AGRICULTURE PRODUCTION PREDICTION USING LINEAR REGRESSION

Project: Rice Production Trend Analysis (Mini Regression Project)

Tools Used: Python, Pandas, Matplotlib, scikit-learn

Dataset: World Food Production (CSV format)

Name: Janani D

Date: May 2025

1. Introduction

This project explores rice production trends over the years using linear regression. The aim is to understand how rice production has evolved globally and predict future values using a simple but powerful machine learning model.

2. Tools & Technologies

- Python
- Pandas
- Matplotlib
- Scikit-learn
- Jupyter Notebook

2. Dataset Description

The dataset used for this project includes food production statistics from various countries and years. The key columns are:

- year: Represents the year of production
- country: Name of the country
- rice production: Quantity of rice produced
- wheat production: Quantity of wheat produced
- vegetable production: Quantity of vegetables produced

The dataset may also contain other numerical columns representing production data from specific years.

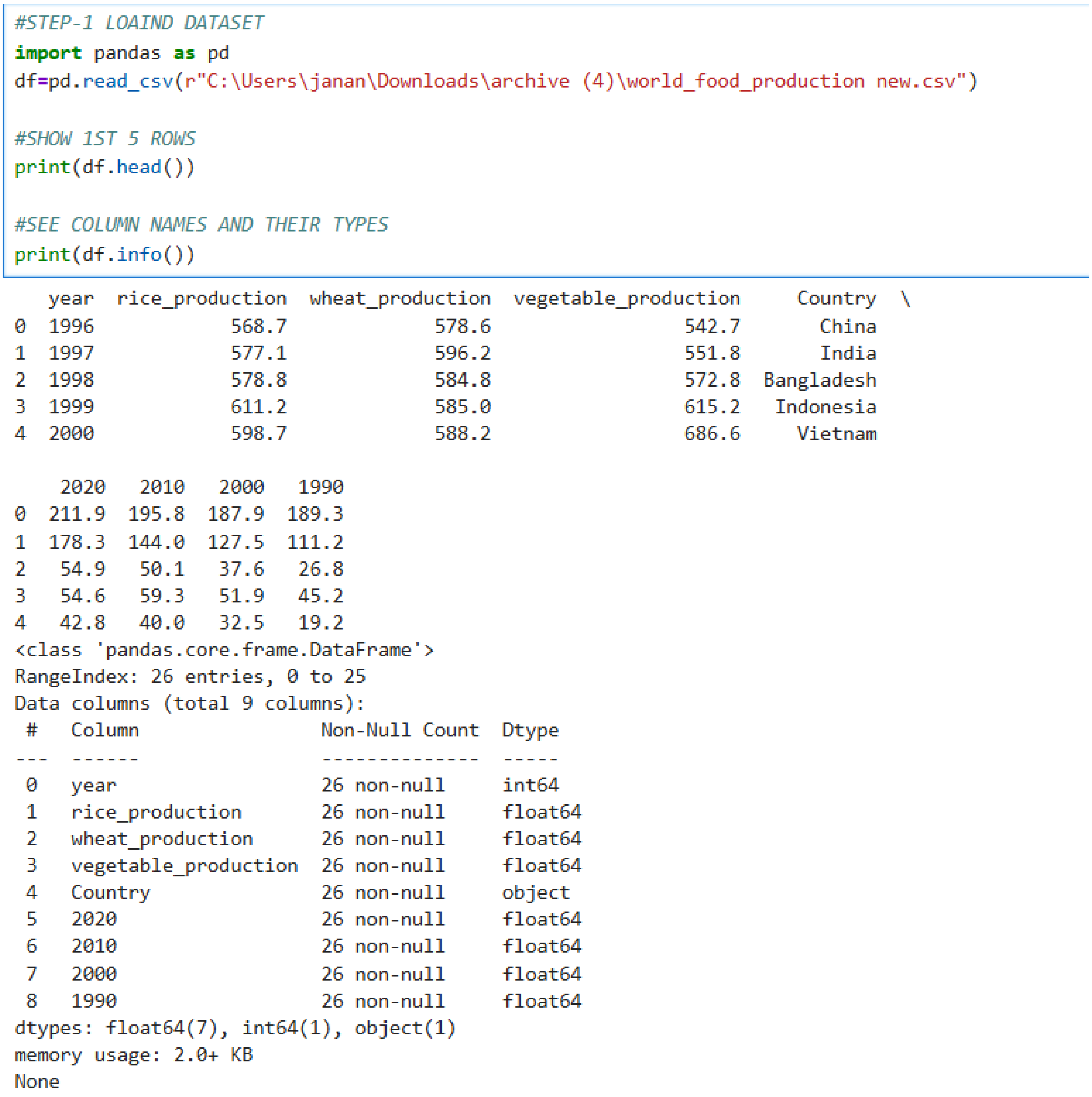
3. Methodology

1. **Data Loading & Cleaning:**
    - Imported the CSV file using Pandas
    - Handled missing values by removing incomplete rows
2. **Feature Selection:**
    - Selected year as the independent variable (X)
    - Selected rice production as the dependent variable (y)
3. **Model Building:**
    - Applied Linear Regression using scikit-learn
    - Trained the model on the data
4. **Visualization:**
    - Created a scatter plot of actual rice production values
    - Plotted the regression line to visualize the trend

4. Insights

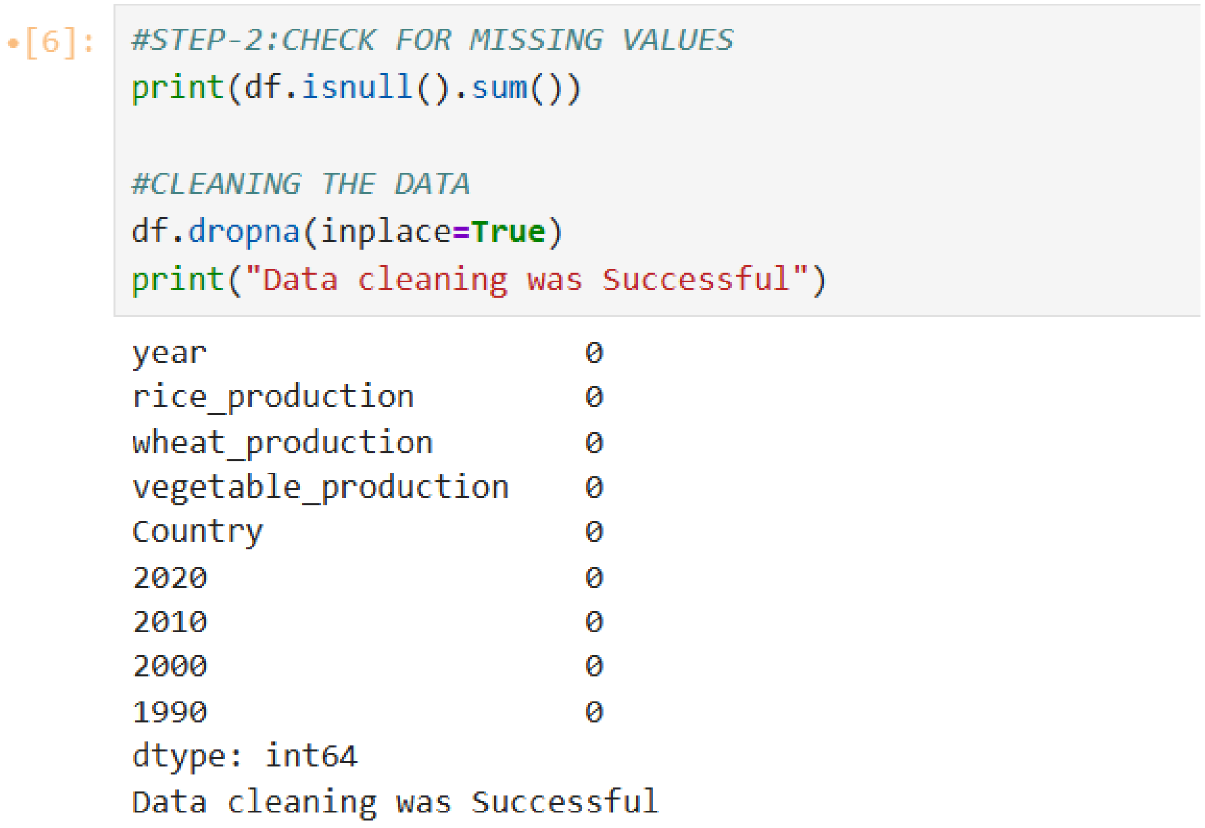
- Rice production has shown a steady increase over the years.
- The regression line indicates a positive upward trend.
- The model can estimate future production levels, such as predicting production for the year 2024-2030.
- Similar analysis can be applied to wheat and vegetable production trends

