climate_change_impact_on_agriculture_2024

In [5]:
```
import pandas as pd
import numpy as np

import warnings
warnings.filterwarnings("ignore")
```

Pandas:

Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data.

Numpy:

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices.

Warnings:

In Python, you can use the warnings module from the standard library to control warnings, such as ignoring (suppressing) warnings or turning matching warnings into exceptions.

Read CSV Files

A simple way to store big data sets is to use CSV files (comma separated files).

In [46]:
```
ccia_ab = pd.read_csv("ccia.csv")
print(ccia_ab)
```

```
      Year    Country             Region  Crop_Type  Average_Temperature_C  \
0     2001      India         West Bengal       Corn                   1.55
1     2024      China               North       Corn                   3.23
2     2001     France       Ile-de-France      Wheat                  21.11
3     2001     Canada            Prairies     Coffee                  27.85
4     1998      India          Tamil Nadu  Sugarcane                   2.19
...    ...        ...                 ...        ...                    ...
9995  2022     France  Nouvelle-Aquitaine     Cotton                  30.48
9996  1999  Australia          Queensland   Soybeans                   9.53
9997  2000  Argentina           Patagonia     Coffee                  31.92
9998  1996     Brazil           Southeast   Soybeans                  13.95
9999  2015      China               South       Corn                  11.78

      Total_Precipitation_mm  CO2_Emissions_MT  Crop_Yield_MT_per_HA  \
0                     447.06             15.22                 1.737
1                    2913.57             29.82                 1.737
2                    1301.74             25.75                 1.719
3                    1154.36             13.91                 3.890
4                    1627.48             11.81                 1.080
...                      ...               ...                   ...
9995                  685.93             17.64                 3.033
9996                 2560.38             10.68                 2.560
9997                  357.76             26.01                 1.161
9998                 1549.52             17.31                 3.348
9999                 1676.25              5.34                 3.710

      Extreme_Weather_Events  Irrigation_Access_%  Pesticide_Use_KG_per_HA  \
0                          8                14.54                    10.08
1                          8                11.05                    33.06
2                          5                84.42                    27.41
3                          5                94.06                    14.38
4                          9                95.75                    44.35
...                      ...                  ...                      ...
9995                       9                27.56                    41.96
9996                       4                77.02                     5.45
9997                      10                78.53                    11.94
9998                       2                42.65                    44.71
9999                       5                46.41                    48.28

      Fertilizer_Use_KG_per_HA  Soil_Health_Index Adaptation_Strategies  \
0                        14.78              83.25      Water Management
1                        23.25              54.02         Crop Rotation
2                        65.53              67.78      Water Management
3                        87.58              91.39         No Adaptation
4                        88.08              49.61         Crop Rotation
...                        ...                ...                   ...
9995                     10.95              43.41         No Adaptation
9996                     82.32              59.39         No Adaptation
9997                     26.00              41.46      Water Management
9998                     25.07              75.10         Crop Rotation
9999                     98.27              59.38      Water Management

      Economic_Impact_Million_USD
0                          808.13
1                          616.22
2                          796.96
3                          790.32
4                          401.72
...                           ...
9995                      1483.06
9996                       829.61
9997                       155.99
9998                      1613.90
9999                       453.14

[10000 rows x 15 columns]
```

Quick Checking DataFrames:

-> .head() -> .tail() -> .sample() -> .info() -> .describe()

In [47]:
```
ccia_ab.head()
```

Out[47]:

	Year	Country	Region	Crop_Type	Average_Temperature_C	Total_Precipitation_mm	CO2_Emissions_MT	Crop_Yield_MT_per_HA
0	2001	India	West Bengal	Corn	1.55	447.06	15.22	1.73
1	2024	China	North	Corn	3.23	2913.57	29.82	1.73
2	2001	France	Ile-de-France	Wheat	21.11	1301.74	25.75	1.71
3	2001	Canada	Prairies	Coffee	27.85	1154.36	13.91	3.89
4	1998	India	Tamil Nadu	Sugarcane	2.19	1627.48	11.81	1.08

In [7]:
```
ccia_ab.tail()
```

Out[7]:

	Year	Country	Region	Crop_Type	Average_Temperature_C	Total_Precipitation_mm	CO2_Emissions_MT	Crop_Yield_MT
9995	2022	France	Nouvelle-Aquitaine	Cotton	30.48	685.93	17.64	
9996	1999	Australia	Queensland	Soybeans	9.53	2560.38	10.68	
9997	2000	Argentina	Patagonia	Coffee	31.92	357.76	26.01	
9998	1996	Brazil	Southeast	Soybeans	13.95	1549.52	17.31	
9999	2015	China	South	Corn	11.78	1676.25	5.34	

In [42]:
```
ccia_ab.sample(10)
```

Out[42]:

	Year	Country	Region	Crop_Type	Average_Temperature_C	Total_Precipitation_mm	CO2_Emissions_MT	Crop_Yield_MT
923	1996	France	Nouvelle-Aquitaine	Vegetables	3.97	793.13	4.16	
7422	2020	France	Grand Est	Sugarcane	21.31	1022.10	29.61	
5187	2010	India	Maharashtra	Soybeans	11.48	1833.56	23.84	
3967	2022	India	Tamil Nadu	Wheat	29.50	2034.73	27.44	
9281	1996	China	East	Sugarcane	30.76	1167.20	11.37	
9138	1999	Brazil	South	Wheat	29.84	993.70	16.63	
3374	2006	Canada	British Columbia	Sugarcane	0.66	2977.20	27.00	
3522	2000	Nigeria	South West	Corn	5.10	1099.87	9.84	
3460	1997	Canada	Quebec	Wheat	24.60	1003.84	12.39	
2925	1997	Nigeria	South East	Cotton	27.30	1906.36	0.95	

In [43]:
```
ccia_ab.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 15 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   Year                         10000 non-null  int64  
 1   Country                      10000 non-null  object 
 2   Region                       10000 non-null  object 
 3   Crop_Type                    10000 non-null  object 
 4   Average_Temperature_C        10000 non-null  float64
 5   Total_Precipitation_mm       10000 non-null  float64
 6   CO2_Emissions_MT             10000 non-null  float64
 7   Crop_Yield_MT_per_HA         10000 non-null  float64
 8   Extreme_Weather_Events       10000 non-null  int64  
 9   Irrigation_Access_%          10000 non-null  float64
 10  Pesticide_Use_KG_per_HA      10000 non-null  float64
 11  Fertilizer_Use_KG_per_HA     10000 non-null  float64
 12  Soil_Health_Index            10000 non-null  float64
 13  Adaptation_Strategies        10000 non-null  object 
 14  Economic_Impact_Million_USD  10000 non-null  float64
dtypes: float64(9), int64(2), object(4)
memory usage: 1.1+ MB
```

In [44]:
```
ccia_ab.describe()
```

Out[44]:

	Year	Average_Temperature_C	Total_Precipitation_mm	CO2_Emissions_MT	Crop_Yield_MT_per_HA	Extreme_Weath
count	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	100
mean	2007.088700	15.241299	1611.663834	15.246608	2.240017	
std	10.084245	11.466955	805.016815	8.589423	0.998342	
min	1990.000000	-4.990000	200.150000	0.500000	0.450000	
25%	1999.000000	5.430000	925.697500	7.760000	1.449000	
50%	2007.000000	15.175000	1611.160000	15.200000	2.170000	
75%	2016.000000	25.340000	2306.997500	22.820000	2.930000	
max	2024.000000	35.000000	2999.670000	30.000000	5.000000	

Data Pre-Processing:

Data preprocessing is a crucial step in any data analysis or machine learning project. It involves cleaning and transforming raw data into a format that is more suitable for analysis. In Python, pandas is a popular library used for data manipulation and preprocessing.

In [8]:
```
ccia_ab.shape
```

Out[8]:
```
(10000, 15)
```

In [9]:
```
ccia_ab.columns
```

Out[9]:
```
Index(['Year', 'Country', 'Region', 'Crop_Type', 'Average_Temperature_C',
       'Total_Precipitation_mm', 'CO2_Emissions_MT', 'Crop_Yield_MT_per_HA',
       'Extreme_Weather_Events', 'Irrigation_Access_%',
       'Pesticide_Use_KG_per_HA', 'Fertilizer_Use_KG_per_HA',
       'Soil_Health_Index', 'Adaptation_Strategies',
       'Economic_Impact_Million_USD'],
      dtype='object')
```

In [12]:
```
ccia_ab.isnull().sum()
```

Out[12]:
```
Year                          0
Country                       0
Region                        0
Crop_Type                     0
Average_Temperature_C         0
Total_Precipitation_mm        0
CO2_Emissions_MT              0
Crop_Yield_MT_per_HA          0
Extreme_Weather_Events        0
Irrigation_Access_%           0
Pesticide_Use_KG_per_HA       0
Fertilizer_Use_KG_per_HA      0
Soil_Health_Index             0
Adaptation_Strategies         0
Economic_Impact_Million_USD   0
dtype: int64
```

In [14]:
```
ccia_ab.describe(include='object')
```

Out[14]:

	Country	Region	Crop_Type	Adaptation_Strategies
count	10000	10000	10000	10000
unique	10	34	10	5
top	Australia	South	Wheat	Water Management
freq	1032	754	1047	2049

In [15]:
```
sel_col = ccia_ab[['Year', 'Country', 'Crop_Type']]
sel_col.head()
```

Out[15]:

	Year	Country	Crop_Type
0	2001	India	Corn
1	2024	China	Corn
2	2001	France	Wheat
3	2001	Canada	Coffee
4	1998	India	Sugarcane

In [16]:
```
ccia_ab = ccia_ab.drop_duplicates()
```

```
ccia_ab.shape
```

Out[16]: (10000, 15)

In [19]:
```
ccia_ab.shape[0]
```

Out[19]: 10000

Data Visulization:

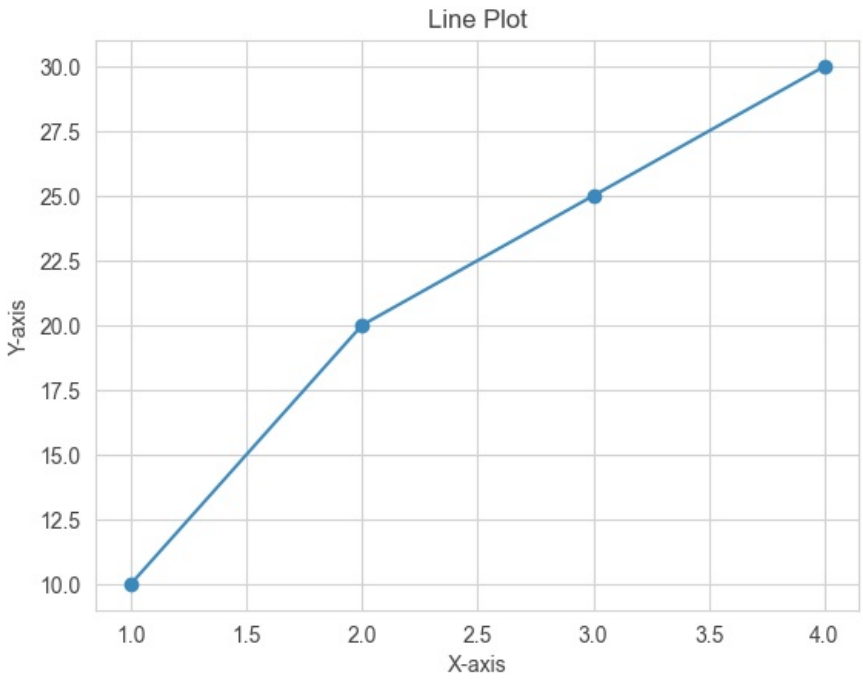
Data visualization is a key aspect of data analysis that helps to understand and interpret data better by presenting it in graphical form. In Python, several libraries can be used for data visualization, with matplotlib, seaborn, and plotly being among the most popular.

In [20]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib

%matplotlib inline

sns.set_style("whitegrid")
```

Lineplot:

A line plot (or line graph) is a basic yet powerful tool used in data visualization to represent data points connected by straight lines. It's particularly useful for showing trends and changes over time or across different categories.

In [60]:
```
plt.plot([1, 2, 3, 4], [10, 20, 25, 30], marker='o')
plt.title('Line Plot')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.grid(True)
plt.show()
```

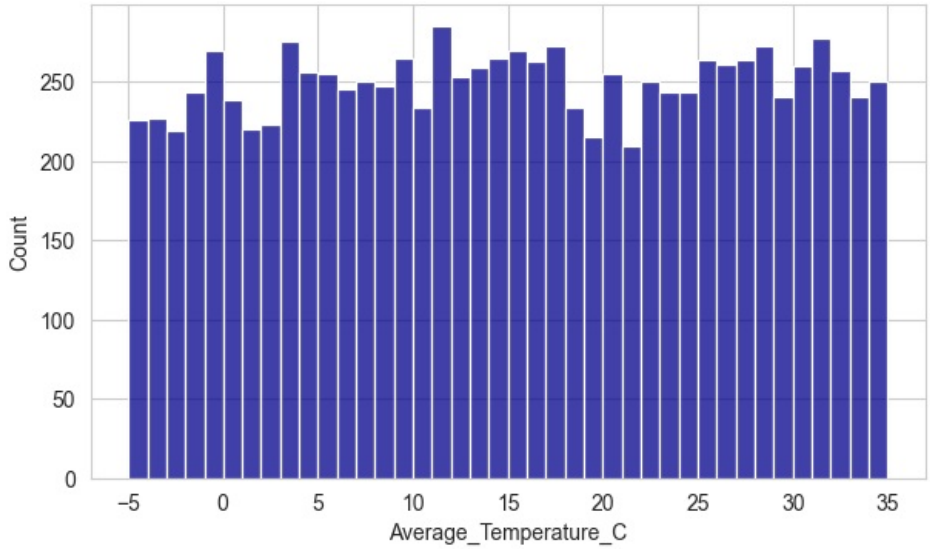


Histplot:

A histogram plot, or histplot in some libraries like Seaborn, is a type of data visualization used to represent the distribution of a continuous variable by dividing the data into bins or intervals and counting the number of observations that fall into each bin. This allows you to see the frequency distribution and overall shape of the data.

In [57]:
```
plt.figure(figsize=(7,4))

sns.histplot(x='Average_Temperature_C', data=ccia_ab, color="darkblue", binwidth=1)
plt.show()
```

<Figure size 700x400 with 0 Axes>



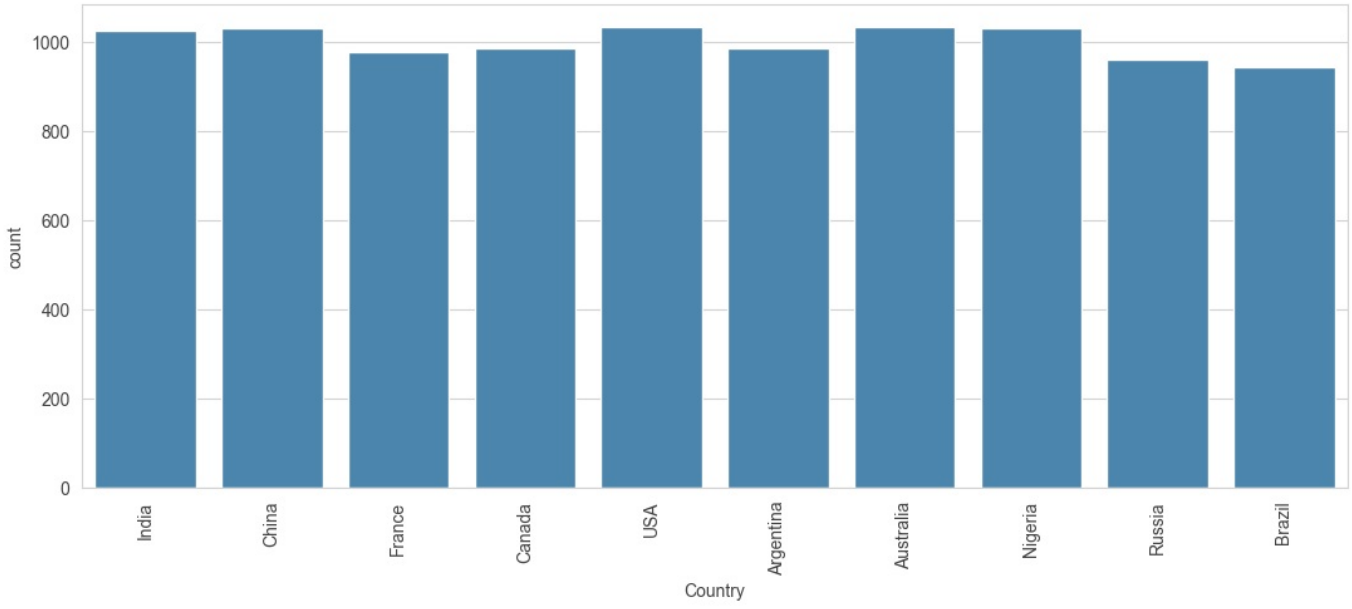
countplot:

A countplot is a type of plot commonly used in data visualization to show the count of observations in each categorical bin or category. It is particularly useful for displaying the frequency distribution of categorical data.

In [24]:
```
plt.figure(figsize=(13,5))

sns.countplot(x='Country', data=ccia_ab)
plt.xticks(rotation=90)
```

Out[24]:
```
([0, 1, 2, 3, 4, 5, 6, 7, 8, 9],
 [Text(0, 0, 'India'),
  Text(1, 0, 'China'),
  Text(2, 0, 'France'),
  Text(3, 0, 'Canada'),
  Text(4, 0, 'USA'),
  Text(5, 0, 'Argentina'),
  Text(6, 0, 'Australia'),
  Text(7, 0, 'Nigeria'),
  Text(8, 0, 'Russia'),
  Text(9, 0, 'Brazil')])
```

In [25]:
```
plt.show()
```



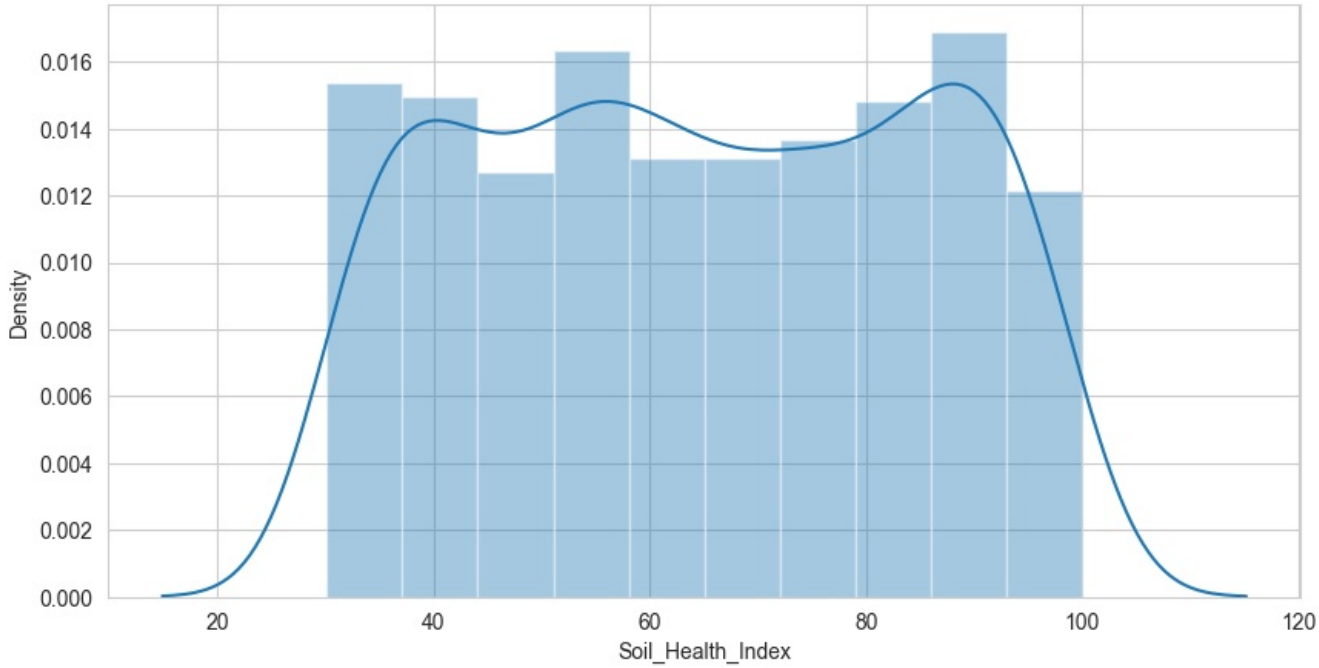
Distplot:

The distplot function was a popular tool in Seaborn for visualizing the distribution of a dataset.

In [28]:
```
plt.figure(figsize=(10,5))

sns.distplot(India_data['Soil_Health_Index'])
```

Out[28]:
```
<Axes: xlabel='Soil_Health_Index', ylabel='Density'>
```

In [29]:
```
plt.show()
```



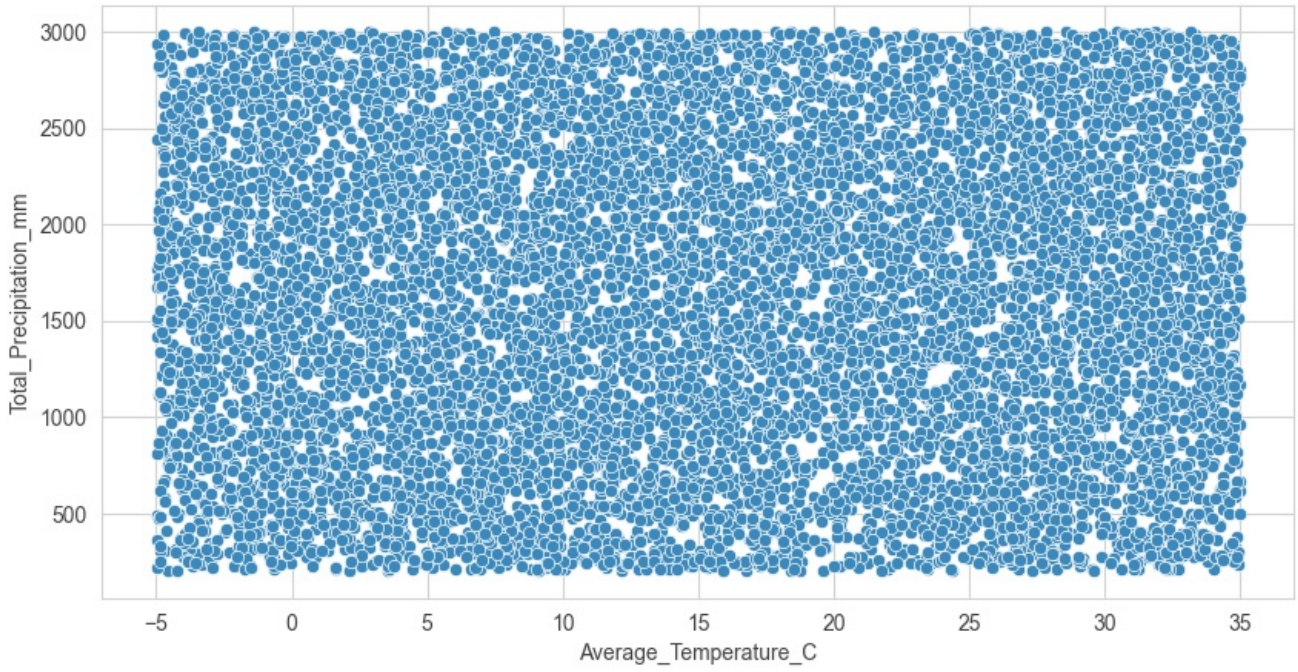
Scatterplot:

A scatterplot is a fundamental data visualization tool used to display the relationship between two continuous variables. Each point on the scatterplot represents a single observation, with its position determined by the values of the two variables.

In [30]:
```
plt.figure(figsize=(10,5))

sns.scatterplot(x='Average_Temperature_C', y='Total_Precipitation_mm', data=ccia_ab)
```

Out[30]:
```
<Axes: xlabel='Average_Temperature_C', ylabel='Total_Precipitation_mm'>
```

In [31]:
```
plt.show()
```



Conclusion:

The project provides valuable insights into agricultural trends and climate impact, aiding decision-makers in the agricultural sector. This project

focuses on developing machine learning models for predicting crop yields and assessing climate change impact on agriculture. It follows a multi-step process, including data collection, preprocessing, feature engineering, model selection, and evaluation.

Thank you