VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI

Inter/Internship Report on "

"A PREDICTIVE MODEL FOR FORECASTING DEMAND AND SUPPLY INFORMATION OF TOP CROPS"

Submitted in partial fulfilment for the award of degree

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by

GAGAN C B

[1AK22CS402]5th SEM

Conducted at

COMPSOFT TECHNOLOGIES

AKSHAYA INSTITUTE OF TECHNOLOGY

Lingapura, Tumkur-Koratagereroad, Tumkur-572106

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the Inter/Intraship report entitled **"A predictive model for forecasting demand and supply information of TOP crops"** carried out by **GAGAN C B** bearing **1AK22CS402** a bonafide student of **Akshaya Institute Of Technology** in partial fulfillment for the award of Bachelor of Engineering in **Computer Science And Engineering** of the **Visvesvaraya Technological University**, Belagavi during the year **2023-24**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The report has been approved as it satisfies the academic requirements in respect of prescribed for the said Degree.

Name & Signature of Guide:
Mr. RAKESH S B.E, M.Tech
Assistant Professor, Dept. of CSE
AIT Tumkur

Name & Signature of HOD
Dr. PUSHPA R Ph.D, MISTE
Professor & Head, Dept. of CSE
AIT Tumkur

Name of the Examiners:

1.....

2. .....

Signature with Date:

1. .....

2. .....

D E C L A R A T I O N

I, **GAGAN C B**, THIRD YEAR STUDENT OF COMPUTER SCIENCE, AKSHAYA INSTITUTE OF TECHNOLOGY, DECLARE THAT THE INTERNSHIP HAS BEEN SUCCESSFULLY COMPLETED IN COMPSOFT TECHNOLOGIES.

THIS REPORT IS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR AWARD OF BACHELOR DEGREE IN COMPUTER SCIENCE, DURING THE ACADEMIC YEAR 2024-2025.

DATE : 30/11/2023

PLACE :AKSHAYA INSTITUTE OF TECHNOLOGY
Lingapura, Tumkur-Koratagereroad, Tumkur-572106

USN : 1AK22CS402

NAME : GAGAN C B

Internship Offer Letter

Date: 25th October, 2023

Name: **GAGAN C B**
USN: **1AK22CS402**
Placement ID: **23OCTMLBONE**

Dear Student,

We would like to congratulate you on being selected for the **Machine Learning with Python (Research Based)** Internship position with **Compsoft Technologies**, effective Start Date **25th October, 2023**, All of us are excited about this opportunity provided to you!

This internship is viewed as being an educational opportunity for you, rather than a part-time job. As such, your internship will include training/orientation and focus primarily on learning and developing new skills and gaining a deeper understanding of concepts of **Machine Learning with Python (Research Based)** through hands-on application of the knowledge you learn while you train with the senior developers. You will be bound to follow the rules and regulations of the company during your internship duration.

Again, congratulations and we look forward to working with you!.

Sincerely,

Nithin K. S
Project Manager
Compsoft Technologies
No. 363, 19th main road,
1st Block Rajajinagar
Bangalore - 560010

A C K N O W L E D G E M E N T

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals and elders. We would like to take this opportunity to thank them all.

We heartily extend our words of gratitude to the technical guide of Compsoft Technologies Pvt. Ltd. , for his valuable advice, encouragement and suggestion given to our team in the course of our Internship project. We convey our gratitude to her for having constantly monitored the development of the mini- project and setting up precise deadlines.

We would like to express our immense gratitude to Head of Department Dr. PUSHPA R, for his unfailing encouragement and suggestions given to us in the course of our work.

We would like to take this opportunity to express our gratitude to the Principal,Dr, K.V Sreenivas Rao, for giving us this opportunity to enrich our knowledge.

Finally, a note of thanks to the Department of Computer Science and Engineering, both teaching and non-teaching staff for their cooperation extended to us

Last but not the least, we acknowledge the support and feedback of our parents, guardians and friends, for their indispensable help always.

USN :1AK22CS402

NAME : GAGAN C B

ABSTRACT

Agriculture is the biggest industry in India and it generates a significant number of employments in the country. The features of weather, geography, and soil of India are diverse. As a consequence, a range of crops are grown in the country. India crops include food grains like rice, wheat, and pulses.

The increase in population will be more in developing countries like India. When the price of any commodities set too high then the suppliers try to produce more goods to make more profit. Conversely, if the supply is less for any commodities, as consumers have to compete with one other to buy the less supplied goods, results in increased price for the commodity, making consumers suffer with the high price.

As there is no synchronization in production and demand for the agricultural commodities, either farmer fail to get good market prices for their products, or consumer suffers high prices due to less production.

Around 42% of the people depend on agriculture for their livelihood. The economic upliftment of farmers happens when there is a seamless transfer of agricultural produce from producers to the consumers. It is evident that there is a huge gap between demand and supply of various crops, due to which both farmers and consumers are facing problems. At present, in India there is no system in place to efficiently manage this demand and supply issue. The potential of present-day technologies like data analytics, machine learning can be exploited to overcome these issues. The available data about the demand, supply, price variation of the crops and other factors affecting the supply chain of agricultural produce can be used to analyse and come up with a model to predict and forecast market variations of agricultural crops.

CONTENTS

Learning Objectives/Internship Objectives

- Internships are generally thought of to be reserved for college students looking to gain experience in a particular field. However, a wide array of people can benefit from Training Internships in order to receive real world experience and develop their skills.

- An objective for this position should emphasize the skills you already possess in the area and your interest in learning more

- Internships are utilized in a number of different career fields, including architecture, engineering, healthcare, economics, advertising and many more.

- Some internship is used to allow individuals to perform scientific research while others are specifically designed to allow people to gain first-hand experience working.

- Utilizing internships is a great way to build your resume and develop skills that can be emphasized in your resume for future jobs. When you are applying for a Training Internship, make sure to highlight any special skills or talents that can make you stand apart from the rest of the applicants so that you have an improved chance of landing the position.

COMPANY PROFILE

A Brief History of Compsoft Technologies

Compsoft Technologies, was incorporated with a goal ”To provide high quality and optimal Technological Solutions to business requirements of our clients”. Every business is a different and has a unique business model and so are the technological requirements. They understand this and hence the solutions provided to these requirements are different as well. They focus on clients requirements and provide them with tailor made technological solutions. They also understand that Reach of their Product to its targeted market or the automation of the existing process into e-client and simple process are the key features that our clients desire from Technological Solution they are looking for and these are the features that we focus on while designing the solutions for their clients.

Sarvamoola Software Services. is a Technology Organization providing solutions for all web design adevelopment, MYSQL, PYTHON Programming, HTML, CSS, ASP.NET andLINQ. Meeting the ever increasing automation requirements, Sarvamoola Software Services. specialize in ERP, Connectivity, SEO Services, Conference Management, effective web promotion and tailor-made software products, designing solutions best suiting clients requirements

Compsoft Technologies, strive to be the front runner in creativity and innovation in software development through their well-researched expertise and establish it as an out of the box software development company in Bangalore, India. As a software development company, they translate this software development expertise into value for their customers through their professional solutions.

They understand that the best desired output can be achieved only by understanding the clients demand better. Compsoft Technologies work with their clients and help them to defiine their exact solution requirement. Sometimes even they wonder that they have completely redefined their solution or new application requirement during the brainstorming session, and here they position themselves as an IT solutions consulting group comprising of high caliber consultants.

They believe that Technology when used properly can help any business to scale and achieve new heights of success. It helps Improve its efficiency, profitability, reliability; to put it in one sentence ” Technology helps you to Delight your Customers” and that is what we want to achieve.

ABOUT THE COMPANY

Compsoft Technologies is a Technology Organization providing solutions for all web design and development, MYSQL, PYTHON Programming, HTML, CSS, ASP.NET and LINQ. Meeting the ever increasing automation requirements, Compsoft Technologies specialize in ERP, Connectivity, SEO Services, Conference Management, effective web promotion and tailor-made software products, designing solutions best suiting clients requirements. The organization where they have a right mix of professionals as a stakeholders to help us serve our clients with best of our capability and with at par industry standards. They have young, enthusiastic, passionate and creative Professionals to develop technological innovations in the field of Mobile technologies, Web applications as well as Business and Enterprise solution.

Motto of our organization is to "Collaborate with our clients to provide them with best bring a cascading a positive effect in their business shape as well". Providing a Complete suite of technical solutions is not just our tag line, it is Our Vision for Our Clients and for Us, We strive hard to achieve it

Products of Compsoft Technologies.

Android Apps

It is the process by which new applications are created for devices running the Android operating system. Applications are usually developed in Java (and/or Kotlin; or other such option) programming language using the Android software development kit (SDK), but other development environments are also available, some such as Kotlin support the exact same Android APIs (and bytecode), while others such as Go have restricted API access.

The Android software development kit includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and zutorials. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS X 10.5.8 or later, and Windows 7 or later. As of March 2015, the SDK is not available on Android itself, but softwaredevelopment is possible by using specialized Android applications.

Web Application

It is a client–server computer program in which the client (including the user interface and client-side logic) runs in a web browser. Common web applications include web mail, online retail sales, online auctions, wikis, instant messaging services and many other functions. web applications use web documents written in a standard format such as HTML and JavaScript,which are supported by a variety of web browsers. Web applications can be considered as a specifific variant of client–server software where the client software is downloaded to the client machine when visiting the relevant web page, using standard

procedures such as HTTP. The Client web software updates may happen each time the web page is visited. During the session, the web browser interprets and displays the pages, and acts as the universal client for any web application. The use of web application frameworks can often reduce the number of errors in a program, both by making the code simpler, and by allowing one team to concentrate on the framework while another focuses on a specifified use case. In applications which are exposed to constant hacking attempts on the Internet, security- related problems can be caused by errors in the program.

Frameworks can also promote the use of best practices such as GET after POST. There are some who view a web application as a two-tier architecture. This can be a "smart" client that performs all the work and queries a "dumb" server, or a "dumb" client that relies on a "smart" server. The client would handle the presentation tier, the server would have the database

Services provided by Compsoft Technologies.

- Core Java and Advanced Java
- Web services and development
- .Dot Net Framework
- Event Management Service
- On The Job Training
- Selenium Testing
- Dot Net Framework
- Python
- Academic Project Guidance
- Software Training

INTRODUCTION

An Introduction to ML

Arthur Samuel, an early American leader in the field of computer gaming and artificial intelligence, coined the term "Machine Learning " in 1959 while at IBM. He defined machine learning as "the field of study that gives computers the ability to learn without being explicitly programmed ". However, there is no universally accepted definition for machine learning. Different authors define the term differently. We give below two more definitions.