5.Visualisations:



```
#STEP-1 LOAIND DATASET
import pandas as pd
df=pd.read_csv(r"C:\Users\janan\Downloads\archive (4)\world_food_production new.csv")

#SHOW 1ST 5 ROWS
print(df.head())

#SEE COLUMN NAMES AND THEIR TYPES
print(df.info())
```

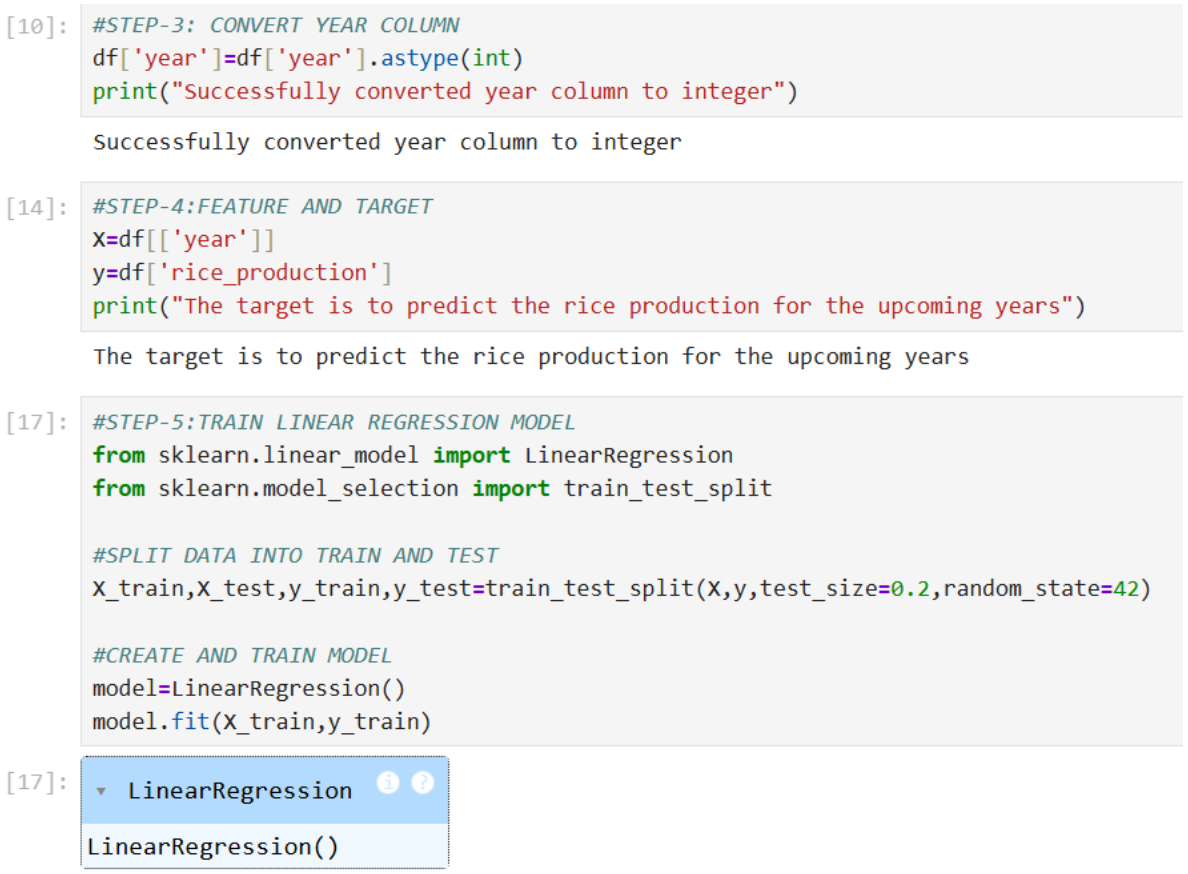
```
   year  rice_production  wheat_production  vegetable_production     Country  \
0  1996            568.7             578.6                 542.7       China   
1  1997            577.1             596.2                 551.8       India   
2  1998            578.8             584.8                 572.8  Bangladesh   
3  1999            611.2             585.0                 615.2   Indonesia   
4  2000            598.7             588.2                 686.6     Vietnam   

    2020   2010   2000   1990  
0  211.9  195.8  187.9  189.3  
1  178.3  144.0  127.5  111.2  
2   54.9   50.1   37.6   26.8  
3   54.6   59.3   51.9   45.2  
4   42.8   40.0   32.5   19.2  
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26 entries, 0 to 25
Data columns (total 9 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   year                  26 non-null     int64  
 1   rice_production       26 non-null     float64
 2   wheat_production      26 non-null     float64
 3   vegetable_production  26 non-null     float64
 4   Country               26 non-null     object 
 5   2020                  26 non-null     float64
 6   2010                  26 non-null     float64
 7   2000                  26 non-null     float64
 8   1990                  26 non-null     float64
dtypes: float64(7), int64(1), object(1)
memory usage: 2.0+ KB
None
```



