CROP PREDICTION USING MACHINE LEARNING

A Report for the ETE Review

S. No	Enrolment Number	Admission Number	Student Name	Degree/ Branch	Sem
1.	201010975	20SCSE1010198	Prakhar Anand	B.Tech.(CSE)	5
2.	201010992	20SCSE1010460	Lokesh Sharma	B.Tech.(CSE)	5

Under the Supervision of

Dr. Arvind Dagur

Professor

School of Computing Science and Engineering

School of Computing Science and Engineering Greater Noida, Uttar Pradesh

Fall 2021 – 2022

DECLARATION

We hereby declare that this project report titled "Crop Prediction Using Machine Learning" is submitted for the award of degree of Bachelor in Technology to the "Department of Computing Science and Engineering".
It is a record of original work done by us under the guidance of **Dr. Arvind Dagur**, Professor, School of Computing Science and Engineering, Galgotias University.
The information and data given in the report is authentic to the best of my knowledge. This project report is not submitted to any other university or institute for the award of any degree, diploma or has been published any time before.

Prakhar Anand 20SCSE1010198
Lokesh Sharma 20SCSE1010460

ACKNOWLEDGEMENT

We would like to extend our gratitude to our guide and mentor Dr. Arvind Dagur, Professor, School of Computing Science and Engineering, Galgotias University, for guiding us throughout the project and giving us suggestions and imparting his ultimate knowledge on us. He gave us his valuable time and helped us to achieve our project goal**.**

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of Prakhar Anand 20SCSE1010198 and Lokesh Sharma 20SCSE1010460 has been held on 24-12-2021 and his/her work is recommended for the award of Bachelor in Technology in Computer Science and Engineering.

Signature of Examiner(s)

Signature of Supervisor(s)

Signature of Project Coordinator

Signature of Dean

Date: 24 December, 2021

Place: Greater Noida

TABLE OF CONTENTS:

List of Figures:

I. ABSTRACT

Agriculture is one of the most essential and widely practiced occupations in India and it has a vital role in the development of our country. Around 60 percent of the total land in the country is used for agriculture to meet the needs of 1.2 billion people, so improving crop production is therefore seen as a significant aspect of agriculture. Basically if we have a piece of land, we need to know what kind of crop can be grown in this area. Agriculture depends on the various soil properties. Production of crops is a difficult task since it involves various factors like soil type, temperature, humidity etc. But now-a-days, food production and prediction is getting depleted due to unnatural climatic changes, which will adversely affect the economy of farmers by getting a poor yield and also help the farmers to remain less familiar in forecasting the future crops. If it is possible to find the crop before sowing it, it would be of great help to the farmers and the other people involved to make appropriate decisions on the storage and business side. The proposed project would solve agricultural problems by monitoring the agricultural area on the basis of soil properties and recommending the most appropriate crop to farmers, thereby helping them to significantly increase productivity and reduce loss. Our project is a recommendation system which makes use of different machine learning techniques like unsupervised learning, predictive modelling, K- Means clustering, Logistic Regression, etc. such that it recommends the suitable crops based on the input soil parameters. The seed data of the crops are collected here, with the appropriate parameters like temperature, humidity and moisture content, which helps the crops to achieve a successful growth. This system thus reduces the financial losses faced by the farmers caused by planting the wrong crops and also it helps the farmers to find new types of crops that can be cultivated in their area.

Keywords: Machine Learning, Unsupervised learning, Predictive Modelling, K-Means Clustering, Logistics Regression.

II. INTRODUCTION

Agriculture has a major role in the lives of every individual. From the olden times itself agriculture is considered to be one of the main practices practiced in India. In olden times, people used to cultivate crops in their own land in order to meet their requirements. Being treated as India's backbone, the agricultural sector has strengthened with the public's needs as the technology is improving. With the rapid population growth, these innovations are very much needed to meet the needs of every person.

Our country had undergone several fluctuations in the price of onions in last couple of years. So most of the farmers decided to cultivate onion in their fields seeing this huge increase in price so that they could make large profits from their land. In some regions for example in Maharashtra this resulted in the abundant supply of onions while many other regions suffered a failed crop production and the farmers lost a large amount of money. This problem occurred due to many unfavourable conditions that prevented the growth of onions. A continuous shortage in the production of onions again in the next few months had a very bad effect on the lives of the common people. This happened because the middle-class people were not able to afford the huge price of onion which is a frequently used commodity in their houses.

The above example leads us to the formulation of our problem statement.

2.1 PROBLEM STATEMENT: The decision of a farmer regarding which type of crop to grow in his land generally depends on his intuition and many other factors such as making huge profits within a short period of time, lack of awareness about the demand in the market and when he overestimates a soil's potential to support the growth of a particular type of crop and many more. A wrong decision that is taken on the farmer's side could put a much bigger pressure on the financial condition of his family resulting in severe loss. For all this reason we can see how much pressure a farmer is under just to decide which crop should he grow in his land.

2.2 SOLUTION: So now the most important aspect is to design a recommendation system that predicts the type of crop that can be grown in a particular land and thereby helping the farmers. With this aim in mind we have decided to develop a system that takes in the soil parameters like N, P, K (Nitrogen, Phosphorus, Potassium) and the pH values and predicts the most

suitable crop that can be grown in that region. The dataset already contains the NPK and pH values of the soil and the appropriate or the most suitable crop that can be grown in given soil.

For our recommendation system, we use machine leaning algorithms. Learning is the concept of improving the regular performance or action that one performs based on the experience and knowledge gain through regular activities. Using this same concept for building a system that can automatically learn and improve through experience. Machine learning is based on the similar concept of learning, here a computer program is said to learn through the knowledge E (experience ie. past data) regarding some tasks T and performance P. Computer program is said to learn if its performance P improves at task T with experience E. In simple words machine learning is the concept where machine learns by it own without needed to be programmed explicitly. It is the application i.e. subset of artificial intelligence that have enabled the system to automatically learn and improve. Programs are also generated in way by integrating input and output of the program. There is methodological difference between machine learning and statistics. One difference is that machine learning focus on the optimization and performance and statistics concentrate on inference.

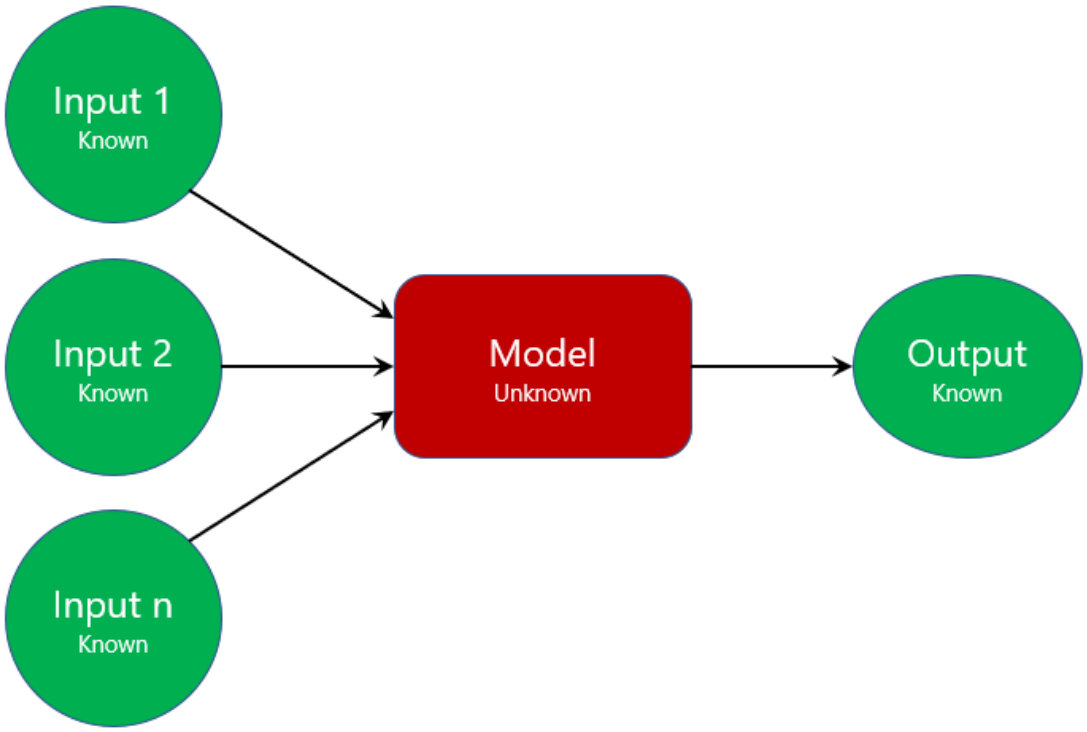


Fig. 1. General flow of Machine Learning

The above figure shows that for the machine learning solution the input and desired output or the target solution we want is to be given to the computer then the program or model is created to work on the test values or new data.

