict((np.array([[60,70,40,70,70,6,400]])))
print('The suggested Crop is:',predication)
The suggested Crop is: ['pigeonpeas']
In [ ]:

# 8 Conclusion

The main purpose of this project is to create a data which is trustworthy. And the data which we generated would become the golden record so that this data could be used across the organization. The data which is generated here is further used for the analysis purposes with the help of Tableau we can visualize the data and get more insights from it. This can also be useful for decision making purposes. The analysis can be used to generate reports using reporting tools like Tableau 10, Power BI, etc. Analysis helps in Agriculture Production.

# 9 References


1. C. Miller, V. N. Saroja, and C. Linder, *ICT Uses for Inclusive Agricultural Value Chains*, Food and agriculture organization of the united nations (FAO), Rome, Italy, 2013.

2. K. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, <Machine learning in agriculture: a review,= *Sensors*, vol. 18, no. 8, p. 2674, 2018.View at: Publisher Site | Google Scholar

3. 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,=2020

4. Kevin Tom Thomas, Varsha S, Merin Mary Saji, Lisha Varghese, Er. Jinu Thomas, <Crop Prediction Using Machine Learning, =2020

5. D. Bhanu Kiran, J. Priyanka K. Sri Poojitha, Afridi Khan, <Crop Yield Pre- diction Using Regression,=2020

6. Rishi R, S Kanagana Suba Raja, Sundaresan E, Srijit V, <Demand Based Crop Recommender System for Farmers,=2017

7. Yogita Masare, Sneha Mahale, Manjusha Kele, Ashvin Upadhyay, Bhushan R. Nanwalkar, <The System for Maximize the Yielding Rate of Crops using Machine Learning Algorithm,=2021

8. Abhijeet Pandhe, Praful Nikam, Vijay Pagare, Pavan Palle, Prof. Dilip Dalgade, <Crop Yield Prediction based on Climatic Parameters, =2019