- Machine learning is programming computers to optimize a performance criterion using example data or past experience. We have a model defined up to some parameters, and learning is the execution of a computer program to optimize the parameters of the model using the training data or past experience. The model may be predictive to make predictions in the future, or descriptive to gain knowledge from data.
- The field of study known as machine learning is concerned with the question of how to construct computer programs that automatically improve with experience.

Machine learning is a subfield of artificial intelligence that involves the development of algorithms and statistical models that enable computers to improve their performance in tasks through experience. These algorithms and models are designed to learn from data and make predictions or decisions without explicit instructions. There are several types of machine learning, including supervised learning, unsupervised learning, and reinforcement learning. Supervised learning involves training a model on labeled data, while unsupervised learning involves training a model on unlabeled data. Reinforcement learning involves training a model through trial and error. Machine learning is used in a wide variety of applications, including image and speech recognition, natural language processing, and recommender systems.

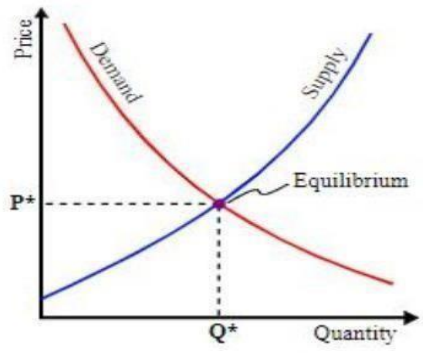
Problem Statement

"A predictive model for forecasting demand and supply information of TOP crops"

Built a python application that analyses the top crops at any given time, depending on the season or demand. You can use the dataset available on the Internet to use

INTRODUCTION

The world's population may reach 6.3 billion and this population growth may surge to 9.5 to 10.0 billion by 2050. The increase in population will be more in developing countries like India than developed countries. The economic growth also has been increased dramatically in the recent years in such countries. The increase in income results in increased demand for more and better food. In India, for instance, the increased income has doubled the expenditure on food commodities. , etc. As shown in Fig 1, the price variation is mainly due to the mismatch in demand and supply of these agricultural products. When the price of any commodity is set too high then the suppliers (farmers) tries to produce more goods to make more profit. When the price of any commodity is set too high then the consumers will tend to purchase less because of high rate, and the suppliers (farmers) incurs the loss. Conversely, if the supply is less for any commodities, as consumers have to compete with one other to buy the less supplied goods, results in increased price for the Hence, in order to reduce the mismatch in demand and supply of food crops effectively, primarily the expected demand for various food commodities needs to be forecasted and guide the farmers accordingly. So there is a need for some system that could guide the farmers in selecting and growing the crops to satisfy the actual demand of the society. This could eliminate the gap between the consumer's demand and producer's supply and reducing the loss for both consumers and farmers. Big data analytics as an emerging trend could help in providing solutions for such problems. An effective forecasting model is proposed and has been implemented in this paper that (i) determine the gap between the demand for and supply of the crops that have to be reduced. (ii) Forecasts the demand of various food commodities that helps the system to guide the farmers in selecting and growing the appropriate crops to satisfy the demand and hence reducing the gap or mismatch between the demand and supply of the crops Karnataka. Also compared the forecasted values of the same commodities for the year 2017 market data and the results have shown the promising equilibrium.The remaining portion of the paper is organized as follows in section II, related work has been discussed. In section III the proposed Demand-Prediction Forecasting Model[DPFM] has been described, section IV illustrates the implementation of the model, section V evaluates the results and finally, section VI concludes the work



Demand, supply and price inflation

SYSTEM ANALYSIS

Existing System:

The necessary data required for this analysis has been gathered from the sources like Ministryof-Agriculture, Agmarknet, Directorate of marketing and Inspection, Ministry-of-Agriculture and Farmers-Welfare, Government of India, National Horticulture Board (NHB) India, HOPCOMS Horticultural Producer's Cooperative Marketing and Processing Society Limited by applying Web scraping methods and stored in the local repository and detailed market survey. Data preprocessing module removes the noise from the collected data sets and builds the missing values before applying forecasting algorithm for better performance. All the data sets collected were integrated into a single dataset. During this process, incomplete information is eliminated and all NA (Not Applicable) values are aggregated to the average value. The data sets collected contains multiple attributes, the required attributes are separated and stored as a separate data frame, and then the data frame is converted to time series data. The pre-processed data has been stored and processed as clusters in distributed mode for effective application of the algorithm and analysis. The Hadoop HDFS and Map Reduce paradigm have been used to provide a distributed data storing and parallel

Proposed System:

The proposed DPFM model is implemented using the integrated R-Hadoop machine learning based prediction modeling that provides scalable and parallel processing environment. Also, a Map-Reduce programming model has been developed in R environment to perform the efficient analysis of the data stored in Hadoop clusters. The server that runs R submits the jobs to Hadoop which in turn distributes the work among m machines in the cluster and gets the result

SOFTWARE REQUIREMENT SPECIFICATIONS

System configurations

The software requirement specification can produce at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by established a complete information description, a detailed functional description, a representation of system behavior, and indication of performance and design constrain, appropriate validate criteria, and other information pertinent to requirements.

Hardware Requirement:

- System : i3 2.4 GHz & above
- Hard Disk : 256GB
- Ram : 4GB

Software Requirements:

- Operating system : Windows 10
- Coding Language : Python
- Application : Jupyter Notebook

. There are a lot of python libraries which could be used to build visualization like *matplotlib, vispy, bokeh, seaborn, pygal, folium, plotly, cufflinks*, and *networkx*. Of the many, *matplotlib* and *seaborn* seems to be very widely used for basic to intermediate level of visualizations.

However, two of the above are widely used for visualization i.e.

- **Matplotlib:** It is an amazing visualization library in Python for 2D plots of arrays, It is a multiplatform data visualization library built on *NumPy* arrays and designed to work with the broader *SciPy* stack. Use the below command to install this library:

```
pip install matplotlib
```

Seaborn: This library sits on top of *matplotlib*. In a sense, it has some flavors of *matplotlib* while from the visualization point, its is much better than *matplotlib* and has added features as well. Use the below command to install this library:

```
pip install seaborn
```

Step-by-step Approach

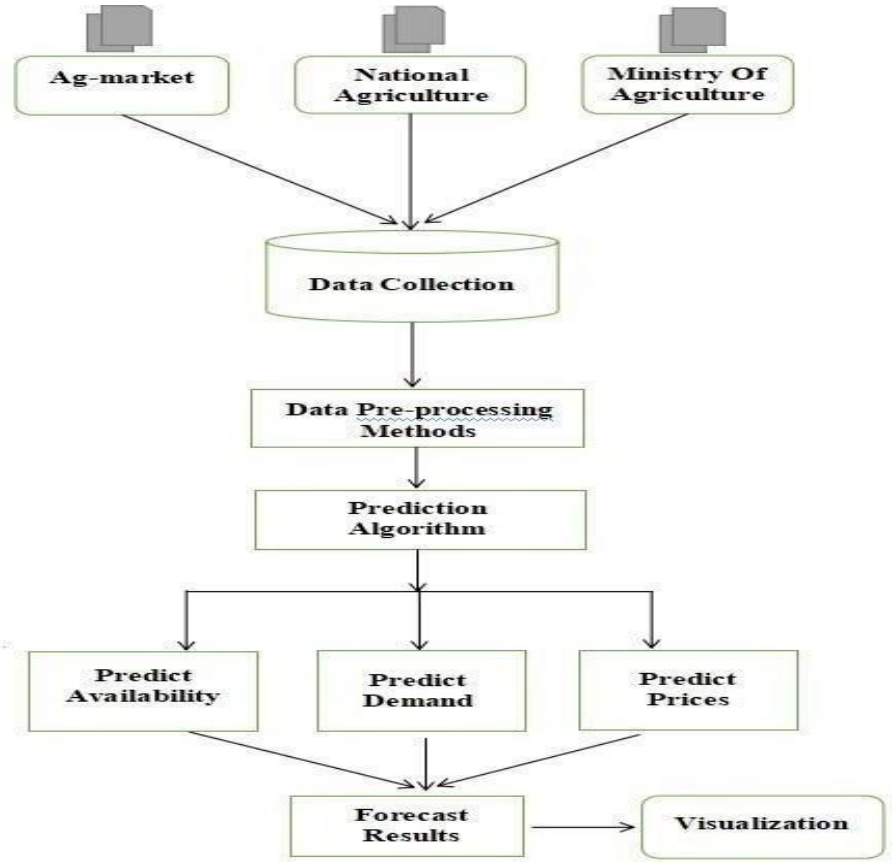
- Import required modules Load the dataset.
- Display the data and constraints of the loaded dataset.

Use different methods to visualize various illustrations from the data

DESIGN ANALYSIS

This consists of modules for data collection, data repository, data pre-processing, clustering, MapReduce and forecasting the demand.

SYSTEM ARCHITECTURE



System Architecture

IMPLEMENTATION

Implementation is the stage where the theoretical design is turned into a working system. The most crucial stage in achieving a new successful system and in giving confidence on the new system for the users that it will work efficiently and effectively.

The system can be implemented only after thorough testing is done and if it is found to work according to the specification. It involves careful planning, investigation of the current system and it constraints

Two major tasks of preparing the implementation are education and training of the users and testing of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required just for implementation.

The implementation phase comprises of several activities. The required hardware and software acquisition is carried out. The system may require some software to be developed. For this, programs are written and tested. The user then changes over to his new fully tested system and the old system is discontinued.

TESTING

The testing phase is an important part of software development. It is the Information zed system will help in automate process of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied. Software testing is carried out in three steps:

1. The first includes unit testing, where in each module is tested to provide its correctness, validity and also determine any missing operations and to verify whether the objectives have been met. Errors are noted down and corrected immediately.

2. Unit testing is the important and major part of the project. So errors are rectified easily in particular module and program clarity is increased. In this project entire system is dividedinto several modules and is developed individually. So unit testing is conducted to individual modules.