There are three common machine learning techniques: supervised, unsupervised, and reinforcement learning. This project uses unsupervised learning clustering techniques for prediction. In unsupervised learning, an AI system will group unsorted information according to similarities and differences even though there are no categories provided. Unsupervised learning algorithms can perform more complex processing tasks than supervised learning systems. Additionally, subjecting a system to unsupervised learning is one way of testing AI. The objective with unsupervised learning is to have the algorithms identify patterns within the training data sets and categorize the input objects based on the patterns that the system itself identifies. The algorithm analyzes the underlying structure of the data sets by extracting useful information or features from them. Thus, these algorithms are expected to develop specific outputs from the unstructured inputs by looking for relationships between each sample or input object.

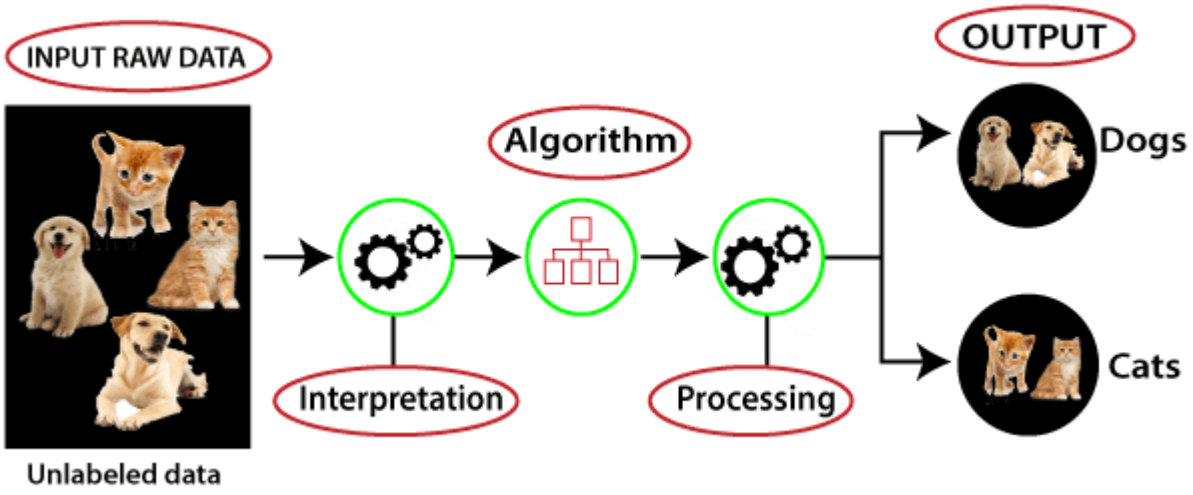


Fig. 2. Unsupervised Learning

Here, we have taken an unlabelled input data, which means it is not categorized and corresponding outputs are also not given. Now, this unlabelled input data is

fed to the machine learning model in order to train it. Firstly, it will interpret the raw data to find the hidden patterns from the data and then will apply suitable algorithms such as k-means clustering, Logistic Regression, etc. Once it applies the suitable algorithm, the algorithm divides the data objects into groups according to the similarities and difference between the objects.

Machine learning (ML) is a game changer for agriculture sector. Machine learning as the part of artificial intelligence, has emerged together with big data technologies and high-performance computing to create new opportunities for data intensive science in the multi-disciplinary agri-technology domain. In the Agriculture field machine learning for instance is not a mysterious trick or magic, it is a set of well defined model that collect specific data and apply specific algorithms to achieve expected results.

This project is intended on precision farming in order to optimize productivity by understanding requirements of Climatic and Soil conditions for the crops. The proposed system will help farmers to cope with Weather unpredictability.

III. LITERATURE SURVEY

The paper hypothesizes analysis of Explorative Data and considers the design of different types of predictive models. A data set is taken as a sample data set, and different regression techniques are tried to recognise and examine each property. Algorithms like K Nearest Neighbors, Naïve Bayes, KNN with Cross Validation were used on the data set to prdict the most suitable crop for cultivation**[1]**

The designed system recommended the most suitable crop for a particular land. Based on weather parameters and soil content such as Rainfall, Temperature, Humidity and pH. This all input data applied to machine learning predictive algorithms like Support Vector Machine (SVM) and Decision tree to identify the pattern among data and then process it as per input conditions. The system recommended the crop for the farmer and also recommended the amount of nutrients to be added for the predicted crop. The system had some other specifications like displaying approximated yield in q/acre, required seed for cultivation in kg/acre and the market price of the crop**[2]**

This paper provides a way for smart agriculture by monitoring the agricultural field which can assist the farmers in increasing productivity to a great extent. It also applies machine learning and prediction algorithm like Multiple Linear Regression to identify the pattern among data and then process it as per input conditions**[3]**

The purpose of this research paper is to propose and implement a rule based system to predict the crop yield production from the collection of past data. This has been achieved by applying association rule mining on agriculture data from 2000 to 2012**[4]**

The project focuses on the creation of a prediction model which may be used to predict the maximum production rate of the crop before sowing. The information about the farmer's state, district, season, land area, crop type is used for the estimating production rate of crops e by applying a machine learning algorithm on that data**[5]**

Based on the Literature Survey:

● About 60 % of the land in the India is used for agriculture in order to fulfil the needs of 1.3 billion people. And population is also increasing day by day. Thus, the modernization of agriculture is needed that will lead the farmers of our country towards profit and solve many problems of them.

● In Existing system farmers are not connected with any technology and analysis.

● In conventional system farmer uses “trial and error” method. Farmer tests on land with different crops, water availability, etc. and after many such “tries”, farmer probably gets the crop production as expected.

● Many papers have done the survey by taking into consideration different parameters.

● There are some systems which helps in selection of crop but no system is accurate.

● Some papers use data mining techniques in predicting the crop yield based on climatic input parameters. But only climatic conditions are not sufficient to predict crop yield.

● Some survey papers have analysed different machine learning algorithms that can be used in prediction of crop.

● There are many review papers on crop prediction that stated different algorithms that can be used for prediction. But no such system exists as of now. So, there is need to implement such system so that farmers will get benefit of it.

IV. DESIGN OF PROPOSED SYSTEM

The Proposed system will predict the most suitable crop for particular land based on soil contents and weather parameters such as Temperature, Humidity, soil PH and Rainfall.

4.1 TECHNOLOGY USED:

1. Machine Learning
2. Unsupervised Learning
3. K Means Clustering

4.2 TOOLS USED:

1. Python: 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.

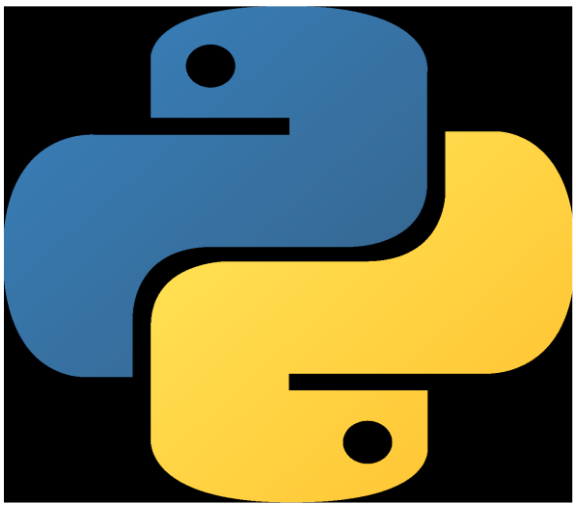


Fig. 3. Python

The Python libraries which have been used in our project are:

A. **Pandas:** pandas is a software library present in the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

B. **NumPy:** NumPy stands for Numerical Python and was created in 2005 by Travis Oliphant. NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices.

