CROP YIELD PREDICTION IN INDIA

Predicting yield helps the state to get an estimate of the crop in a certain year to control the price rates.This model focuses on predicting the crop yield in advance by analyzing factors like location, season, and crop type through machine learning techniques on previously collected datasets.

In [1]:
```
# importing necessary libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [2]:
```
# Loading dataset

df=pd.read_csv(r"C:\Users\cool_adarsh\Downloads\crop_production-csv.csv")
df
```

Out[2]:

	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]:
```
# finding rows and columns

df.shape
```

Out[3]:
```
(246091, 7)
```

In [4]:
```
# dataset columns

df.columns
```

Out[4]:
```
Index(['State_Name', 'District_Name', 'Crop_Year', 'Season', 'Crop', 'Area',
       'Production'],
      dtype='object')
```

In [5]:
```
# statistical inference of the dataset

df.describe()
```

Out[5]:

	Crop_Year	Area	Production
count	246091.000000	2.460910e+05	2.423610e+05
mean	2005.643018	1.200282e+04	5.825034e+05
std	4.952164	5.052340e+04	1.706581e+07
min	1997.000000	4.000000e-02	0.000000e+00
25%	2002.000000	8.000000e+01	8.800000e+01
50%	2006.000000	5.820000e+02	7.290000e+02
75%	2010.000000	4.392000e+03	7.023000e+03
max	2015.000000	8.580100e+06	1.250800e+09

In [6]:
```
# Checking missing values of the dataset in each column
df.isnull().sum()
```

Out[6]:
```
State_Name          0
District_Name       0
Crop_Year           0
Season              0
Crop                0
Area                0
Production       3730
dtype: int64
```

In [7]:
```
# Dropping missing values
df = df.dropna()
df
```

Out[7]:

	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

242361 rows × 7 columns

In [8]:
```
#checking
df.isnull().values.any()
```

Out[8]:
```
False
```

In [9]:
```
# Displaying State Names present in the dataset

df.State_Name.unique()
```

Out[9]:
```
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 [19]:
```
# Adding a new column Yield which indicates Production per unit Area.

df['Yield'] = (df['Production'] / df['Area'])
df.head(10)
```

```
C:\Users\cool_adarsh\AppData\Local\Temp\ipykernel_3708\4144987489.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['Yield'] = (df['Production'] / df['Area'])
```

Out[19]:

	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production	Yield
0	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Arecanut	1254.0	2000.0	1.594896
1	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Other Kharif pulses	2.0	1.0	0.500000
2	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Rice	102.0	321.0	3.147059
3	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Banana	176.0	641.0	3.642045
4	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Cashewnut	720.0	165.0	0.229167
5	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Coconut	18168.0	65100000.0	3583.223250
6	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Dry ginger	36.0	100.0	2.777778
7	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Sugarcane	1.0	2.0	2.000000
8	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Sweet potato	5.0	15.0	3.000000
9	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Tapioca	40.0	169.0	4.225000

In [ ]:
```
# Visualizing the features

ax = sns.pairplot(df)
ax
```

In [20]:
```
# Dropping unnecessary columns

data = df.drop(['State_Name'], axis = 1)
data
```

Out[20]:

	District_Name	Crop_Year	Season	Crop	Area	Production	Yield
0	NICOBARS	2000	Kharif	Arecanut	1254.0	2000.0	1.594896
1	NICOBARS	2000	Kharif	Other Kharif pulses	2.0	1.0	0.500000
2	NICOBARS	2000	Kharif	Rice	102.0	321.0	3.147059
3	NICOBARS	2000	Whole Year	Banana	176.0	641.0	3.642045
4	NICOBARS	2000	Whole Year	Cashewnut	720.0	165.0	0.229167
...	...	...	...	...	...	...	...
246086	PURULIA	2014	Summer	Rice	306.0	801.0	2.617647
246087	PURULIA	2014	Summer	Sesamum	627.0	463.0	0.738437
246088	PURULIA	2014	Whole Year	Sugarcane	324.0	16250.0	50.154321
246089	PURULIA	2014	Winter	Rice	279151.0	597899.0	2.141848
246090	PURULIA	2014	Winter	Sesamum	175.0	88.0	0.502857

242361 rows × 7 columns

In [21]:
```
data.corr()
```

```
C:\Users\cool_adarsh\AppData\Local\Temp\ipykernel_3708\2627137660.py:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
  data.corr()
```

