Virtual Agricultural IoT Platform

Report By:

- *Kaamna Kandpal*
- *Saikarthik Reddy*
- *Shubham Kumar*

24 May, 2024

Introduction:

Agriculture is a critical sector that sustains the global population, but it faces numerous challenges including climate variability, water scarcity, and the need for increased productivity. To address these issues, integrating modern technology into agricultural practices has become essential. This project report presents the development and implementation of a Virtual Agricultural IoT Platform, designed to simulate and optimize farming processes using Internet of Things (IoT) technologies and Machine Learning (ML) algorithms.

Project Overview:

The Virtual Agricultural IoT Platform is an innovative system that leverages simulated sensors to monitor key agricultural parameters such as soil moisture, temperature, and crop growth. The primary objective of this platform is to provide farmers and agronomists with actionable insights to improve crop management and enhance yield predictability. By using a combination of IoT and ML, the platform aims to offer a comprehensive solution for precision agriculture.

Key Components

1. Simulated Sensors:
   - Soil Moisture Sensors: Monitor the moisture levels in the soil to inform irrigation needs.
   - Temperature Sensors: Track ambient and soil temperature to ensure optimal growing conditions.
   - Growth Monitoring Sensors: Measure and record crop growth stages and health indicators.

2. Data Collection and Integration:
   - The platform integrates data from various simulated sensors to provide a holistic view of the agricultural environment.
   - Real-time data acquisition and processing enable timely decision-making.

3. Machine Learning Algorithms:
   - Predictive Models: Utilize historical and real-time data to predict optimal planting times, irrigation schedules, and crop yields.
   - Environmental Analysis: Assess the impact of different environmental conditions on crop performance and suggest adaptive strategies.

Objectives

Develop a Virtual Agricultural IoT Platform incorporating simulated sensors for soil moisture, temperature, and crop growth. Utilize machine learning (ML) algorithms to predict optimal planting times, irrigation schedules, and crop yields under varying environmental conditions.

- Optimized Planting Schedules: Use predictive analytics to determine the best planting times based on weather forecasts and soil conditions.
- Efficient Irrigation Management: Develop irrigation schedules that conserve water while ensuring crops receive adequate hydration.
- Enhanced Yield Prediction: Provide accurate yield forecasts to assist in planning and resource allocation.

Creating a virtual agricultural IoT platform with simulated sensors involves several key components. Here's an outline of the steps and technologies required to build such a system:

1. System Architecture
Components:

- Simulated Sensors: Virtual representations of soil moisture, temperature, and crop growth sensors.

- IoT Hub: Centralized platform to collect data from sensors.
- Data Storage: Database to store sensor data.

- Machine Learning Engine: To process data and provide predictions.

- User Interface: Dashboard for farmers to view data and insights.

2. Simulated Sensors
Types of Simulated Sensors:

- Soil Moisture Sensors: Simulate varying moisture levels based on weather data.

- Temperature Sensors: Simulate ambient temperature changes throughout the day.

- Crop Growth Sensors: Simulate crop development stages using growth models.

Tools:

- Simulation Frameworks: Use tools like SimPy (Python) for simulating sensor data.

- Random Data Generation: Create realistic data variations using random functions within expected ranges.

3. IoT Hub Components:
- Data Collection: Collect data from simulated sensors via MQTT or HTTP protocols.

- Edge Processing: Perform initial data processing at the edge before sending to the cloud.

4. Data Storage Components:
- Database: Store sensor data and ML predictions.

- Time-Series Database: For efficient storage and retrieval of time-stamped data.

5. Machine Learning Engine
Predictive Models:
- Optimal Planting Times: Use historical weather data and crop yield records to predict the best planting times.

- Irrigation Schedules: Model soil moisture levels and forecast irrigation needs based on weather predictions.

- Crop Yield Predictions: Use growth stage data and environmental factors to predict yields.

6. User Interface
Components:
- Dashboard: Visualize sensor data, ML predictions, and alerts.
- Notifications: Alert users about critical conditions (e.g., low moisture levels).

Tools:
- Web Frameworks: Use frameworks like React.js or Angular for frontend development.
- Chart Libraries: D3.js or Chart.js for visualizing data.

Conclusion

Building a virtual agricultural IoT platform requires integrating several technologies to simulate real-world agricultural conditions. By using simulated sensors and advanced ML algorithms, the platform can provide valuable insights to optimize farming practices, ultimately leading to increased efficiency and yields.

Implementation Steps

1. Platform Architecture Design:
   - Define system architecture including cloud infrastructure, data flow, and user interface components.

2. Virtual Sensor Development:
   - Develop software models to simulate sensor data for soil moisture, temperature, and crop growth.

   - Ensure sensors can replicate realistic agricultural conditions and variations.

3. Machine Learning Model Development:
   - Collect and preprocess historical agricultural data for training ML models.
   - Develop and test ML algorithms for prediction tasks.
   - Validate models using simulated datasets and refine for accuracy.

4. User Interface Design:
   - Create wireframes and prototypes for the dashboard.
   - Develop the front-end application using frameworks like React or Angular.
   - Implement backend services to handle data retrieval and processing.

5. Testing and Validation:
   - Conduct extensive testing of the platform with various virtual scenarios.

   - Validate predictions and recommendations against known agricultural practices and outcomes.

6. Deployment and Monitoring:
   - Deploy the platform on a cloud service like AWS, Azure, or Google Cloud.
   - Implement monitoring tools to track system performance and sensor data accuracy.

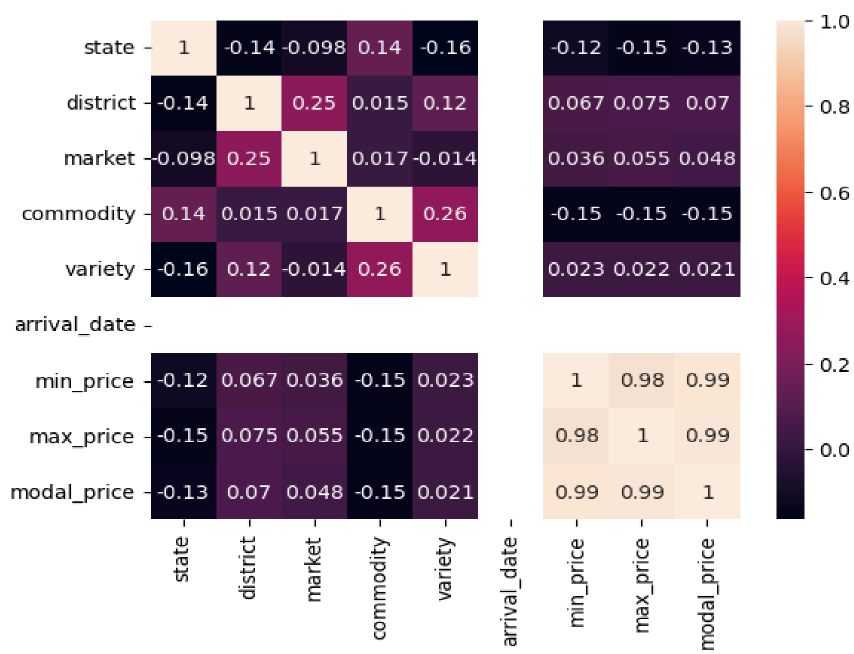
7. User Training and Support:
   - Provide training materials and sessions for users.
   - Establish a support system for troubleshooting and assistance.

Market Segmentation

Here we have consumer dataset in csv file where features are `state district market commodity variety arrival_date min_price max_price modal_price.`

`Data is then normalized used StandardScaler().`

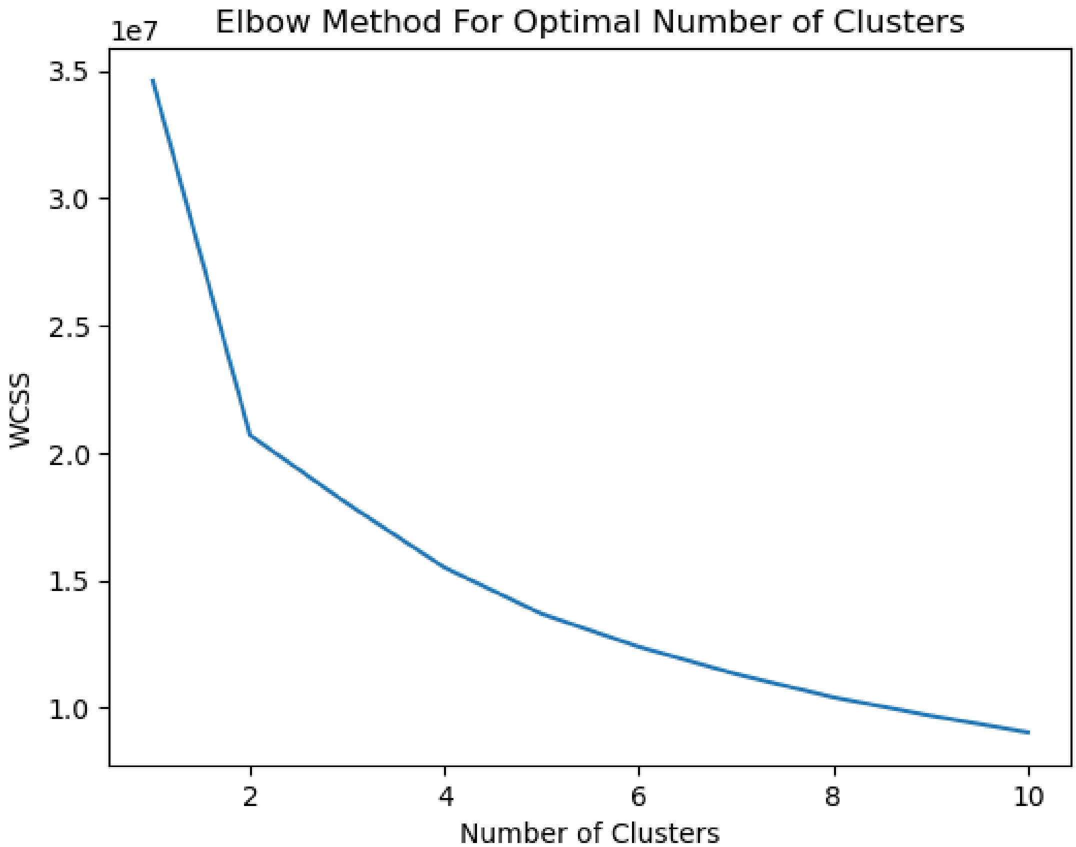
This is the heatmap, the features in the dataset are not highly correlated so there is no need to remove any feature.



