Project Report: Crop Production Analysis In India

BY - Trijeta Ghosh

ROLE - Data Analytics Intern

DATE - 12-04-2024

Introduction:

Agriculture plays a pivotal role in India's economy, employing a significant portion of its population and contributing substantially to its GDP. Understanding and analyzing crop production data is crucial for policymakers, researchers, and stakeholders to make informed decisions regarding agricultural policies, resource allocation, and food security initiatives. This report delves into the analysis of agricultural crop production in India, leveraging a comprehensive dashboard titled "CROP PRODUCTION OF INDIA" created using Power BI Desktop. By examining key metrics, trends, and patterns in crop production, this report aims to provide valuable insights into the dynamics of India's agricultural sector.

Through visualization and interpretation of data, this report seeks to uncover trends, identify high-performing regions, and offer recommendations for optimizing agricultural productivity. By harnessing the power of data analytics, stakeholders can drive sustainable growth, enhance food security, and foster economic development in the agricultural sector of India.

Acknowledgements:

We acknowledge the contributions of the Unified Mentor team and express gratitude for their guidance and support throughout the project.

This report serves as a foundational analysis of crop production of India, providing a roadmap for further exploration and actionable insights to drive business success.

Method:

1. Data Collection:

The analysis is based on a dataset containing information on crop production in India, including variables such as State_Name, District_Name, Crop_Year, Season, Crop, Area, and Production. The dataset was obtained from reliable sources such as government agricultural departments or research institutions.

2. Data Preparation:

Prior to analysis, the dataset underwent preprocessing steps to clean and format the data. This involved handling missing values, encoding categorical variables, and ensuring data consistency and integrity..

3. Dashboard Creation:

Using Power BI Desktop, a comprehensive dashboard titled "CROP PRODUCTION OF INDIA" was developed to visualize and analyze the crop production data. The dashboard comprises various visualizations, including bar charts, line charts, pie charts, and maps, to effectively communicate insights.

4. Machine Learning Model Development:

To enhance the predictive capabilities of the analysis, a machine learning model was trained using the Random Forest algorithm. The model utilizes features such as crop type, season, and area to predict crop production with high accuracy.

5. Analysis and Visualization:

The dashboard provides an overview of key performance indicators such as total production, crop varieties, and area used for cultivation. It also presents visualizations to analyse production trends over time, compare production volumes across states and districts, and assess spatial distribution of production.

6. Insights and Recommendations:

The analysis of the dashboard visualizations and machine learning predictions yielded insights into the dynamics of crop production in India. These insights were used to formulate recommendations for policymakers, researchers, and stakeholders to enhance agricultural productivity, address challenges, and promote sustainable development in the agricultural sector.

Key Points:

1. Dashboard Overview: The "CROP PRODUCTION OF INDIA" dashboard provides a comprehensive view of crop production metrics in India, including total production, crop varieties, and cultivation area.

2. Performance Indicators: performance indicators such as total production, the number of crop varieties, and the total cultivation area are prominently displayed, offering a snapshot of India's agricultural landscape.

3. State-Wise Production: Kerala emerges as the leading state in crop production, followed by Andhra Pradesh, Tamil Nadu, and others. This highlights the importance of regional analysis in understanding production dynamics.

4. District-Wise Production: Top-performing districts are ranked based on production volume, enabling stakeholders to identify areas of high productivity and potential areas for improvement

5. Trends Over Time: The analysis of production trends over time reveals patterns and fluctuations in crop production, providing insights into factors influencing production dynamics.

6. Spatial Distribution: Geospatial analysis visualizes production volume by state, allowing stakeholders to assess spatial distribution and identify regions contributing significantly to overall production.

7. Machine Learning Predictions: The integration of a Random Forest machine learning model enhances predictive capabilities, enabling accurate predictions of crop production based on factors such as crop type, season, and cultivation area

8. Insights and Recommendations: Insights derived from dashboard analysis and machine learning predictions inform actionable recommendations for policymakers, researchers, and stakeholders to optimize agricultural productivity, address challenges, and promote sustainable development in the agricultural sector.