3. The second step includes Integration testing. It need not be the case, the software whose modules when run individually and showing perfect results, will also show perfect results when run as a whole.

coding and data analysis

CODE:-

```
import numpy as np
import pandas as pd
import os
for dirname, _, filenames in os.walk('/Agriculture Crop/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))


import numpy as np
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly as py
import os

[249]
df=pd.read_csv('./Datafile.csv')
[250]
df.head()
[251]
df.tail()
[252]
#Load the data using pandas read functions
d1 = pd.read_csv("./datafile1.csv")
d2 = pd.read_csv("./datafile.csv")
d3 = pd.read_csv("./datafile (2).csv")
d4 = pd.read_csv("./datafile (3).csv")
[253]
d1 = pd.read_csv("./datafile1.csv")
d1 = d1.rename(columns={
    'Crop': 'Crop',
    'State': 'State',
    'Cost of Cultivation (`/Hectare) A2+FL': 'Cost_A2_FL',
    'Cost of Cultivation (`/Hectare) C2': 'Cost_C2',
    'Cost of Production (`/Quintal) C2': 'Cost_Production',
    'Yield (Quintal/ Hectare) ': 'Yield'
})

d1.head().style.set_properties(**{'background-
color':'lightblue','color':'black','border-color':'#8b8c8c'})
[254]
#check the shape of the data
print(f' The dataset contains {d1.shape[1]} columns and {d1.shape[0]} r
ows')
```

```
# Print the column names
print(d1.columns)
 The dataset contains 6 columns and 49 rows
Index(['Crop', 'State', 'Cost_A2_FL', 'Cost_C2', 'Cost_Production',
'Yield'], dtype='object')

[255]
d1.info()
d1.describe()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 49 entries, 0 to 48
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Crop             49 non-null     object
 1   State            49 non-null     object
 2   Cost_A2_FL       49 non-null     float64
 3   Cost_C2          49 non-null     float64
 4   Cost_Production  49 non-null     float64
 5   Yield            49 non-null     float64
dtypes: float64(4), object(2)
memory usage: 2.4+ KB

[256]
d1.isnull().sum().sum()
0
[257]
#checking the null values in the data
d1.isna().sum()/len(d1)*100
Crop               0.0
State              0.0
Cost_A2_FL         0.0
Cost_C2            0.0
Cost_Production    0.0
Yield              0.0
dtype: float64
[258]
d1.fillna(0, inplace=True)
[259]
#Lets check the duplicate values in the data
print('The duplicate values in the data is', d1.duplicated().sum())
The duplicate values in the data is 0

[260]
# Find the Average yield of the top 10 yield in the data
# Group by 'Yield' and calculate the mean for specific columns
avg_yield = d1.groupby('Yield')[['Cost_A2_FL', 'Cost_C2', 'Cost_Product
ion']].mean()

# Display the top 10 rows with a heatmap-style background
avg_yield.head(10).style.background_gradient(cmap='Pastel1')
[261]
```

```
def state1(row):
    if 'Andhra Pradesh' in row['Recommended Zone']:
        return 1
def state2(row):
    if 'Tamil Nadu' in row['Recommended Zone']:
        return 1
def state3(row):
    if 'Gujarat' in row['Recommended Zone']:
        return 1
def state4(row):
    if 'Orissa' in row['Recommended Zone']:
        return 1
def state5(row):
    if 'Punjab' in row['Recommended Zone']:
        return 1
def state6(row):
    if 'Haryana' in row['Recommended Zone']:
        return 1
def state7(row):
    if 'Uttar Pradesh' in row['Recommended Zone']:
        return 1
def state8(row):
    if 'Rajasthan' in row['Recommended Zone']:
        return 1
def state9(row):
    if 'Karnataka' in row['Recommended Zone']:
        return 1
def state10(row):
    if 'Madhya Pradesh' in row['Recommended Zone']:
        return 1
def state11(row):
    if 'West Bengal' in row['Recommended Zone']:
        return 1
[262]
plt.figure(figsize=(12,6))
k=px.sunburst(d1,path=['State','Crop'],values='Yield',
              hover_data=['Yield'], color_continuous_scale='Blues')  #
Specify the color scale here)
k.update_layout(title='Best Yield Capacity Crop')
k.show()

 [263]
cols = d1.columns
d1.groupby('Crop')[cols[:-1]].sum().plot(kind='bar', figsize=(12,6)),
plt.title('Cost of Cultivation vs. Cost of Production by Crop')
Text(0.5, 1.0, 'Cost of Cultivation vs. Cost of Production by Crop')

[264]
import plotly.graph_objects as go
import pandas as pd
```

```
# Extract unique crop names from the dataset
crops = d2['Crop'].unique()

# Limit the number of crops to display (e.g., the first six crops)
crops_to_display = crops[:6]

# Create traces for each crop to display
traces = []
for crop in crops_to_display:
    trace = go.Scatter(
        x=d2.columns[1:],
        y=d2.loc[d2['Crop'] == crop, d2.columns[1:]].values.flatten(),
        mode='lines+markers',
        name=crop,
    )
    traces.append(trace)

# Create the layout
layout = go.Layout(
    title='Crop Growth Over Time',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Percentage'),
)

# Create the figure
fig = go.Figure(data=traces, layout=layout)

# Show the figure
fig.show()

[265]
import matplotlib.pyplot as plt
x=[20,6,8,10]
y=[1,5,900,11]
# plt.plot(x,y, marker='*', markersize=10, linestyle='dashed', linewidt
h=1, color='black')
plt.subplot(2,2,1)
plt.plot(x,y, '^:g')
plt.xlabel('Cost ')
plt.ylabel('Cost V2')
plt.title('agriculture-crop')
plt.legend(['Cost of Production by Crop', 'B'])
plt.grid()

plt.subplot(2,2,2)
plt.plot(x,y, '^-g')
plt.xlabel('Cost ')
plt.ylabel('Cost V2')
plt.title('agriculture-crop')
plt.legend(['Cost of Production by Crop', 'B'])
plt.grid()
```

```
plt.subplot(2,2,3)
plt.bar(x, y, width=0.2, color='r')
plt.xlabel('Cost ')
plt.ylabel('Cost V2')
plt.title('agriculture-crop')
plt.legend(['Cost of Production by Crop', 'B'])
plt.grid()


plt.subplot(2,2,4)


plt.show()


[266]
x = [2,2,3,50,3,6]
print(x)
plt.hist(x, color='r')

plt.xlabel('Cost')
plt.ylabel('Cost v2 ')
plt.title('agriculture crop')
plt.legend(['Cost of Production by Crop', 'B'])
plt.grid()
plt.show()
[2, 2, 3, 50, 3, 6]


[267]
plt.pie(d1["State"].value_counts(),autopct="%.f%%",labels=d1["State"].u
nique())
plt.show()

[268]
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
[269]
data = pd.read_csv("./datafile1.csv")
[270]
data.head()
[271]
crop_production_data = pd.read_csv("./datafile (2).csv")
[272]
crop_production_data.head()
[273]
fig,axs = plt.subplots(figsize=(10,6))
crop_wise_yield = data.groupby(['Crop']).sum()['Yield (Quintal/ Hectare
) ']
```

```
plt.plot(crop_wise_yield)
crop_wise_production = data.groupby(['Crop']).sum()['Cost of Production
 (`/Quintal) C2']/10
plt.plot(crop_wise_production)
plt.xticks(rotation ='vertical')
plt.legend()
No artists with labels found to put in legend.  Note that artists whose
label start with an underscore are ignored when legend() is called with
no argument.

<matplotlib.legend.Legend at 0x22123e10920>

[274]
state_crop_yield = data.groupby(['State'])
index = list(state_crop_yield.indices.keys())
state_crop_yield.sum()[['Cost of Production (`/Quintal) C2', 'Yield (Qu
intal/ Hectare) ']].plot(kind='bar',figsize=(12,7))
<Axes: xlabel='State'>

[275]
recommended_zone = pd.read_csv('./datafile (3).csv')
[276]
recommended_zone.drop('Unnamed: 4',axis=1,inplace=True)
recommended_zone.dropna(inplace=True)
[277]
recommended_zone.info()
<class 'pandas.core.frame.DataFrame'>
Index: 50 entries, 0 to 75
Data columns (total 4 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Crop                     50 non-null     object
 1   Variety                  50 non-null     object
 2   Season/ duration in days  50 non-null     object
 3   Recommended Zone         50 non-null     object
dtypes: object(4)
memory usage: 2.0+ KB

[278]
recommended_zone.head()
[279]
def state1(row):
    if 'Andhra Pradesh' in row['Recommended Zone']:
        return 1
def state2(row):
    if 'Tamil Nadu' in row['Recommended Zone']:
        return 1
def state3(row):
    if 'Gujarat' in row['Recommended Zone']:
        return 1
def state4(row):
    if 'Orissa' in row['Recommended Zone']:
```

```
        return 1
def state5(row):
    if 'Punjab' in row['Recommended Zone']:
        return 1
def state6(row):
    if 'Haryana' in row['Recommended Zone']:
        return 1
def state7(row):
    if 'Uttar Pradesh' in row['Recommended Zone']:
        return 1
def state8(row):
    if 'Rajasthan' in row['Recommended Zone']:
        return 1
def state9(row):
    if 'Karnataka' in row['Recommended Zone']:
        return 1
def state10(row):
    if 'Madhya Pradesh' in row['Recommended Zone']:
        return 1
def state11(row):
    if 'West Bengal' in row['Recommended Zone']:
        return 1
[280]
recommended_zone['Andhra Pradesh'] = recommended_zone.apply(state1,axis
=1)
recommended_zone['Tamil Nadu']=recommended_zone.apply(state2,axis=1)
recommended_zone['Gujarat']=recommended_zone.apply(state3,axis=1)
recommended_zone['Orissa']=recommended_zone.apply(state4,axis=1)
recommended_zone['Punjab']=recommended_zone.apply(state5,axis=1)
recommended_zone['Haryana']=recommended_zone.apply(state6,axis=1)
recommended_zone['Uttar Pradesh']=recommended_zone.apply(state7,axis=1)
recommended_zone['Rajasthan']=recommended_zone.apply(state8,axis=1)
recommended_zone['Karnataka']=recommended_zone.apply(state9,axis=1)
recommended_zone['Madhya Pradesh']=recommended_zone.apply(state10,axis=
1)
recommended_zone['West Bangal']=recommended_zone.apply(state11,axis=1)
[281]
recommended_zone.fillna(0).head()
[282]
dataframe = recommended_zone.groupby('Crop').sum().plot(kind='bar',figs
ize=(15,7))
dataframe
<Axes: xlabel='Crop'>

[283]
crop_production_data.columns = ['Crop', 'Production 2006-
07', 'Production 2007-08',
       'Production 2008-09', 'Production 2009-10', 'Production 2010-
11',
       'Area 2006-07', 'Area 2007-08', 'Area 2008-09', 'Area 2009-10',
       'Area 2010-11', 'Yield 2006-07', 'Yield 2007-08', 'Yield 2008-
```

```
09',
       'Yield 2009-10', 'Yield 2010-11']
[284]
plt.subplots(figsize=(15,6))
plt.scatter(x='Crop',y='Production 2006-
07',data = crop_production_data)
plt.xticks(rotation=90)
plt.show()
```