Now, WCSS is used to find sum of the squares distance between points in a cluster and the cluster centroid.

```
wcss = []  # Within-cluster sum of squares
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, random_state=42)
    kmeans.fit(df)
    wcss.append(kmeans.inertia_)
plt.plot(range(1, 11), wcss)
plt.xlabel('Number of Clusters')
plt.ylabel('WCSS')
plt.title('Elbow Method For Optimal Number of Clusters')
plt.show()
```

Then, optimum number of centroid is found with Elbow method to be used in KMeans.

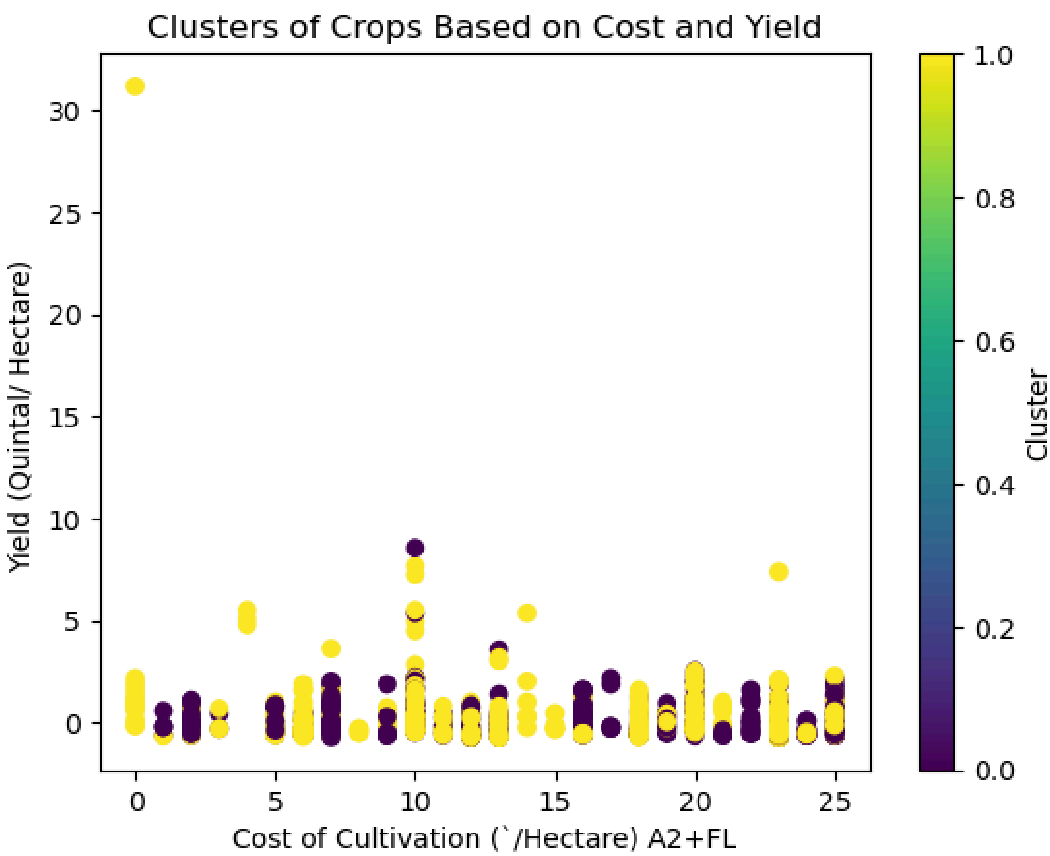


Using KMeans clustering and Silhoute Score, where optimium number of cluster is 2.

```
silhoute_scores=[]
for i in range(2, 11):
    kmeans = KMeans(n_clusters = i, random_state = 42)
    kmeans.fit(df)
    silhoute_scores.append(silhouette_score(df,kmeans.labels_))
silhoute_scores
```

```
kmeans=KMeans(n_clusters=2,random_state=2)
kmeans.fit(df)
df_org['Labels']=kmeans.labels_
df_org
```

The code performs KMeans clustering on a dataset for cluster counts from 2 to 10, storing silhouette scores to evaluate clustering quality. It then fits a final KMeans model with 2 clusters to the data and adds the resulting cluster labels to the original data frame. The updated data frame with cluster labels is printed at the end.



This clustering analysis helps in market segmentation for farmers by grouping crops with similar cost and yield characteristics. It allows farmers to identify which crops are likely to have similar economic and production profiles, aiding in decision-making regarding crop selection, resource allocation, and market strategy. Farmers can focus on crops in the same cluster to optimize their farming practices and marketing approaches for better profitability and efficiency.

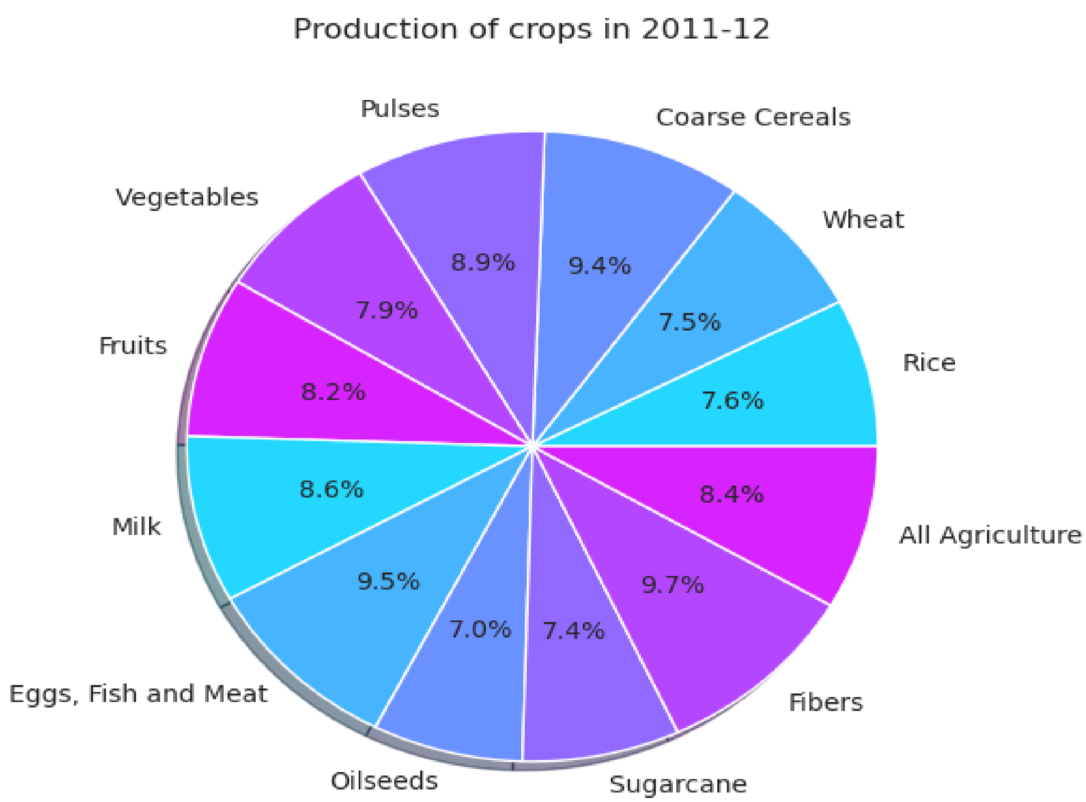
Farmers Market Segmentation

We have crop production data of India to perform basic EDA on this dataset and for visualization of data.

Q1. What was the share of crops in total prodcution in 2011-12?

```
plt.figure(figsize=(12,6))
sns.set_style('white')
color=sns.color_palette('cool')
plt.pie(df4['2011-12'],
        labels=df4['Crop'],
        autopct='%0.1f%%',
        shadow= True,
        colors=color)
plt.title('Production of crops in 2011-12')
plt.show()
```

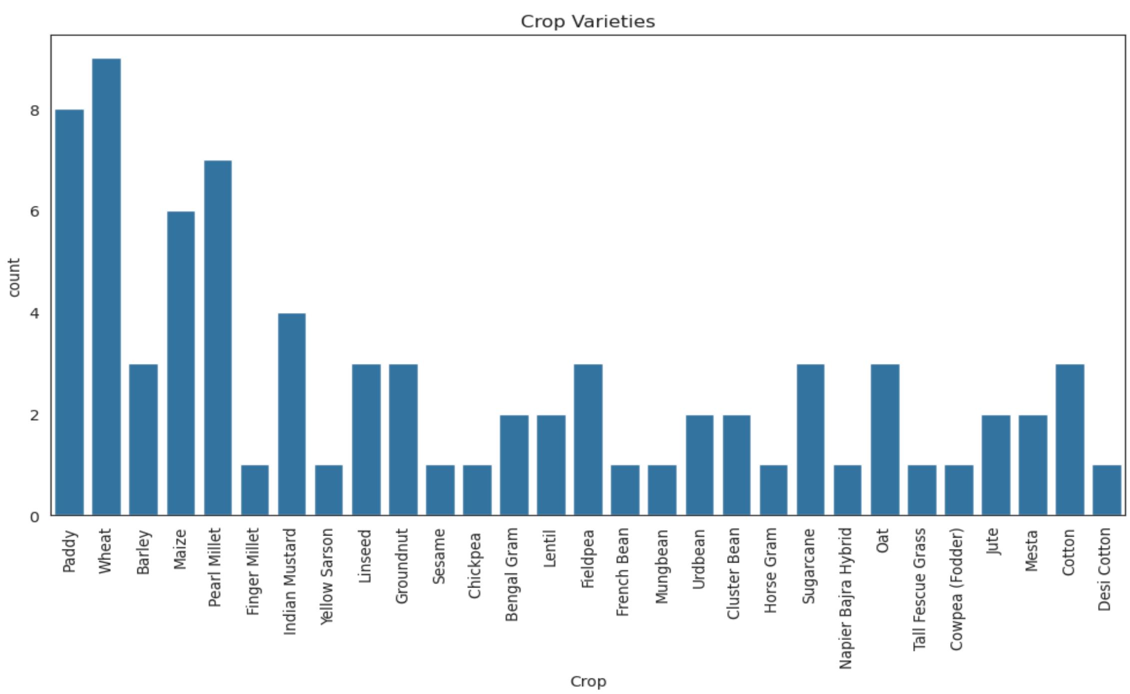
We have plotted a figure to show a pie chart showing us share of different crops in production in the year 2011-12.