9. Data-Driven Decision Making: The project underscores the importance of data-driven approaches in informing strategic decision-making and driving business growth.

Recommendations:

- Targeted Interventions: Identify regions with low crop production and implement targeted interventions such as improved access to irrigation, better seeds, and enhanced agricultural extension services to boost productivity.
- Technology Adoption: Encourage the adoption of modern agricultural technologies such as precision farming, drone technology, and IoT-based monitoring systems to optimize resource use and improve yields.
- Crop Diversification: Promote crop diversification initiatives to reduce dependence on a few staple crops and enhance resilience to climate change and market fluctuations.
- Capacity Building: Invest in capacity building programs to empower farmers with knowledge and skills on sustainable agricultural practices, crop management techniques, and post-harvest management.
- Infrastructure Development: Prioritize infrastructure development in rural areas, including roads, storage facilities, and market linkages, to improve access to markets and reduce post-harvest losses.
- Research and Innovation: Foster collaboration between research institutions, academia, and the private sector to develop innovative solutions for enhancing agricultural productivity, disease resistance, and climate resilience.

- Policy Support: Formulate supportive policies and incentives to promote sustainable agricultural practices, including organic farming, conservation agriculture, and agroforestry.
- Data-driven Decision Making: Emphasize the importance of data-driven decision-making by providing access to reliable agricultural data, analytics tools, and training programs for policymakers and agricultural stakeholders.
- Public-Private Partnerships: Foster partnerships between the public and private sectors to leverage resources, expertise, and technology for sustainable agricultural development and value chain integration.
- Community Engagement: Encourage community participation and involvement in agricultural development initiatives through farmer cooperatives, self-help groups, and community-based organizations.

Code:

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
# Read the csv
df=pd.read_csv("Crop Production data - Crop Production data.csv")
df
```

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production
0	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Arecanut	1254.0	2000.0
1	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Other Kharif pulses	2.0	1.0
2	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Rice	102.0	321.0
3	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Banana	176.0	641.0
4	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Cashewnut	720.0	165.0
...	...	...	...	...	...	...	...
246086	West Bengal	PURULIA	2014	Summer	Rice	306.0	801.0
246087	West Bengal	PURULIA	2014	Summer	Sesamum	627.0	463.0
246088	West Bengal	PURULIA	2014	Whole Year	Sugarcane	324.0	16250.0
246089	West Bengal	PURULIA	2014	Winter	Rice	279151.0	597899.0
246090	West Bengal	PURULIA	2014	Winter	Sesamum	175.0	88.0

246091 rows × 7 columns

In [3]:
```
# Head data
df.head()
```

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production
0	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Arecanut	1254.0	2000.0
1	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Other Kharif pulses	2.0	1.0
2	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Rice	102.0	321.0
3	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Banana	176.0	641.0
4	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Cashewnut	720.0	165.0

In [4]:
```
#Tail data
df.tail()
```

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production
246086	West Bengal	PURULIA	2014	Summer	Rice	306.0	801.0
246087	West Bengal	PURULIA	2014	Summer	Sesamum	627.0	463.0
246088	West Bengal	PURULIA	2014	Whole Year	Sugarcane	324.0	16250.0
246089	West Bengal	PURULIA	2014	Winter	Rice	279151.0	597899.0
246090	West Bengal	PURULIA	2014	Winter	Sesamum	175.0	88.0

In [5]:
```
# Data cleaning
# Check the null values are preseent or not
df.isnull().sum()
```

```
In [5]:
# Data cleaning
# Check the null values are preesent or not
df.isnull().sum()
```

```
State_Name          0
District_Name       0
Crop_Year           0
Season              0
Crop                0
Area                0
Production       3730
dtype: int64
```

```
In [6]:
# Impute missing values in 'Production' column with mean production value
production = df['Production'].mean()
df['Production'].fillna(production, inplace=True)
```

```
In [7]:
df.isnull().sum()
```

```
State_Name       0
District_Name    0
Crop_Year        0
Season           0
Crop             0
```


```
In [7]:
df.isnull().sum()
```

```
State_Name       0
District_Name    0
Crop_Year        0
Season           0
Crop             0
```