OUTPUTS:-

[1]:
```
import numpy as np
import pandas as pd
import os
for dirname, _, filenames in os.walk('/Agriculture Crop/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

[2]:
```
import numpy as np
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly as py
import os
```

[3]:
```
df=pd.read_csv('./Datafile.csv')
```

[4]:
```
df.head()
```

[4]:

	Crop	2004-05	2005-06	2006-07	2007-08	2008-09	2009-10	2010-11	2011-12
0	Rice	100.0	101.0	99.0	105.0	112.0	121.0	117.0	110.0
1	Wheat	100.0	101.0	112.0	115.0	117.0	127.0	120.0	108.0
2	Coarse Cereals	100.0	107.0	110.0	115.0	113.0	123.0	122.0	136.0
3	Pulses	100.0	108.0	134.0	124.0	124.0	146.0	137.0	129.0
4	Vegetables	100.0	109.0	103.0	118.0	113.0	124.0	128.0	115.0

[6]:
```
#Load the data using pandas read functions
d1 = pd.read_csv("./datafile1.csv")
d2 = pd.read_csv("./datafile.csv")
d3 = pd.read_csv("./datafile (2).csv")
d4 = pd.read_csv("./datafile (3).csv")
```

[7]:
```
d1 = pd.read_csv("./datafile1.csv")
d1 = d1.rename(columns={
    'Crop': 'Crop',
    'State': 'State',
    'Cost of Cultivation (`/Hectare) A2+FL': 'Cost_A2_FL',
    'Cost of Cultivation (`/Hectare) C2': 'Cost_C2',
    'Cost of Production (`/Quintal) C2': 'Cost_Production',
    'Yield (Quintal/ Hectare) ': 'Yield'
})

d1.head().style.set_properties(**{'background-color':'lightblue','color':'black','border-color':'#8b8c8c'})
```

[7]:

	Crop	State	Cost_A2_FL	Cost_C2	Cost_Production	Yield
0	ARHAR	Uttar Pradesh	9794.050000	23076.740000	1941.550000	9.830000
1	ARHAR	Karnataka	10593.150000	16528.680000	2172.460000	7.470000
2	ARHAR	Gujarat	13468.820000	19551.900000	1898.300000	9.590000
3	ARHAR	Andhra Pradesh	17051.660000	24171.650000	3670.540000	6.420000
4	ARHAR	Maharashtra	17130.550000	25270.260000	2775.800000	8.720000

```
# Find the Average yield of the top 10 yield in the data
# Group by 'Yield' and calculate the mean for specific columns
avg_yield = d1.groupby('Yield')[['Cost_A2_FL', 'Cost_C2', 'Cost_Production']].mean()

# Display the top 10 rows with a heatmap-style background
avg_yield.head(10).style.background_gradient(cmap='Pastel1')
```

Yield	Cost_A2_FL	Cost_C2	Cost_Production
1.320000	6440.640000	7868.640000	5777.480000
3.010000	5483.540000	8266.980000	2614.140000
4.050000	6204.230000	9165.590000	2068.670000
4.710000	13647.100000	17314.200000	3484.010000
5.900000	6684.180000	13209.320000	2228.970000
6.420000	17051.660000	24171.650000	3670.540000
6.700000	10780.760000	15371.450000	2261.240000
6.830000	8552.690000	12610.850000	1691.660000
7.470000	10593.150000	16528.680000	2172.460000
8.050000	12985.950000	18679.330000	2277.680000

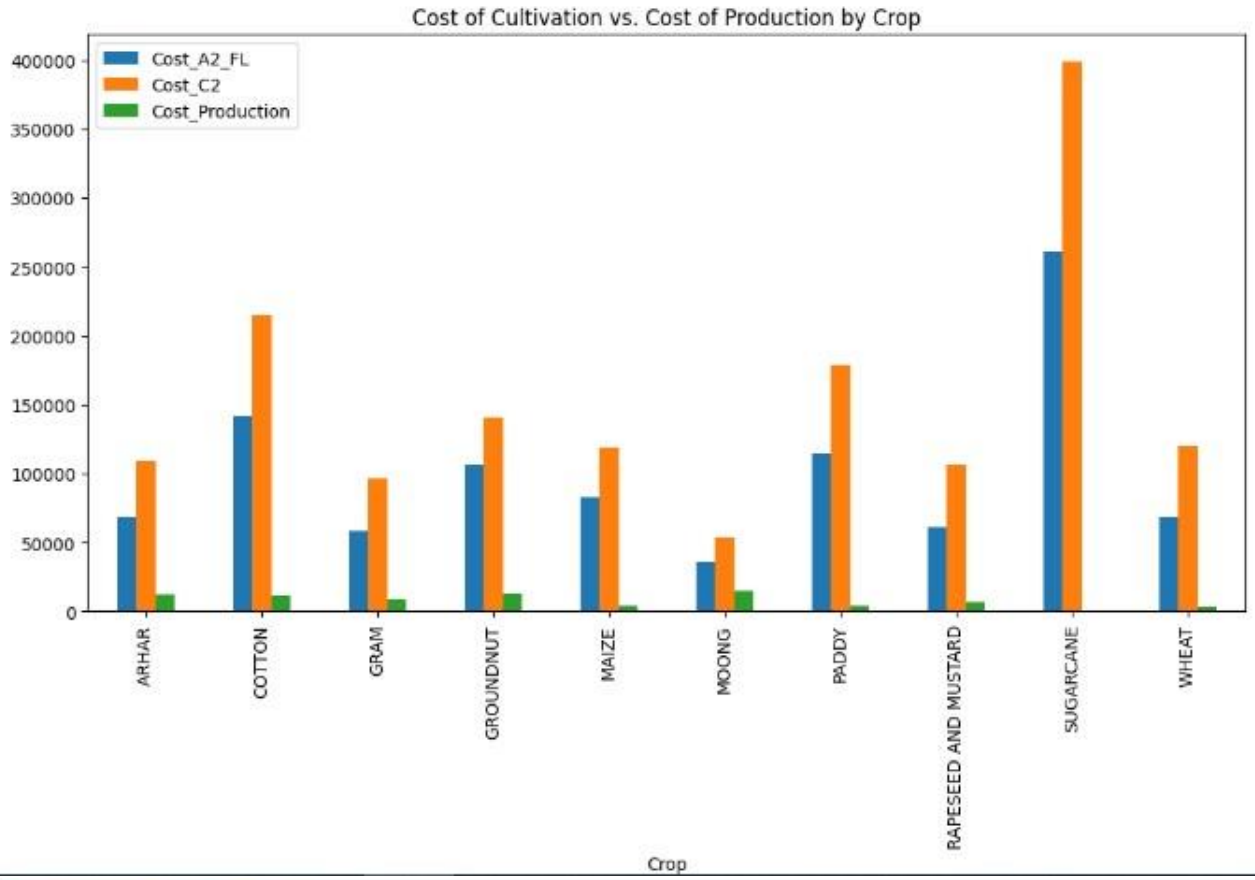
```
plt.figure(figsize=(12,6))
k=px.sunburst(d1,path=['State','Crop'],values='Yield',
              hover_data=['Yield'], color_continuous_scale='Blues')  # Specify the color scale here)
k.update_layout(title='Best Yield Capacity Crop')
k.show()
```

Best Yield Capacity Crop

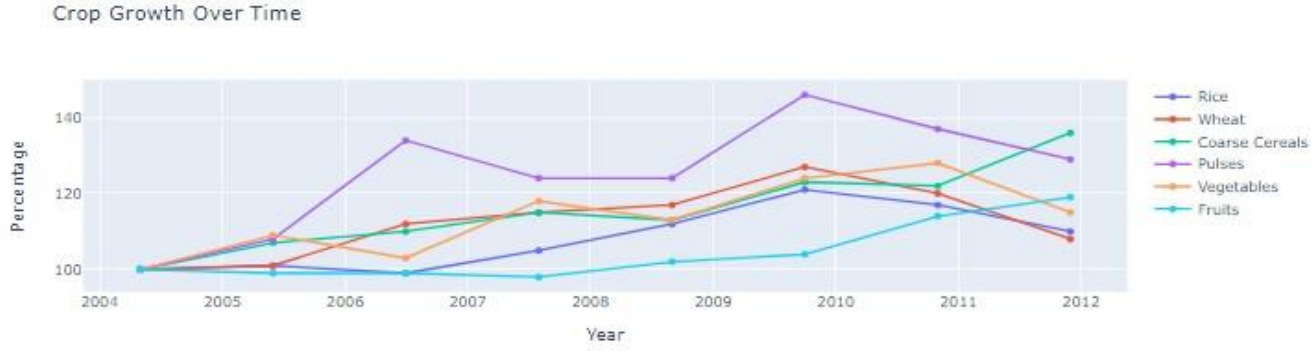