Above pie chart displays the distribution of crop production in 2011-12 across various categories like pulses, cereals, wheat, rice, and others, each segment showing the percentage of total production.

Coarse cereals (9.4%) and fibers (9.7%) are the largest categories, while sugarcane (7.0%) and oilseeds (7.4%) are the smallest, indicating a diverse agricultural production landscape with a focus on cereals and fibers.

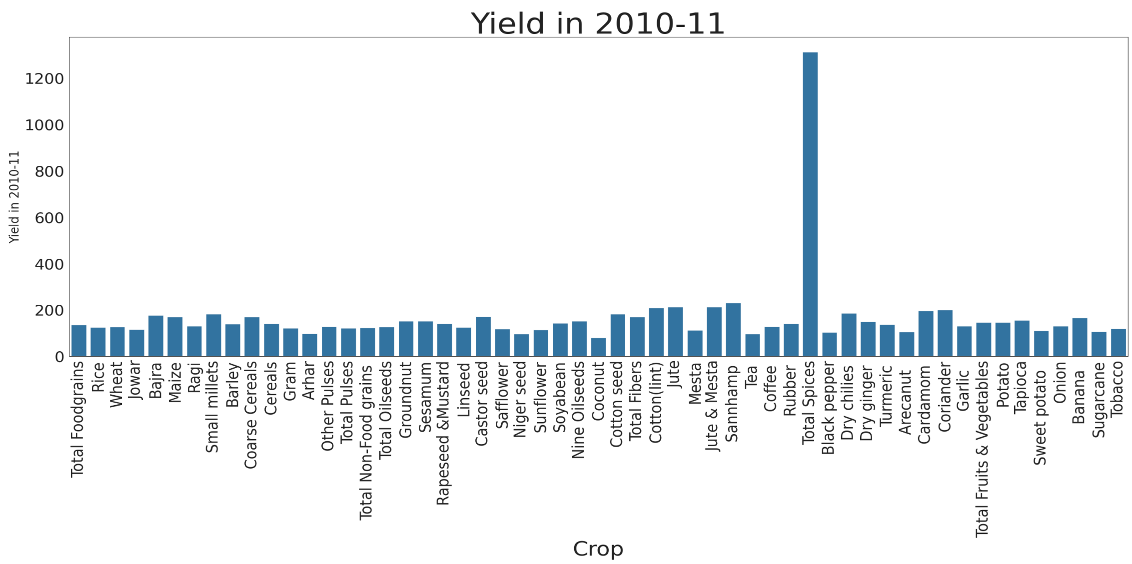
Q2. What crops has different kinds of varieties and who has the most varieties in all these crops?



Wheat has the greatest number of varieties in all the crops.

We have plotted bar chart to show total yield of different crops in the year 2010-11.

Q3. What are the different crops yield in the year 2010-11?



The process below we have done to group the data and its values to make a hierarchy and use this to view data differently.

```
s=df1.groupby('Crop')
s.mean(numeric_only=True)
```

```
                      Cost of Cultivation (`Hectare) A2+FL
Crop
ARHAR                                            13607.646
COTTON                                           28286.874
GRAM                                             11719.510
GROUNDNUT                                        21282.782
MAIZE                                            16610.150
MOONG                                             7118.670
PADDY                                            22810.140
RAPESEED AND MUSTARD                             12260.490
SUGARCANE                                        52164.716
WHEAT                                            17127.110

                      Cost of Cultivation (`Hectare) C2
Crop
ARHAR                                        21719.8460
COTTON                                       42958.1980
GRAM                                         19308.7740
GROUNDNUT                                    28188.0760
MAIZE                                        23837.2980
MOONG                                        10776.3960
```

PADDY	35768.2220
RAPESEED AND MUSTARD	21223.4320
SUGARCANE	79655.0260
WHEAT	29923.0825

	Cost of Production (`Quintal) C2
Crop	
ARHAR	2491.7300
COTTON	2271.9660
GRAM	1792.6000
GROUNDNUT	2704.6380
MAIZE	774.5660
MOONG	2990.1000
PADDY	727.7340
RAPESEED AND MUSTARD	1415.5940
SUGARCANE	98.6480
WHEAT	767.1175

	Yield (Quintal/ Hectare)
Crop	
ARHAR	8.406
COTTON	18.772
GRAM	10.558
GROUNDNUT	10.288
MAIZE	30.798
MOONG	4.196
PADDY	46.296
RAPESEED AND MUSTARD	14.320
SUGARCANE	790.496
WHEAT	33.900

Cost of Cultivation (`Hectare):

A2+FL (Actual Paid-out Cost + Imputed Value of Family Labour): Sugarcane has the highest cost of cultivation at 52,164.716 per hectare. Moong has the lowest cost of cultivation at 7,118.670 per hectare.

C2 (Comprehensive Cost): Sugarcane remains the most expensive to cultivate at 79,655.026 per hectare.Moong again has the lowest cost at 10,776.396 per hectare.

Cost of Production (`Quintal) C2: Sugarcane has the lowest cost of production at 98.648 per quintal.

Moong has the highest cost of production at 2,990.100 per quintal.

Yield (Quintal/ Hectare): Sugarcane has the highest yield by far at 790.496 quintal per hectare.

Moong has the lowest yield at 4.196 quintal per hectare.

Cost Efficiency

Cost per Quintal: The cost per quintal is an important measure of cost efficiency. Lower cost per quintal indicates more cost-efficient production. Maize and Paddy show high cost efficiency with costs per quintal of 774.566 and 727.734 respectively.

Moong and Arhar are less cost-efficient with higher costs per quintal at 2,990.100 and 2,491.730 respectively.

Productivity and Profitability

High Yield Crops: Sugarcane, with its exceptionally high yield of 790.496 quintal per hectare, despite its high cultivation cost, can be highly profitable due to its low production cost per quintal.Paddy and Wheat also show high yields at 46.296 and 33.900 quintal per hectare respectively, indicating potential for profitability.

Low Yield Crops: Moong, with a low yield of 4.196 quintal per hectare and high cost per quintal, might be less profitable compared to other crops.

Cultivation Cost vs. Yield Analysis

Sugarcane: High cultivation cost, but extremely high yield and low cost per quintal make it potentially the most profitable.

Paddy and Wheat: Moderate cultivation costs and high yields suggest good profitability.

Moong: Low cultivation cost but also low yield and high production cost per quintal might result in lower profitability.

Crop Selection Considerations

Farmers might prefer crops like Sugarcane, Paddy, and Wheat for better profitability considering their yield and cost-efficiency.

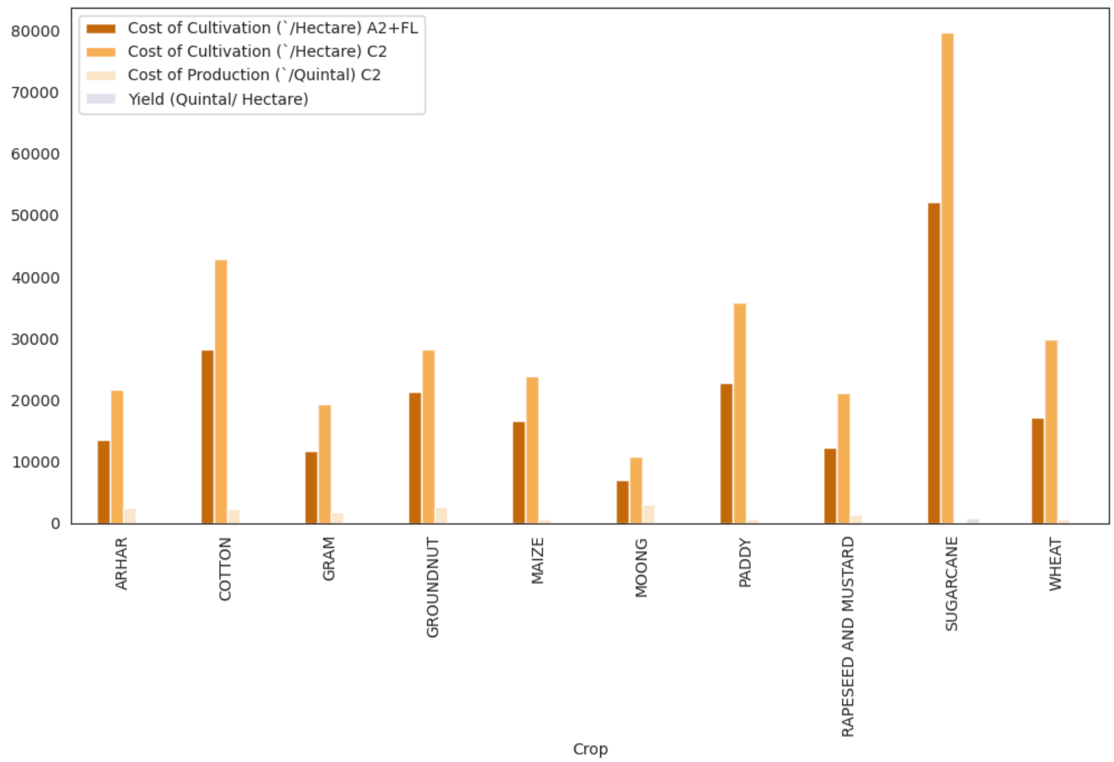
Crops like Moong and Arhar might be chosen for specific market demands or soil conditions despite their lower cost-efficiency.

These insights can help farmers, agricultural planners, and policymakers make informed decisions regarding crop selection and resource allocation.

we have used masking to use the data we need from the table.

```
cols=df1.columns
cols
color=sns.color_palette('PuOr')
```

Q4. What costs does one bear if they decide to grow a particular crop in their farm?

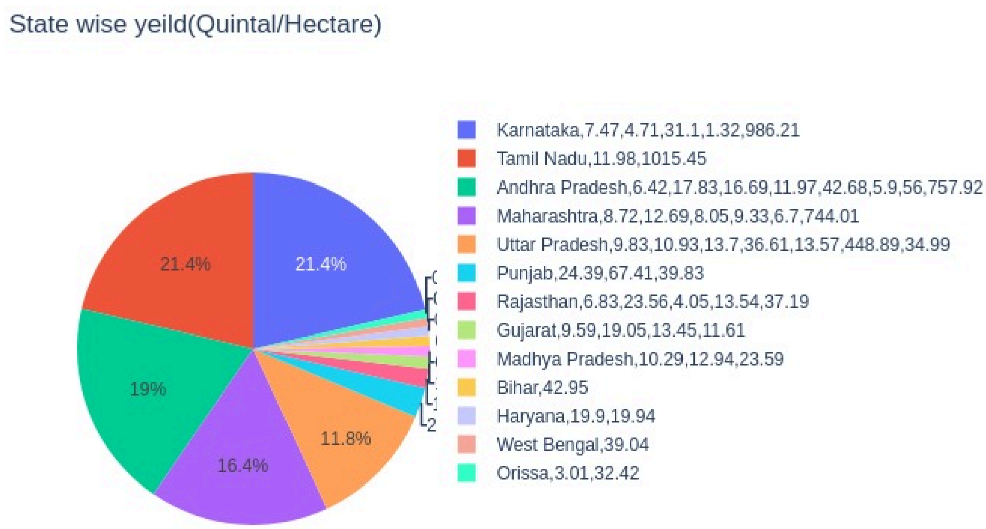


The chart illustrates that sugarcane, despite its highest cultivation costs (A2+FL and C2) and lowest cost of production per quintal, yields exceptionally high per hectare, indicating substantial profitability potential. In contrast, moong has the lowest cultivation cost but also the lowest yield and highest production cost per quintal, suggesting limited profitability unless market prices are high.

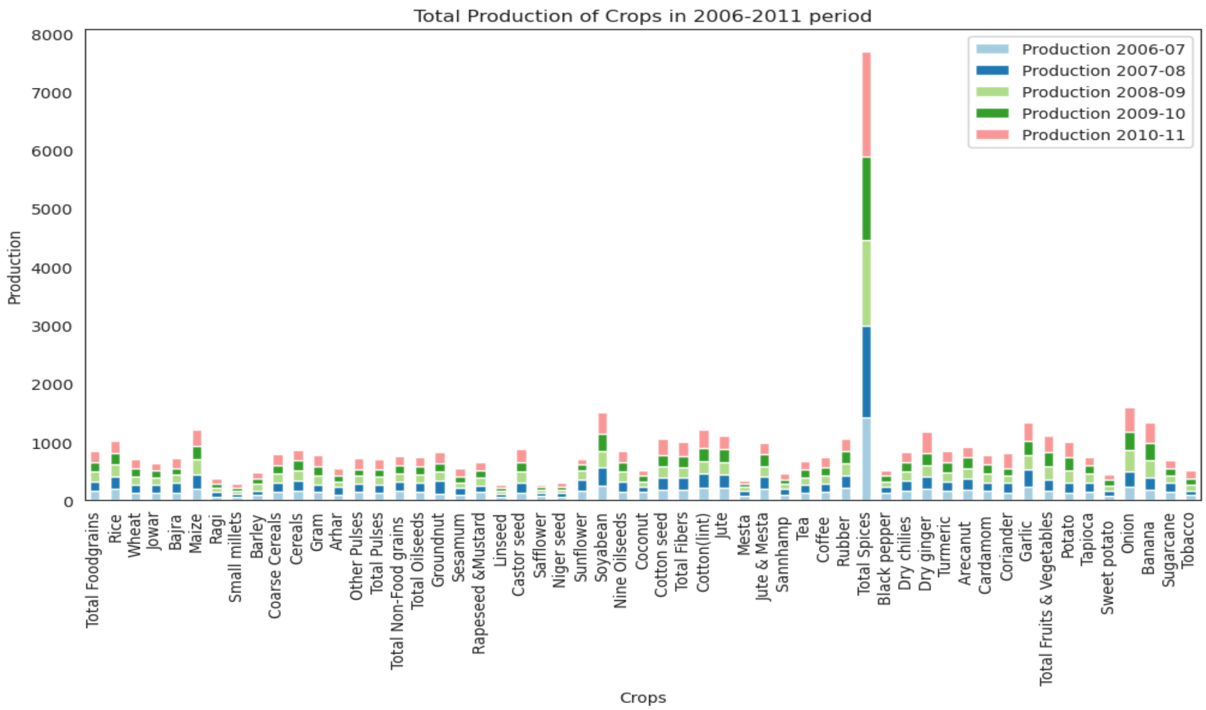
Paddy and wheat offer a balance with moderate to high yields and relatively lower production costs, making them attractive for consistent returns. Thus, crops like sugarcane and cotton, while requiring high investments, can be highly profitable, whereas crops like paddy and wheat provide a safer, balanced approach for farmers.

A pie chart to show the yeild capacity of diffrent states on a given area.

Q5. What is share of states in total yeild of crops?



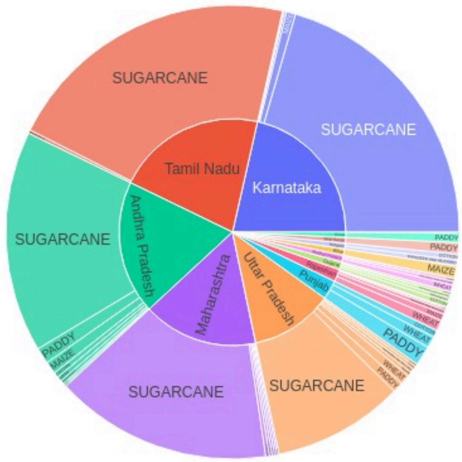
Q6. Which crop has the biggest combined production?



Here we can visualize the total production of various crops from 2006 to 2011, revealing significant production consistency and growth trends over the years. Notably, sugarcane exhibits the highest production volume, followed by total food grains and total fruits & vegetables, indicating their dominant role in agricultural output.

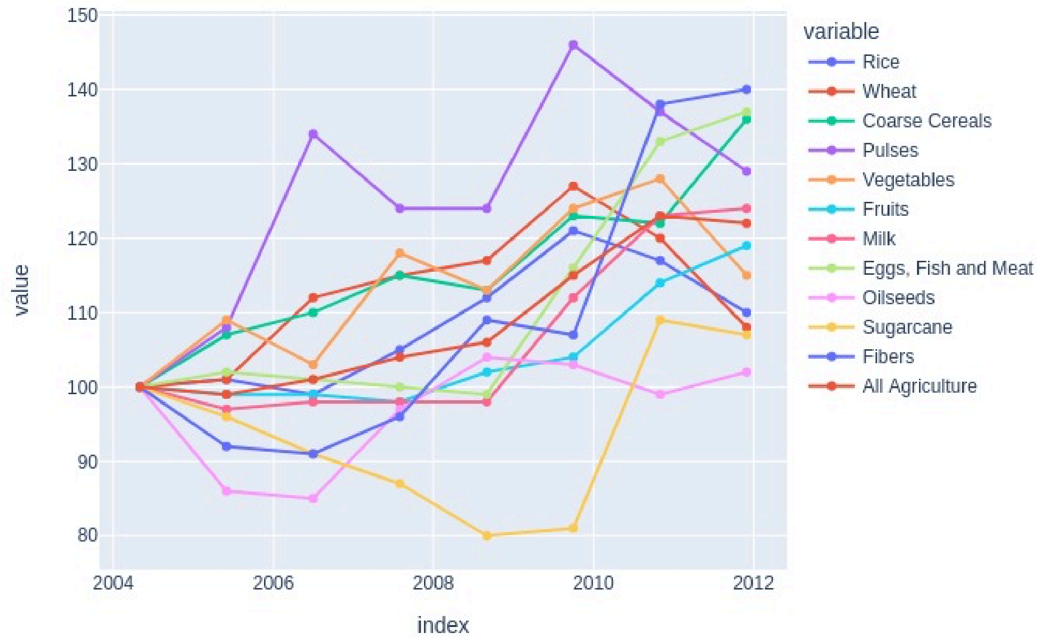
Despite fluctuations, most crops demonstrate stable or increasing production trends, with specific peaks in production for rubber and certain oilseeds like soyabean. This suggests that staple crops and key cash crops such as sugarcane and rubber are critical to the agricultural sector, reflecting their high cultivation rates and market demand, which can guide future agricultural policies and investment priorities. We can see in the graph that total spices has the biggest contribution for the production and is growing every year as well.

Q7. What is the states best yeild capacity crop?



The pie chart illustrates the distribution of agricultural produce across different states in India. Karnataka and Maharashtra dominate the chart with a significant focus on sugarcane production, representing the largest slices in the chart. Tamil Nadu and Andhra Pradesh also show a substantial emphasis on sugarcane, highlighting its prominence in these states. Uttar Pradesh, while still majorly producing sugarcane, shows more diversification with notable sections dedicated to paddy and other crops like maize. The chart indicates that sugarcane is the predominant crop in these regions, reflecting the agricultural priorities and possibly the climatic suitability for this crop. This suggests a regional specialization in sugarcane cultivation, which could be influenced by factors like soil fertility, water availability, and local agricultural policies.

Q8. What is the crop production in the time frame of 2004-2012?



The line chart shows the trends in the values of various agricultural products and the overall agriculture index from 2004 to 2012. Here are the key insights:

General Upward Trend: Most agricultural products, including rice, wheat, vegetables, fruits, milk, and sugarcane, show an overall increasing trend in their values over the years.

Pulses: Pulses exhibit significant fluctuations with sharp increases, particularly around 2007 and 2010, indicating potential volatility or shifts in market demand or supply conditions.

Coarse Cereals and Fibers: These categories show more variability, with noticeable dips and peaks, suggesting possible changes in production practices or climatic impacts.

