**OPTIMIZING AGRICULTURAL PRODUCTION – PREDICTIVE MODELLING PROJECT**

**Problem Statement:**

Building a Predictive Model so as to suggest the most suitable crops to grow based on the available climatic and soil conditions.

## Goal:

* To achieve precision farming by optimizing the agricultural production. The project is intended on Precision Farming.
* To Optimize Productivity
* By Understanding requirements of climatic and soil conditions for crops.
* Helps us to cope up with weather unpredictability.

## About the Dataset Used (data.csv) -

* The Dataset Consists of 22 Unique Crops such as Maize, Wheat, Mango, Watermelon, etc.
* The dataset consists of climatic conditions required to grow the crops.
* Temperature, Humidity, Rainfall.

The dataset also consists of soil conditions required to grow the crops

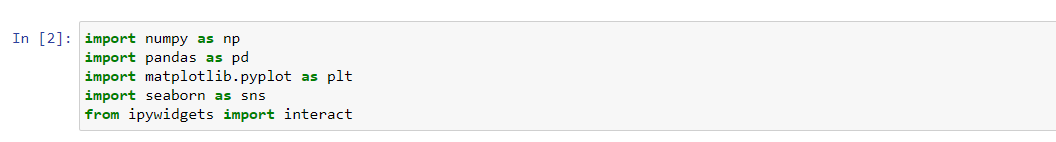
* N: The Ratio of Nitrogen Content in Soil.
* P: The Ratio of Phosphorus Content in Soil.
* K: The Ratio of Potassium Content in Soil.
* pH: pH of the Soil

## Libraries Used In the Project -

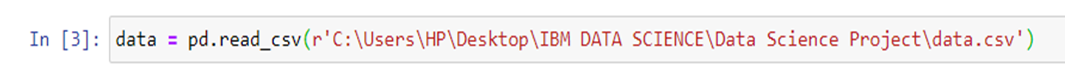
* Numpy: Used for Mathematical Operations.
* Pandas: Used for Data Frame Operations.
* Seaborn and Matplotlib: Used for Data Visualizations.
* Ipywidgets: Used for Interactive Analysis.
* Sklearn: Used for Machine Learning Algorithms

**Step by step procedure for optimizing Agricultural production:**

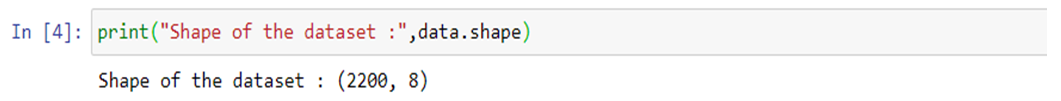
**Step 1:** Import necessary libraries such as numpy, pandas, Matplotlib, sea born. These libraries help in analyzing, visualizing and predicting the data from a dataset.



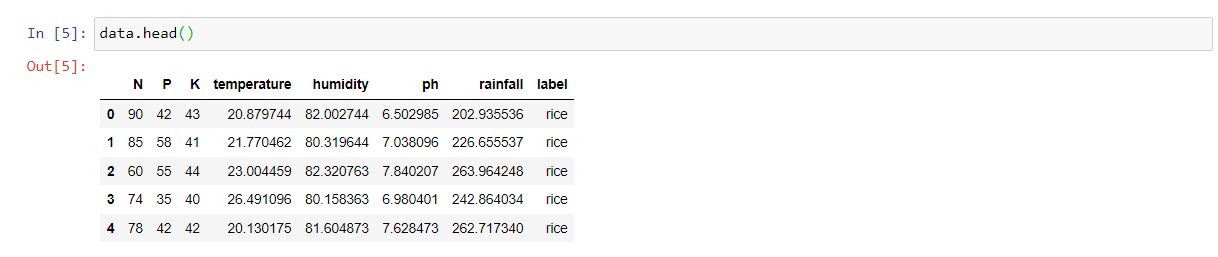
**Step 2:** Import the data set to Jupyter notebook using the following command.



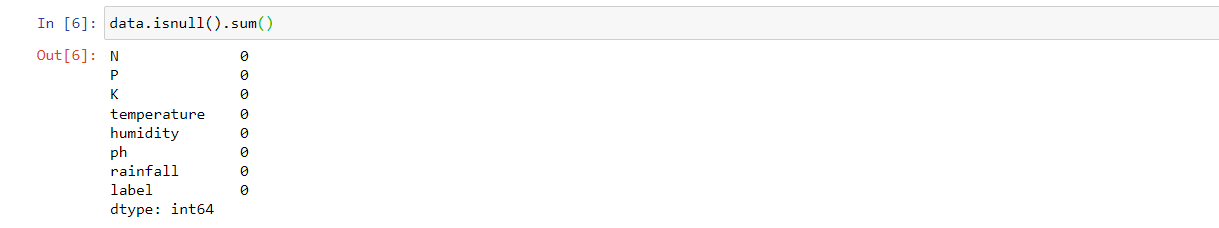
**Step 3:** Check the shape of the data set to know how many rows and columns present in the data set.



**Step 4:** We have used the crops dataset which shows the details of the crop cultivation and the values of the dataset is viewed using the following command. View the first 5 rows of the data set using the following command

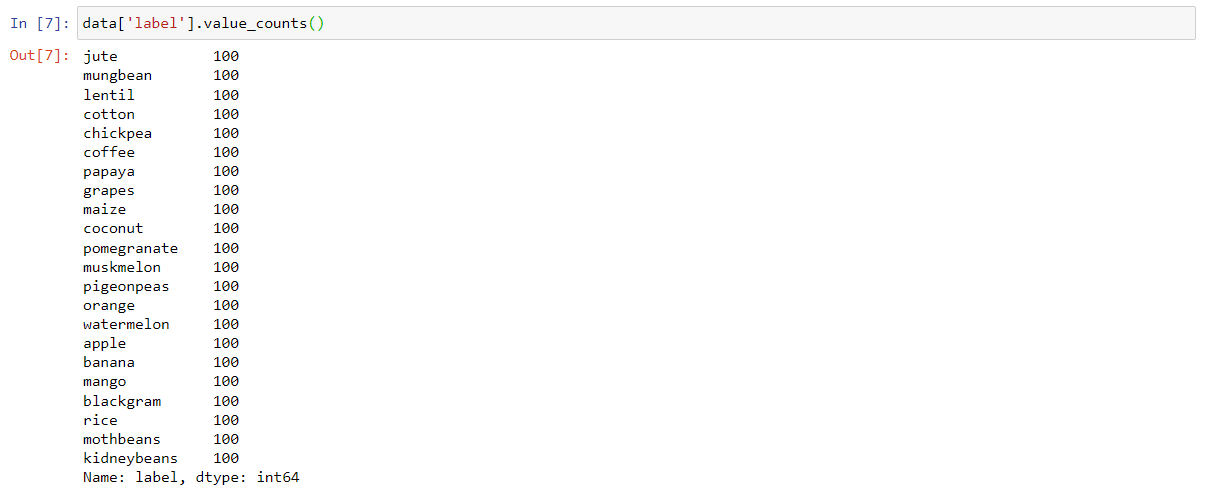


**Step 5:** Check if there is any missing values available in the dataset using the following command. **isnull()** is the command used to check the number of missing values.

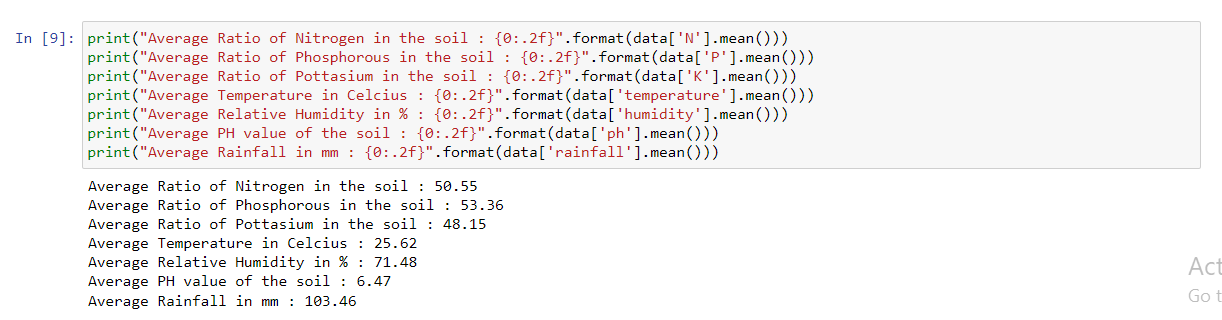


**Step 6:** If there is any duplicate values or null values present in the database, replace those values with the mean value using the command **data.fillna(data.mean())**

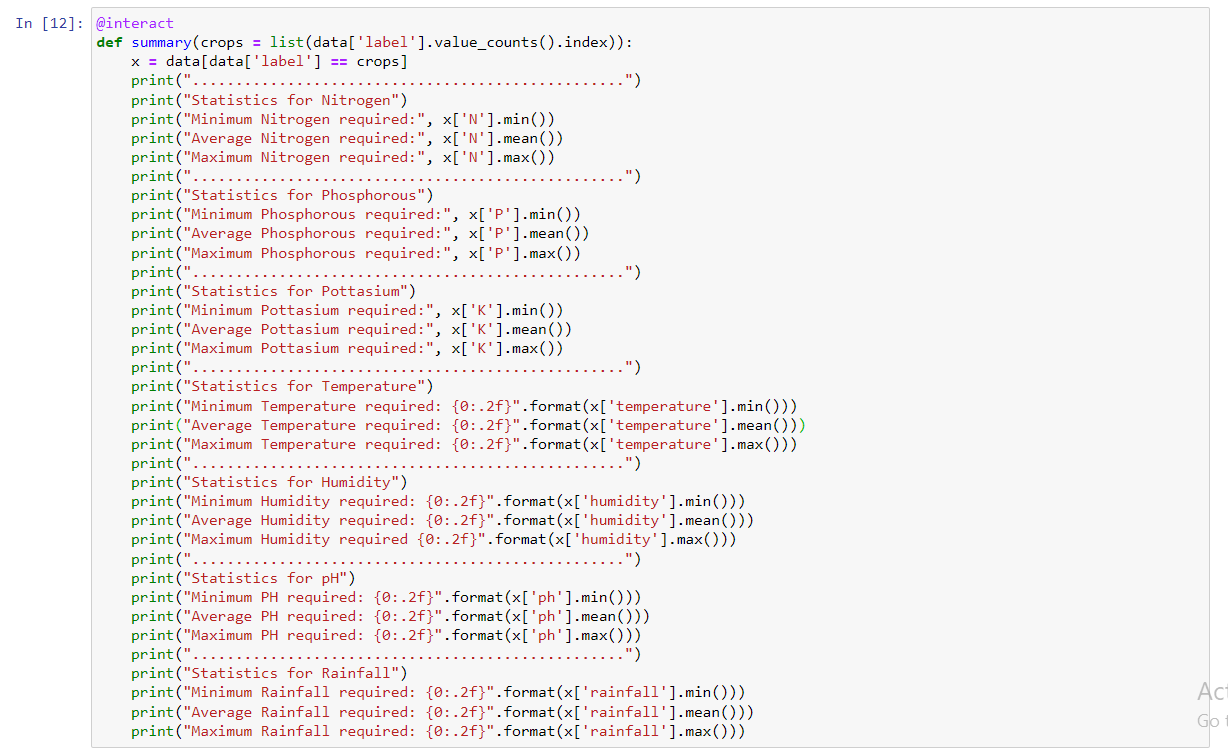
**Step 7:** After checking the missing values, null values and duplicate values check for the number of values present using the **value\_counts()** method. This shows the presence of missing values, null values and duplicate values. This process of removing or replacing the duplicate or null values in a database is known as Data Cleaning.



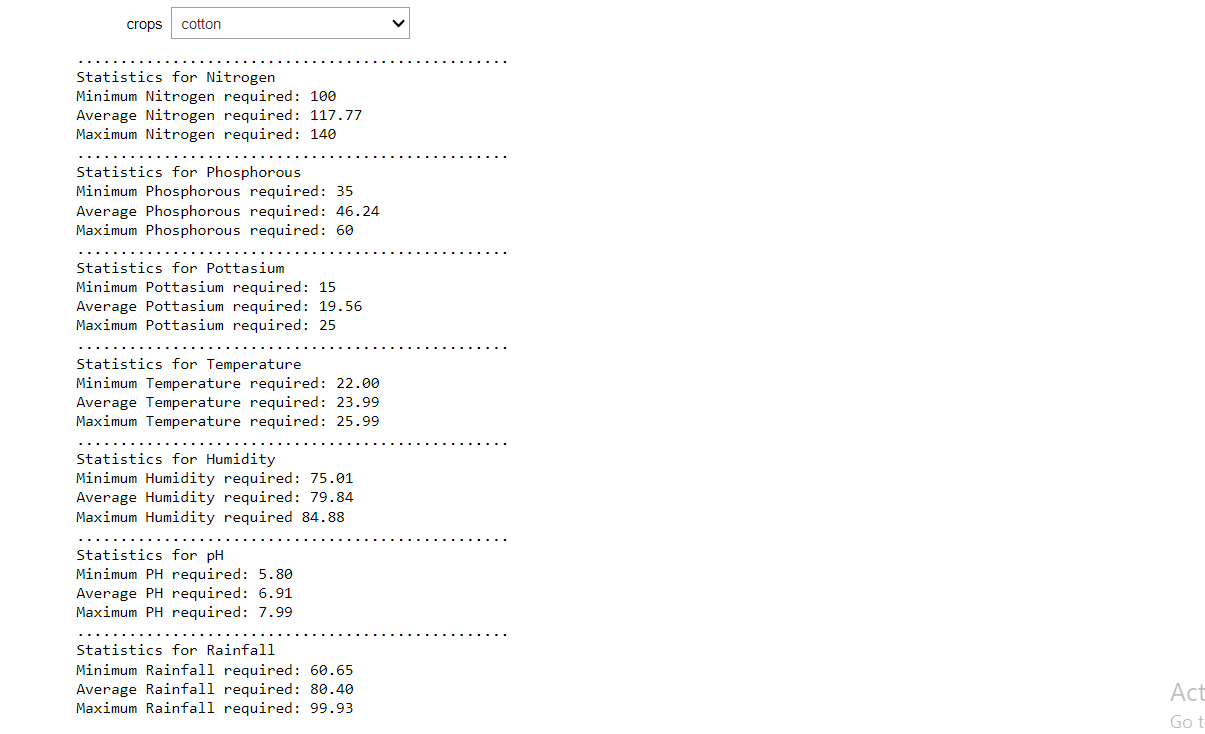
**Step 8:** Check the average ratio of individual elements present in the soil. Here we can view the mean value of elements required for the crop. Based on the values provided we can cultivate the crops.



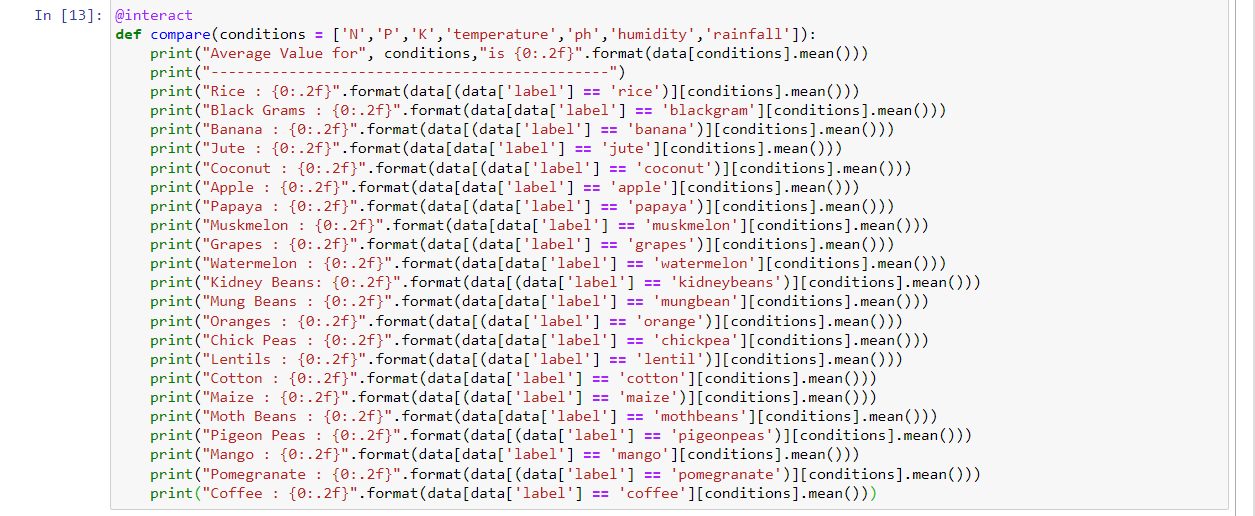
**Step 9:** Check the detailed statistics of the crop table. The statistics table shows the minimum, maximum and average values of individual elements present in the table. Here we can view individual quantity of elements required to grow a particular crop using the drop down menu.



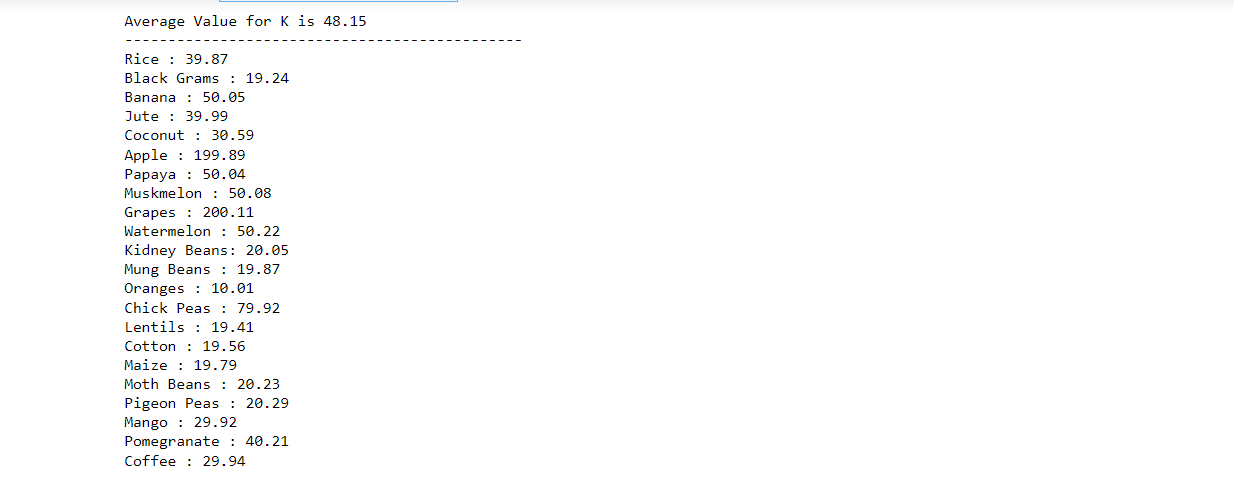
And the output for the above process is mentioned below.



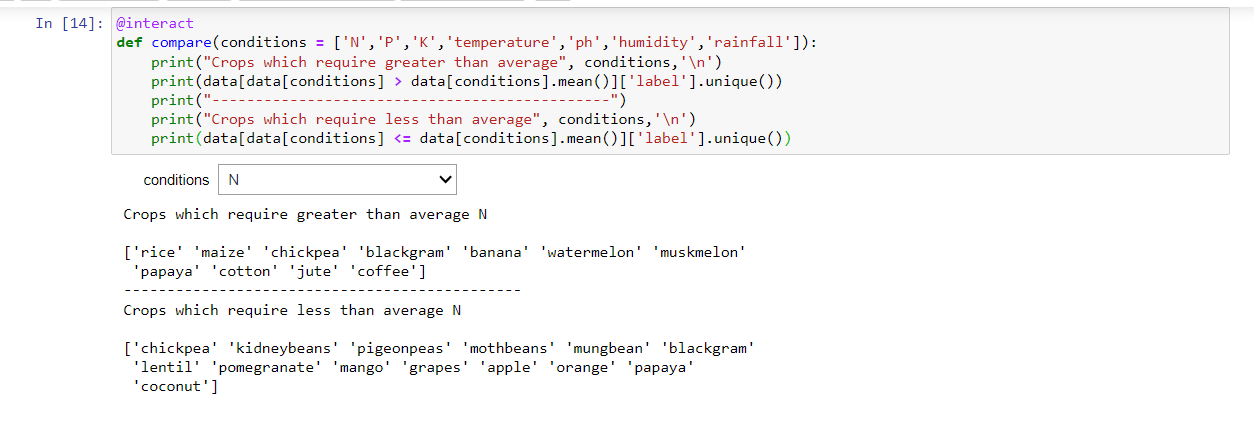
**Step 10:** In this step we could get the average requirement value and conditions to grow a particular crop. This can be achieved by selecting the crop from the dropdown menu and the average required values are provided.



The following output shows the exact process of the above mentioned code



**Step 11:** Here the crops which require greater than and lesser than average conditions values are displayed. And also can view the average value for individual condition can be viewed from drop down menu

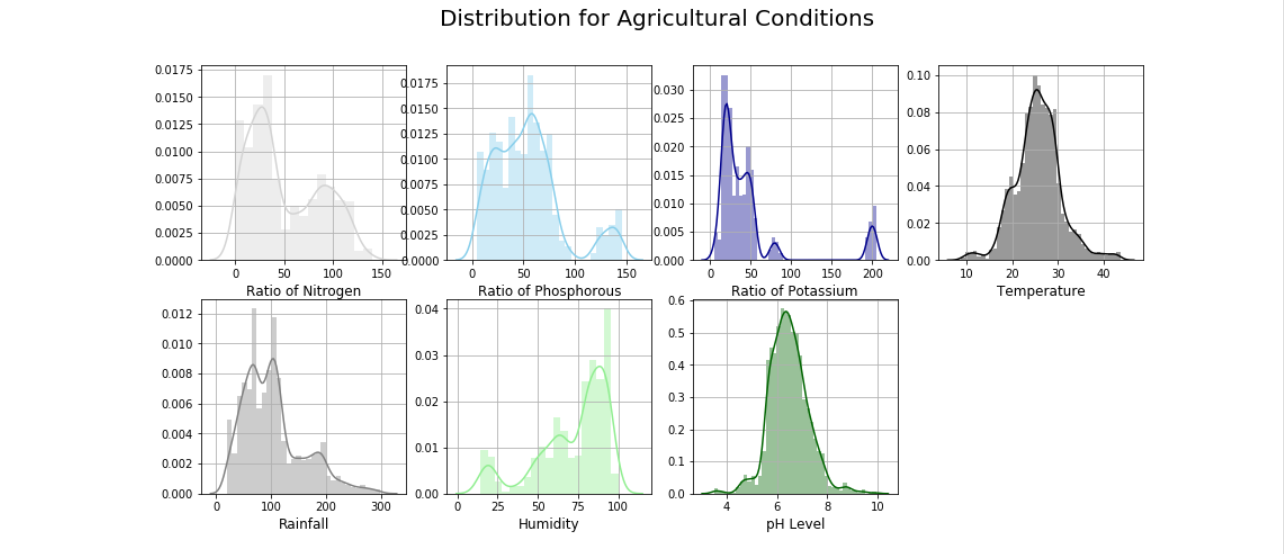


**Step 12:** In this step we’ll plotting conditions via graph so that we can compare the distribution of agricultural conditions for each element present in the data.

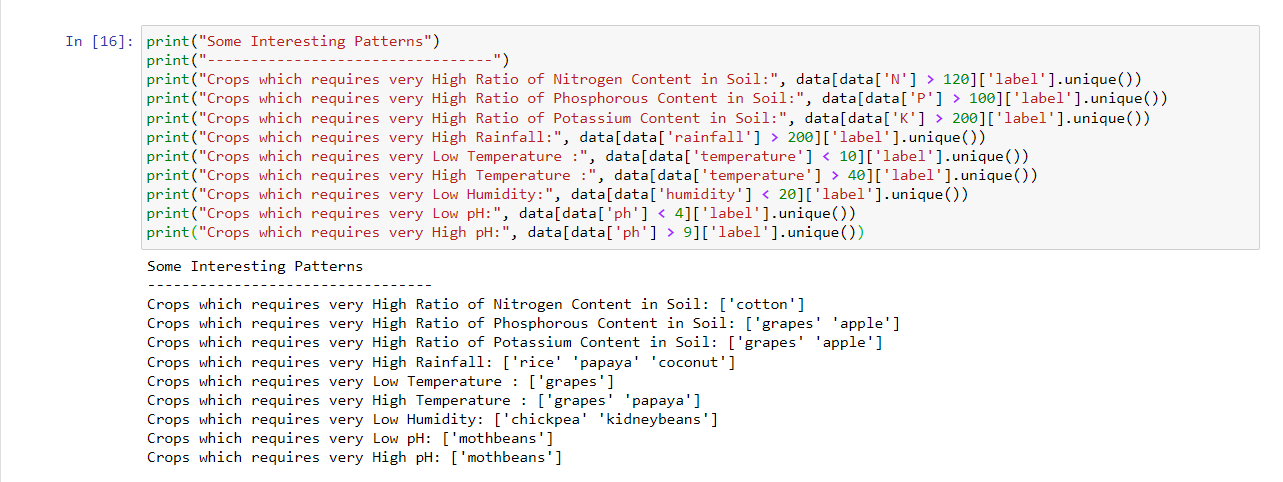
This helps to view the value of individual elements present.



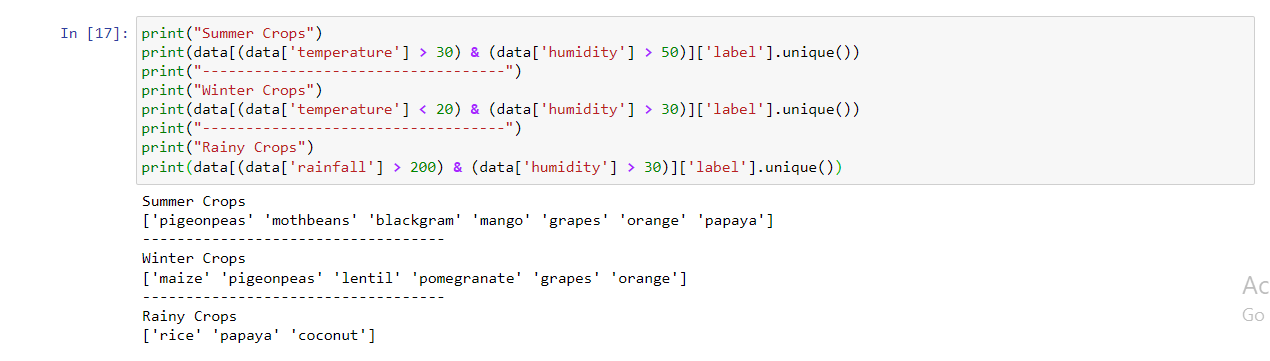
Output for the graph is follows.



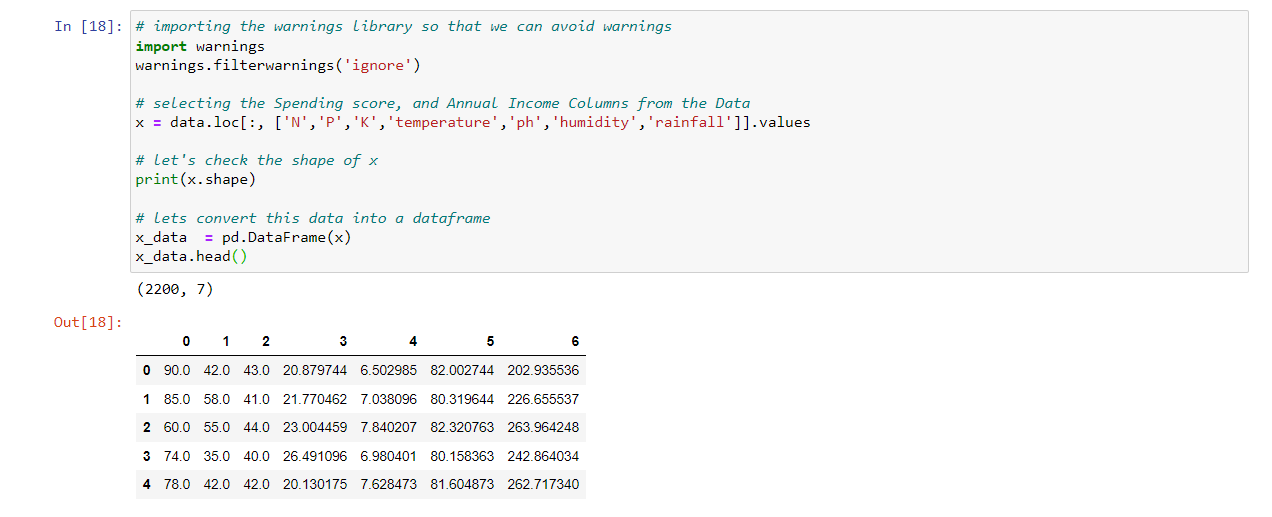
**Step 13:** Same way we can view the crop types which requires high and low ratio values of individual elements in the soil. Here we’ll be fetching the crops based on the particular soil conditions. This filtration helps in cultivating right type of crop which is suitable for the soil type.



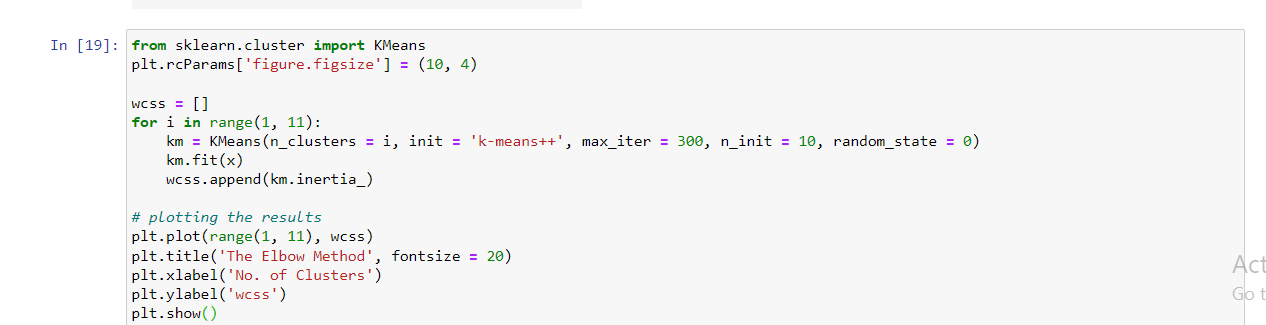
**Step 14:** Next step check the type of crops should be grown according to the season. Here we’ll be categorizing the seasonal crops and these categories are created by declaring the temperature and humidity range. And these values are fetched by **unique ()** function. This helps in cultivating suitable type of crop for the season.

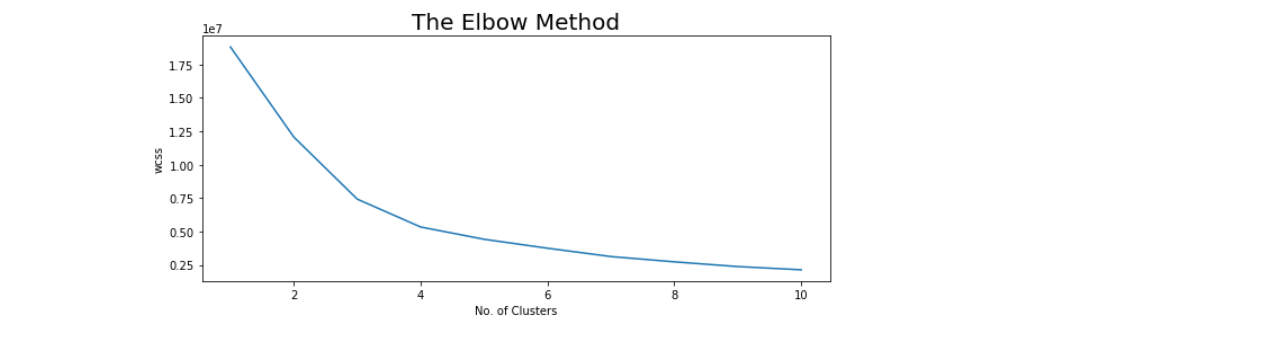


**Step 15:** Import warnings so that we can avoid warnings and also view the first 5 values of the data available in data frame.

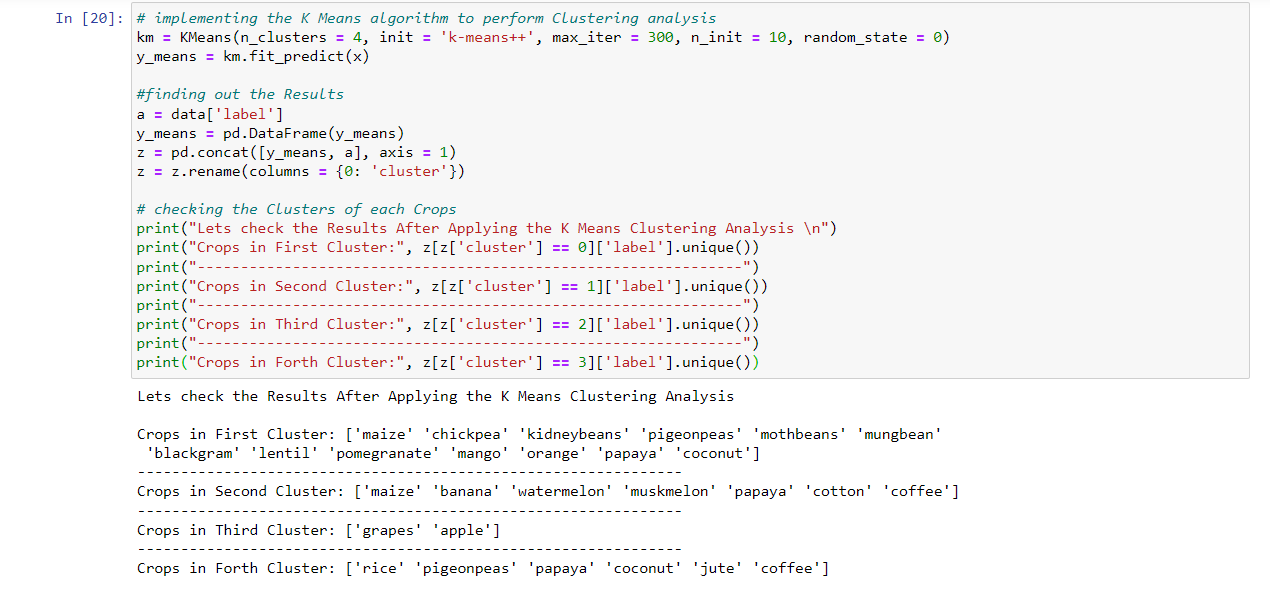


**Step 16:** Import **sklearn** library to cluster data using k means method to remove the labels column and to check the values of the data and visualize the data values via graph.

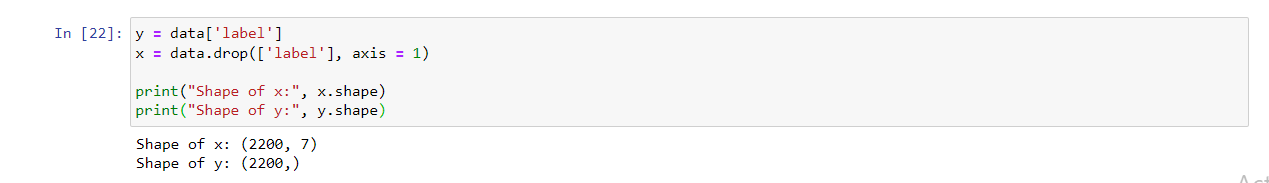




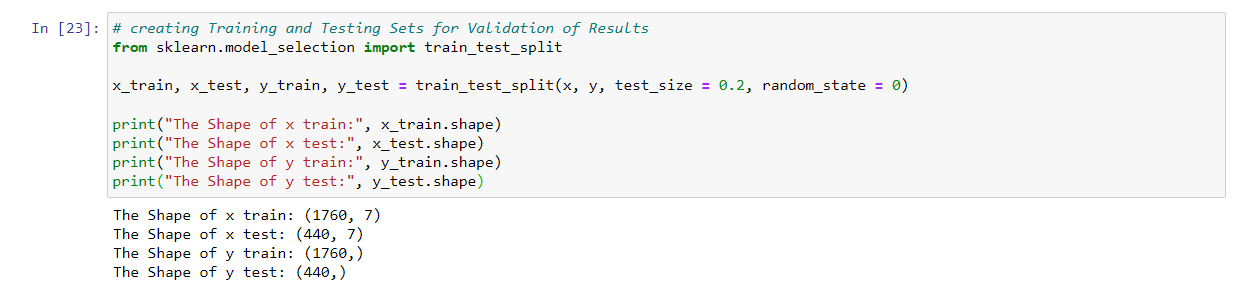
**Step 17:** Implement K Means algorithm to perform clustering analysis and check the clusters for each crop



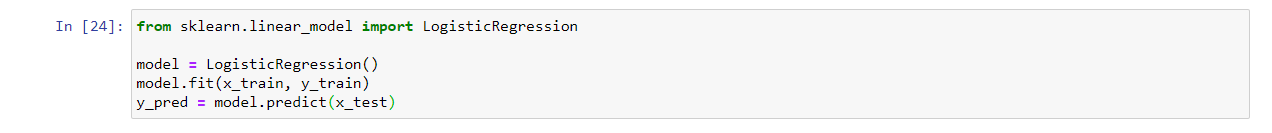
**Step 18:** Now split the data set for predictive modelling. For that we have to check the shape of the data.



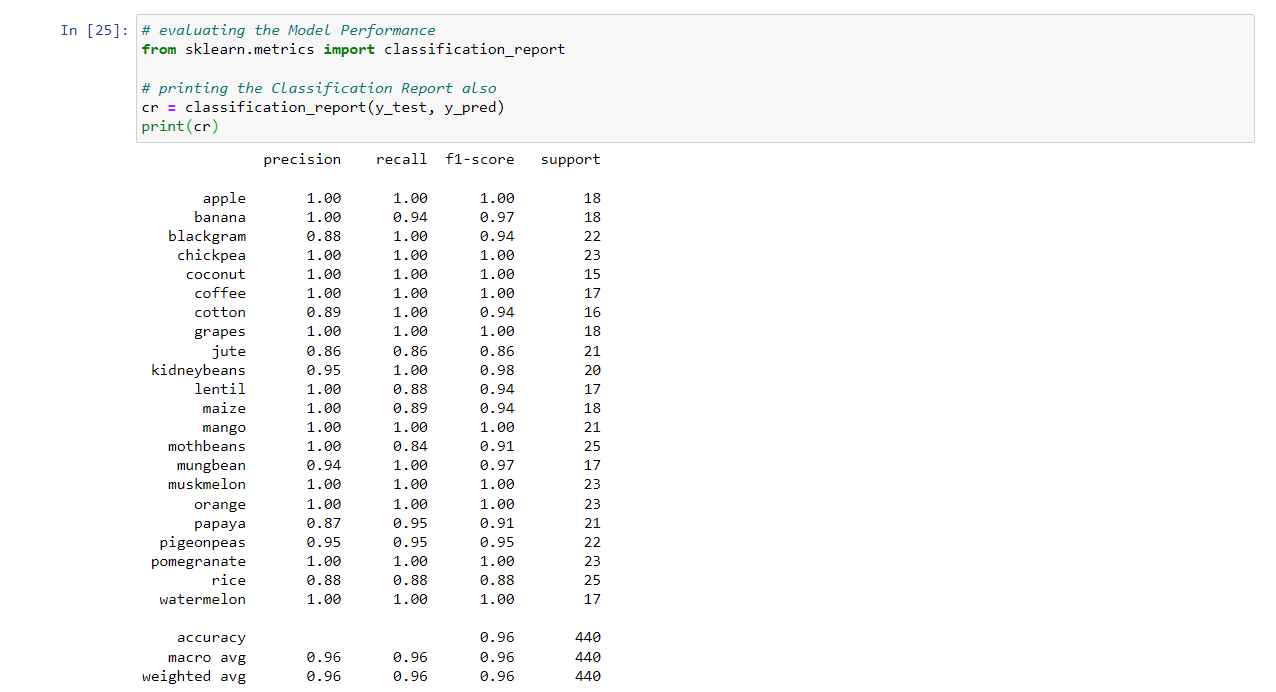
**Step 19:** Then create a training set and testing set of data to validate results. For this process ***sklearn.model\_selection*** library is used



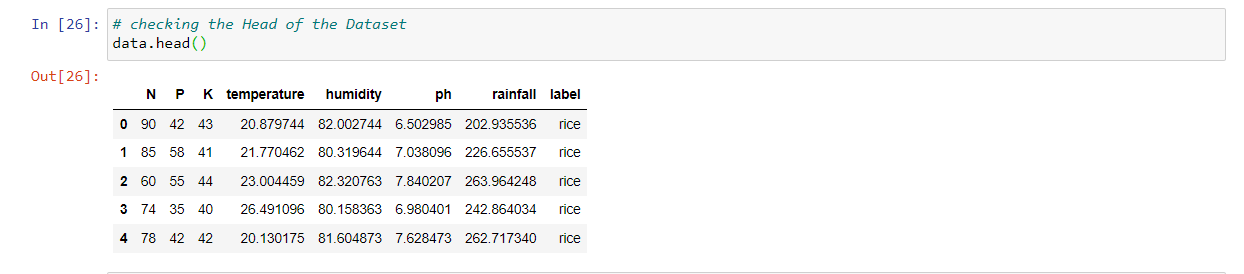
**Step 20:** Create a predictive model by importing ***Logistic Regression*** library.



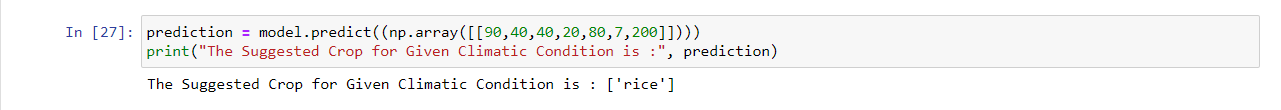
**Step 21:** Evaluate the model performance and print the classification report using ***classification\_report*** library. This library helps in classifying details about each crop individually.

****

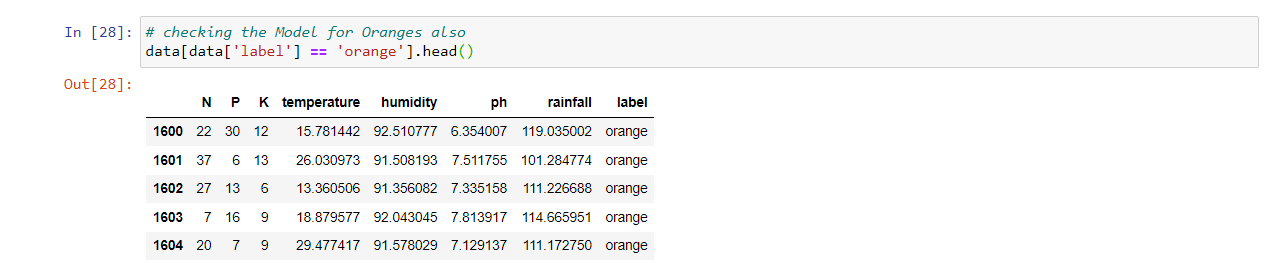
**Step 22:** View the first 5 values of the data set to predict the values using the prediction model



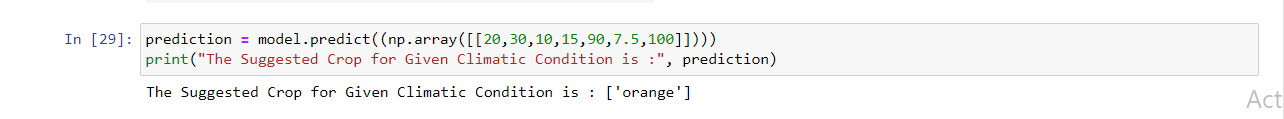
**Step 23:** Declare the function to predict values from the dataset. Here we have passed a range of values related to parameter values in an array format. And the condition fetches the crop type based on the values.



**Step 24:** Now check the values for particular item. Here we have declared a command to view the parameters for the crop orange.



**Step 25:** Also check for some real time predictions by passing data values in an array format.



**GitHub link:** https://github.com/Raajasree/Optimizing-Agricultural-Production/blob/main/Optimizing%20Agricultural%20Production.ipynb

### Upload project directly into Github without using Notebook:

1. Click on **File -> Download as -> Notebook(*.ipynb*)**
2. Make a **new repository** into Github.
3. Click **Add Files -> Create New File.**
4. Browse through your directory and **upload your file** (example filename: *GeeksForGeeks.ipynb*) and click Open.
5. **Commit** the changes.

Pwd for pycharm account - pycharmforrajsri

**GitHub pwd:** raajasree12@hub