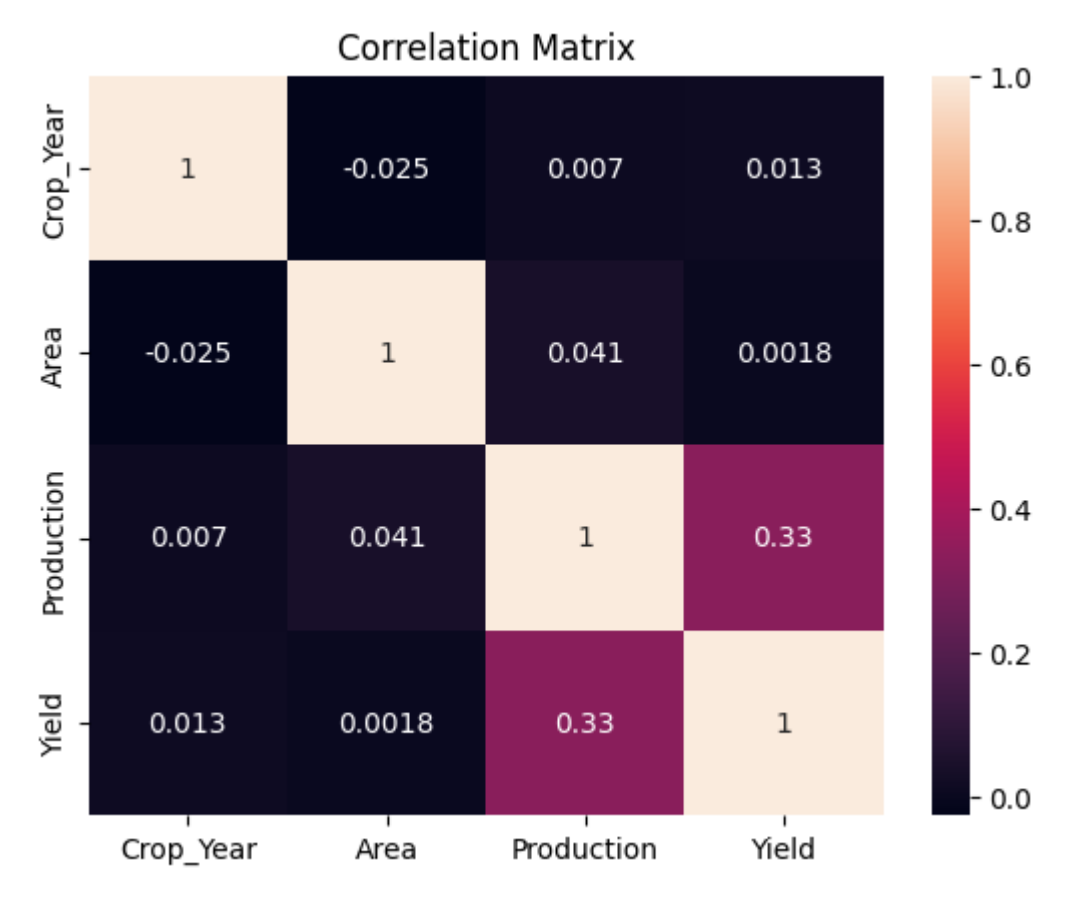
Out[21]:

	Crop_Year	Area	Production	Yield
Crop_Year	1.000000	-0.025305	0.006989	0.013499
Area	-0.025305	1.000000	0.040587	0.001822
Production	0.006989	0.040587	1.000000	0.330961
Yield	0.013499	0.001822	0.330961	1.000000

In [22]:
```
sns.heatmap(data.corr(), annot =True)
plt.title('Correlation Matrix')
```

```
C:\Users\cool_adarsh\AppData\Local\Temp\ipykernel_3708\1522484008.py:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.
  sns.heatmap(data.corr(), annot =True)
```

Out[22]:
```
Text(0.5, 1.0, 'Correlation Matrix')
```



In [23]:
```
dummy = pd.get_dummies(data)
dummy
```

Out[23]:

	Crop_Year	Area	Production	Yield	District_Name_24 PARAGANAS NORTH	District_Name_24 PARAGANAS SOUTH	District_Name_ADILABAD	District_Name_AGAR MALWA	District_Name_AGRA	District_Name_AHMADABAD	...	Crop_Turmeric	Crop_Turnip	Crop_Urad	Crop_Varagu	Crop_Water Melon
0	2000	1254.0	2000.0	1.594896	0	0	0	0	0	0	...	0	0	0	0	0
1	2000	2.0	1.0	0.500000	0	0	0	0	0	0	...	0	0	0	0	0
2	2000	102.0	321.0	3.147059	0	0	0	0	0	0	...	0	0	0	0	0
3	2000	176.0	641.0	3.642045	0	0	0	0	0	0	...	0	0	0	0	0
4	2000	720.0	165.0	0.229167	0	0	0	0	0	0	...	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
246086	2014	306.0	801.0	2.617647	0	0	0	0	0	0	...	0	0	0	0	0
246087	2014	627.0	463.0	0.738437	0	0	0	0	0	0	...	0	0	0	0	0
246088	2014	324.0	16250.0	50.154321	0	0	0	0	0	0	...	0	0	0	0	0
246089	2014	279151.0	597899.0	2.141848	0	0	0	0	0	0	...	0	0	0	0	0
246090	2014	175.0	88.0	0.502857	0	0	0	0	0	0	...	0	0	0	0	0

242361 rows × 780 columns

Splitting dataset into Training and Test Data

In [28]:
```
from sklearn.model_selection import train_test_split

x = dummy.drop(["Production","Yield"], axis = 1)
y = dummy["Production"]

# Splitting data set - 25% test dataset and 75%

x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.25, random_state=5)

print("x_train :",x_train.shape)
print("x_test :",x_test.shape)
print("y_train :",y_train.shape)
print("y_test :",y_test.shape)
```

```
x_train : (181770, 778)
x_test : (60591, 778)
y_train : (181770,)
y_test : (60591,)
```