•[6]:

```python
#STEP-2:CHECK FOR MISSING VALUES
print(df.isnull().sum())

#CLEANING THE DATA
df.dropna(inplace=True)
print("Data cleaning was Successful")
```

```
year                    0
rice_production         0
wheat_production        0
vegetable_production    0
Country                 0
2020                    0
2010                    0
2000                    0
1990                    0
dtype: int64
Data cleaning was Successful
```



[10]:

```python
#STEP-3: CONVERT YEAR COLUMN
df['year']=df['year'].astype(int)
print("Successfully converted year column to integer")
```

```
Successfully converted year column to integer
```

[14]:

```python
#STEP-4:FEATURE AND TARGET
X=df[['year']]
y=df['rice_production']
print("The target is to predict the rice production for the upcoming years")
```

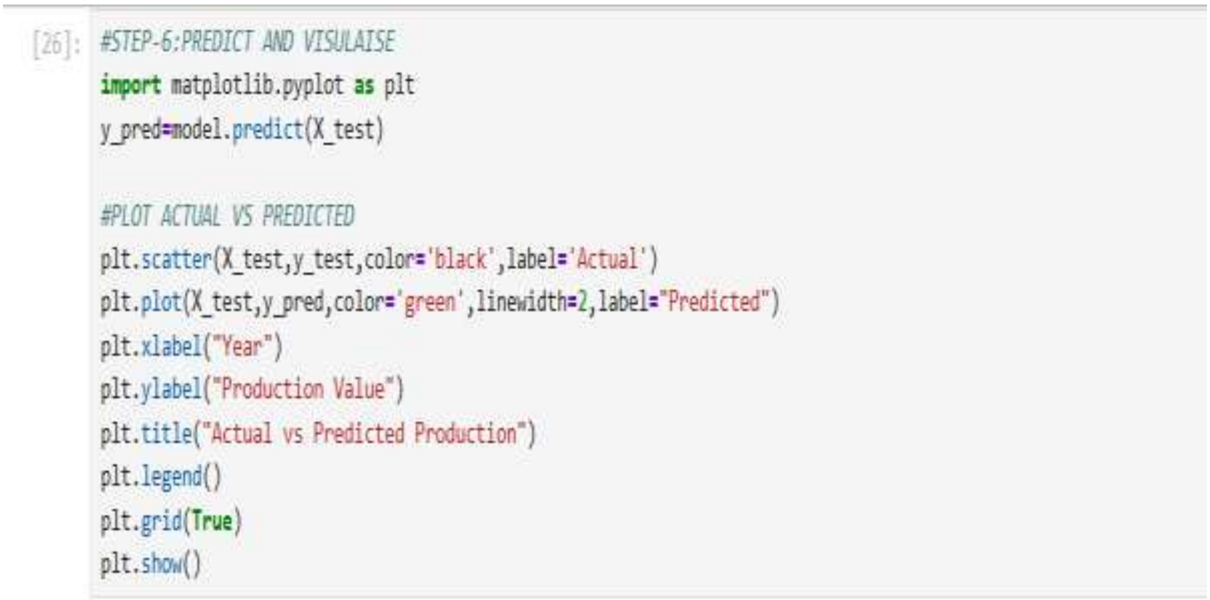
```
The target is to predict the rice production for the upcoming years
```

[17]:

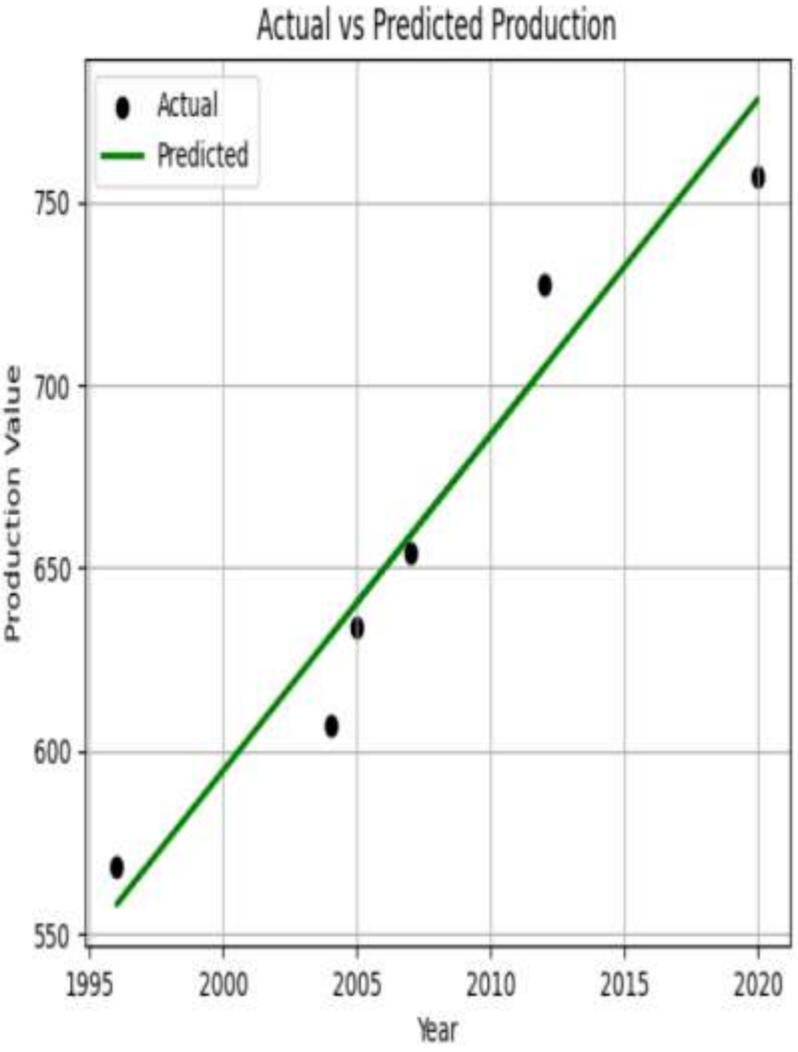
```python
#STEP-5:TRAIN LINEAR REGRESSION MODEL
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

#SPLIT DATA INTO TRAIN AND TEST
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=42)

#CREATE AND TRAIN MODEL
model=LinearRegression()
model.fit(X_train,y_train)
```

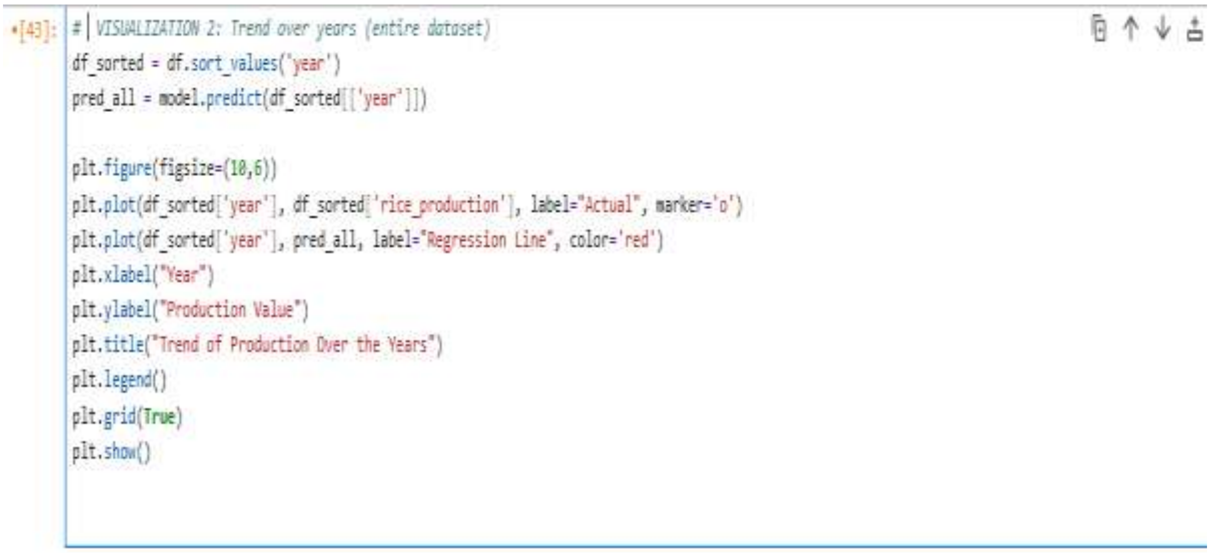
[17]:

```
▾ LinearRegression
LinearRegression()
```

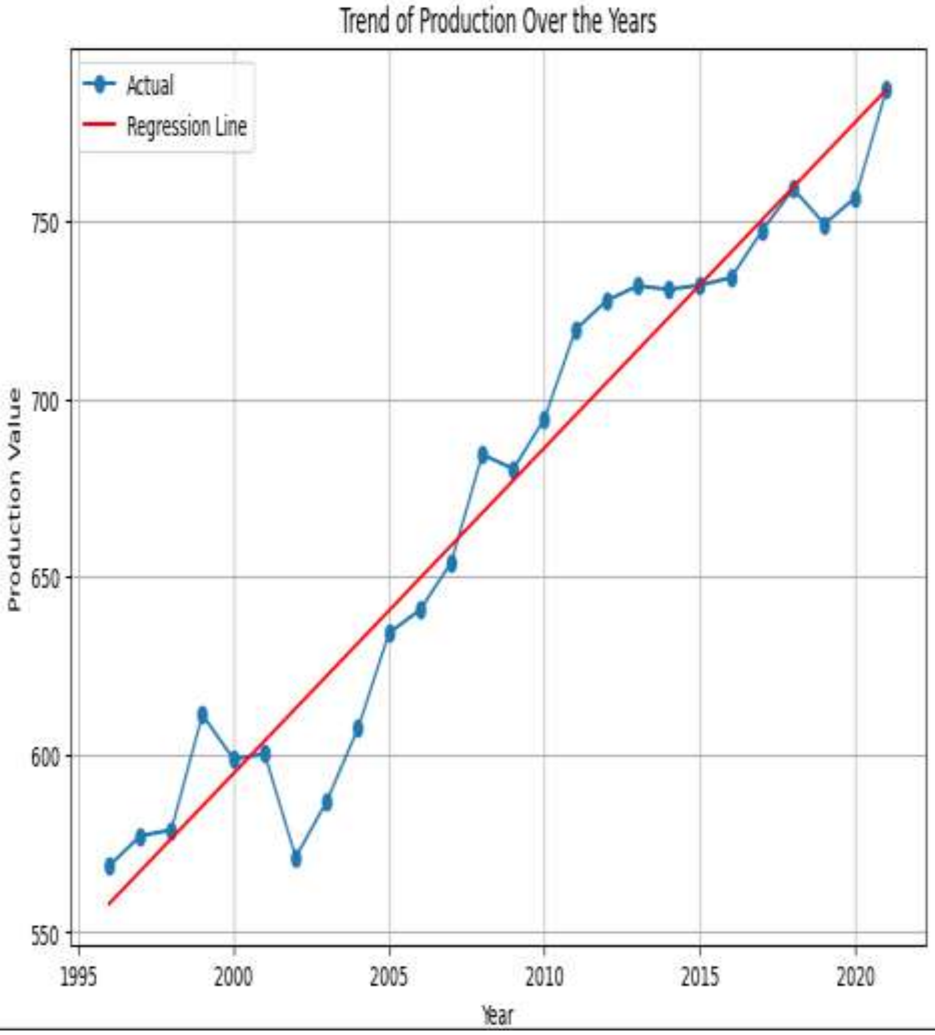


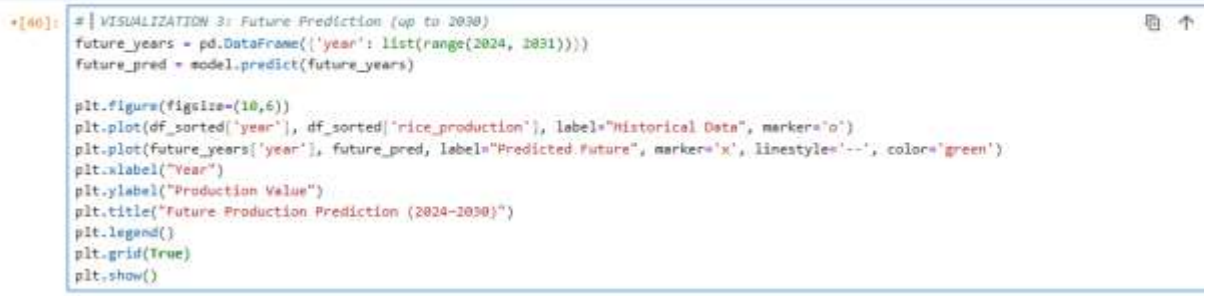
```
#STEP-6:PREDICT AND VISULAISE
import matplotlib.pyplot as plt
y_pred=model.predict(X_test)

#PLOT ACTUAL VS PREDICTED
plt.scatter(X_test,y_test,color='black',label='Actual')
plt.plot(X_test,y_pred,color='green',linewidth=2,label="Predicted")
plt.xlabel("Year")
plt.ylabel("Production Value")
plt.title("Actual vs Predicted Production")
plt.legend()
plt.grid(True)
plt.show()
```



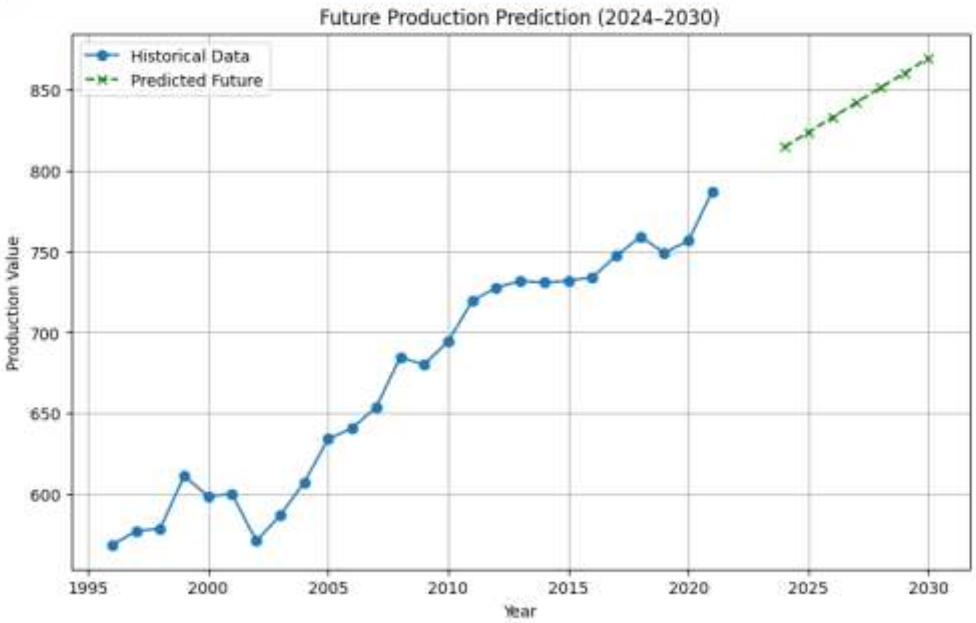


```
•[43]: # VISUALIZATION 2: Trend over years (entire dataset)
       df_sorted = df.sort_values('year')
       pred_all = model.predict(df_sorted[['year']])

       plt.figure(figsize=(10,6))
       plt.plot(df_sorted['year'], df_sorted['rice_production'], label="Actual", marker='o')
       plt.plot(df_sorted['year'], pred_all, label="Regression Line", color='red')
       plt.xlabel("Year")
       plt.ylabel("Production Value")
       plt.title("Trend of Production Over the Years")
       plt.legend()
       plt.grid(True)
       plt.show()
```





```
•[46]: # VISUALIZATION 3: Future Prediction (up to 2030)
future_years = pd.DataFrame({'year': list(range(2024, 2031))})
future_pred = model.predict(future_years)

plt.figure(figsize=(10,6))
plt.plot(df_sorted['year'], df_sorted['rice_production'], label="Historical Data", marker='o')
plt.plot(future_years['year'], future_pred, label="Predicted Future", marker='x', linestyle='--', color='green')
plt.xlabel("Year")
plt.ylabel("Production Value")
plt.title("Future Production Prediction (2024-2030)")
plt.legend()
plt.grid(True)
plt.show()
```



6.Conclusion

This mini-project successfully demonstrates how linear regression can be used to analyze historical data and predict future values. Such analysis is vital in agriculture planning, resource management, and policy-making. As a data analyst, understanding and applying regression techniques like this is a crucial skill.