```
<Figure size 1200x600 with 0 Axes>
```

[17]:
```
cols = d1.columns
d1.groupby('Crop')[cols[:-1]].sum().plot(kind='bar', figsize=(12,6)),
plt.title('Cost of Cultivation vs. Cost of Production by Crop')
```

[17]: Text(0.5, 1.0, 'Cost of Cultivation vs. Cost of Production by Crop')



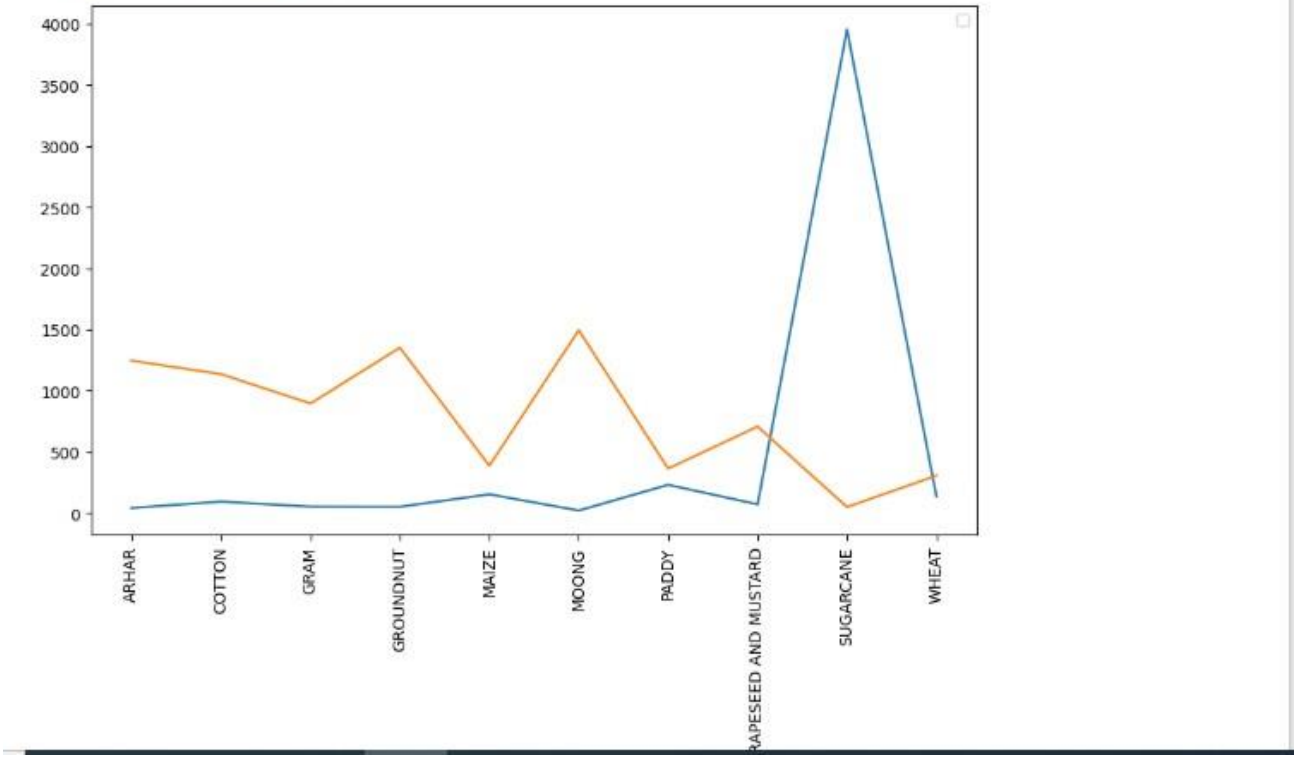
[18]:
```
import plotly.graph_objects as go
import pandas as pd
# Extract unique crop names from the dataset
crops = d2['Crop'].unique()
# Limit the number of crops to display (e.g., the first six crops)
crops_to_display = crops[:6]
# Create traces for each crop to display
traces = []
for crop in crops_to_display:
    trace = go.Scatter(
        x=d2.columns[1:],
        y=d2.loc[d2['Crop'] == crop, d2.columns[1:]].values.flatten(),
        mode='lines+markers',
        name=crop,
    )
    traces.append(trace)
# Create the layout
layout = go.Layout(
    title='Crop Growth Over Time',
    xaxis=dict(title='Year'),
    yaxis=dict(title='Percentage'),
)
# Create the figure
fig = go.Figure(data=traces, layout=layout)
# Show the figure
fig.show()
```



```
fig,axs = plt.subplots(figsize=(10,6))
crop_wise_yield = data.groupby(['Crop']).sum()['Yield (Quintal/ Hectare) ']
plt.plot(crop_wise_yield)
crop_wise_production = data.groupby(['Crop']).sum()['Cost of Production (`/Quintal) C2']/10
plt.plot(crop_wise_production)
plt.xticks(rotation ='vertical')
plt.legend()
```

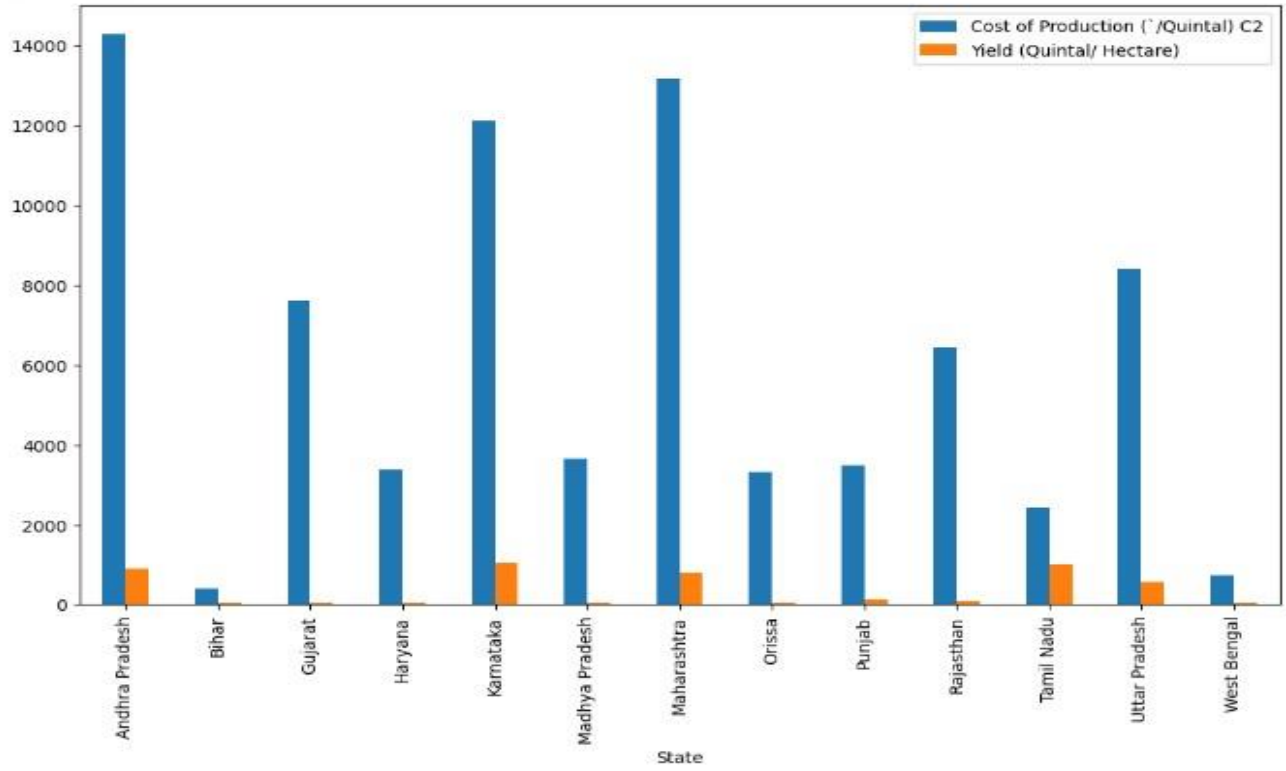
```
No artists with labels found to put in legend.  Note that artists whose label start with an underscore are ignored when legend() is called with no argu
ment.
```

```
<matplotlib.legend.Legend at 0x20b507db980>
```




```
state_crop_yield = data.groupby(['State'])
index = list(state_crop_yield.indices.keys())
state_crop_yield.sum()[['Cost of Production (`/Quintal) C2', 'Yield (Quintal/ Hectare) ']].plot(kind='bar',figsize=(12,7))
```