C. **Matplotlib:** Matplotlib is a python library used to create 2D graphs and plots by using python scripts. It has a module named pyplot which makes things easy for plotting by providing feature to control line styles, font properties, formatting axes etc. It supports a very wide variety of graphs and plots namely - histogram, bar charts, power spectra, error charts etc. It is used along with NumPy.

D. **Seaborn:** Seaborn is an open-source Python library built on top of matplotlib. It is used for data visualization and exploratory data analysis. Seaborn works easily with dataframes and the Pandas library. The graphs created can also be customized easily.

E. **Ipywidets:** ipywidgets, also known as jupyter-widgets or simply widgets, are interactive HTML widgets for Jupyter notebooks and the IPython kernel. Notebooks come alive when interactive widgets are used. Users gain control of their data and can visualize changes in the data.

F. **Scikit-learn (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.

2. Jupyter Notebook: The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modelling, data visualization, machine learning, and much more.

Fig. 4. Jupyter

4.3 FLOW IN THE PROPOSED SYSTEM:

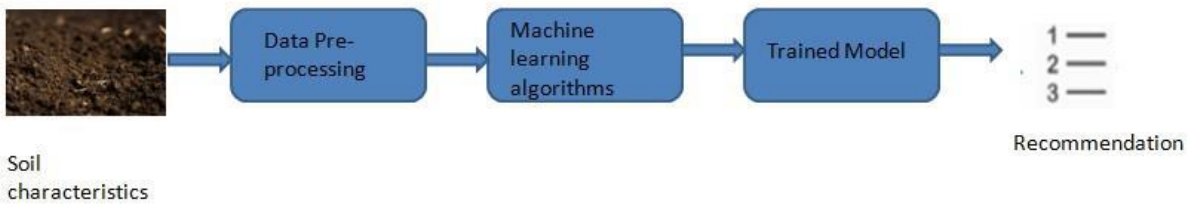


Fig. 5. Flow in the proposed system

A. Acquisition of training dataset:

The accuracy of a machine learning algorithm may depend on the number of parameters used and to the extent of correctness of the dataset. Our dataset contains the N, P, K, and pH values of different kinds of soils as attributes and it also contains the corresponding crops that can be grown in that soil as label.

Thus, by using an appropriate machine learning algorithm we can train the dataset to predict the most suitable crop that can be grown under the given input parameters.

The data set used in our project was obtained from Kaggle and is titled "Crop_recommendation" and is a CSV file. A Comma Separated Value (CSV) file is a delimited text file that uses a comma to separate values. Each line of the file is a data record. Each record consists of one or more fields, separated by commas. The use of the comma as a field separator is the source of the name for this file format. A CSV file typically stores tabular data (numbers and text) in plain text, in which case each line will have the same number of fields. So in order to use this dataset in Python, we have to import the .csv file.

After the .csv file is imported, in order to read the .csv file using Python, we use the command:

```
In [11]: # Lets check the head of the dataset
         data.head()
```

Out[11]:

	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

B. Data preprocessing:

Data preprocessing is the second step and it contains two steps. The first step being Data Cleaning in which, the original dataset which can contain lots of missing values, so initially all these should be removed. Missing values are denoted by a dot in the dataset and their presence can deteriorate the value of entire data and it can reduce the performance. So, to solve this problem we replace these values with large negative values which will be treated as outliers by the model. Generating the class labels is the second step. Since we are using a supervised learning method, for each entry in the dataset there should be a class label which is created during the preprocessing step.