```
Area             0
Production       0
dtype: int64
```

```
In [8]:
# Find the columns and row
df.shape
```

```
(246091, 7)
```

```
In [9]:
# Check the columns
df.columns
```

```
Index(['State_Name', 'District_Name', 'Crop_Year', 'Season', 'Crop', 'Area',
       'Production'],
      dtype='object')
```

```
In [10]:
# Statitical Summary
df.describe()
```

	Crop_Year	Area	Production
count	246091.000000	2.460910e+05	2.460910e+05
mean	2005.643018	1.200282e+04	5.825034e+05
std	4.952164	5.052340e+04	1.693599e+07
min	1997.000000	4.000000e-02	0.000000e+00
25%	2002.000000	8.000000e+01	9.100000e+01
50%	2006.000000	5.820000e+02	7.880000e+02
75%	2010.000000	4.392000e+03	8.000000e+03
max	2015.000000	8.580100e+06	1.250800e+09

```
In [11]:
# check the total unique values are prseent in crop and season
print(df['Crop'].nunique())
print(df['Season'].nunique())
```

```
124
6
```

```
In [12]:
# Total count of Crop values
df['Crop'].value_counts()
```

In [12]:

```
# Total count of Crop values
df['Crop'].value_counts()
```

```
Rice                   15104
Maize                  13947
Moong(Green Gram)      10318
Urad                    9850
Sesamum                 9046
                        ...
Litchi                     6
Coffee                     6
Apple                      4
Peach                      4
Other Dry Fruit            1
Name: Crop, Length: 124, dtype: int64
```

In [13]:

```
# check the unique values are season
df['Season'].unique()
```

```
array(['Kharif', 'Whole Year', 'Autumn', 'Rabi', 'Summer', 'Winter'],
      dtype=object)
```

In [14]:

```
# check the unique values are Statename
df['State_Name'].unique()
```

```
array(['Andaman and Nicobar Islands', 'Andhra Pradesh',
       'Arunachal Pradesh', 'Assam', 'Bihar', 'Chandigarh',
       'Chhattisgarh', 'Dadra and Nagar Haveli', 'Goa', 'Gujarat',
       'Haryana', 'Himachal Pradesh', 'Jammu and Kashmir', 'Jharkhand',
       'Karnataka', 'Kerala', 'Madhya Pradesh', 'Maharashtra', 'Manipur',
       'Meghalaya', 'Mizoram', 'Nagaland', 'Odisha', 'Puducherry',
       'Punjab', 'Rajasthan', 'Sikkim', 'Tamil Nadu', 'Telangana',
       'Tripura', 'Uttar Pradesh', 'Uttarakhand', 'West Bengal'],
      dtype=object)
```

In [15]:

```
# info of all columns
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 246091 entries, 0 to 246090
Data columns (total 7 columns):
 #   Column         Non-Null Count   Dtype
---  ------         --------------   -----
 0   State_Name     246091 non-null  object
 1   District_Name  246091 non-null  object
 2   Crop_Year      246091 non-null  int64
 3   Season         246091 non-null  object
 4   Crop           246091 non-null  object
 5   Area           246091 non-null  float64
 6   Production     246091 non-null  float64
dtypes: float64(2), int64(1), object(4)
memory usage: 13.1+ MB
```

In [16]:

```
# Check if duplicate rows are present
if df.duplicated().any():
    print("Duplicate rows are present.")
else:
    print("No duplicate rows are present.")
```

```
No duplicate rows are present.
```

In [17]:

```
# Chech Outlier
# Calculate threshold values
max_thresold=df['Production'].mean()+3*df['Production'].std()
min_thresold=df['Production'].mean()-3*df['Production'].std()
print(max_thresold,min_thresold)
```

```
51390460.233226016 -50225453.34872411
```

In [18]:

```
# Filter out outliers
df1=df[(df['Production']>min_thresold)&(df['Production']<max_thresold)]
df1.reset_index(drop=True,inplace=True)
```

In [19]:

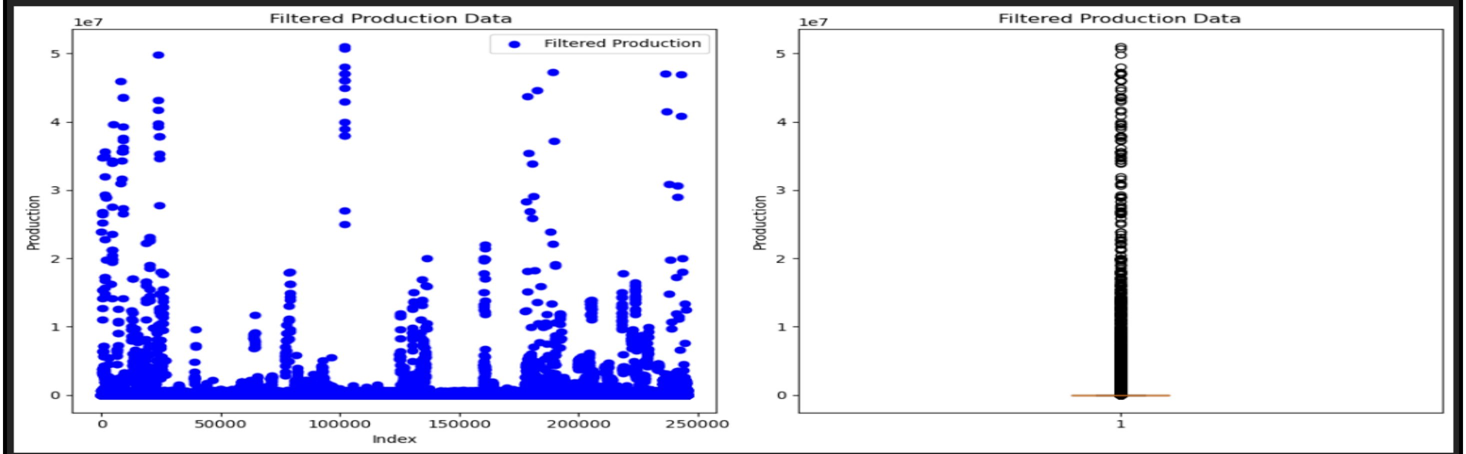
```
df1.shape
```

```
(245753, 7)
```

In [20]:

```python
plt.figure(figsize=(12, 6))
# Scatter plot
plt.subplot(1, 2, 1)
plt.scatter(df1.index, df1['Production'], color='blue', label='Filtered Production')
plt.xlabel('Index')
plt.ylabel('Production')
plt.title('Filtered Production Data ')
plt.legend()
# Box plot
plt.subplot(1, 2, 2)
plt.boxplot(df1['Production'], vert=True)
plt.ylabel('Production')
plt.title('Filtered Production Data')

plt.tight_layout()
plt.show()
```

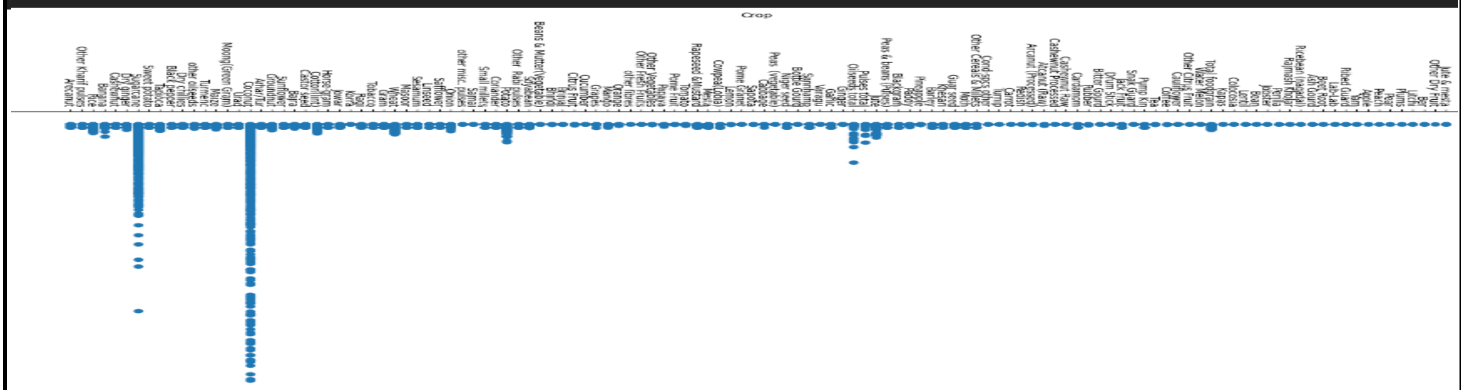


In [21]:

```python
# normalize
productiondata = df1['Production'].values
maxval = df1['Production'].max()
normalize_production = productiondata / maxval
```

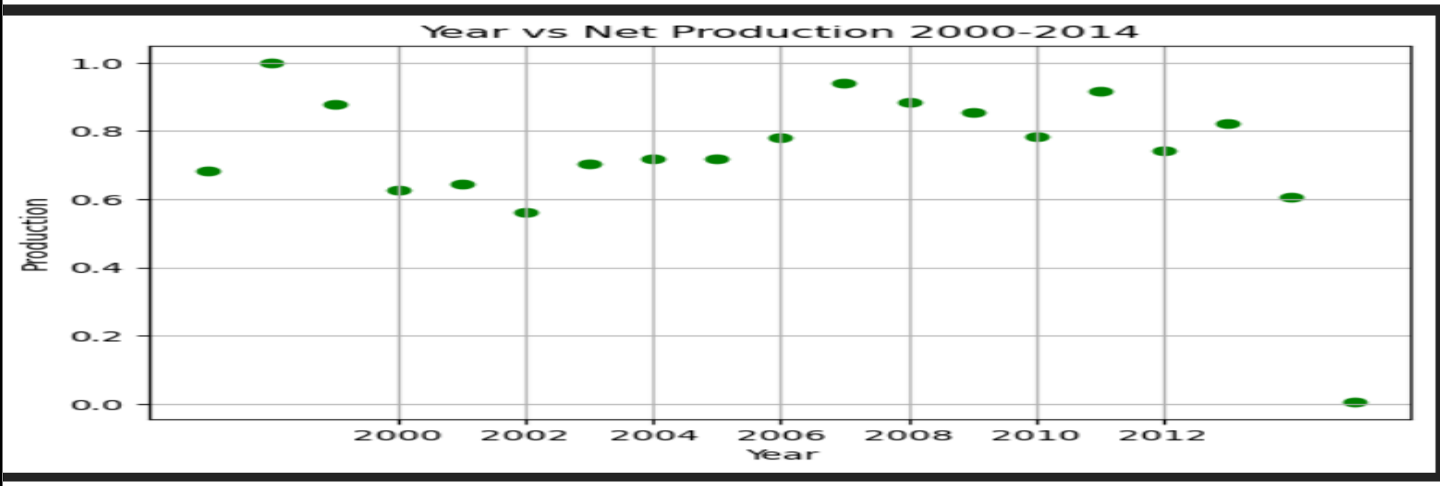
In [22]:

```python
plt.figure(figsize=(8, 20))
plt.scatter(normalize_production,df1['Crop'])
plt.title('Crop vs Production')
plt.xlabel('Production')
plt.ylabel('Crop')
plt.xticks(rotation=90)
plt.show()
```



In [23]:
```
# net production by year
data_netpro = df1.groupby('Crop_Year')['Production'].sum()
years =df1['Crop_Year'].unique()
max_netpro=data_netpro.max()
# normalising Net Production
data_netpro = np.array(data_netpro)
data_netpro = data_netpro/max_netpro
```

In [24]:
```
plt.scatter(years, data_netpro, color='green')
plt.title('Year vs Net Production 2000-2014')
plt.ylabel('Production')
plt.xlabel('Year')
plt.xticks(np.arange(2000, 2014, 2))
plt.grid(True)
plt.show()
```

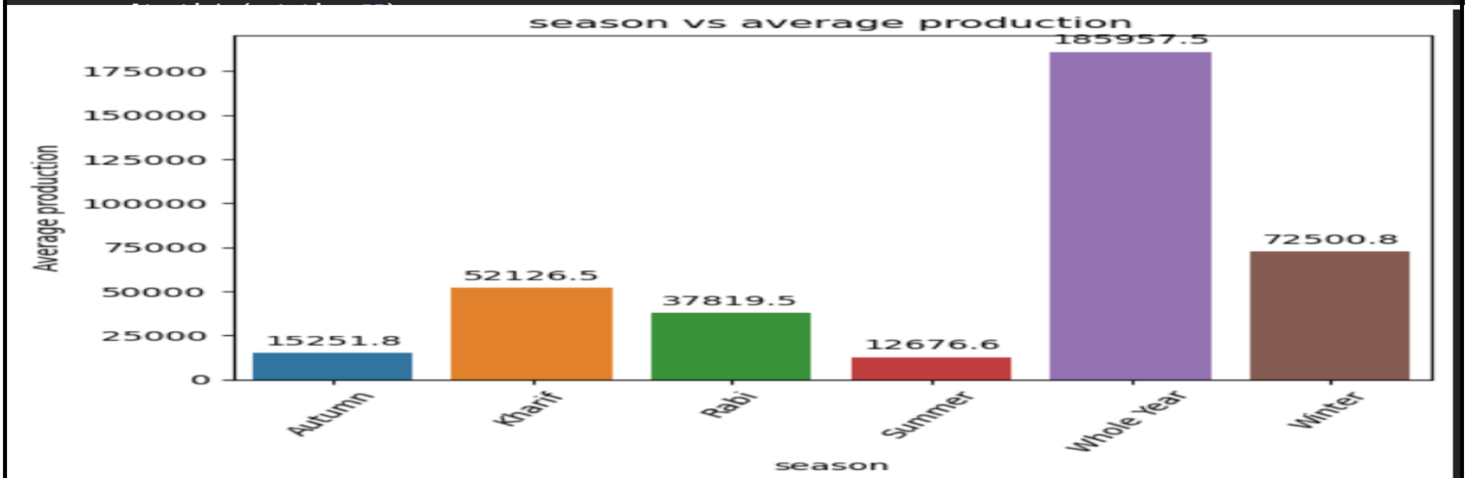


In [26]:
```
# season vs average production
season_production=df1.groupby('Season')['Production'].mean()
season_production
```

```
Season
Autumn          15251.835446
Kharif          52126.460218
Rabi            37819.545581
Summer          12676.573566
Whole Year     185957.467642
Winter          72500.801536
Name: Production, dtype: float64
```

In [27]:
```
# plot bar
ax = sns.barplot(x='Season', y='Production', data=season_production.reset_index())
plt.xlabel('season')
plt.ylabel('Average production')
plt.title('season vs average production')

for p in ax.patches:
    ax.annotate(format(p.get_height(), '.1f'),
                (p.get_x() + p.get_width() / 2., p.get_height()),
                ha='center', va='center',
                xytext=(0, 9),
                textcoords='offset points')
```



```
In [28]:
# make model to predict and analysis
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error
```

```
In [29]:
df1.head(3)
```

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production
0	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Arecanut	1254.0	2000.0
1	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Other Kharif pulses	2.0	1.0
2	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Rice	102.0	321.0

```
In [30]:
# Data Preprocessing
# Encode categorical variables
data_encoded = pd.get_dummies(df.drop(columns=['State_Name', 'District_Name', 'Crop_Year', 'Crop']), column:
X = data_encoded.drop(columns=['Production'])
y = data_encoded['Production']
```

```
In [31]:
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
In [32]:
# Model Training
model = RandomForestRegressor(random_state=42)
model.fit(X_train, y_train)
```

```
RandomForestRegressor(random_state=42)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.

On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
In [33]:
# Model Evaluation
train_preds = model.predict(X_train)
test_preds = model.predict(X_test)
```

```
In [36]:
train_rmse = mean_squared_error(y_train, train_preds)
test_rmse = mean_squared_error(y_test, test_preds)
print("Train RMSE:", train_rmse)
print("Test RMSE:", test_rmse)
```

```
Train RMSE: 529783819390l1.125
Test RMSE: 207551213835908.3
```

```
In [37]:
# Prediction
new_data = X_test.iloc[[0]]
prediction = model.predict(new_data)
print("Predicted Production:", prediction)
```

```
Predicted Production: [621.72964762]
```

Dashboard:

CROP PRODUCTION OF INDIA
Season
All
Crop_Year
All
Performance Indicator
141.18bn
Production
246K
Total crop
2.95bn
TotalArea
Production Vs Year
Production by year
5.9bn
5.6bn
6.0bn
6.2bn
5.9bn
4.9bn
4.6bn
0.0bn
5bn
0bn
Production
1995
2000
2005
2010
2015
year
Top 10 State wise Production
Production by State_Name
Production 0.59bn
97.88bn
49.23bn
Kerala 98bn
Andhra... 17bn
Tamil ... 12bn
Uttar P... 3bn
Assam 2bn
West B... 1bn
Mahar... 1bn
Karnat... 1bn
Andam... 1bn
Punjab 1bn
State
0bn 50bn 100bn
Production
TotalArea and TotalProduction
Total Area and Production
Total area and T...
Total Area
Production
Top 10 District wise Production
Sum of Production by District_Name
15bn 15bn 10bn 10bn 10bn 8bn 8bn 7bn 6bn 5bn
10bn
0bn
Sum of Produc...
KOZHIKODE
MALAPPU...
THIRUVA...
THRISSUR
KANNUR
EAST GOD...
KASARAG...
KOLLAM
PALAKKAD
ERNAKUL...
State
Production of state of India
Production by State
AFGHANISTAN
PAKISTAN
INDIA
MYANMAR
THAILAND
©2024 OSM ©2024 TomTom
Microsoft Azure

Future Scope:

- **Big Data Integration**: Harness the power of big data analytics by integrating data from multiple sources such as satellite imagery, weather forecasts, soil health data, and market trends to develop comprehensive decision support systems for farmers and policymakers.
- **Precision Agriculture**: Embrace precision agriculture technologies including remote sensing, drones, and IoT sensors to enable real-time monitoring of crops, soil conditions, and weather patterns, leading to more efficient resource utilization and sustainable crop management practices.
- **Climate Resilience**: Develop strategies and technologies to enhance the resilience of crops to climate change-induced stresses such as drought, floods, and extreme temperatures, including the breeding of climate-resilient crop varieties and the adoption of climate-smart agricultural practices.
- **Digital Agriculture Platforms**: Invest in the development of digital agriculture platforms and mobile applications that provide farmers with access to timely information, advisory services, market linkages, and financial services, empowering them to make informed decisions and improve productivity.
- **Vertical Farming and Urban Agriculture:** Explore the potential of vertical farming, hydroponics, and urban agriculture to address urban food security challenges and reduce the environmental footprint of agriculture by maximizing crop yields in limited space and reducing transportation costs.
- **Sustainable Supply Chains:** Strengthen sustainable supply chains by promoting traceability, transparency, and ethical sourcing practices, ensuring fair remuneration for farmers, and minimizing the environmental impact of agricultural production and distribution.
- **Capacity Building and Extension Services:** Invest in capacity building programs, vocational training, and extension services to equip farmers, agri-entrepreneurs, and rural youth with the skills and knowledge needed to adopt modern agricultural practices, agribusiness management, and value chain development.
- **Policy Support and Regulatory Framework**: Enact supportive policies, regulatory frameworks, and incentives that foster innovation, investment, and

entrepreneurship in the agricultural sector, while ensuring social equity, environmental sustainability, and food sovereignty.

Conclusion:

the analysis of crop production in India reveals key insights into production trends, regional variations, and predictive modeling. By leveraging data-driven analytics and machine learning, we've gained a deeper understanding of agricultural dynamics. Looking ahead, embracing innovation and evidence-based strategies can drive sustainable growth and resilience in India's agricultural sector.

In essence, the analysis presented in this report serves as a foundation for informed decision-making, collaboration, and collective action towards a resilient, productive, and equitable agricultural sector in India. By harnessing the power of data and technology, we can address the challenges facing agriculture and unlock new opportunities for growth, prosperity, and resilience in the years to come.