```
<Axes: xlabel='State'>
```



[29]:
```
recommended_zone = pd.read_csv('./datafile (3).csv')
```

[30]:
```
recommended_zone.drop('Unnamed: 4',axis=1,inplace=True)
recommended_zone.dropna(inplace=True)
```

[31]:
```
recommended_zone.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 50 entries, 0 to 75
Data columns (total 4 columns):
 #   Column                   Non-Null Count  Dtype 
---  ------                   --------------  ----- 
 0   Crop                     50 non-null     object
 1   Variety                  50 non-null     object
 2   Season/ duration in days  50 non-null     object
 3   Recommended Zone         50 non-null     object
dtypes: object(4)
memory usage: 2.0+ KB
```

[32]:
```
recommended_zone.head()
```

[32]:

	Crop	Variety	Season/ duration in days	Recommended Zone
0	Paddy	Chinsurah Rice (IET 19140)	Medium	Andhra Pradesh, Tamil Nadu, Gujarat, Orissa, a...
2	Paddy	IGKVR-1 (IET 19569)	Mid-early	Chhattisgarh, Madhya Pradesh and Orissa under ...
3	Paddy	IGKVR-2 (IET 19795)	Medium	Chhattisgarh, Bihar and Orissa under both irr...
4	Paddy	CR Dhan 401 (REETA)	145-150	Orissa, West Bengal, Tamil Nadu and Andhra Pra...
5	Paddy	CR Dhan 601 (IET 18558)	160	Boro Area of Orissa, West Bengal and Assam.

[34]:
```
recommended_zone['Andhra Pradesh'] = recommended_zone.apply(state1,axis=1)
recommended_zone['Tamil Nadu']=recommended_zone.apply(state2,axis=1)
recommended_zone['Gujarat']=recommended_zone.apply(state3,axis=1)
recommended_zone['Orissa']=recommended_zone.apply(state4,axis=1)
recommended_zone['Punjab']=recommended_zone.apply(state5,axis=1)
recommended_zone['Haryana']=recommended_zone.apply(state6,axis=1)
recommended_zone['Uttar Pradesh']=recommended_zone.apply(state7,axis=1)
recommended_zone['Rajasthan']=recommended_zone.apply(state8,axis=1)
recommended_zone['Karnataka']=recommended_zone.apply(state9,axis=1)
recommended_zone['Madhya Pradesh']=recommended_zone.apply(state10,axis=1)
recommended_zone['West Bangal']=recommended_zone.apply(state11,axis=1)
```

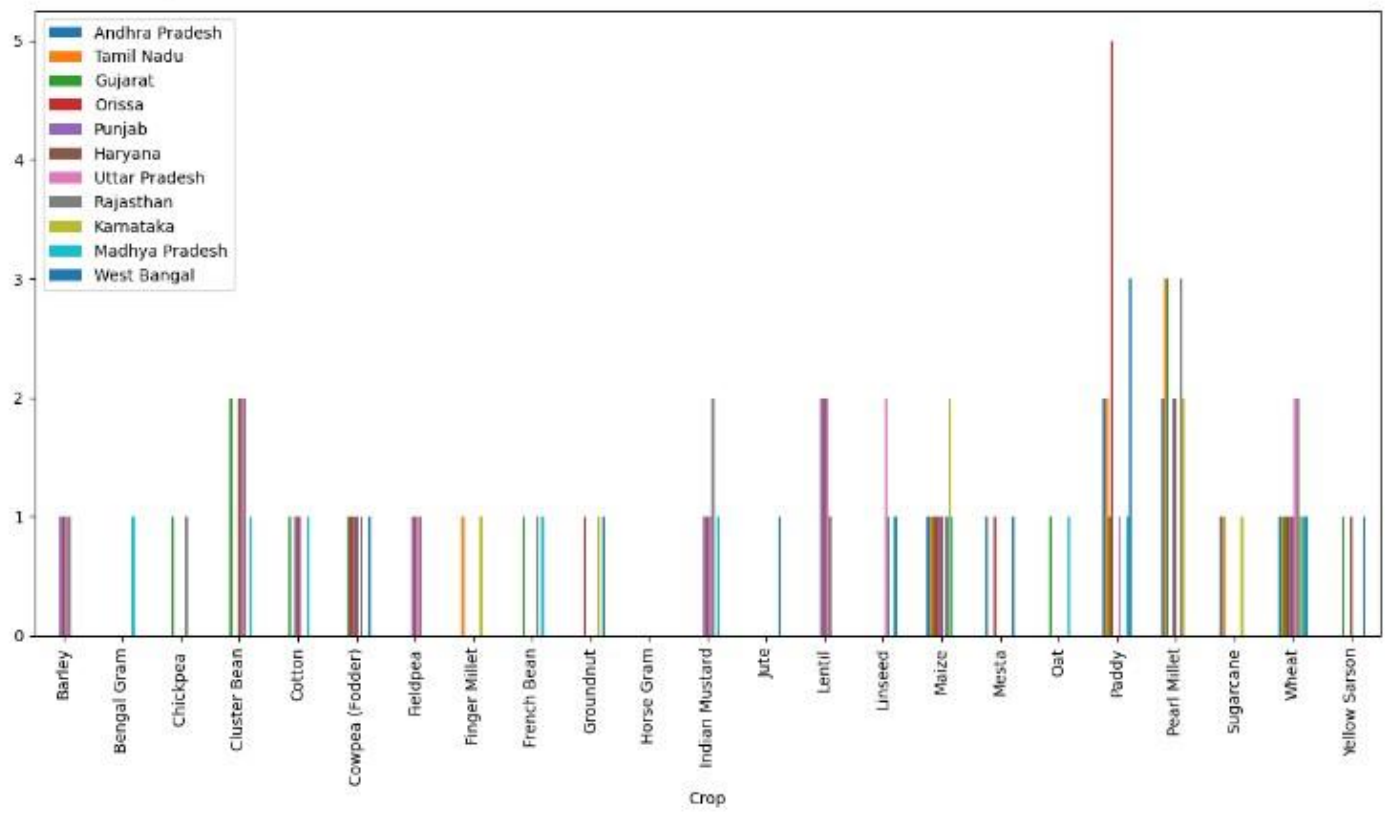
[35]:
```
recommended_zone.fillna(0).head()
```

[35]:

	Crop	Variety	Season/ duration in days	Recommended Zone	Andhra Pradesh	Tamil Nadu	Gujarat	Orissa	Punjab	Haryana	Uttar Pradesh	Rajasthan	Karnataka	Madhya Pradesh	West Bangal
0	Paddy	Chinsurah Rice (IET 19140)	Medium	Andhra Pradesh, Tamil Nadu, Gujarat, Orissa, a...	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
2	Paddy	IGKVR-1 (IET 19569)	Mid-early	Chhattisgarh, Madhya Pradesh and Orissa under ...	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
3	Paddy	IGKVR-2 (IET 19795)	Medium	Chhattisgarh, Bihar and Orissa under both irr...	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	Paddy	CR Dhan 401 (REETA)	145-150	Orissa, West Bengal, Tamil Nadu and Andhra Pra...	1.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
5	Paddy	CR Dhan 601 (IET 18558)	160	Boro Area of Orissa, West Bengal and Assam.	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

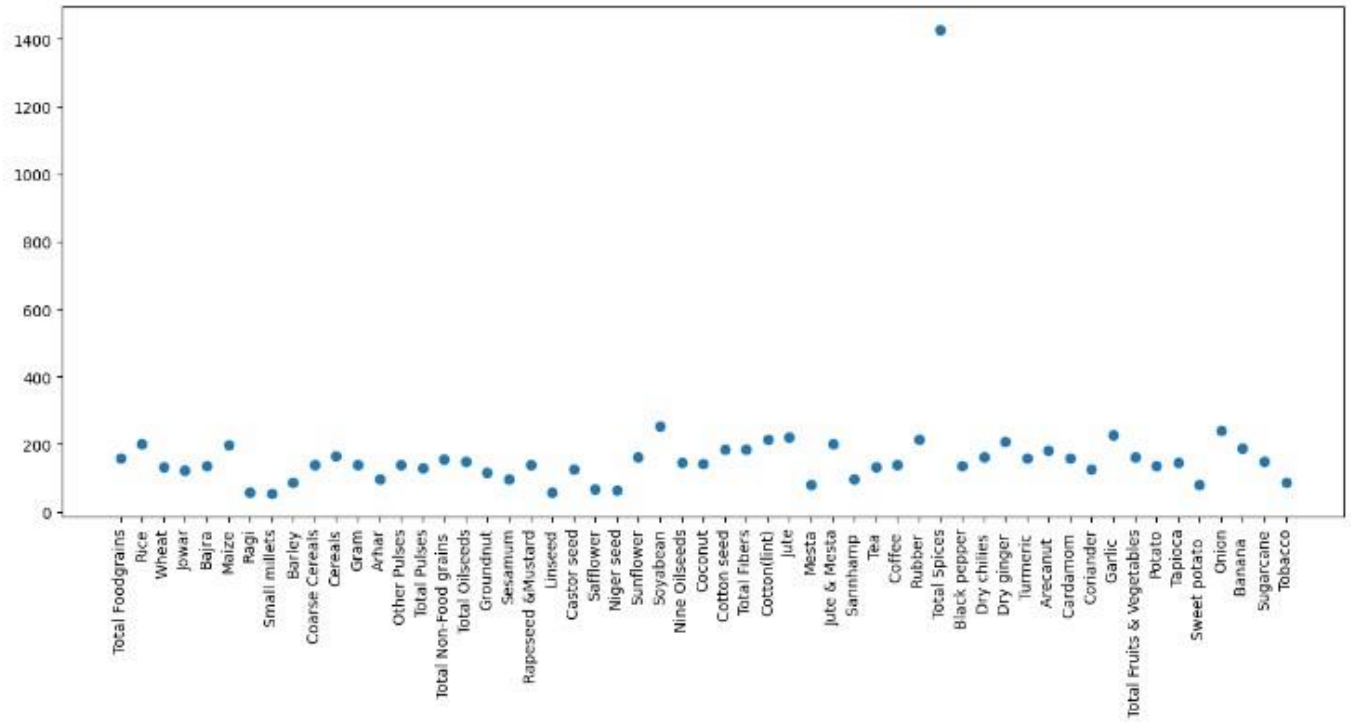
```
dataframe = recommended_zone.groupby('Crop').sum().plot(kind='bar',figsize=(15,7))
dataframe
```

```
<Axes: xlabel='Crop'>
```

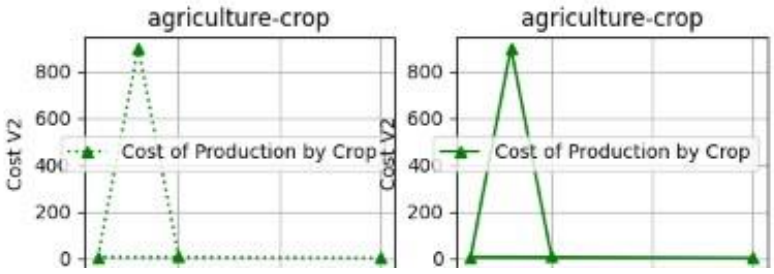



```
crop_production_data.columns = ['Crop', 'Production 2006-07', 'Production 2007-08',
       'Production 2008-09', 'Production 2009-10', 'Production 2010-11',
       'Area 2006-07', 'Area 2007-08', 'Area 2008-09', 'Area 2009-10',
       'Area 2010-11', 'Yield 2006-07', 'Yield 2007-08', 'Yield 2008-09',
       'Yield 2009-10', 'Yield 2010-11']
```

```
plt.subplots(figsize=(15,6))
plt.scatter(x='Crop',y='Production 2006-07',data = crop_production_data)
plt.xticks(rotation=90)
plt.show()
```



[19]:
```
import matplotlib.pyplot as plt
x=[20,6,8,10]
y=[1,5,900,11]
# plt.plot(x,y, marker='*', markersize=10, linestyle='dashed', linewidth=1, color='black')
plt.subplot(2,2,1)
plt.plot(x,y, '^:g')
plt.xlabel('Cost ')
plt.ylabel('Cost V2')
plt.title('agriculture-crop')
plt.legend(['Cost of Production by Crop', 'B'])
plt.grid()

plt.subplot(2,2,2)
plt.plot(x,y, '^-g')
plt.xlabel('Cost ')
plt.ylabel('Cost V2')
plt.title('agriculture-crop')
plt.legend(['Cost of Production by Crop', 'B'])
plt.grid()

plt.subplot(2,2,3)
plt.bar(x, y, width=0.2, color='r')
plt.xlabel('Cost ')
plt.ylabel('Cost V2')
plt.title('agriculture-crop')
plt.legend(['Cost of Production by Crop', 'B'])
plt.grid()


plt.subplot(2,2,4)


plt.show()
```



In [82]:
```
plt.tick_params(labelsize=10)
sns.heatmap(df_copy.corr(),annot=True)
```

Out[82]:
```
<AxesSubplot:>
```

WEBSITE OF BULING

Building a website using HTML is a great way to create a basic structure for your web content. HTML (Hypertext Markup Language) is the standard language for creating web pages. Here's a simple guide to help you get started:

1. **Create a New HTML File:**
   Open a text editor like Notepad (on Windows) or TextEdit (on macOS) and create a new file. Save it with a `.html` extension, for example, `index.html`.
2. **Basic HTML Structure:**
   Start your HTML file with the basic structure, including the HTML, head, and body tags.

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1
    <title>Your Website Title</title>
</head>
<body>

<!-- Your content goes here -->

</body>
</html>
```

3. **Add Content:**
   Within the `<body>` tags, you can add various HTML elements to structure your content. For example:

```html
<h1>Hello, World!</h1>
<p>This is a simple website built with HTML.</p>
```

4. **Headings, Paragraphs, and Lists:**
   Use headings (`<h1>`, `<h2>`, etc.) for titles, paragraphs (`<p>`) for text, and lists (`<ul>`, `<ol>`, `<li>`) for organized information.

```html
<h2>About Us</h2>
<p>Welcome to our website! We are...</p>

<h3>Our Team</h3>
<ul>
    <li>John Doe</li>
    <li>Jane Smith</li>
</ul>
```

Links:

Create links using the `<a>` tag.

```html
<a href="https://www.example.com">Visit our main site</a>
```

6. **Images:**

   Include images with the `<img>` tag.

```html
<img src="image.jpg" alt="Description of the image">
```

7. **HTML Forms:**

   If you want to create a form, use the `<form>` tag and include various form elements like `<input>`, `<textarea>`, and `<button>`.

```html
<form>
    <label for="name">Name:</label>
    <input type="text" id="name" name="name">

    <label for="email">Email:</label>
    <input type="email" id="email" name="email">

    <button type="submit">Submit</button>
</form>
```

8. **Testing Your Website:**

   Save your HTML file and open it in a web browser to see how it looks. Simply double-click the HTML file, and it will open in your default browser.

Remember, this is just a very basic introduction to HTML. To enhance your website further, you may want to learn CSS (Cascading Style Sheets) for styling and layout, and JavaScript for adding interactivity. Additionally, many modern websites use more advanced tools and frameworks like Bootstrap or React to streamline development.

CSS

Building a website using CSS (Cascading Style Sheets) allows you to control the visual presentation and layout of your HTML content. Here's a basic guide on how to use CSS to style a simple HTML page:

Link CSS to HTML:

In your HTML file (between the `<head>` tags), link to your CSS file using the `<link>` tag. Create a new CSS file (e.g., `styles.css`) to store your styles.

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1
    <title>Your Website Title</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>

<!-- Your content goes here -->

</body>
</html>
```

2. **Basic CSS Styling:**

   Open your `styles.css` file and start adding styles. For example, you can change the background color, text color, and font:

```css
body {
    background-color: #f4f4f4;
    color: #333;
    font-family: Arial, sans-serif;
}

h1 {
    color: #007bff;
}

p {
    font-size: 16px;
    line-height: 1.6;
}
```

3. **Selectors and Properties:**

   CSS uses selectors to target HTML elements and properties to define their style. In the example above, `body`, `h1`, and `p` are selectors, and `background-color`, `color`, `font-family`, `font-size`, and `line-height` are properties.

4. **Box Model:**

   Understand the box model, which consists of content, padding, border, and margin. You can control these aspects to adjust the spacing and layout of your elements.

```css
/* Example of adjusting the box model */
div {
    width: 300px;
    padding: 20px;
    border: 1px solid #ccc;
    margin: 10px;
}
```

5. **Classes and IDs:**

   Use classes (`.`) and IDs (`#`) to target specific elements for styling. This allows you to apply styles to specific elements without affecting others.

```html
<h2 class="highlight">Important Heading</h2>
<p id="special-paragraph">This paragraph is special.</p>
```

```css
.highlight {
    color: #ff0000;
}

#special-paragraph {
    font-weight: bold;
}
```

. **Flexbox and Grid (Optional):**

   Learn about CSS Flexbox and Grid for more advanced layout options. They provide powerful tools for creating responsive and complex layouts.

```css
/* Example of using Flexbox */
.container {
    display: flex;
    justify-content: space-between;
}
```

Remember that CSS is a powerful tool for styling, and there is much more to explore. Consider resources like Mozilla Developer Network (MDN) or W3Schools for more in-depth tutorials and references. Additionally, tools like browser developer tools can help you inspect and experiment with styles in real-time

JavaScript (JS)

Building a website using JavaScript (JS) allows you to add interactivity and dynamic behavior to your web pages. Here's a basic guide on how to use JavaScript in conjunction with HTML and CSS:

1. **Link JavaScript to HTML:**
   Similar to CSS, you need to link your JavaScript file to your HTML file. Include the following line inside the `<head>` section of your HTML file:

```html
<script src="script.js" defer></script>
```

   The `defer` attribute ensures that the script is executed after the HTML document has been completely parsed.

2. **Basic JavaScript Code:**
   Create a new JavaScript file (e.g., `script.js`) and start adding basic JavaScript code. For example, you can use `console.log` to print messages to the browser's console.

```javascript
console.log("Hello, world!");
```

3. **DOM Manipulation:**

   The Document Object Model (DOM) is a programming interface for web documents. JavaScript can be used to manipulate the DOM to dynamically update the content of your HTML page.

```javascript
// Change the text content of an element
document.getElementById("myElement").textContent = "New Text";

// Change the style of an element
document.querySelector(".myClass").style.color = "red";
```

4. **Event Handling:**

   JavaScript can respond to user actions (events) such as clicks, keypresses, etc. Use event listeners to execute functions when these events occur.

```javascript
// Add a click event listener to a button
document.getElementById("myButton").addEventListener("click", function
    alert("Button clicked!");
});
```

5. **Asynchronous Operations:**

   JavaScript can handle asynchronous operations, such as fetching data from a server. Use functions like `fetch` to make HTTP requests.

```javascript
// Fetch data from an API
fetch("https://api.example.com/data")
    .then(response => response.json())
    .then(data => console.log(data))
    .catch(error => console.error("Error fetching data:", error));
```

6. **Working with Forms:**

   JavaScript can be used to validate form input and handle form submissions.

```javascript
// Validate a form before submission
document.getElementById("myForm").addEventListener("submit", function(
    var input = document.getElementById("myInput").value;
    if (input === "") {
        alert("Please enter a value!");
        event.preventDefault(); // Prevent form submission
    }
});
```

7. **Modern JavaScript (ES6+):**
   Familiarize yourself with modern JavaScript features introduced in ECMAScript 6 (ES6) and later, such as arrow functions, `let` and `const` declarations, destructuring, and more.

```javascript
// Example of using arrow functions
const add = (a, b) => a + b;
```

Remember, JavaScript is a versatile language with a wide range of capabilities. As you become more comfortable with the basics, you can explore frameworks like React, Angular, or Vue.js for building more sophisticated and dynamic web applications. Additionally, always consider best practices, such as organizing your code into functions and modules, to keep it maintainable.

CROP ANALYZER WEBSITE CODE

CODE OF HTML,CSS,JS

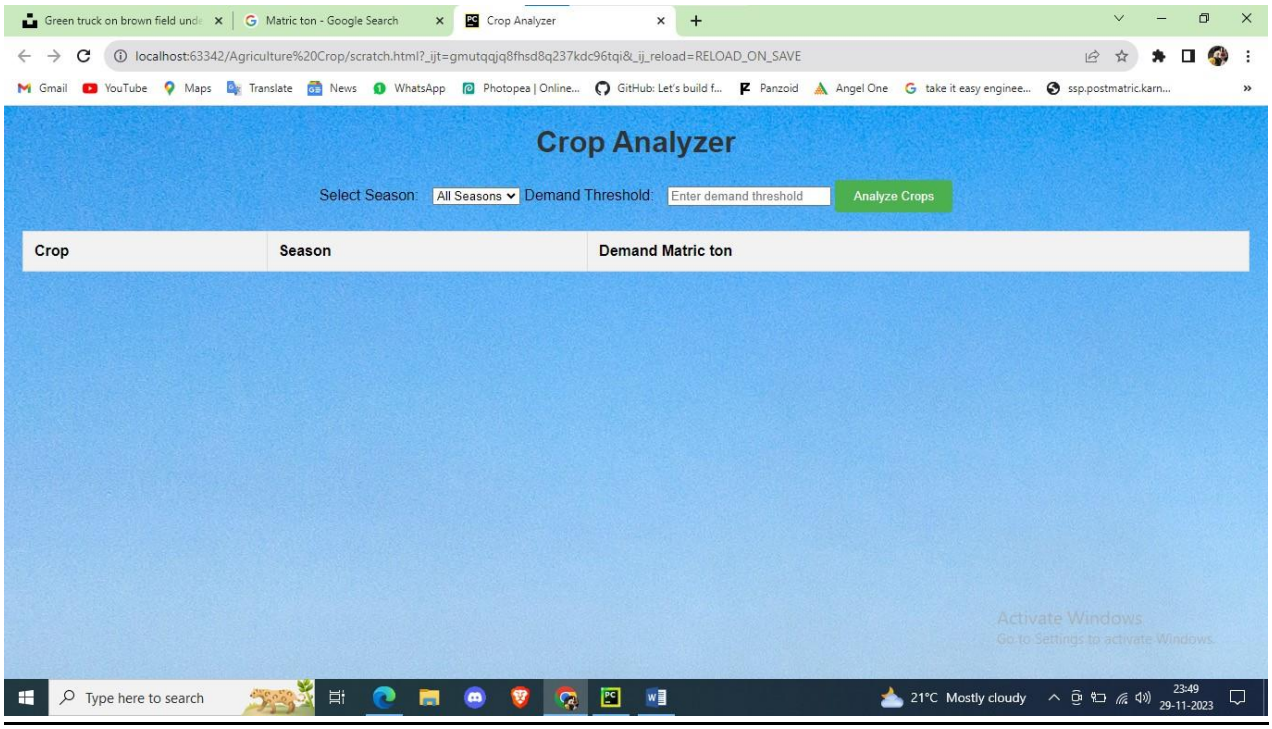
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Crop Analyzer</title>

    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 20px;
        }

        h1 {
            text-align: center;
            color: #333;
        }

        table {
            width: 100%;
            border-collapse: collapse;
            margin-top: 20px;
        }

        th, td {
            border: 1px solid #ddd;
            padding: 12px;
            text-align: left;
        }

        th {
            background-color: #f2f2f2;
```