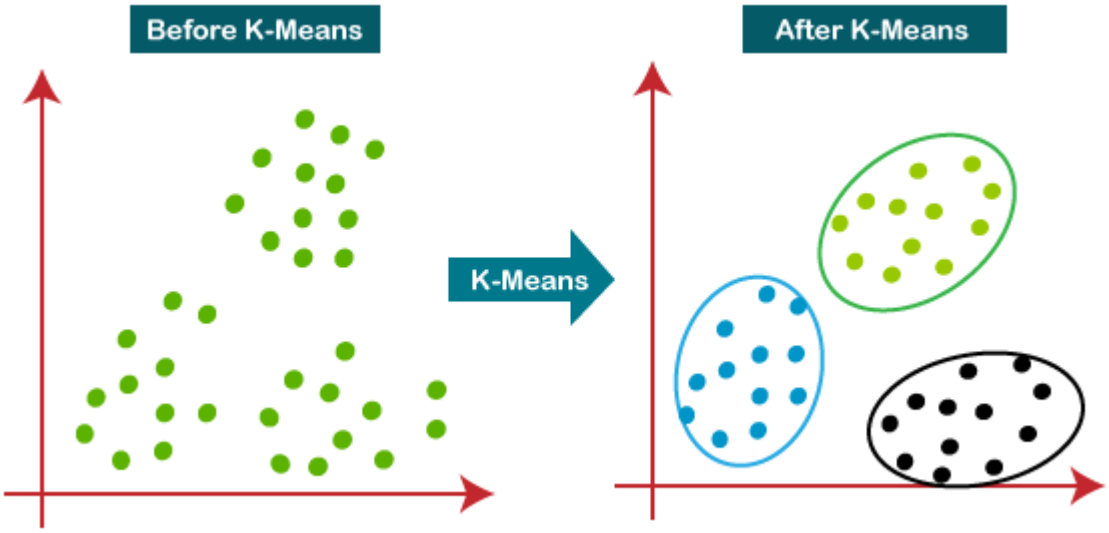
```
In [12]: # Lets check if there is any missing value present in the dataset
         data.isnull().sum()

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

Here we can use either the Fill-NA function to replace the missing values with statistical values such as mean, median or mode or the drop-NA function to remove missing values. Both these functions are present in the pandas library in Python.

C. Machine Learning Algorithm:

The different algorithms used are as follows:

1. **K Means Clustering:** K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabelled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.



- It is an iterative algorithm that divides the unlabelled dataset into different clusters in such a way that each dataset belongs only one group that has similar properties. It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabelled dataset on its own without the need for any training.

- It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

- The algorithm takes the unlabelled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best possible cluster. The value of k should be predetermined in this algorithm.

- The k-means clustering algorithm mainly performs two tasks:
  1) Determines the best value for K center points or centroids by an iterative process.
  2) Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

  Hence, each cluster has data points with some commonalities, and it is away from other clusters.

- If k is given, the K-means algorithm can be executed in the following steps:

i. Partition of objects into k non-empty subsets
ii. Identifying the cluster centroids (mean point) of the current partition.
iii. Assigning each point to a specific cluster
iv. Compute the distances from each point and allot points to the cluster where the distance from the centroid is minimum.
v. After re-allotting the points, find the centroid of the new cluster formed.

- The below diagram explains the working of the K-means Clustering Algorithm:

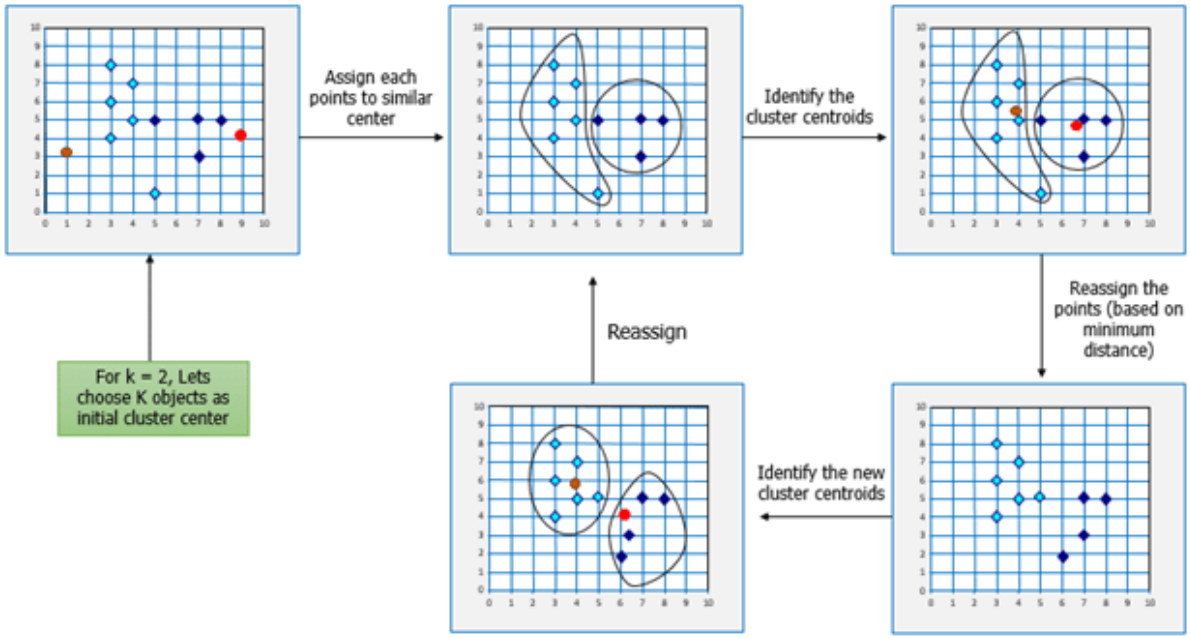


Fig. K-Means Clustering Algorithm Flow

So for a particular example, each iteration can be shown as:

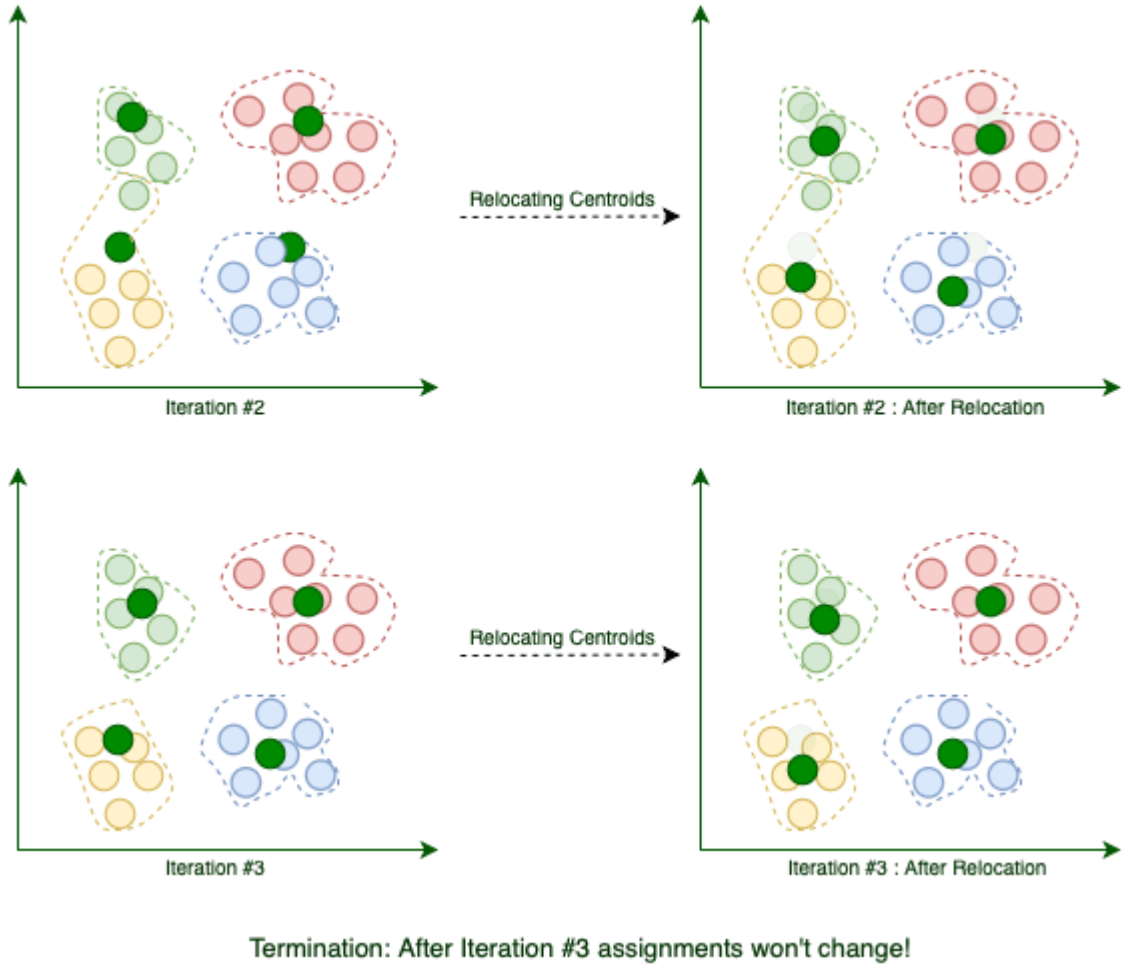


Fig: Iterations in K-Means Clustering

- The performance of the K-means clustering algorithm depends upon highly efficient clusters that it forms. But choosing the optimal number of clusters is a big task. There are some different ways to find the optimal number of clusters, but here we are discussing the most appropriate method to find the number of clusters or value of K. The method is given below:

ELBOW METHOD:

The Elbow method is one of the most popular ways to find the optimal number of clusters. This method uses the concept of WCSS value. **WCSS** stands for **Within Cluster Sum of Squares**, which defines the total variations within a cluster. The formula to calculate the value of WCSS (for 3 clusters) is given below:

$$WCSS= \sum_{P_i \text{ in Cluster1}} distance(P_i\ C_1)^2 + \sum_{P_i \text{ in Cluster2}} distance(P_i\ C_2)^2 + \sum_{P_i \text{ in CLuster3}} distance(P_i\ C_3)^2 \quad (1)$$

In the above formula of WCSS,

$\Sigma_{P_i \text{ in Cluster1}} \text{distance}(P_i\ C_1)^2$: is the sum of the square of the distances between each data point and its centroid within a cluster1 and the same for the 2 others.

To measure the distance between data points and centroid, we can use any method such as Euclidean distance or Manhattan distance.

To find the optimal value of clusters, the elbow method follows the below steps:

- It executes the K-means clustering on a given dataset for different K values (ranges from 1-10).
- For each value of K, calculates the WCSS value.
- Plots a curve between calculated WCSS values and the number of clusters K.
- The sharp point of bend or a point of the plot looks like an arm, then that point is considered as the best value of K.

Since the graph shows the sharp bend, which looks like an elbow, hence it is known as the elbow method.

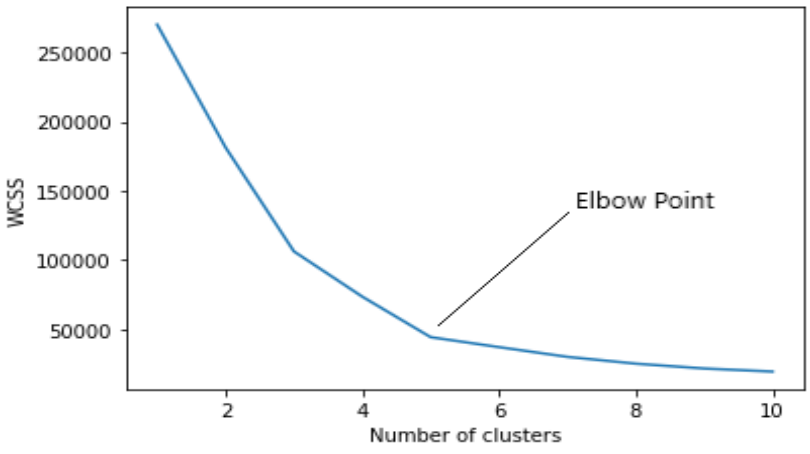
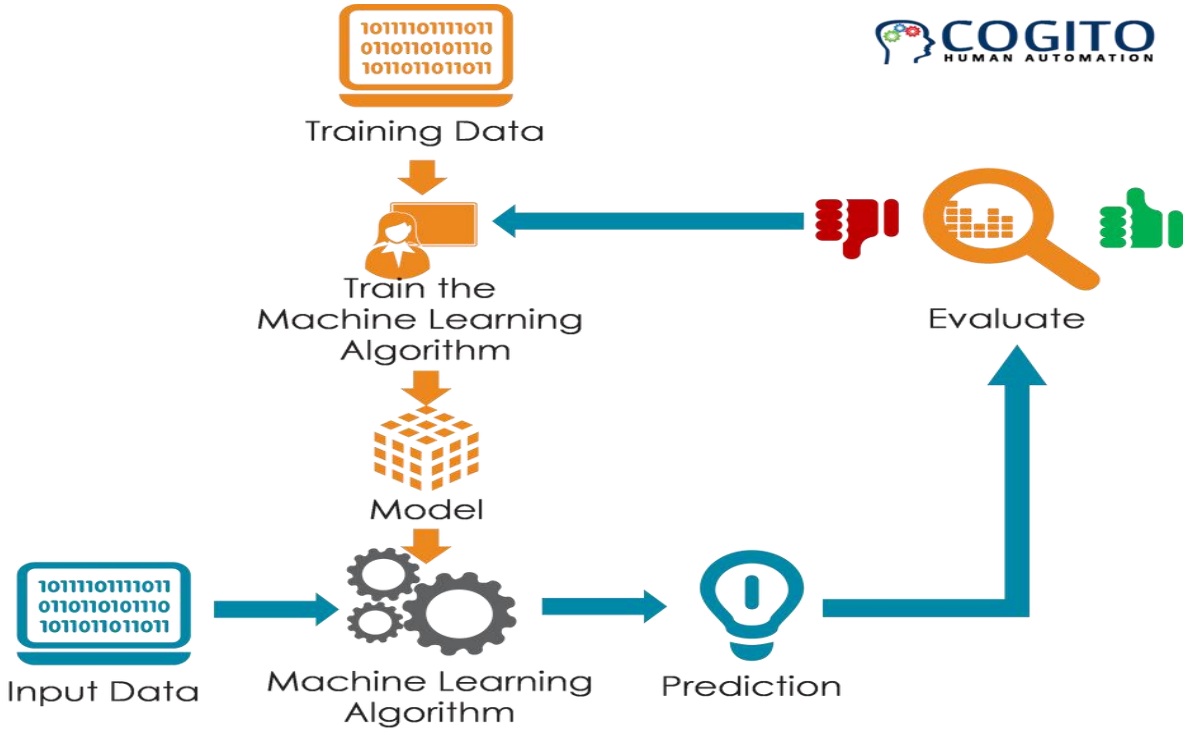


Fig. Elbow Method

2. Trained model:

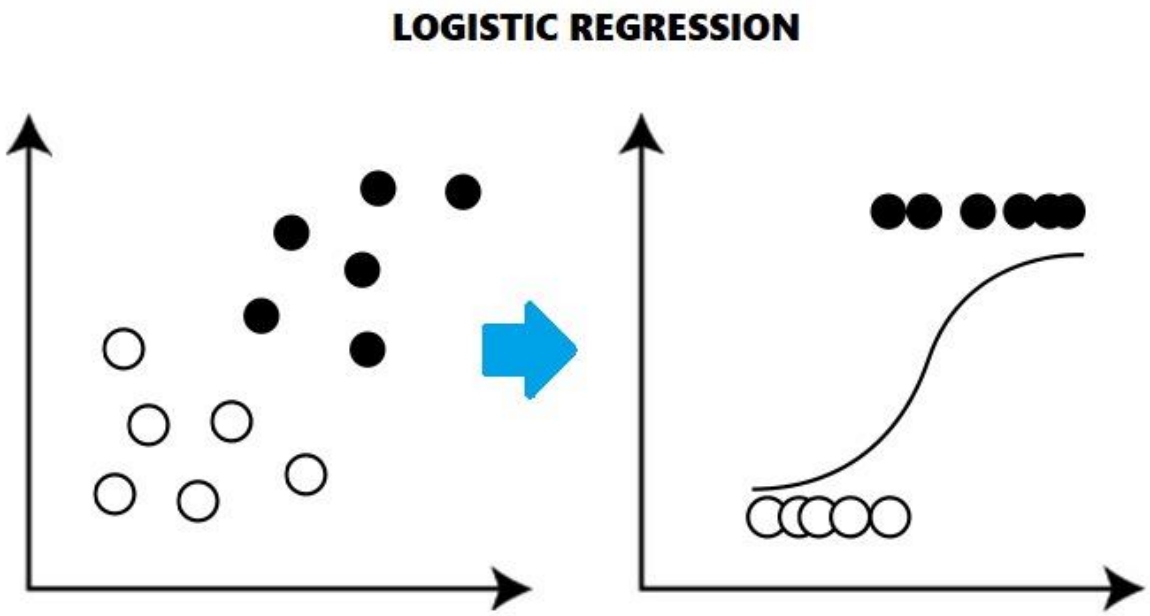
Trained models are obtained after applying the dataset to the machine learning algorithms. Our project suggests a crop prediction system which is based on the

K-Means Clustering algorithm. Soil properties such Nitrogen, Phosphorus, Potassium, pH value, etc. are given as input to the model. The algorithm will look for a crop which will have the value closest to the inputted values. The output will be all the crops which are suitable for the inputted values. The result is calculated based on the most comparable or closest values. Because of its higher convergence speed and simplicity this algorithm is preferred over other algorithms. The input for the algorithm is the soil properties such as Nitrogen, Phosphorus, Potassium, pH value, etc.



3. Logistic Regression:

Logistic regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.



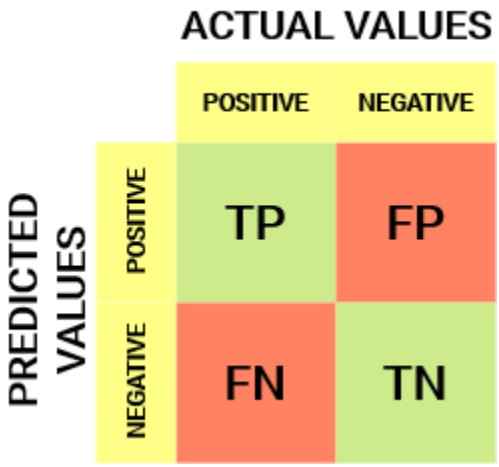
4. Evaluation Metrics:

The idea of building machine learning models works on a constructive feedback principle. You build a model, get feedback from metrics, make improvements and continue until you achieve a desirable accuracy. Evaluation metrics explain the performance of a model. An important aspect of evaluation metrics is their capability to discriminate among model results.

Simply building a predictive model is not our motive. It's about creating and selecting a model which gives high accuracy on out of sample data. Hence, it is crucial to check the accuracy of your model prior to computing predicted values.

In our project, we have used "Confusion Matrix" as the evaluation metric:

A Confusion matrix is an N X N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.



- The target variable has two values: Positive or Negative.
- The columns represent the actual values of the target variable.
- The rows represent the predicted values of the target variable.
- **True Positive (TP):**
  i. The predicted value matches the actual value
  ii. The actual value was positive and the model predicted a positive value
- **True Negative (TN):**
  i. The predicted value matches the actual value
  ii. The actual value was negative and the model predicted a negative value
- **False Positive (FP) – Type 1 error:**
  i. The predicted value was falsely predicted
  ii. The actual value was negative but the model predicted a positive value
- **False Negative (FN) – Type 2 error:**
  i. The predicted value was falsely predicted
  ii. The actual value was positive but the model predicted a negative value

V. IMPLEMENTATION

```
#for manipulations
import numpy as np
import pandas as pd

#for data visualization
import matplotlib.pyplot as plt
import seaborn as sns

#for interactivity
from ipywidgets import interact

#Let's read the data set
data = pd.read_csv("Crop_recommendation.csv")
#Let's Check the shape of the dataset
print("Shape of the dataset: ", data.shape)
#Let's check the head of the dataset
data.head()
#Let's check if there is any missing value present in the dataset i.e data cleaning
#isnull() returns boolean value and sum() returns the count
data.isnull().sum()
#Let's check the unique crops present in this Dataset and their occurence
data['label'].value_counts()
#Let's check the unique crops present in this Dataset and their occurence
data['label'].value_counts()
#Let's check the summary statistics for each Factor

@interact
def summary(crops = list(data['label'].value_counts().index)):
    x = data[data['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 Phosphorus")
print("Minimum Phosphorus required : ", x['P'].min())
print("Average Phosphorus required : ", x['P'].mean())
print("Maximum Phosphorus 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()))
#Let's check the summary statistics for each of the Crops


@interact
def compare(conditions = ['N', 'P', 'K', 'temperature', 'humidity', 'ph', 'rainfall']):
    print("Average value for", conditions, "is {0:.2f}".format(data[conditions].mean()))
    print("-----------------------------------------")
    print("Rice : {0:.2f}".format(data[(data['label'] == 'rice')][conditions].mean()))
    print("Black Grams : {0:.2f}".format(data[(data['label'] == 'blackgram')][conditions].mean()))
    print("Banana : {0:.2f}".format(data[(data['label'] == 'banana')][conditions].mean()))
    print("Jute : {0:.2f}".format(data[(data['label'] == 'jute')][conditions].mean()))
    print("Coconut : {0:.2f}".format(data[(data['label'] == 'coconut')][conditions].mean()))
    print("Apple : {0:.2f}".format(data[(data['label'] == 'apple')][conditions].mean()))
    print("Papaya : {0:.2f}".format(data[(data['label'] == 'papaya')][conditions].mean()))
    print("Muskmelon : {0:.2f}".format(data[(data['label'] == 'muskmelon')][conditions].mean()))
    print("Grapes : {0:.2f}".format(data[(data['label'] == 'grapes')][conditions].mean()))
    print("Watermelon : {0:.2f}".format(data[(data['label'] == 'watermelon')][conditions].mean()))
    print("Kidney Beans : {0:.2f}".format(data[(data['label'] == 'kidneybeans')][conditions].mean()))
    print("Mung Beans : {0:.2f}".format(data[(data['label'] == 'mungbean')][conditions].mean()))
    print("Oranges : {0:.2f}".format(data[(data['label'] == 'orange')][conditions].mean()))
    print("Chick Peas : {0:.2f}".format(data[(data['label'] == 'chickpea')][conditions].mean()))
    print("Lentils : {0:.2f}".format(data[(data['label'] == 'lentil')][conditions].mean()))
    print("Cotton : {0:.2f}".format(data[(data['label'] == 'cotton')][conditions].mean()))
```

```
    print("Maize : {0:.2f}".format(data[(data['label'] == 'maize')][conditions].mean()))
    print("Moth Beans : {0:.2f}".format(data[(data['label'] == 'mothbeans')][conditions].mean()))
    print("Pigeon Peas : {0:.2f}".format(data[(data['label'] == 'pigeonpeas')][conditions].mean()))
    print("Mango : {0:.2f}".format(data[(data['label'] == 'mango')][conditions].mean()))
    print("Pomegranate : {0:.2f}".format(data[(data['label'] == 'pomegranate')][conditions].mean()))
    print("Coffee : {0:.2f}".format(data[(data['label'] == 'coffee')][conditions].mean()))
@interact
def compare(conditions = ['N', 'P', 'K', 'temperature', 'humidity', 'ph', 'rainfall']):
    print("Crops which require greater than average", conditions, '\n')
    print(data[data[conditions] > data[conditions].mean()]['label'].unique())
    print("-----------------------------------------")
    print("Crops which require equal to or smaller than average", conditions, '\n')
    print(data[data[conditions] <= data[conditions].mean()]['label'].unique())
plt.figure(figsize=(15,8))

plt.subplot(2, 4, 1)
sns.distplot(data['N'], color = 'lightgrey')
plt.xlabel("Ratio of Nitrogen", fontsize = 12)
plt.grid()

plt.subplot(2, 4, 2)
sns.distplot(data['P'], color = 'lightblue')
plt.xlabel("Ratio of Phosphorus", fontsize = 12)
plt.grid()

plt.subplot(2, 4, 3)
sns.distplot(data['K'], color = 'darkblue')
plt.xlabel("Ratio of Potassium", fontsize = 12)
```

```
plt.grid()

plt.subplot(2, 4, 4)
sns.distplot(data['temperature'], color = 'black')
plt.xlabel("Temperature", fontsize = 12)
plt.grid()

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

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

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

plt.suptitle("Distribution for Agricultural Conditions", fontsize = 20)
plt.show()
# Some Interseting Patterns

print("Crops which requires very High rainfall:",data[data['rainfall'] > 200]['label'].unique())
print("Crops which requires very Low rainfall:",data[data['rainfall'] < 40]['label'].unique())
print("Crops which requires very High ratio of Nitrogen Content in soil :",data[data['N'] >
120]['label'].unique())
```

```
print("Crops which requires very High ratio of Phosphorous Content in soil :",data[data['P'] > 100]['label'].unique())

print("Crops which requires very High ratio of Potassium Content in soil :",data[data['K'] > 200]['label'].unique())

print("Crops which requires very High Rainfall :",data[data['rainfall'] > 200]['label'].unique())

print("Crops which requires very Low Rainfall:",data[data['rainfall'] < 40]['label'].unique())

print("Crops which requires very Low Temperature :",data[data['temperature'] < 10]['label'].unique())

print("Crops which requires very High Temperature :",data[data['temperature'] > 40]['label'].unique())

print("Crops which requires very Low Humidity :",data[data['humidity'] < 20]['label'].unique())

print("Crops which requires very Low pH :",data[data['ph'] < 4]['label'].unique())

print("Crops which requires very High pH :",data[data['ph'] > 8]['label'].unique())

# By using dataset, Find the best suitable Crop for Summer Season,Winter Season and Rainy Season

print("Summer Crops")

print(data[(data['temperature'] > 30) & (data['humidity'] > 50)]['label'].unique())

print("----------------------------------------------------------------------")

print("Winter Crops")

print(data[(data['temperature'] < 20) & (data['humidity'] > 30)]['label'].unique())

print("----------------------------------------------------------------------")

print("Rainy Crops")

print(data[(data['rainfall'] > 200) & (data['humidity'] > 30)]['label'].unique())

