Project Title:

Predictive Shelter Demand Analysis for Disaster Preparedness using Historical Event Data

Problem Statement:

Natural disasters such as floods, cyclones, droughts, and earthquakes cause large-scale human displacement every year. One of the major challenges during these events is the absence of predictive systems to estimate shelter needs and evacuation requirements. This often results in delayed responses, inadequate facilities, and increased vulnerability of affected populations.

Description:

This project utilizes the Natural Disasters Emergency Events Database to analyze historical disaster patterns and apply machine learning techniques to predict shelter demand and identify high-risk regions. The outcomes will support policymakers and disaster management agencies in evacuation planning, infrastructure readiness, and sustainable disaster risk reduction strategies.

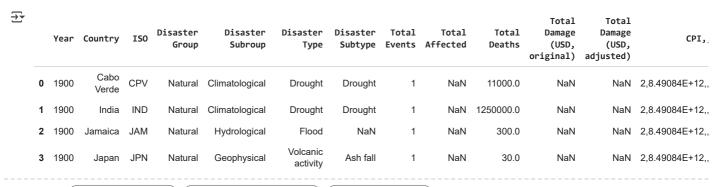
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split,cross_val_score, GridSearchCV
from sklearn.preprocessing import StandardScaler, LabelEncoder, MinMaxScaler
from sklearn.ensemble import RandomForestRegressor,GradientBoostingRegressor
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.tree import DecisionTreeRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.cluster import KMeans, DBSCAN
from sklearn.decomposition import PCA
import pickle
```

Load the dataset

```
df = pd.read_csv("natural_disasters.csv",sep=";")
```

Display first 5 rows

df.head()



New interactive sheet

Explore and understand the dataset

```
# Check structure
df.info()
```

Next steps:

<<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10431 entries, 0 to 10430
Data columns (total 13 columns):
Column No

Generate code with df

Data	columns (total 13 columns):		
#	Column	Non-Null Count	Dtype
0	Year	10431 non-null	int64
1	Country	10431 non-null	object
2	ISO	10431 non-null	object
3	Disaster Group	10431 non-null	object
4	Disaster Subroup	10431 non-null	object
5	Disaster Type	10431 non-null	object
6	Disaster Subtype	8298 non-null	object
7	Total Events	10431 non-null	int64
8	Total Affected	7586 non-null	float64
9	Total Deaths	7375 non-null	float64

View recommended plots

```
10 Total Damage (USD, original) 3834 non-null float64
11 Total Damage (USD, adjusted) 3830 non-null float64
12 CPI,, 10431 non-null object
dtypes: float64(4), int64(2), object(7)
memory usage: 1.0+ MB
```

Check missing values

df.isnull().sum()

₹	0
Year	0
Country	0
ISO	0
Disaster Group	0
Disaster Subroup	0
Disaster Type	0
Disaster Subtype	2133
Total Events	0
Total Affected	2845
Total Deaths	3056
Total Damage (USD, or	iginal) 6597
Total Damage (USD, adj	justed) 6601
CPI,,	0

dtype: int64

df.describe()

<u>-</u>	Year	Total Events	Total Affected	Total Deaths	Total Damage (USD, original)	Total Damage (USD, adjusted)
cour	t 10431.000000	10431.000000	7.586000e+03	7.375000e+03	3.834000e+03	3.830000e+03
mea	n 1995.609625	1.446649	1.125969e+06	3.107711e+03	1.122262e+09	1.748704e+09
std	22.001186	1.246589	9.760891e+06	7.255589e+04	6.792339e+09	9.115319e+09
min	1900.000000	1.000000	1.000000e+00	1.000000e+00	2.000000e+03	2.469000e+03
25%	1986.000000	1.000000	1.200000e+03	6.000000e+00	1.000000e+07	2.020927e+07
50%	2001.000000	1.000000	1.141400e+04	2.300000e+01	6.800000e+07	1.469247e+08
75%	2011.000000	1.000000	1.193045e+05	9.000000e+01	4.000000e+08	7.847767e+08
max	2023.000000	20.000000	3.300000e+08	3.700000e+06	2.100000e+11	2.732184e+11

print(df.columns.tolist())

['Year', 'Country', 'ISO', 'Disaster Group', 'Disaster Subroup', 'Disaster Type', 'Disaster Subtype', 'Total Events', 'Total Affecte

print(df['Disaster Type'].unique())

```
['Drought' 'Flood' 'Volcanic activity' 'Earthquake' 'Storm'
'Mass movement (dry)' 'Landslide' 'Wildfire' 'Insect infestation'
'Extreme temperature ' 'Fog' 'Animal accident' 'Glacial lake outburst']
```

Distribution of affected population
df['Total Affected'].describe()



	Total Affected
count	7.586000e+03
mean	1.125969e+06
std	9.760891e+06
min	1.000000e+00
25%	1.200000e+03
50%	1.141400e+04
75%	1.193045e+05
max	3.300000e+08

dtype: float64

Correlation check for numeric columns
df.corr(numeric_only=True)