In [29]:
```
print(x_train)
print(y_train)
```

```
        Crop_Year    Area  District_Name_24 PARAGANAS NORTH  \
201072       2013    16.0                                 0
191897       1998  5400.0                                 0
43814        2000  2968.0                                 0
32815        2013   211.0                                 0
62249        2006  1700.0                                 0
...           ...     ...                               ...
236131       2000   207.0                                 0
127145       2007    39.0                                 0
20536        2005    43.0                                 0
18709        2011  2489.0                                 0
35767        1999    67.0                                 0

        District_Name_24 PARAGANAS SOUTH  District_Name_ADILABAD  \
201072                                 0                       0
191897                                 0                       0
43814                                  0                       0
32815                                  0                       0
62249                                  0                       0
...                                  ...                     ...
236131                                 0                       0
127145                                 0                       0
20536                                  0                       0
18709                                  0                       0
35767                                  0                       0

        District_Name_AGAR MALWA  District_Name_AGRA  District_Name_AHMADABAD  \
201072                         0                   0                        0
191897                         0                   0                        0
43814                          0                   0                        0
32815                          0                   0                        0
62249                          0                   0                        0
...                          ...                 ...                      ...
236131                         0                   0                        0
127145                         0                   0                        0
20536                          0                   0                        0
18709                          0                   0                        0
35767                          0                   0                        0

        District_Name_AHMEDNAGAR  District_Name_AIZAWL  ...  Crop_Turmeric  \
201072                         0                     0  ...              0
191897                         0                     0  ...              0
43814                          0                     0  ...              0
32815                          0                     0  ...              0
62249                          0                     0  ...              0
...                          ...                   ...  ...            ...
236131                         0                     0  ...              0
127145                         0                     0  ...              0
20536                          0                     0  ...              0
18709                          0                     0  ...              0
35767                          0                     0  ...              0

        Crop_Turnip  Crop_Urad  Crop_Varagu  Crop_Water Melon  Crop_Wheat  \
201072            0          0            0                 0           0
191897            0          0            0                 0           0
43814             0          0            0                 0           0
32815             0          0            0                 0           0
62249             0          0            0                 0           0
...             ...        ...          ...               ...         ...
236131            0          0            0                 0           0
127145            0          0            0                 0           0
20536             0          0            0                 0           0
18709             0          0            0                 0           0
35767             0          0            0                 0           0

        Crop_Yam  Crop_other fibres  Crop_other misc. pulses  \
201072         0                  0                        0
191897         0                  0                        0
43814          0                  0                        0
32815          0                  0                        0
62249          0                  0                        0
...          ...                ...                      ...
236131         0                  0                        0
127145         0                  0                        0
20536          0                  0                        0
18709          0                  0                        0
35767          0                  0                        0

        Crop_other oilseeds
201072                    0
191897                    0
43814                     0
32815                     0
62249                     0
...                     ...
236131                    0
127145                    0
20536                     0
18709                     0
35767                     0

[181770 rows x 778 columns]
201072      11.0
191897    2000.0
43814     2555.0
32815      175.0
62249     1400.0
           ...
236131     139.0
127145      44.0
20536       27.0
18709     4779.0
35767       81.0
Name: Production, Length: 181770, dtype: float64
```

Decision Tree

In [31]:
```
# Training model
from sklearn.tree import DecisionTreeRegressor
regressor = DecisionTreeRegressor(random_state = 5)
regressor.fit(x_train,y_train)

# Predicting results
decisiontree_predict = regressor.predict(x_test)
decisiontree_predict
```

Out[31]:
```
array([3900.,  700., 2377., ...,   84., 9656.,  100.])
```

In [32]:
```
regressor.score(x_test,y_test)
```

Out[32]:
```
0.9564181540365296
```

In [33]:
```
# Calculating R2 score :

from sklearn.metrics import r2_score
r2 = r2_score(y_test,decisiontree_predict)
print("R2 score : ",r2)
```

```
R2 score :  0.9564181540365296
```

Random Forest

In [35]:
```
from sklearn.ensemble import RandomForestRegressor
model = RandomForestRegressor(n_estimators = 11)
model.fit(x_train,y_train)
rf_predict = model.predict(x_test)
rf_predict
```

Out[35]:
```
array([ 4160.8       ,   628.72727273,  2232.72727273, ...,
         209.83636364, 11117.        ,   118.18181818])
```

In [36]:
```
model.score(x_test,y_test)
```

Out[36]:
```
0.9671788850897047
```

In [37]:
```
# Calculating R2 score

from sklearn.metrics import r2_score
r1 = r2_score(y_test,rf_predict)
print("R2 score : ",r1)
```

```
R2 score :  0.9671788850897047
```

R2 score: This is pronounced as R-squared, and this score refers to the coefficient of determination.

This tells us how well the unknown samples will be predicted by our model.

In [ ]: