SRI KRISHNA COLLEGE OF TECHNOLOGY
(AN AUTONOMOUS INSTITUTION)

KOVAIPUDUR, COIMBATORE 641042

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AGRICULTURAL DATA ANALYSIS USING PYTHON IN JUPYTER NOTEBOOK

A Mini Project Report

(Subject : 17CS318 DATA ANALYTICS)

Submitted by

DURAIPAANDIYAAN AP (17TUCS203)

SARATH S (17TUCS206)

SATHISH R (17TUCS208)

TABLE OF CONTENTS

ABSTRACT

Agriculture is important for human survival because it serves the basic need. A well-known fact that the majority of population (≥55%) in India is into agriculture. Due to variations in climatic conditions, there exist bottlenecks for increasing the crop production in India. It has become challenging task to achieve desired targets in Agriculture based crop yield. Various factors are to be considered which have direct impact on the production, productivity of the crops. Crop yield prediction is one of the important factors in agriculture practices. Farmers need information regarding crop yield before sowing seeds in their fields to achieve enhanced crop yield. The use of technology in agriculture has increased in recent year and data analytics is one such trend that has penetrated into the agriculture field. The main challenge in using big data in agriculture is identification of effectiveness of big data analytics. Efforts are going on to understand how big data analytics can agriculture productivity. This project focuses on the analysis of the agriculture data and finding optimal parameters to maximize the crop production using data mining techniques like PAM(Partition Around Medoids), CLARA(Clustering Large Applications), DBSCAN(Density-based spatial clustering of applications) and Multiple Linear Regression. Mining the large amount of existing crop, and its quantity, cultivation, production, yield and analysing new or non-experimental data optimizes the production and makes agriculture more resilient to climatic change of different states of our country.

INTRODUCTION:

Today, India ranks second worldwide in the farm output. Agriculture is demographically the broadest economic sector and plays a significant role in the overall socio-economic fabric of India. Agriculture is a unique business crop production which is dependent on many climate and economy factors. Some of the factors on which agriculture is dependent are soil, climate, cultivation, irrigation, fertilizers, temperature, rainfall, harvesting, pesticide weeds and other factors. Historical crop yield information is also important for supply chain operation of companies engaged in industries. These industries use agricultural products as raw material, livestock, food, animal feed, chemical, poultry, fertilizer, pesticides, seed and paper. An accurate estimate of crop production and risk helps these companies in planning supply chain decision like production scheduling. Business such as seed, fertilizer, agrochemical and agricultural machinery industries plan production and marketing activities based on crop production estimates. There are 2 factors which are helpful for the farmers and the government in decision making namely,

- It helps farmers in providing the historical crop yield record with a forecast reducing the risk management.
- It helps the government in making crop insurance policies and policies for supply chain operation.

Data mining technique plays a vital role in the analysis of data. Data mining is the computing process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database system. Unsupervised (clustering) and supervised (classifications) are two different types of learning methods in the data mining. Clustering is the process of examining a collection of “data points,” and grouping the data points into “clusters” according to some distance measure. The goal is that data points in the same cluster have a small distance from one another, while data points in different clusters are at a large distance from one another. Cluster analysis divides data into well-formed groups. Well-formed clusters should capture the “natural” structure of the data. This paper focuses on PAM, CLARA and DBSCAN clustering methods. These methods are used to categorize the different districts of Karnataka which are having similar crop production.

TOOLS USED

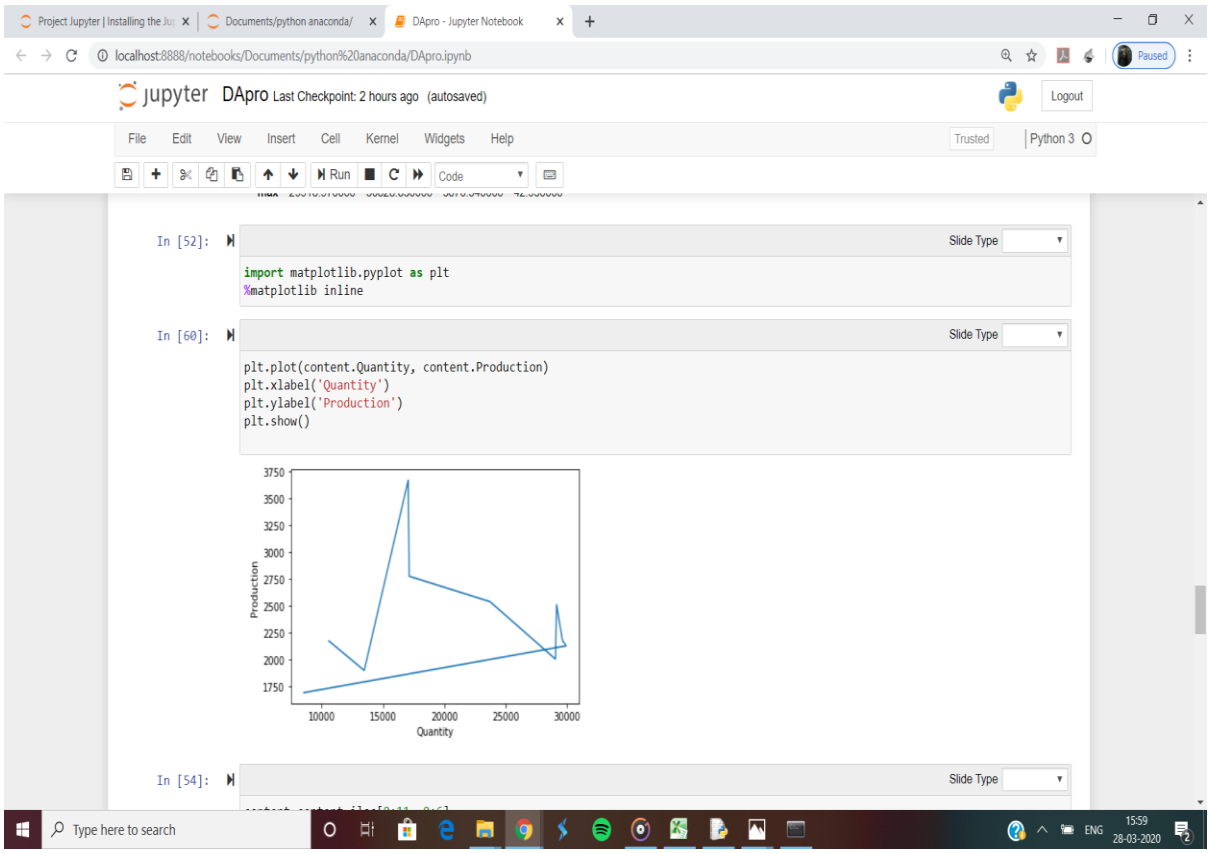
1). JUPYTER NOTEBOOK :

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. Project Jupyter is a nonprofit organization created to "develop open-source software, open-standards, and services for interactive computing across dozens of programming languages". Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more. Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter 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. Jupyter Notebook (formerly IPython Notebooks) is a web- based interactive computational environment for creating Jupyter notebook documents. The "notebook" term can colloquially make reference to many different entities, mainly the Jupyter web application, Jupyter Python web server, or Jupyter document format depending on context. A Jupyter Notebook document is a JSON document, following a versioned schema, and containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots and rich media, usually ending with the ".ipynb" extension. A Jupyter Notebook can be converted to a number of open standard output formats through "Download As" in the web interface, via the nbconvert library or "jupyter nbconvert" command line interface in a shell. To simplify visualisation of Jupyter notebook documents on the web, the nbconvert library is provided as a service through NbViewer which can take a URL to any publicly available notebook document, convert it to HTML on the fly and display it to the user.

Project Jupyter is a nonprofit organization created to "develop open-source software, open-standards, and services for interactive computing across dozens of programming languages". Spun-off from IPython in 2014 by Fernando Pérez, Project Jupyter supports execution environments in several dozen languages. Project Jupyter's name is a reference to the three core programming languages supported by Jupyter, which are Julia, Python and R, and also a homage to Galileo's notebooks recording the discovery

of the moons of Jupiter. Project Jupyter has developed and supported the interactive computing products Jupyter Notebook, JupyterHub, and JupyterLab, the next-generation version of Jupyter Notebook.

WORKING EXAMPLE ON JUPYTER :



2) ANACONDA PYTHON :

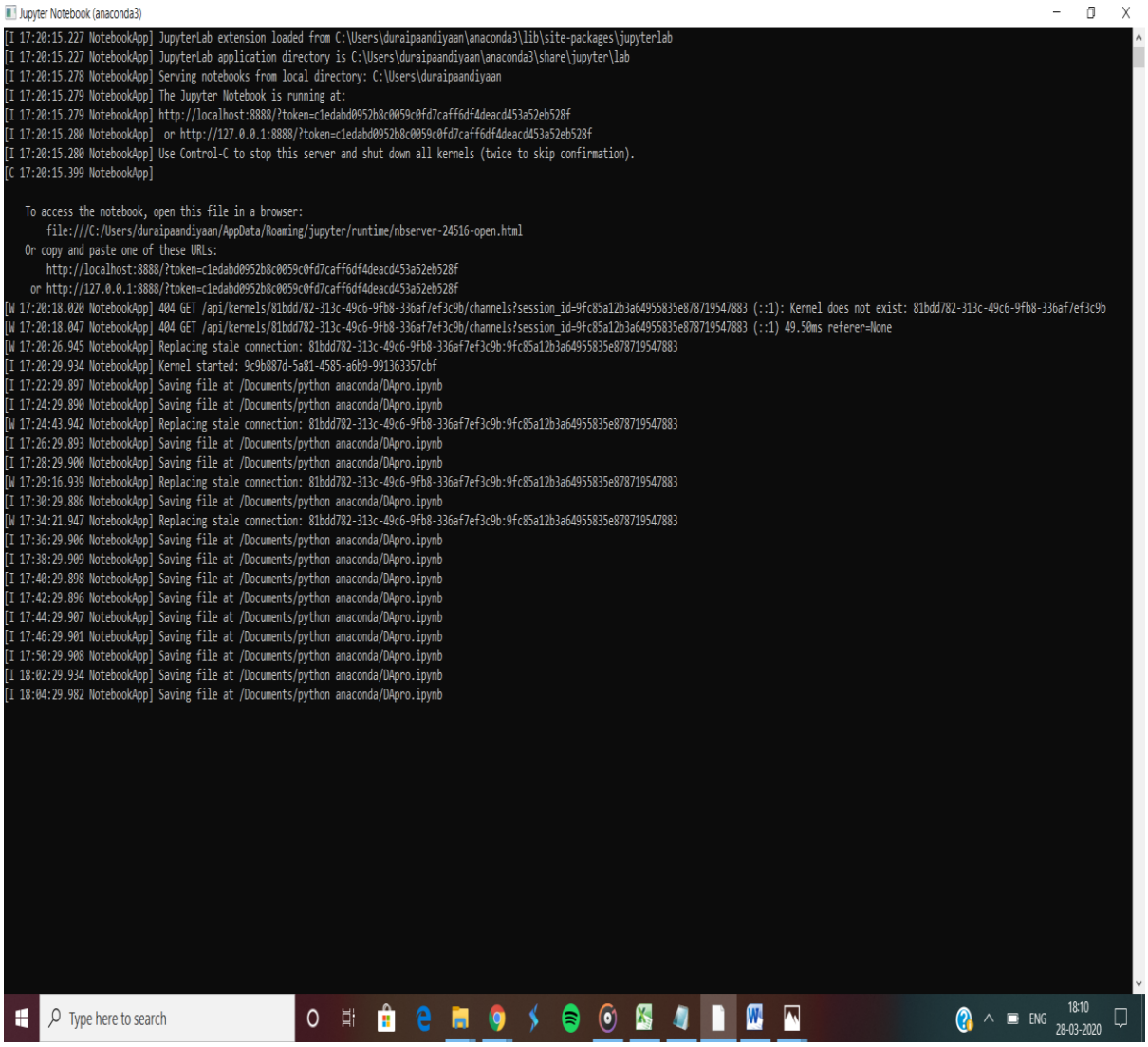
Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Package versions are managed by the package management system conda. The Anaconda distribution includes data-science packages suitable for Windows, Linux, and MacOS.Anaconda Enterprise is an enterprise-ready, secure and scalable data science platform that empowers teams to govern data science assets, collaborate and deploy their data science projects.

Anaconda distribution comes with 1,500 packages selected from PyPI as well as the conda package and virtual environment manager. It also includes a GUI, Anaconda Navigator, as a graphical alternative to the command line interface (CLI).The big difference between conda and the pip package manager is in how package dependencies are managed. Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.

The following applications are available by default in Navigator :

- JupyterLab
- Jupyter Notebook
- QtConsole[12]
- Spyder
- Glue
- Orange
- RStudio
- Visual Studio Code

TIMELOG OF PYTHON ANACONDA :



DATASET DESCRIPTION

Agriculture Production in India from 2001-2014. This Dataset Describes the Agriculture Crops Cultivation/Production in India. This Dataset can solves the problems of various crops Cultivation/production in India. Across The Globe India Is The Second Largest Country having People more than 1.3 Billion. Many People Are Dependent On The Agriculture And it is the Main Resource. In Agriculture Cultivation / Production Having More Problems. I want to solve the Big problem in India and useful to many more people. This Dataset Describes the Agriculture Crops Cultivation/Production in India .It contain information like crop ,variety ,state ,quantity ,production ,season .

Crops : string,
State : string,
Cultivation : Integer,
Quantity: Integer,
Production : Integer,
Yield : Integer,
place(State,crops)

This Dataset to be downloaded From kaggle. It shown about various production in India .this Data given here is big Data. From this project we are here to give an analyze of crops in agriculture season over various States with its production , cultivation , Yields.

EXAMPLE :

Crops	State	Quantity	Cultivation	Production	Yield
ARHAR	Karnataka	10593.15	16528.68	2172.46	7.47
ARHAR	Gujarat	13468.82	19551.9	1898.3	9.59
ARHAR	Andhra Pradesh	17051.66	24171.65	3670.54	6.42
ARHAR	Maharashtra	17130.55	25270.26	2775.8	8.72
COTTON	Maharashtra	23711.44	33116.82	2539.47	12.69
COTTON	Maharashtra	23711.44	33116.82	2539.47	12.69
COTTON	Punjab	29047.1	50828.83	2003.76	24.39
COTTON	Andhra	29140.77	44756.72	2509.99	17.83

	Pradesh				
COTTON	Gujarat	29616.09	42070.44	2179.26	19.05
COTTON	Haryana	29918.97	44018.18	2127.35	19.9
GRAM	Rajasthan	8552.69	12610.85	1691.66	6.83
GRAM	Madhya Pradesh	9803.89	16873.17	1551.94	10.29
GRAM	Uttar Pradesh	12833.04	21618.43	1882.68	10.93
GRAM	Maharashtra	12985.95	18679.33	2277.68	8.05
GRAM	Andhra Pradesh	14421.98	26762.09	1559.04	16.69
GROUNDNUT	Karnataka	13647.1	17314.2	3484.01	4.71
GROUNDNUT	Andhra Pradesh	21229.01	30434.61	2554.91	11.97
GROUNDNUT	Tamil Nadu	22507.86	30393.66	2358	11.98
GROUNDNUT	Andhra Pradesh	21229.01	30434.61	2554.91	11.97
GROUNDNUT	Gujarat	22951.28	30114.45	1918.92	13.45
GROUNDNUT	Maharashtra	26078.66	32683.46	3207.35	9.33
MAIZE	Bihar	13513.92	19857.7	404.43	42.95
MAIZE	Karnataka	13792.85	20671.54	581.69	31.1

DATASET LINK :

https://www.kaggle.com/srinivas1/agricuture-crops-production-in-india.

EXISTING SYSTEM

The Existing System uses Exploratory Data Analysis (EDA) and Data Virtualization in R for Agriculture crops prediction analysis. It analyses the dataset to summarize their main characteristics, often with visual methods. It mainly focuses more narrowly on checking assumptions required for model fitting and hypothesis testing, and handling missing values and making transformations of variables as needed. The dataset does not undergo pre- processing for removal of noise values and to remove unwanted data. The traditional analysis method mainly depends on database system therefore using a traditional database will not be efficient.

PROPOSED SYSTEM

The Proposed System analyses the Agriculture crops prediction dataset using python in jupyter notebook. This project uses pandas and seaborn library in jupyter notebook. Pandas is an open source library that allows you perform data manipulation in Python. Pandas library is built on top of Numpy, meaning Pandas needs Numpy to operate. Pandas provide an easy way to create, manipulate and wrangle the data. Pandas is also an elegant solution for time series data. Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. In this proposed work, binning and regression methods are used for cleaning noisy data to improve the mining results. The posteriori distributions is calculated as w= q*h/N where q is the dataset and h is the record considered and N is count of records in dataset. Every component is estimated by a parameter (Pk) and its probability with every other parameter that goes under different clusters (Ck).

IMPLEMENTATION AND RESULTS

Step 1: Open Jupyter Notebook, CreateNewNotebook.
Step 2: Import the dataset into the notebook.
Step 3: Import the required libraries.
Step 4: Read the dataset and create a dataframe.
Step 5: Analyse the given dataset using various techniques.
Step 6: Visualize the results.

CODE :

```
#import the libraries
import numpy as np
import pandas as pd
import seaborn as sns


#importing matplotlib libraries
import matplotlib.pyplot as plt
get_ipython().run_line_magic('matplotlib', 'inline')
#reading and displaying the csv form of data set
content = pd.read_csv('agri.csv')
content
```

#**printing the type of content**
type(content)

#**printing head and tail of dataset**
content.head()
content.tail()

#**renaming the column**
content = content.rename(columns={'Crops':'INCrops'})
content.head(5)

#**retriveing the specific values**
single_State = content[content.State == 'Maharashtra']
single_State

#**retriveing single and multiple columns**
single_column = content.INCrops
single_column
more_columns = content[{'State','Yield'}]
more_columns

#**retriveing values using specified loc**
content.loc [{2,5},'Production']
content.loc[2:7,{'Production','INCrops','State'}]

#**retriveing values using index positions**
content.iloc[2:7,1:3]

#**describing about the dataset using describe()**
content.describe()

#**lineplot**
plt.plot(content.Quantity, content.Production)
plt.xlabel('Quantity')
plt.ylabel('Production')
plt.show()

#**reducing the rows in dataset**
```
content=content.iloc[0:11 ,0:6]
content.head(11)
```

#**A clear line plot of "crop production and quantity"**
```
plt.plot(content.Production, content.Quantity)
plt.xlabel('Production')
plt.ylabel('Quantity')
plt.show()
```

#**histogram of "crop yield"**
```
plt.hist(content.Yield)
plt.xlabel('Yield')
plt.ylabel('Frequency')
plt.title('Yield of Agricultural Crops in India')
plt.show()
```

#**scatter plot of "particular states and their crop cultivation"**
```
plt.scatter(content.State,content.Cultivation)
plt.xlabel('State')
plt.ylabel('Cultivation')
plt.title('Scatter plot of production and cultivation')
plt.xticks(rotation=65)
plt.show()
```

#**Analysis of "crop yield and profuction"**
```
data = pd.read_csv('agri.csv')
fig,axs = plt.subplots(figsize=(10,6))
crop_wise_yield = data.groupby(['Crops']).sum()['Yield']
plt.plot(crop_wise_yield)
crop_wise_production = data.groupby(['Crops']).sum()['Production']/10
plt.plot(crop_wise_production)
plt.xticks(rotation ='vertical')
plt.legend()
```

#**Analysis of "each state's crop cultivation and production"**
```
state_crop_cult = data.groupby(['State'])
index = list(state_crop_cult.indices.keys())
state_crop_cult.sum()[['Production', 'Cultivation']].plot(kind='bar',figsize=(12,7))
```

#**“Analysis of agriculture data of india”**
dataframe = data.groupby('Crops').sum().plot(kind='bar',figsize=(15,7))
dataframe

IMPLEMENTATION AND OUTPUT :

1)
content = pd.read_csv('agri.csv')
content

Output:

Out[7]:

	Crops	State	Quantity	Cultivation	Production	Yield
0	ARHAR	Karnataka	10593.15	16528.68	2172.46	7.47
1	ARHAR	Gujarat	13468.82	19551.90	1898.30	9.59
2	ARHAR	Andhra Pradesh	17051.66	24171.65	3670.54	6.42
3	ARHAR	Maharashtra	17130.55	25270.26	2775.80	8.72
4	COTTON	Maharashtra	23711.44	33116.82	2539.47	12.69
5	COTTON	Maharashtra	23711.44	33116.82	2539.47	12.69
6	COTTON	Punjab	29047.10	50828.83	2003.76	24.39
7	COTTON	Andhra Pradesh	29140.77	44756.72	2509.99	17.83
8	COTTON	Gujarat	29616.09	42070.44	2179.26	19.05
9	COTTON	Haryana	29918.97	44018.18	2127.35	19.90
10	GRAM	Rajasthan	8552.69	12610.85	1691.66	6.83
11	GRAM	Madhya Pradesh	9803.89	16873.17	1551.94	10.29
12	GRAM	Uttar Pradesh	12833.04	21618.43	1882.68	10.93
13	GRAM	Maharashtra	12985.95	18679.33	2277.68	8.05
14	GRAM	Andhra Pradesh	14421.98	26762.09	1559.04	16.69
15	GROUNDNUT	Karnataka	13647.10	17314.20	3484.01	4.71
16	GROUNDNUT	Andhra Pradesh	21229.01	30434.61	2554.91	11.97
17	GROUNDNUT	Tamil Nadu	22507.86	30393.66	2358.00	11.98
18	GROUNDNUT	Andhra Pradesh	21229.01	30434.61	2554.91	11.97
19	GROUNDNUT	Gujarat	22951.28	30114.45	1918.92	13.45
20	GROUNDNUT	Maharashtra	26078.66	32683.46	3207.35	9.33
21	MAIZE	Bihar	13513.92	19857.70	404.43	42.95
22	MAIZE	Karnataka	13792.85	20671.54	581.69	31.10

2) type(content)

output:

```
pandas.core.frame.DataFrame
```

3) content.head()

output:

	Crops	State	Quantity	Cultivation	Production	Yield
0	ARHAR	Karnataka	10593.15	16528.68	2172.46	7.47
1	ARHAR	Gujarat	13468.82	19551.90	1898.30	9.59
2	ARHAR	Andhra Pradesh	17051.66	24171.65	3670.54	6.42
3	ARHAR	Maharashtra	17130.55	25270.26	2775.80	8.72
4	COTTON	Maharashtra	23711.44	33116.82	2539.47	12.69

4) content = content.rename(columns={'Crops':'INCrops'})
content.head(5)

output:

	INCrops	State	Quantity	Cultivation	Production	Yield
0	ARHAR	Karnataka	10593.15	16528.68	2172.46	7.47
1	ARHAR	Gujarat	13468.82	19551.90	1898.30	9.59
2	ARHAR	Andhra Pradesh	17051.66	24171.65	3670.54	6.42
3	ARHAR	Maharashtra	17130.55	25270.26	2775.80	8.72
4	COTTON	Maharashtra	23711.44	33116.82	2539.47	12.69

5) single_column = content.INCrops
single_column

Output:

```
0          ARHAR
1          ARHAR
2          ARHAR
3          ARHAR
4         COTTON
5         COTTON
6         COTTON
7         COTTON
8         COTTON
9         COTTON
10          GRAM
11          GRAM
12          GRAM
13          GRAM
14          GRAM
15     GROUNDNUT
16     GROUNDNUT
17     GROUNDNUT
18     GROUNDNUT
19     GROUNDNUT
20     GROUNDNUT
21         MAIZE
22         MAIZE
Name: INCrops, dtype: object
```

6) more_columns = content[{'State','Yield'}]
more_columns

Output:

	Yield	State
0	7.47	Karnataka
1	9.59	Gujarat
2	6.42	Andhra Pradesh
3	8.72	Maharashtra
4	12.69	Maharashtra
5	12.69	Maharashtra
6	24.39	Punjab
7	17.83	Andhra Pradesh
8	19.05	Gujarat
9	19.90	Haryana
10	6.83	Rajasthan
11	10.29	Madhya Pradesh
12	10.93	Uttar Pradesh
13	8.05	Maharashtra
14	16.69	Andhra Pradesh
15	4.71	Karnataka
16	11.97	Andhra Pradesh
17	11.98	Tamil Nadu
18	11.97	Andhra Pradesh
19	13.45	Gujarat
20	9.33	Maharashtra
21	42.95	Bihar
22	31.10	Karnataka

7) content.loc [{2,5},'Production']

Output:

```
2    3670.54
5    2539.47
Name: Production, dtype: float64
```

8) content.loc[2:7,{'Production','INCrops','State'}]

Output:

	Production	State	INCrops
2	3670.54	Andhra Pradesh	ARHAR
3	2775.80	Maharashtra	ARHAR
4	2539.47	Maharashtra	COTTON
5	2539.47	Maharashtra	COTTON
6	2003.76	Punjab	COTTON

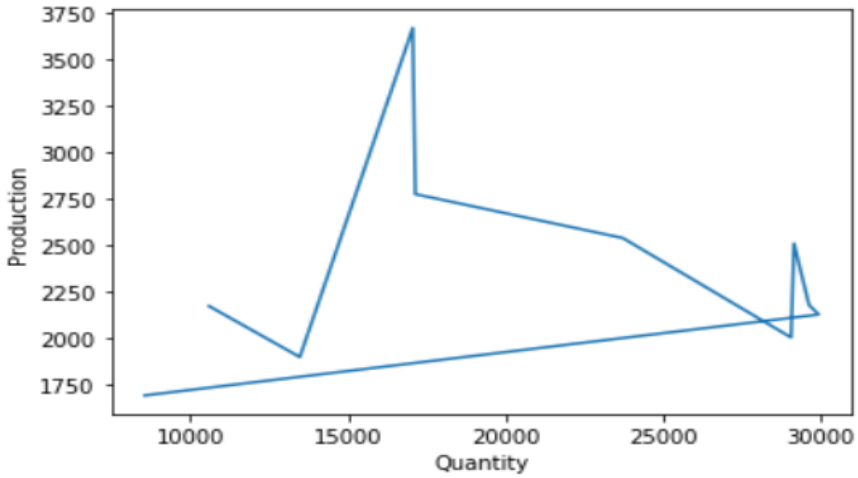
9) content.iloc[2:7,1:3]

Output:

	State	Quantity
2	Andhra Pradesh	17051.66
3	Maharashtra	17130.55
4	Maharashtra	23711.44
5	Maharashtra	23711.44

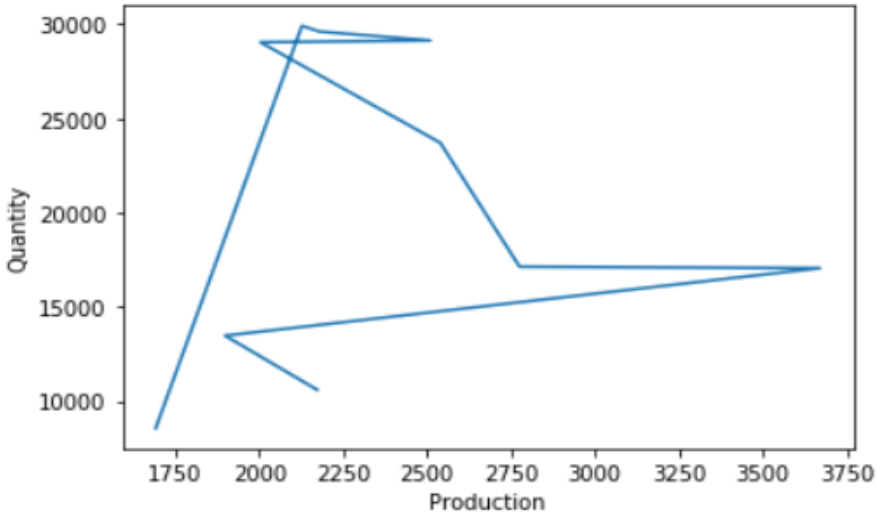
10)
```
plt.plot(content.Quantity, content.Production)
plt.xlabel('Quantity')
plt.ylabel('Production')
plt.show()
```

Output:



11).
```
plt.plot(content.Production, content.Quantity)
plt.xlabel('Production')
plt.ylabel('Quantity')
plt.show()
```

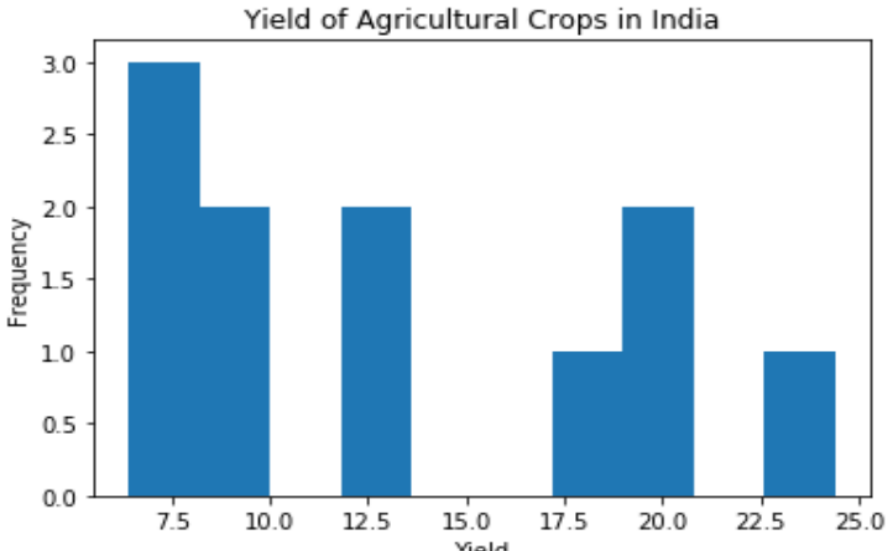
Output:



12)

```
plt.hist(content.Yield)
plt.xlabel('Yield')
plt.ylabel('Frequency')
plt.title('Yield of Agricultural Crops in India')
plt.show()
```

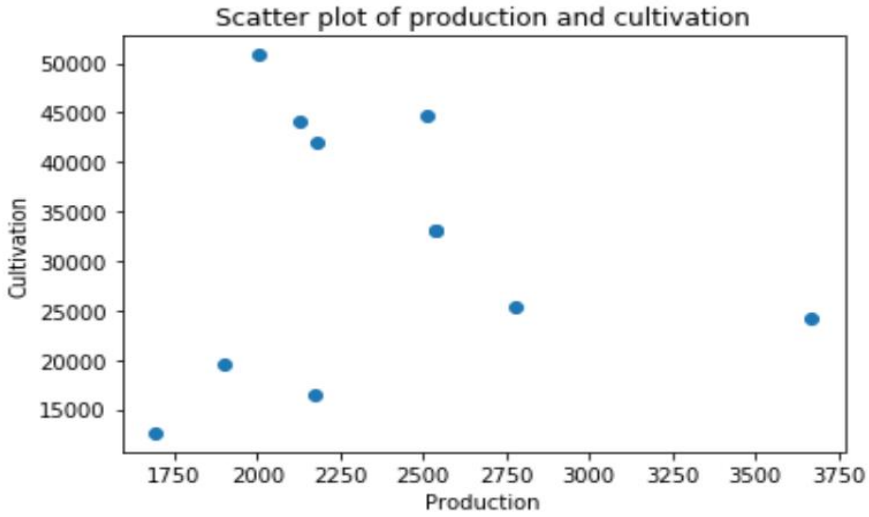
Output:



13)

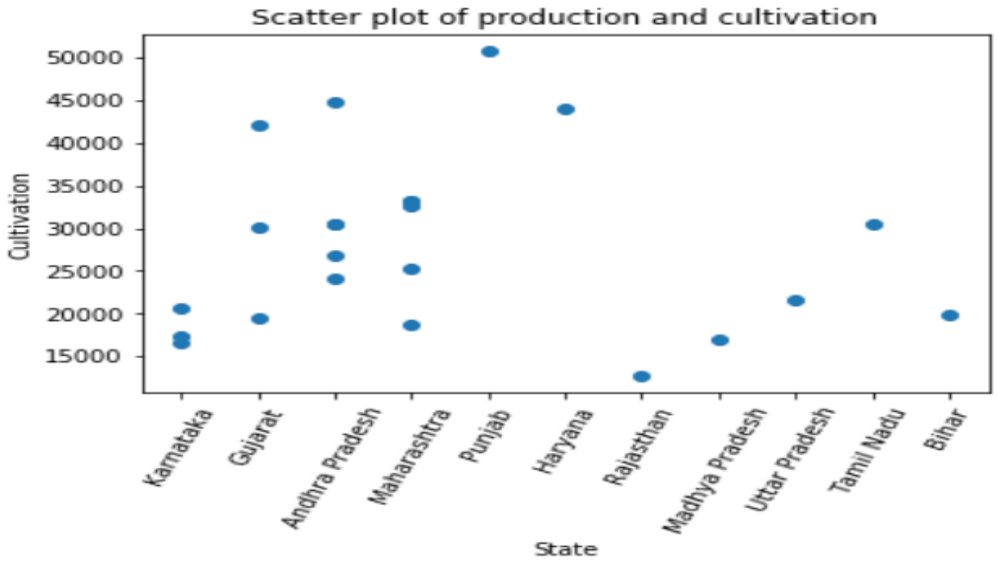
```
plt.scatter(content.Production,content.Cultivation)
plt.xlabel('Production')
plt.ylabel('Cultivation')
plt.title('Scatter plot of production and cultivation')
plt.show()
```

Output:



14)
```
plt.scatter(content.State,content.Cultivation)
plt.xlabel('State')
plt.ylabel('Cultivation')
plt.title('Scatter plot of production and cultivation')
plt.xticks(rotation=90)
plt.show()
```

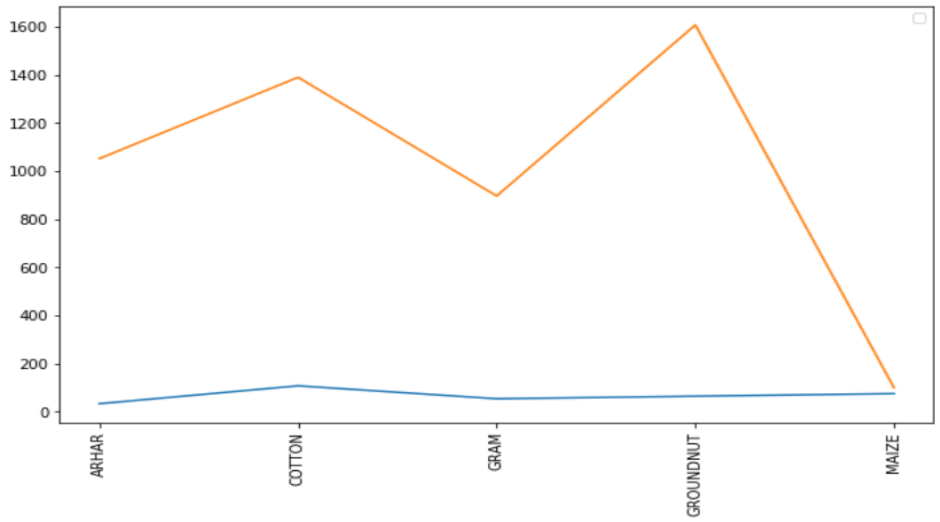
Output:



15) .
```
data = pd.read_csv('agri.csv')
fig,axs = plt.subplots(figsize=(10,6))
crop_wise_yield = data.groupby(['Crops']).sum()['Yield']
plt.plot(crop_wise_yield)
crop_wise_production = data.groupby(['Crops']).sum()['Production']/10
plt.plot(crop_wise_production)
plt.xticks(rotation ='vertical')
plt.legend()
```

output :

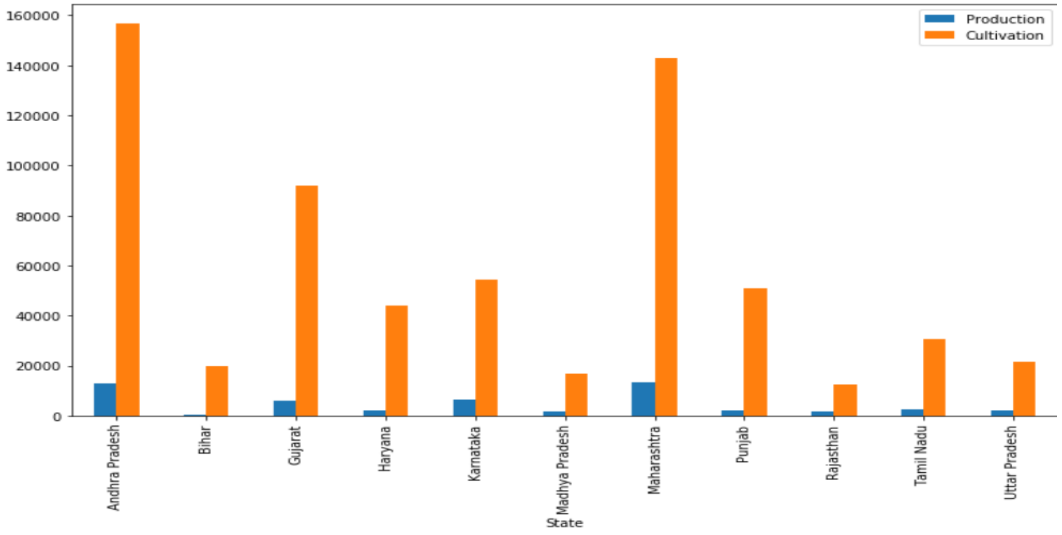
```
Out[85]:<matplotlib.legend.Legend at 0x1452c701688>
```



16).
state_crop_cult = data.groupby(['State'])
index = list(state_crop_cult.indices.keys())
state_crop_cult.sum()[['Production', 'Cultivation']].plot(kind='bar',figsize=(12,7))

Output:

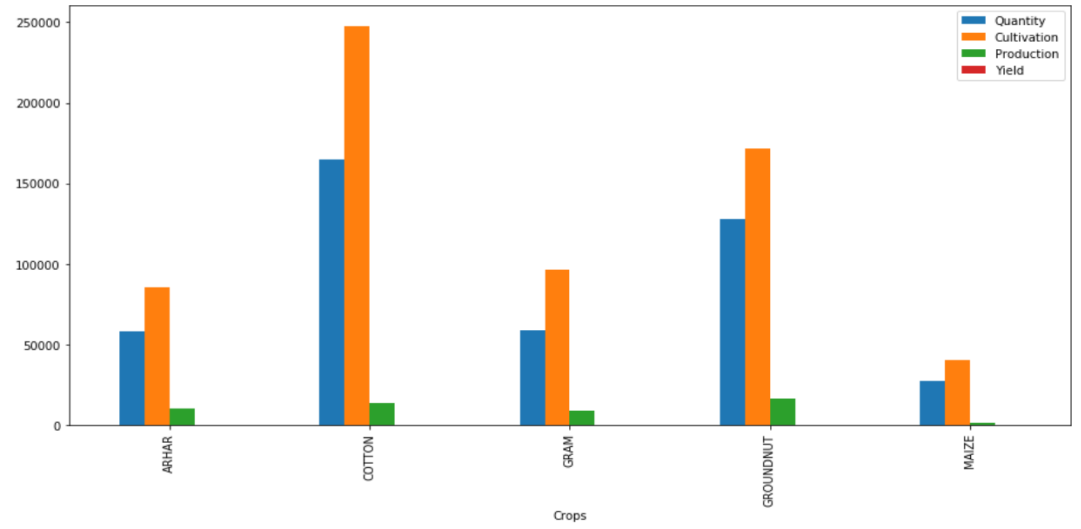
```
<matplotlib.axes._subplots.AxesSubplot at 0x1452dcdb3c8>
```



```
17)
```
dataframe = data.groupby('Crops').sum().plot(kind='bar',figsize=(15,7))
dataframe

Output:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1452dd0f1c8>
```



CONCLUSION:

As a result of penetration of technology into agriculture field, there is a marginal improvement in the productivity. The innovations have led to new concepts like digital agriculture, smart farming, precision agriculture etc.In the literature,it has been observed that analysis has been done on agriculture soils,hidden patterns discovery using data set related to climatic conditions and crop yields data. The activities of agriculture field are numerous like weather forecasting,soil quality assessment,seeds selection,crop yield prediction etc.In this survey, the specific activity, crop yield prediction has been surveyed and the major trends have been identified.

The rice crop yield prediction has been done in the state of Maharashtra using data mining techniques in one of the works. The analysis has been done using machine learning framework WEKA.In the work carried out in, various algorithms applied in the assessment crop yield and mechanism for knowledge discovery has been discussed. The challenges and opportunities in the field of Big Data analytics in agriculture has been discussed in with a case study of Netherlands. Fuzzy logic designs have been used in optimizing the crop yields and the same has been explained in the research work in. A case study of Nebraska - USA and at a national scale for Argentina and Kenya has been done and presented in.The remote sensing technology for identification and measurement of the causes of yield gaps and their impact on final crop yield is presented in.

It can be concluded that the research in the field of agriculture with reference to using IT trends like data analytics is in its infancy. As the food is the basic need of humans, the requirement of getting the maximum yields using optimal resource will become the necessity in near future as a result of growing population. The survey outcomes indicate the need for improved techniques in crop yield analytics. There exists a lot of research scope in this research area.

REFERENCE:

1) https://get.anaconda.com/distribution/tutorial/
2) https://jupyter.org/
3) https://matplotlib.org/tutorials
4) https://docs.python.org/
5) https://docs.python.org/3.8/tutorial
6) https://www.kaggle.com

Dhivya B H, Manjula R, Siva Bharathi S, Madhumathi R. A Survey on Crop Yield Prediction based on Agricultural Data, International Journal of Innovative Research in Science, Engineering and Technology. 2017