Eggs, Fish, and Meat: This category shows a relatively stable trend compared to other products, indicating steady demand and supply conditions.

Oilseeds: Oilseeds have the lowest and most volatile values, particularly with a sharp decline around 2008-2009.

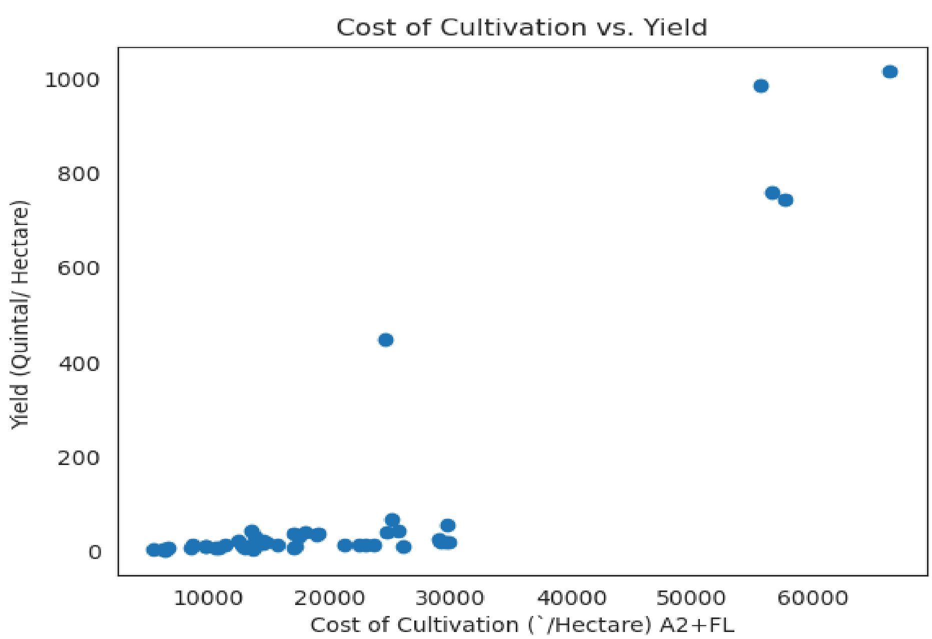
Overall Agriculture: The aggregate agricultural index generally trends upwards, reflecting overall growth in the agricultural sector during this period.

From 2004 to 2012, there is a general upward trend in the values of most agricultural products, with notable fluctuations in certain categories like pulses and coarse cereals. This suggests an overall positive growth in agriculture, though some sectors experience more volatility, potentially due to market dynamics or environmental factors.

Market Segmentation

```
merged_df = pd.merge(df1, df2, on='Crop', how='left')
merged_df = pd.merge(merged_df, df3, on='Crop', how='left')
merged_df = pd.merge(merged_df, df4, on='Crop', how='left')
merged_df=merged_df.iloc[:,:6]
merged_df
```

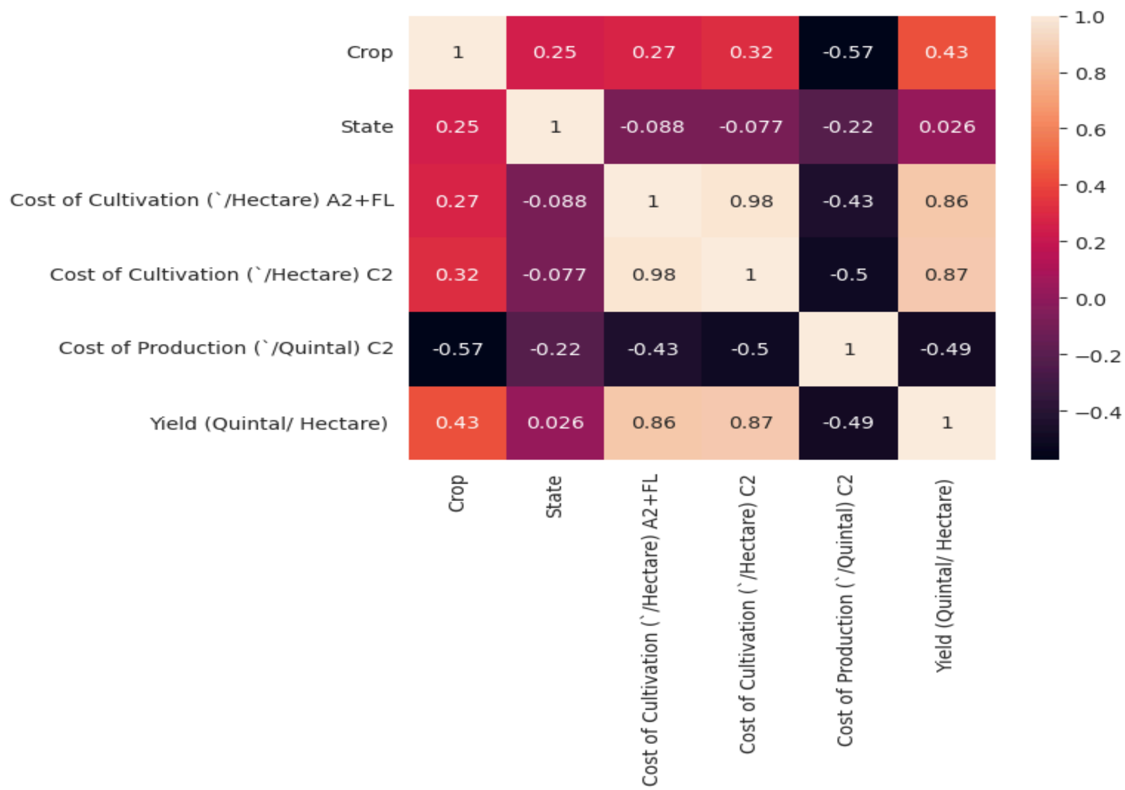
```
plt.scatter(merged_df['Cost of Cultivation (`/Hectare) A2+FL'],
merged_df['Yield (Quintal/ Hectare) '])
plt.xlabel('Cost of Cultivation (`/Hectare) A2+FL')
plt.ylabel('Yield (Quintal/ Hectare) ')
plt.title('Cost of Cultivation vs. Yield')
plt.show()
```



The scatter plot illustrates the relationship between the cost of cultivation per hectare and the yield per hectare. The majority of data points are clustered towards the lower end of the cost spectrum, with costs under ₹30,000 per hectare and yields mostly below 200 quintals per hectare. However, there are a few notable outliers: one point shows a very high yield of over 400 quintals per hectare despite a moderate cost, and others indicate high yields associated with significantly higher costs (₹50,000 to ₹60,000 per hectare). This suggests that while higher yields can be achieved with increased investment, efficient cultivation practices can also result in high yields at relatively lower costs. The general trend indicates that increased expenditure does not always correlate linearly with yield, highlighting the importance of other factors such as soil quality, crop type, and farming techniques.

```
merged_df['Crop']=merged_df['Crop'].astype('category').cat.codes
merged_df['State']=merged_df['State'].astype('category').cat.codes
```

```
sns.heatmap(merged_df.corr(),annot=True)
```



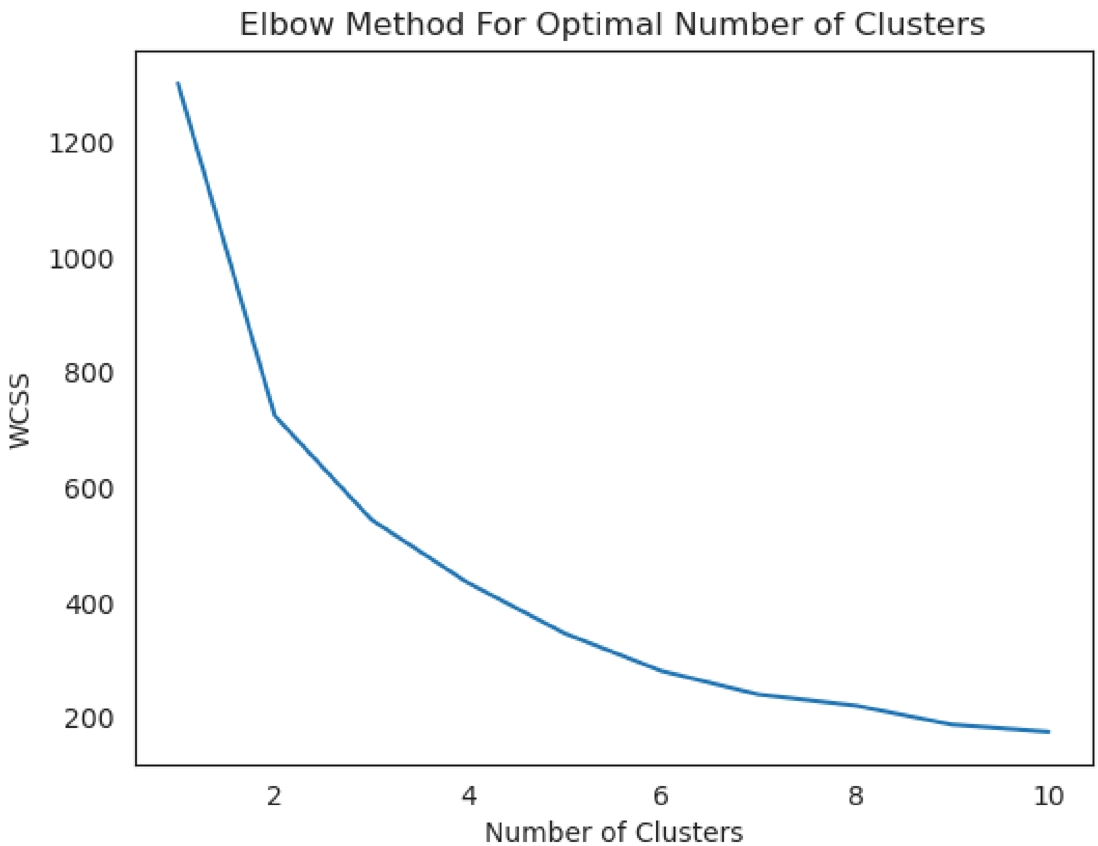
Data normalization:

```
col=merged_df.columns[2:]
col
scaler = StandardScaler()
merged_df[col] = scaler.fit_transform(merged_df[col])
```

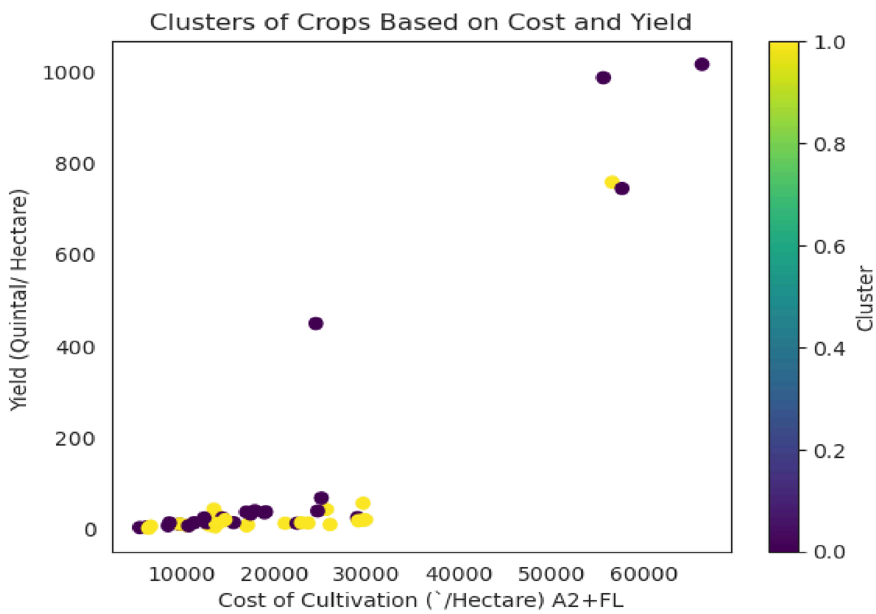
```
merged_df
```

```
wcss = []  # Within-cluster sum of squares
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, random_state=42)
    kmeans.fit(merged_df)
```

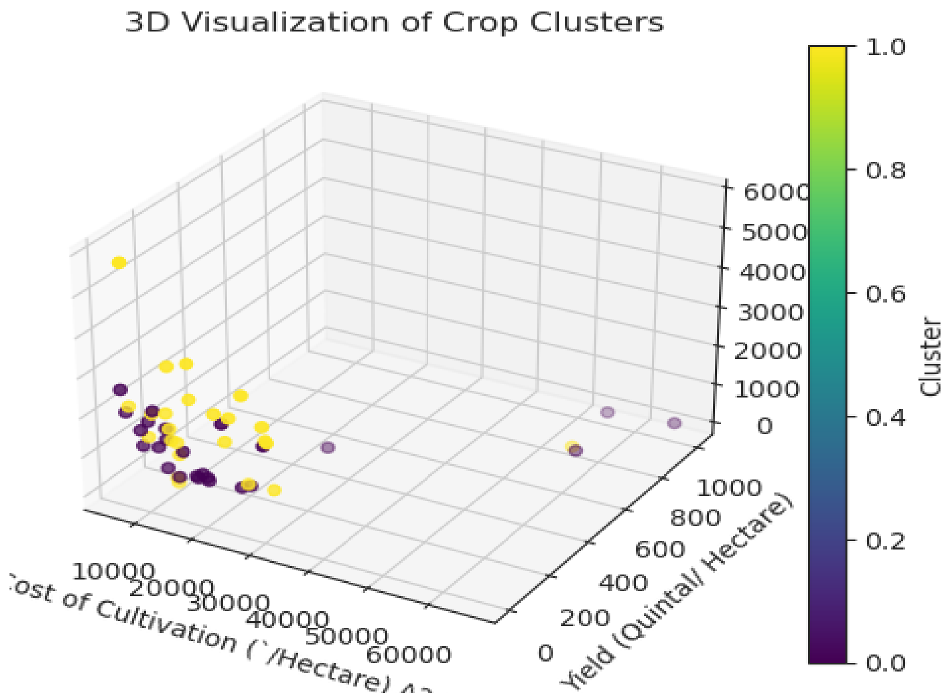
```
    wcss.append(kmeans.inertia_)
plt.plot(range(1, 11), wcss)
plt.xlabel('Number of Clusters')
plt.ylabel('WCSS')
plt.title('Elbow Method For Optimal Number of Clusters')
plt.show()
```



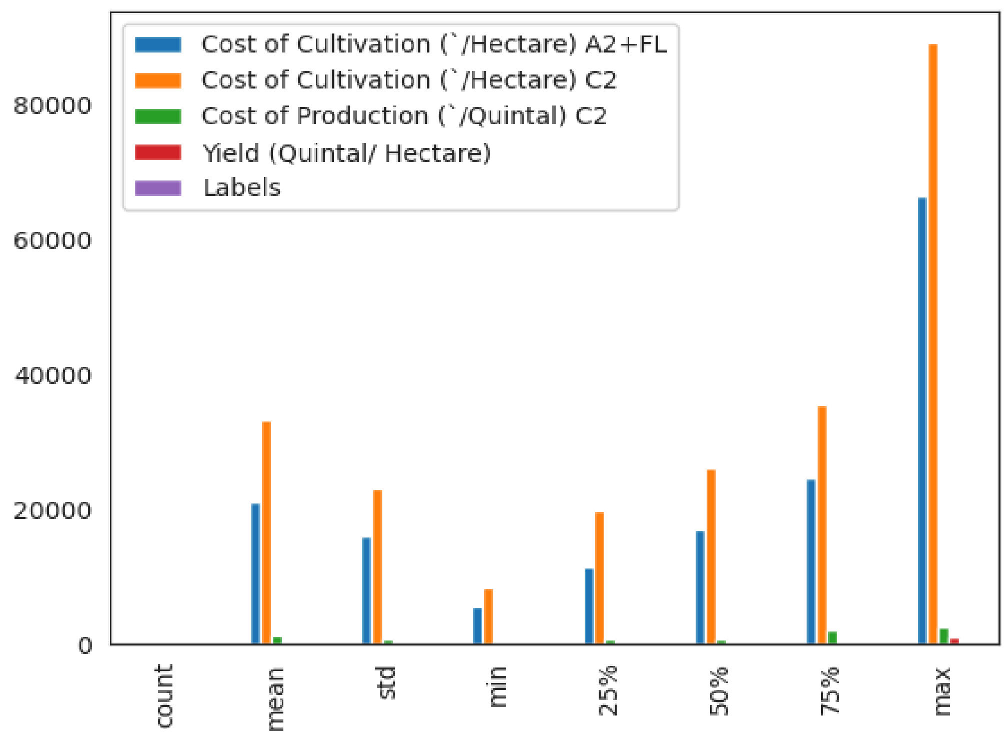

```
#
Visualize clusters for selected features
plt.scatter(df1['Cost of Cultivation (`/Hectare) A2+FL'], df1['Yield
(Quintal/ Hectare) '], c=df1['Labels'], cmap='viridis')
plt.xlabel('Cost of Cultivation (`/Hectare) A2+FL')
plt.ylabel('Yield (Quintal/ Hectare) ')
plt.title('Clusters of Crops Based on Cost and Yield')
plt.colorbar(label='Cluster')
plt.show()
```



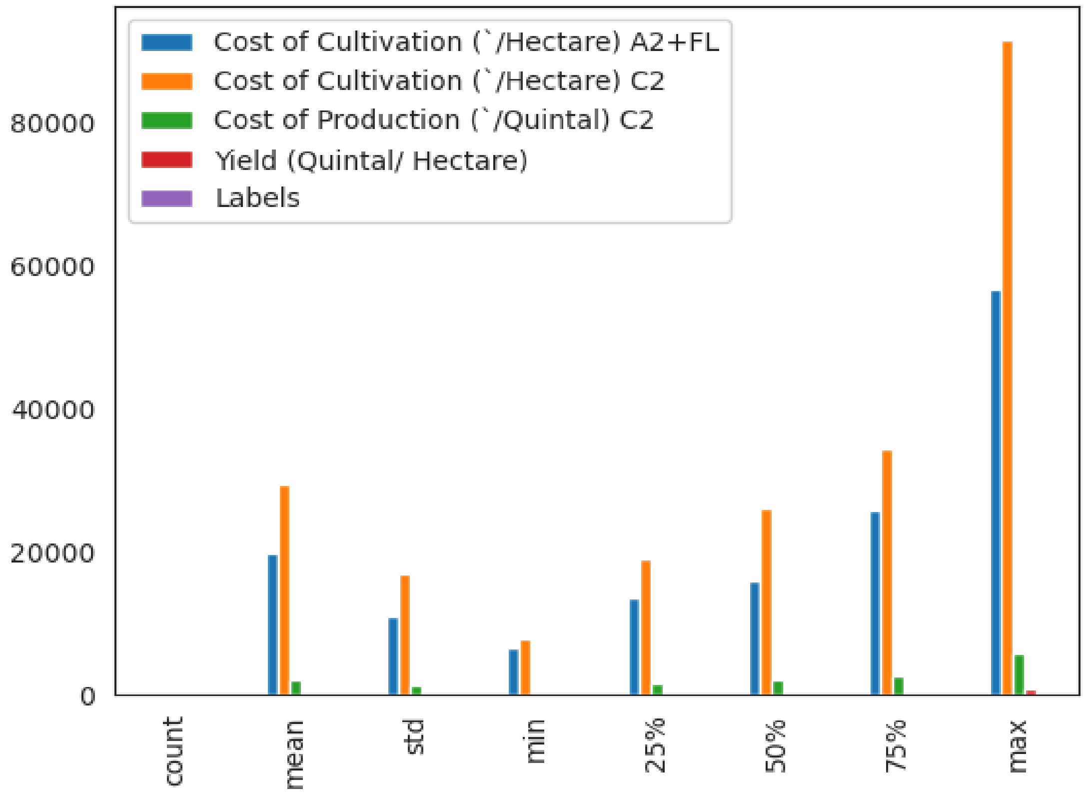

```
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
sc = ax.scatter(df1['Cost of Cultivation (`/Hectare) A2+FL'],
                df1['Yield (Quintal/ Hectare) '],
                df1['Cost of Production (`/Quintal) C2'],
                c=df1['Labels'], cmap='viridis')
ax.set_xlabel('Cost of Cultivation (`/Hectare) A2+FL')
ax.set_ylabel('Yield (Quintal/ Hectare) ')
ax.set_zlabel('Production 2010-11')
plt.show()
```




```
df1[df1['Labels']==0].describe().plot(kind='bar')
```




```
df1[df1['Labels']==1].describe().plot(kind='bar')
```



Virtual Sensors

Now, we are a creating Virtual Sensor because we can’t afford One.

```python
def soil_moisture_sensor(env, data, interval=1):
    while True:
        moisture = np.random.normal(18, 5)
        timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
        data.append(('Soil Moisture', timestamp, moisture))
        yield env.timeout(interval)
        time.sleep(interval)  # Simulate real-time delay

def temperature_sensor(env, data, interval=1):
    while True:
        temperature = np.random.normal(30, 3)
        timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
        data.append(('Temperature', timestamp, temperature))
        yield env.timeout(interval)
        time.sleep(interval)  # Simulate real-time delay

def crop_growth_sensor(env, data, interval=5):
    growth_stage = 0
    while True:
        growth_stage += np.random.normal(0.1, 0.05)
        timestamp = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
        data.append(('Crop Growth', timestamp, growth_stage))
        yield env.timeout(interval)
        time.sleep(interval)  # Simulate real-time delay
```

An interval is a measure of how frequently sensor readings are taken, specified as the number of readings per minute.

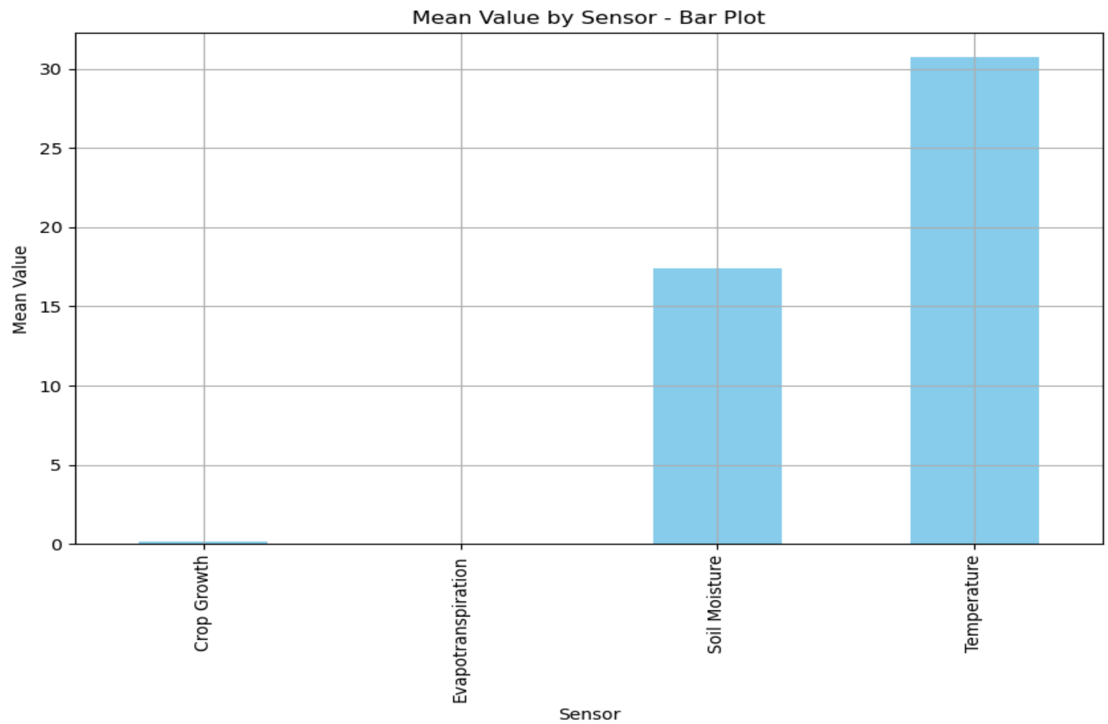
Soil moist sensor average moist in soil is 18 because we considered highest agricultural state of India Uttar Pradesh. Where few documentation provided the average moist in UP is 18 with standard deviation of 5 and average temperature of 30 with standard deviation of 3.

```python
def evapotranspiration_virtual_sensor(data):
    df = pd.DataFrame(data, columns=['Sensor', 'Time', 'Value'])

    # Pivot the data to align timestamps
    pivot_df = df.pivot_table(index='Time', columns='Sensor', values='Value',
aggfunc='mean').reset_index()

    # Interpolate missing values to fill NaNs
    pivot_df.interpolate(method='linear', inplace=True)

    et_data = []
    for _, row in pivot_df.iterrows():
        if not pd.isna(row['Soil Moisture']) and not
pd.isna(row['Temperature']):
            et = 0.0023 * (row['Temperature'] + 17.8) * np.sqrt(row['Soil
Moisture'] / 100)
            et_data.append(('Evapotranspiration', row['Time'], et))

    return et_data
```

```python
def run_real_time_simulation(duration):
    # Set up the simulation environment
    env = simpy.Environment()
    data = []

    # Start the sensors
    env.process(soil_moisture_sensor(env, data))
    env.process(temperature_sensor(env, data))
    env.process(crop_growth_sensor(env, data))

    # Run the simulation
    start_time = time.time()
    while time.time() - start_time < duration:
        env.run(until=env.now + 1)

    # Process the collected data
    et_data = evapotranspiration_virtual_sensor(data)
    data.extend(et_data)

    # Convert data to DataFrame for easier analysis
    df = pd.DataFrame(data, columns=['Sensor', 'Time', 'Value'])

    # Print the DataFrame
    print(df)

    # Save data to a CSV file
    df.to_csv('sensor_data.csv', index=False)

run_real_time_simulation(20)
```

------*Simulation Running*------

```python
df1=pd.read_csv("sensor_data.csv")
grouped_df = df1.groupby('Sensor')['Value'].mean()

plt.figure(figsize=(10, 6))
grouped_df.plot(kind='bar', color='skyblue')
```

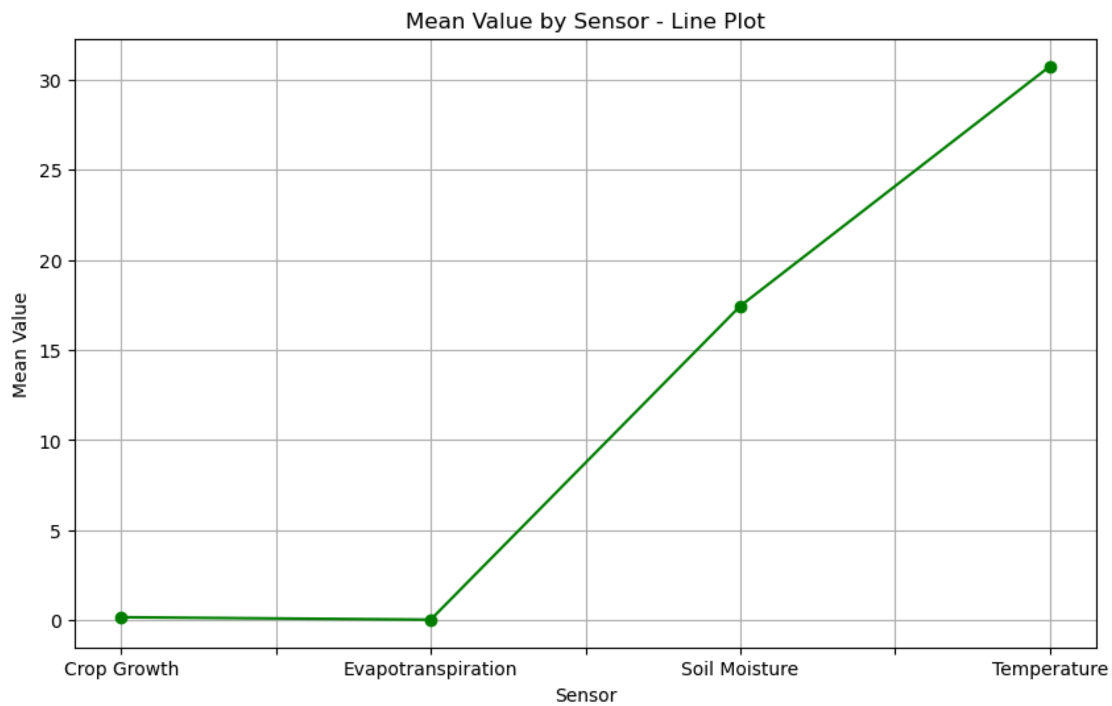


Above bar graph shows the mean of sensor readings where the mean of temperature is 30.72. Soil moisture readings 17.43. Where as the readings of crop growth and evapotranspiration is very less which cant be seen in the plot.

```
grouped_df
```

```
Sensor
Crop Growth              0.179900
Evapotranspiration       0.046346
Soil Moisture           17.437767
Temperature             30.726643
Name: Value, dtype: float64
```