from sklearn.cluster import KMeans


#Removing the Labels column

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


#Selecting all values of data

x = x.values
```

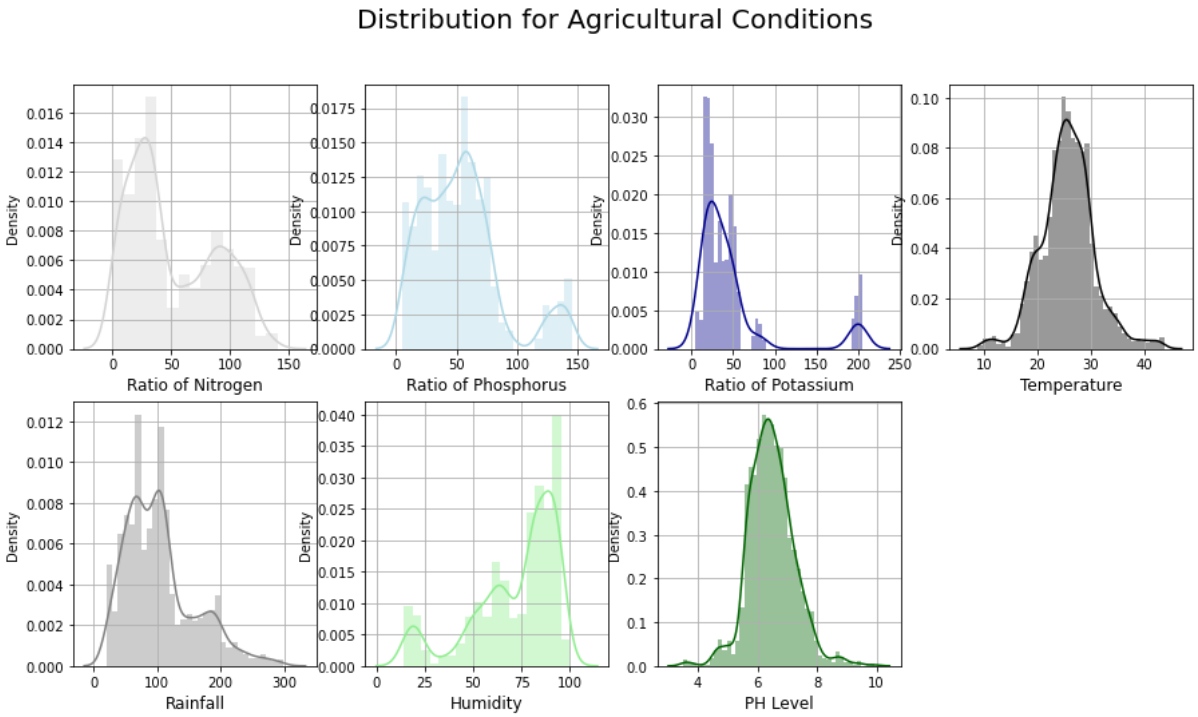
```
#Checking the shape
print(x.shape)
#Determining Optimum number of Clusters within Dataset by using K-means Clustering
plt.rcParams['figure.figsize'] = (10,4)

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_)

#Plotting the Results
plt.plot(range(1,11),wcss)
plt.title('The Elbow Method',fontsize = 20)
plt.xlabel('No. of Cluster')
plt.ylabel('wcss')
plt.show()
#Implementing K-means 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 results
a = data['label']
y_means = pd.DataFrame(y_means)
z = pd.concat([y_means, a],axis = 1)
z = z.rename(columns = {0: 'cluster'})

#Checking Clusters of Each crop
print("Checking results after applying K-means Clustering Analysis \n")
```

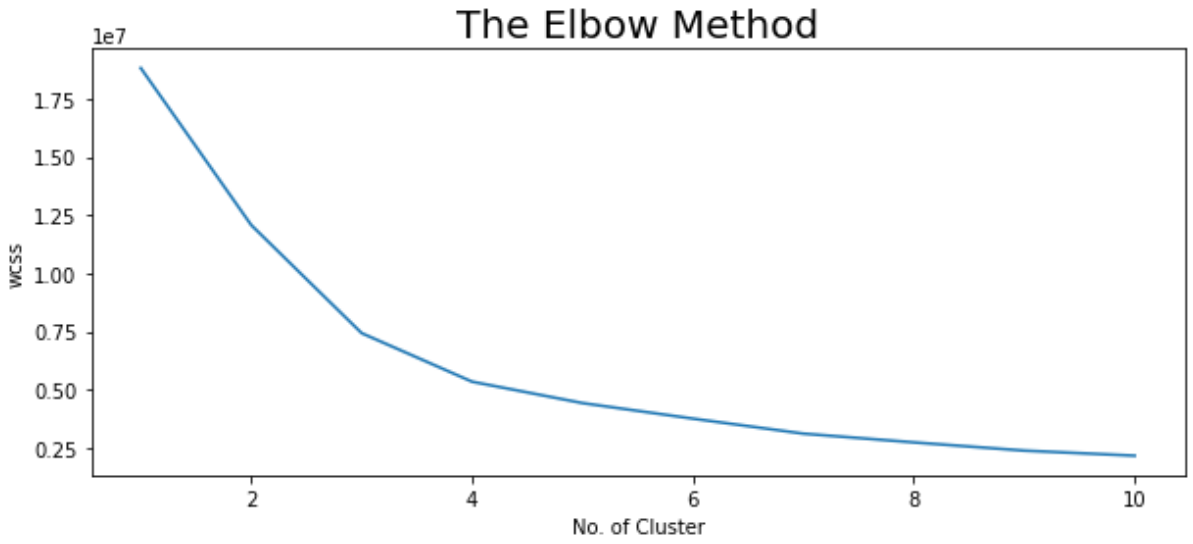
```
print("Crops in First Cluster:", z[z['cluster'] == 0]['label'].unique())
print("-----------------------------------------------------------")
print("Crops in Second Cluster:", z[z['cluster'] == 1]['label'].unique())
print("-----------------------------------------------------------")
print("Crops in Third Cluster:", z[z['cluster'] == 2]['label'].unique())
print("-----------------------------------------------------------")
print("Crops in Forth Cluster:", z[z['cluster'] == 3]['label'].unique())
#Splitting dataset for Predictive Modelling
y = data['label']
x = data.drop(['label'],axis = 1)

print("Shape of x:", x.shape)
print("Shape of y:", y.shape)
#Training and Testing Sets for Validation 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)
#Creating a Predictive model
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(x_train, y_train)
y_pred = model.predict(x_test)
#Evaluating the Model Performance
from sklearn.metrics import confusion_matrix
```

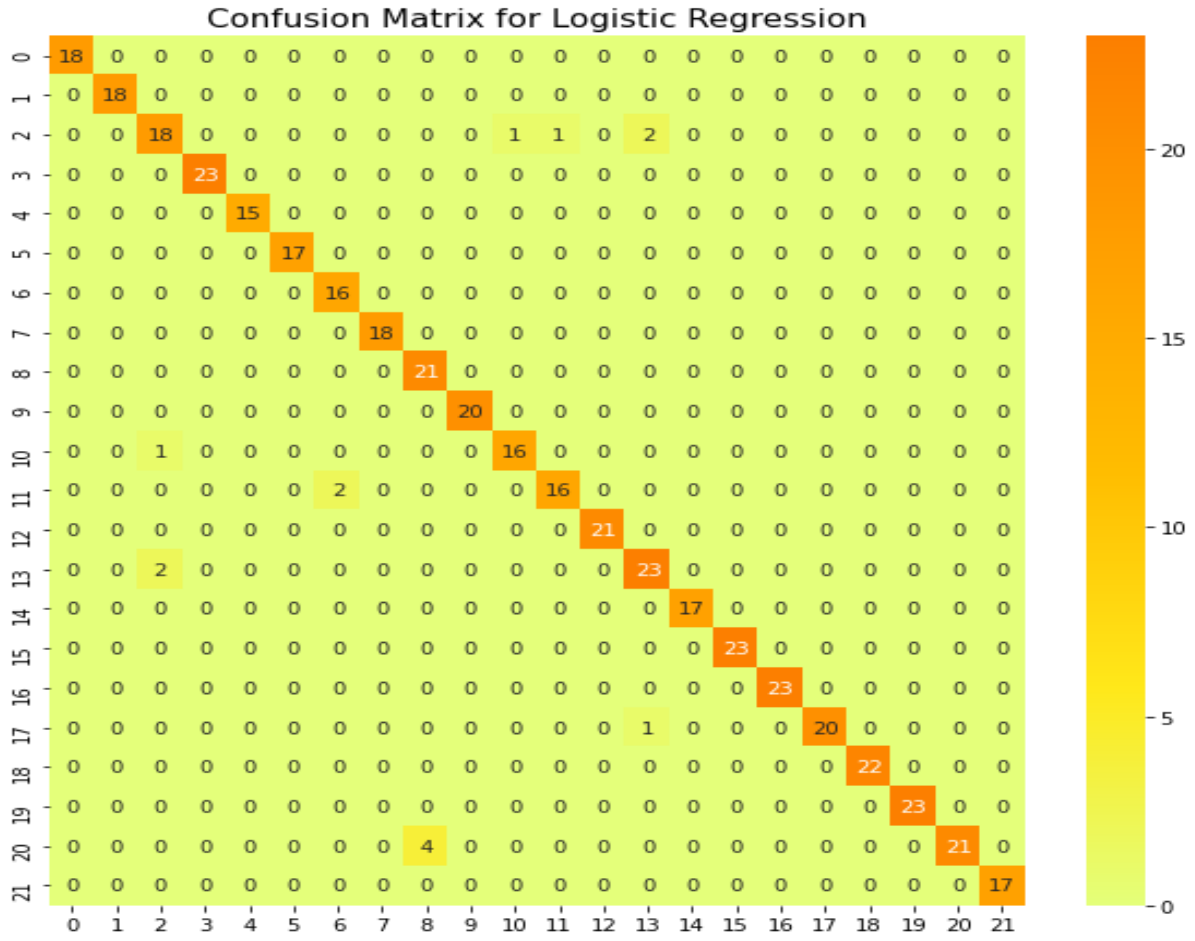
```
#Evaluating Model Performance by using confusion matrix
plt.rcParams['figure.figsize'] = [10,10]
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot = True, cmap = 'Wistia')
plt.title('Confusion Matrix for Logistic Regression', fontsize=15)
plt.show()
from sklearn.metrics import classification_report