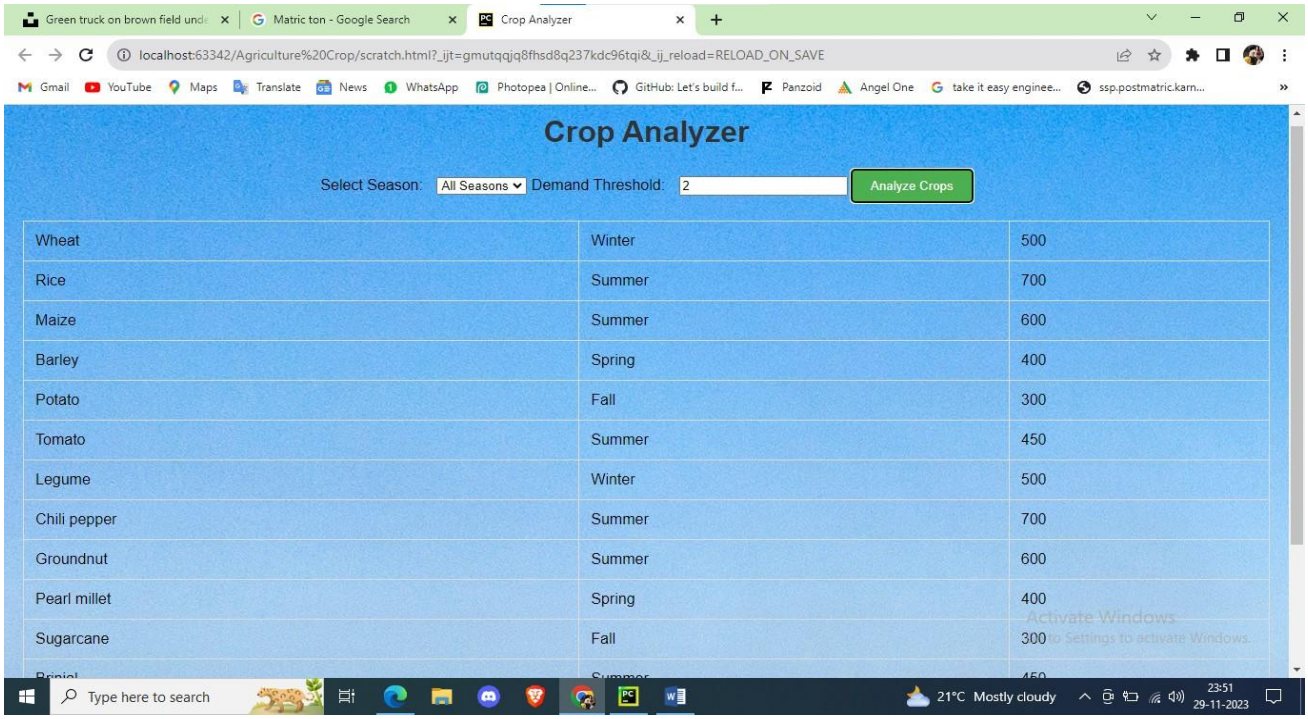
```
        }

        form {
            margin-top: 20px;
            text-align: center;
        }

        label {
            margin-right: 10px;
        }

        button {
            padding: 10px 20px;
            background-color: #4CAF50;
            color: white;
            border: none;
            border-radius: 4px;
            cursor: pointer;
        }

        button:hover {
            background-color: #45a049;
        }
    </style>
</head>
<body>
<body background="land crop2.jpg"></body>
    <h1>Crop Analyzer</h1>

    <form id="cropForm">
        <label for="season">Select Season:</label>
        <select id="season" name="season">
            <option value="">All Seasons</option>
            <option value="Winter">Winter</option>
            <option value="Spring">Spring</option>
            <option value="Summer">Summer</option>
            <option value="Fall">Fall</option>
        </select>

        <label for="demand">Demand Threshold:</label>
        <input type="number" id="demand" name="demand"
placeholder="Enter demand threshold">

        <button type="button" onclick="analyzeCrops()">Analyze
Crops</button>
    </form>

    <table id="cropTable">
        <tr>
            <th>Crop</th>
            <th>Season</th>
            <th>Demand Matric ton</th>
        </tr>
```

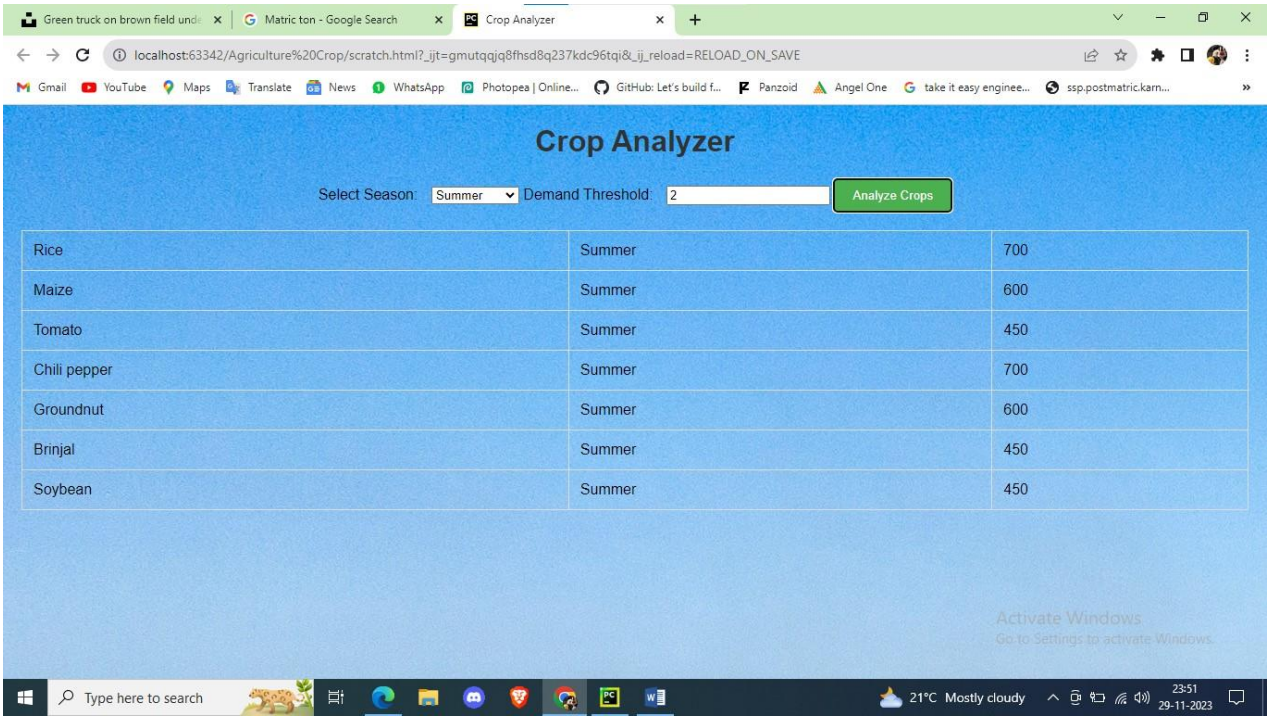
```
        <!-- Table rows will be populated dynamically using
JavaScript -->
    </table>

    <script>
        function analyzeCrops() {
            var season = document.getElementById('season').value;
            var demand = document.getElementById('demand').value;

            // Perform AJAX request or use mock data to fetch and
display results
            // For simplicity, using mock data here
            var mockData = [
                { Crop: 'Wheat', Season: 'Winter', Demand: 500 },
                { Crop: 'Rice', Season: 'Summer', Demand: 700 },
                { Crop: 'Maize', Season: 'Summer', Demand: 600 },
                { Crop: 'Barley', Season: 'Spring', Demand: 400
},
                { Crop: 'Potato', Season: 'Fall', Demand: 300 },
                { Crop: 'Tomato', Season: 'Summer', Demand: 450
},
                { Crop: 'Legume', Season: 'Winter', Demand: 500
},
                { Crop: 'Chili pepper', Season: 'Summer', Demand:
700 },
                { Crop: 'Groundnut', Season: 'Summer', Demand:
600 },
                { Crop: 'Pearl millet', Season: 'Spring', Demand:
400 },
                { Crop: 'Sugarcane', Season: 'Fall', Demand: 300
},
                { Crop: 'Brinjal', Season: 'Summer', Demand: 450
},
                { Crop: 'Millets', Season: 'Spring', Demand: 400
},
                { Crop: 'Mung bean', Season: 'Fall', Demand: 300
},
                { Crop: 'Soybean', Season: 'Summer', Demand: 450
}
            ];

            // Filter data based on user input
            var filteredData = mockData.filter(function (crop) {
                return (!season || crop.Season === season) &&
(!demand || crop.Demand >= demand);
            });

            // Populate the table with the filtered data
            var tableBody =
document.getElementById('cropTable').getElementsByTagName('tbody'
)[0];
            tableBody.innerHTML = ''; // Clear existing rows
```

```
            filteredData.forEach(function (crop) {
                var row = tableBody.insertRow();
                var cell1 = row.insertCell(0);
                var cell2 = row.insertCell(1);
                var cell3 = row.insertCell(2);

                cell1.innerHTML = crop.Crop;
                cell2.innerHTML = crop.Season;
                cell3.innerHTML = crop.Demand;
            });
        }
    </script>

</body>
</html>
```

WEBSITE OUTPUT




Green truck on brown field und
Matric ton - Google Search
Crop Analyzer
localhost:63342/Agriculture%20Crop/scratch.html?_ijt=gmutqqjq8fhsd8q237kdc96tqi&_ij_reload=RELOAD_ON_SAVE
Gmail YouTube Maps Translate News WhatsApp Photopea | Online... GitHub: Let's build f... Panzoid Angel One take it easy enginee... ssp.postmatric.karn...
Crop Analyzer
Select Season: All Seasons Demand Threshold: 2 Analyze Crops
Wheat Winter 500
Rice Summer 700
Maize Summer 600
Barley Spring 400
Potato Fall 300
Tomato Summer 450
Legume Winter 500
Chili pepper Summer 700
Groundnut Summer 600
Pearl millet Spring 400
Sugarcane Fall 300
Activate Windows
Go to Settings to activate Windows.
Type here to search
21°C Mostly cloudy
23:51
29-11-2023


Green truck on brown field und
Matric ton - Google Search
Crop Analyzer
localhost:63342/Agriculture%20Crop/scratch.html?_ijt=gmutqqjq8fhsd8q237kdc96tqi&_ij_reload=RELOAD_ON_SAVE
Gmail YouTube Maps Translate News WhatsApp Photopea | Online... GitHub: Let's build f... Panzoid Angel One take it easy enginee... ssp.postmatric.karn...
Crop Analyzer
Select Season: Summer Demand Threshold: 2 Analyze Crops
Rice Summer 700
Maize Summer 600
Tomato Summer 450
Chili pepper Summer 700
Groundnut Summer 600
Brinjal Summer 450
Soybean Summer 450
Activate Windows
Go to Settings to activate Windows.
Type here to search
21°C Mostly cloudy
23:51
29-11-2023

CONCLUSION

The proposed system is about the collection of massive dataset collection, the problems in collecting the dataset and also includes collecting the dataset both in the internet as well as in the traditional way.

The massive development in computer technology brings need of most of the data so they produced the system of collecting more number of dataset.

By considering this dataset, a Supply-Demand Prediction forecasting model has been developed in this work that guides the farmers in selecting the appropriate crops to grow.

This in turn suffices the actual demand of the society, minimizing the loss for both farmers as well as consumers at peak times.

This model helps in achieving equilibrium in demand and supply of TOP crops that effectively solve the current crisis.

REFERENCES

1. Department of Agriculture welfare https://agricoop.nic.in/en/all-india-cropsituation

2. Top ten India Crops https://www.mapsofindia.com/top-ten/india-crops/

3. Indian Crops https://www.fao.org/india/fao-in-india/india-at-a-glance/en/

4. Production Of Important Crops in Three Largest Producing State https://www.indiabudget.gov.in/economicsurvey/doc/stat/

5. Production Of Important Crops in Three Largest Producing States https://www.safalta.com/blog/major-crops-and-their-states-in-india

6. Major Cropping Seasons https://byjus.com/free-ias-prep/major-croppingseasons-in-india/

7. Major Crops in India https://unacademy.com/content/upsc/studymaterial/indian-geography/major-crops-in-india/

8. Top 10 Agriculture States https://www.tractorjunction.com/blog/top-10agriculture-states-in-india/

9. Agricultural market (2021) http://agmarknet.gov.in/PriceTrends/Default.aspx. Accessed