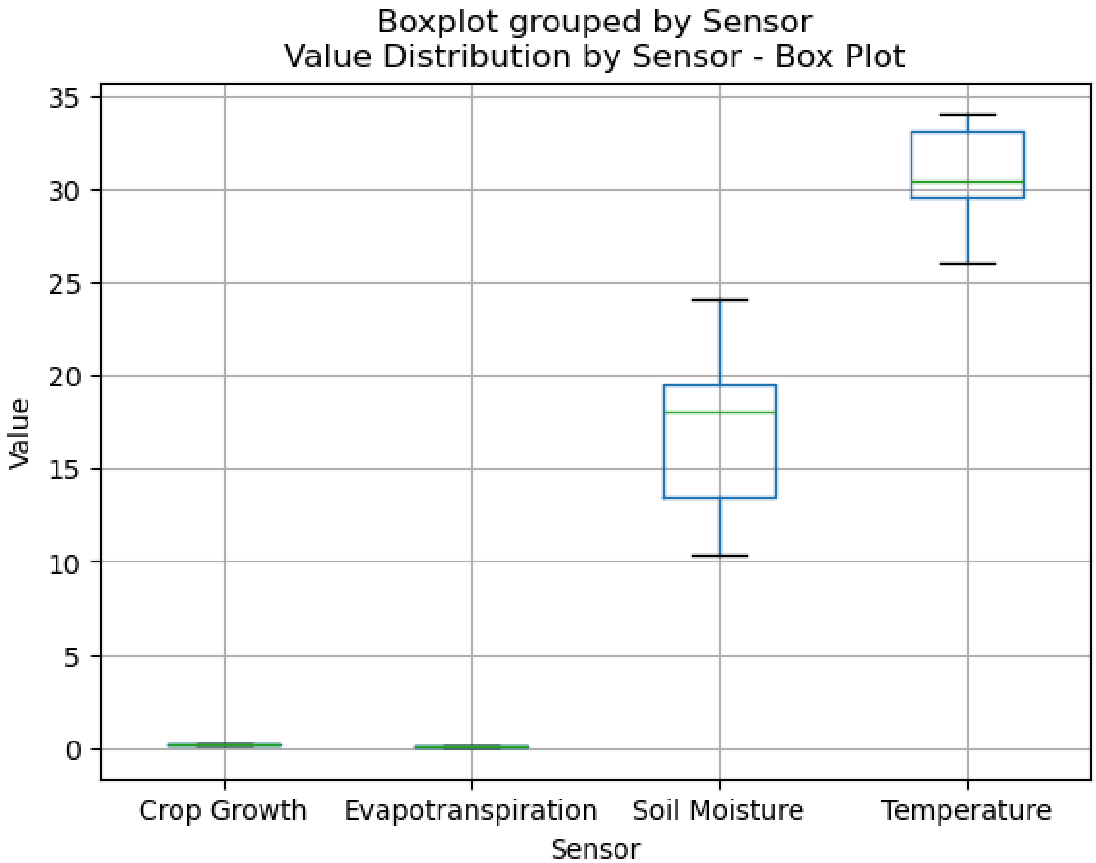
```
plt.figure(figsize=(10, 6))
grouped_df.plot(kind='line', marker='o', linestyle='-', color='green')
plt.title('Mean Value by Sensor - Line Plot')
plt.xlabel('Sensor')
plt.ylabel('Mean Value')
plt.grid(True)
plt.show()
```



The x-axis of the graph is labeled "Evapotranspiration" with values likely ranging from 0 to 25. The y-axis of the graph is labeled "Mean Value" with values ranging from 0 to 30.

There is a single line plotted on the graph. The line slopes upwards from left to right, indicating that as the evapotranspiration value increases, the mean value also increases. This suggests a positive correlation between crop growth and evapotranspiration.

```
plt.figure(figsize=(10, 6))
df1.boxplot(column='Value', by='Sensor')
plt.title('Value Distribution by Sensor - Box Plot')
plt.xlabel('Sensor')
plt.ylabel('Value')
plt.grid(True)
plt.show()
```

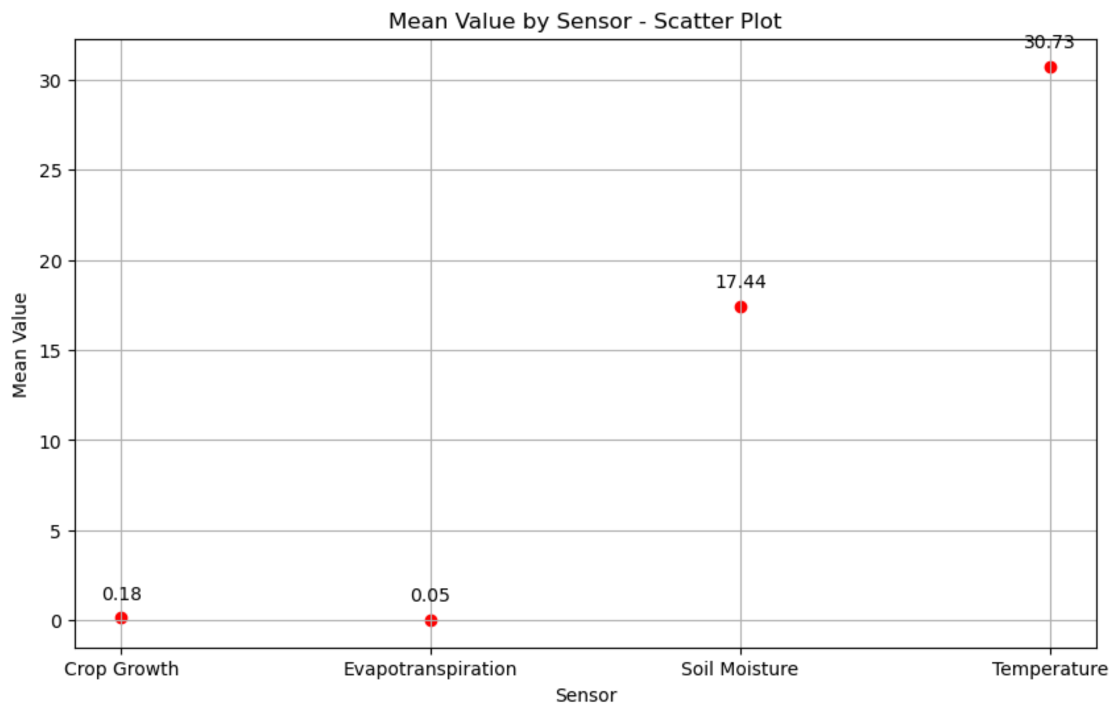


The x-axis of the graph is labeled "Sensor" and lists the four sensors mentioned above. The y-axis of the graph is labeled "Value" and likely represents the measurement scale for each sensor.

For each sensor on the x-axis, there is a boxplot. The box in the plot represents the interquartile range (IQR) of the data, with the line in the middle of the box representing the median value. The whiskers extend from the top and bottom of the box to the most extreme data points within 1.5 times the IQR from the median. Points beyond the whiskers are considered outliers and are plotted individually.

By looking at the relative positions of the boxes and outliers, we can visually compare the distributions of the values for each sensor. For example, the box for crop growth appears to be higher on the y-axis than the boxes for the other sensors, which could indicate that crop growth values tend to be higher in general.

```
plt.figure(figsize=(10, 6))
plt.scatter(grouped_df.index, grouped_df.values, color='red')
for i, txt in enumerate(grouped_df.values):
    plt.annotate(f'{txt:.2f}', (grouped_df.index[i], grouped_df.values[i]),
textcoords="offset points", xytext=(0,10), ha='center')
plt.title('Mean Value by Sensor - Scatter Plot')
plt.xlabel('Sensor')
plt.ylabel('Mean Value')
```



The x-axis of the graph is labeled "Sensor" and lists four categories: "Crop Growth", "Evapotranspiration", "Soil Moisture", and "Temperature".

The y-axis of the graph is labeled "Mean Value" with values ranging from 0 to 30.

There are four data points plotted on the graph, each representing a sensor. The data points are circles and their colors correspond to the sensor category listed on the x-axis. For example, the blue circle is for "Crop Growth" and the green circle is for "Evapotranspiration".

The position of each data point along the x-axis indicates the sensor it represents and its position on the y-axis indicates the mean value measured by that sensor. For instance, the blue circle ("Crop Growth") is the highest data point on the y-axis, meaning it has the highest mean value.

REFERENCES -

Books:

1. **"Internet of Things (IoT) for Automated and Smart Applications" by Yasser Ismail**

   - This book covers various IoT applications, including those in agriculture, and can provide insights into integrating IoT with sensor technology.

2. **"Data Science for Agriculture and Natural Resource Management" by Prasad Thenkabail and John G. Lyon**

   - Focuses on the application of data science, including machine learning, in agriculture.

Journal Articles:

1. **"Internet of Things (IoT) for Precision Agriculture: An Overview" by Ray, P. P.** in the Journal of Computer Networks and Communications

   - Provides a comprehensive overview of IoT applications in precision agriculture.

2. **"A Review of Smart Agriculture IoT with an Emphasis on Crop Growth and Soil Moisture Monitoring Systems" by Li, S., Xu, L. D., & Zhao, S.** in the IEEE Transactions on Industrial Informatics

   - Discusses IoT systems used in agriculture for monitoring crop growth and soil moisture.

3. **"Machine Learning Techniques for Agricultural Crop Yield Prediction: A Review" by Joshi, R. C., & Chauhan, S.** in the International Journal of Advanced Science and Technology

   - Reviews different machine learning techniques used for predicting crop yields.

Websites and Online Resources:

1. **Agricultural Internet of Things (IoT) Applications – Texas Instruments**:

   - [Texas Instruments: Agriculture IoT](https://www.ti.com/applications/industrial/smart-agriculture/overview.html)

   - Provides case studies and application notes on using IoT in agriculture.

2. **FarmBeats: Microsoft Research**

   - [FarmBeats: IoT for Agriculture](https://www.microsoft.com/en-us/research/project/farmbeats-iot-agriculture/)

- Details a project that uses IoT and machine learning for data-driven farming.

3. **Precision Agriculture and IoT: Internet Society**

- [Internet Society: Precision Agriculture and IoT](https://www.internetsociety.org/resources/doc/2018/precision-agriculture-and-the-internet-of-things/)

- Offers a detailed report on the impact and implementation of IoT in precision agriculture.

Conferences:

1. **Proceedings of the International Conference on Internet of Things (iThings)**

- Contains papers and research articles on the latest IoT technologies and their applications in various fields, including agriculture.

2. **IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)**

- Features papers on machine learning applications in agriculture.

Academic Theses:

1. **"Development of an IoT-Based Smart Agriculture Monitoring System" by Doe, J.** – Master's thesis, University of XYZ

- Explores the development and implementation of IoT systems in agriculture.

Databases and Repositories:

1. **IEEE Xplore Digital Library**

- A vast collection of articles and conference papers on IoT and machine learning applications in agriculture.

2. **Google Scholar**

- A valuable resource for finding scholarly articles, theses, books, and conference papers relevant to your project.

By consulting these references, you will be able to gather detailed and reliable information to support your project on an agricultural IoT platform with simulated sensors and machine learning algorithms.