#Printing Classification Report
cr = classification_report(y_test, y_pred)
print(cr)
# Our model is ready,now we use model to predict Crop name
data.head()
prediction = model.predict((np.array([[90,
                                       40,
                                       40,
                                       20,
                                       80,
                                       7,
                                       200]])))
print("The suggested Crop for Given Climatic condition is :", prediction)
```

VI. RESULT

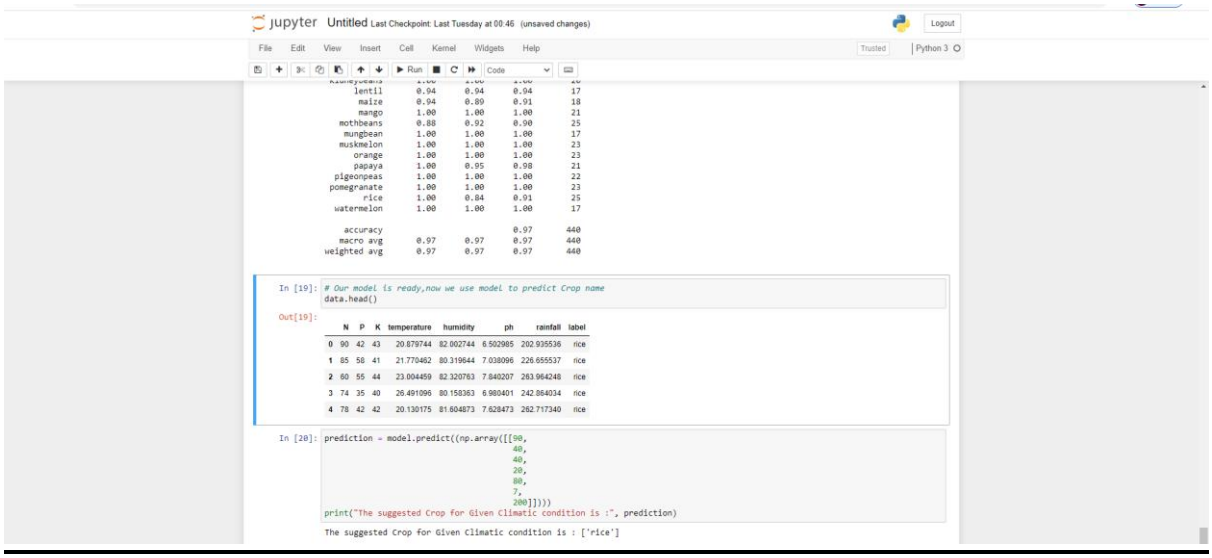
The proposed system recommends the best suitable crop for particular land by considering parameters as annual rainfall, temperature, humidity and soil pH. Other parameters are have to be entered by the user. In the output section the system displays a suitable crop, required seeds/acre, market price and approximate yield of the recommended crop and also the system takes NPK values in the input section to display the required NPK for the recommended crop. There are many advancements for our project compared to the previous papers. In our project we are taking a large dataset therefore we can get the details regarding a greater number of crops. So more number of crops that can be grown in different soil conditions can be predicted. We have used different machine learning models in our project. Different models show different accuracies so we can select the best among them in order to do the accurate predictions. In this manner we get the results in a speedy way. We have built our project in such a way that it is easily accessible to all the farmers and with the advancement in technology we can incorporate more features into it. Since we are using the machine learning model of higher predicting accuracy, our project gives best results. The occurrence of natural disasters like flood and soil erosion can change the overall composition of the soil and our recommendation system provides a better way to predict the suitable crops in the changed soil conditions.


Distribution for Agricultural Conditions
Density
Ratio of Nitrogen
Ratio of Phosphorus
Ratio of Potassium
Temperature
Rainfall
Humidity
PH Level


The Elbow Method
1e7
WCSS
No. of Cluster



Here we have taken a sample input providing all the required parameters and the suggested crop that we get is: **‘RICE’** as can be seen in the given output:



VI. CONCLUSION AND FUTURE SCOPE CONCLUSION:

6.1 CONCLUSION: Presently our farmers are not effectively using technology and analysis, so there may be a chance of wrong selection of crop for cultivation that will reduce their income. To reduce those type of loses we have developed a farmer friendly system, that will predict which would be the best suitable crop for particular land and this system will also provide information about required nutrients to add up, required seeds for cultivation, expected yield and market price. So, this makes the farmers to take right decision in selecting the crop for cultivation such that agricultural sector will be developed by innovative idea.

6.2 FUTURE SCOPE: We have to collect all required data by giving GPS locations of a land and by taking access from Rain forecasting system of by the government, we can predict crops by just giving GPS location. Also, we can develop the model to avoid over and under crisis of the food. In the future, this system can be implemented further using IOT to get the real time values of the soil. In the farm, the sensors can be installed to collect information about the current soil conditions, and the systems can therefore increase the accuracy of correctness of the results. Hence, farming can be done in a smart way.

VII. REFERENCES:

1. Kevin Tom Thomas , Varsha S , Merin Mary Saji , Lisha Varghese , Er. Jinu Thomas **"Crop Prediction using Machine Learning".**

2. Nischitha K, Dhanush Vishwakarma, Mahendra N, Manjuraju M.R, Ashwini **"Crop Prediction using Machine Learning Approaches"**

3. **"CROP YIELD PREDICTION USING K-MEANS CLUSTERING"** Capstone Design Spring 2020 Amine Bouighoulouden Dr. Ilham Kissani.

4. **"Crop prediction based on soil and environmental characteristics using feature selection techniques"** by A. Suruliandi,G. Mariammal & S.P. Raja

5. **"Crop Yield Prediction Using Supervised Machine Learning Algorithm"** Hardik Joshi, Monika Gawade, Manasvi Ganu, Prof. Priya Porwal.

6. **https://github.com/yrnigam/Crop_Sowing_Prediction**

7. **https://github.com/deepanshu13-eng/Machine-learning-with-k-nearest-neighbors**

8. **https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning**

9. **https://www.python.org/doc/essays/blurb/**

10. **https://realpython.com/jupyter-notebook-introduction/**

11. **https://en.wikipedia.org/wiki/Pandas_(software)#:~:text=pandas%20is%20a%20software%20library,the%20three%2Dclause%20BSD%20license.**

12. https://www.w3schools.com/python/numpy/numpy_intro.asp

13. https://www.tutorialspoint.com/python_data_science/python_matplotlib.htm

14. https://www.section.io/engineering-education/seaborn-tutorial/#:~:text=Seaborn%20is%20an%20open%2Dsource,can%20also%20be%20customized%20easily.

15. https://ipywidgets.readthedocs.io/en/latest/

16. https://www.tutorialspoint.com/scikit_learn/index.htm

17. https://inblog.in/K-Means-Clustering-IuDFy5MF26

18. https://oden.io/glossary/model-training/

19. https://www.equiskill.com/understanding-logistic-regression/

20. https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/what-is-logistic-regression/

21. https://www.cogitotech.com/ml-model-validation-services

22. https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/

23. https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/