Importing Libraries

In [2]:

```
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
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_rep
```

Load Dataset

In [3]:

```
df = pd.read_excel("capstone dataset.xlsx")
print(df.head())
print(df.info())
print("\nMissing values:\n", df.isnull().sum())
```

```
     City Name Country  Urban Population (Millions)  \
0      Chennai   India                          1.4   
1        Delhi   India                         10.8   
2       Mumbai   India                         11.5   
3     New York     USA                         13.6   
4  Los Angeles     USA                         13.8   

   Population Growth Rate (%)  Green Space (%)  Air Quality Index (PM2.5)  \
0                         0.8               44                         99   
1                         1.2               24                         64   
2                         2.7               13                         50   
3                         0.8               27                         30   
4                         2.8               41                         85   

   CO2 Emissions (per capita)  Access to Clean Water (%)  \
0                         4.9                         98   
1                         5.6                         93   
2                         1.8                         96   
3                         4.9                         99   
4                         0.9                         98   

   Waste Recycling Rate (%)  Public Transport Usage (%)  \
0                        41                          59   
1                        59                          64   
2                        46                          71   
3                        42                          75   
4                        52                          51   

   Energy Consumption (kWh per capita)  
0                                 1594  
1                                 1006  
2                                 1352  
3                                 1652  
4                                  927  
<class 'pandas.DataFrame'>
RangeIndex: 82 entries, 0 to 81
Data columns (total 11 columns):
 #   Column                               Non-Null Count  Dtype  
---  ------                               --------------  -----  
 0   City Name                            82 non-null     str    
 1   Country                              82 non-null     str    
 2   Urban Population (Millions)          82 non-null     float64
 3   Population Growth Rate (%)           82 non-null     float64
 4   Green Space (%)                      82 non-null     int64  
 5   Air Quality Index (PM2.5)            82 non-null     int64  
 6   CO2 Emissions (per capita)           82 non-null     float64
 7   Access to Clean Water (%)            82 non-null     int64  
 8   Waste Recycling Rate (%)             82 non-null     int64  
 9   Public Transport Usage (%)           82 non-null     int64  
 10  Energy Consumption (kWh per capita)  82 non-null     int64  
dtypes: float64(3), int64(6), str(2)
memory usage: 7.2 KB
None

Missing values:
 City Name                              0
Country                                0
Urban Population (Millions)            0
Population Growth Rate (%)             0
Green Space (%)                        0
```

```
Air Quality Index (PM2.5)             0
CO2 Emissions (per capita)            0
Access to Clean Water (%)             0
Waste Recycling Rate (%)              0
Public Transport Usage (%)            0
Energy Consumption (kWh per capita)   0
dtype: int64
```

Data Cleaning

In [4]:

```
df = df.drop_duplicates()
```

Encoding Categorical Data

In [5]:

```
le_city = LabelEncoder()
le_country = LabelEncoder()

df["City Name"] = le_city.fit_transform(df["City Name"])
df["Country"] = le_country.fit_transform(df["Country"])
```

Feature Engineering

In [6]:

```
df["Sustainability_Score"] = (
    df["Green Space (%)"] +
    df["Access to Clean Water (%)"] +
    df["Waste Recycling Rate (%)"] +
    df["Public Transport Usage (%)"]
) - (
    df["Air Quality Index (PM2.5)"] +
    df["CO2 Emissions (per capita)"]
)

df["Sustainable"] = df["Sustainability_Score"].apply(
    lambda x: 1 if x > df["Sustainability_Score"].median() else 0
)
```

Feature Selection

In [7]:

```
X = df.drop(["Sustainability_Score", "Sustainable"], axis=1)
y = df["Sustainable"]
```

Train Test Split

In [8]:

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

Feature Scaling

In [9]:

```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Training ML Model

In [10]:

```
model = LogisticRegression()
model.fit(X_train, y_train)
```

Out[10]:

```
▾ LogisticRegression ⓘ ?

▶ Parameters
```

Predictions

In [11]:

```
y_pred = model.predict(X_test)
```

Evaluation

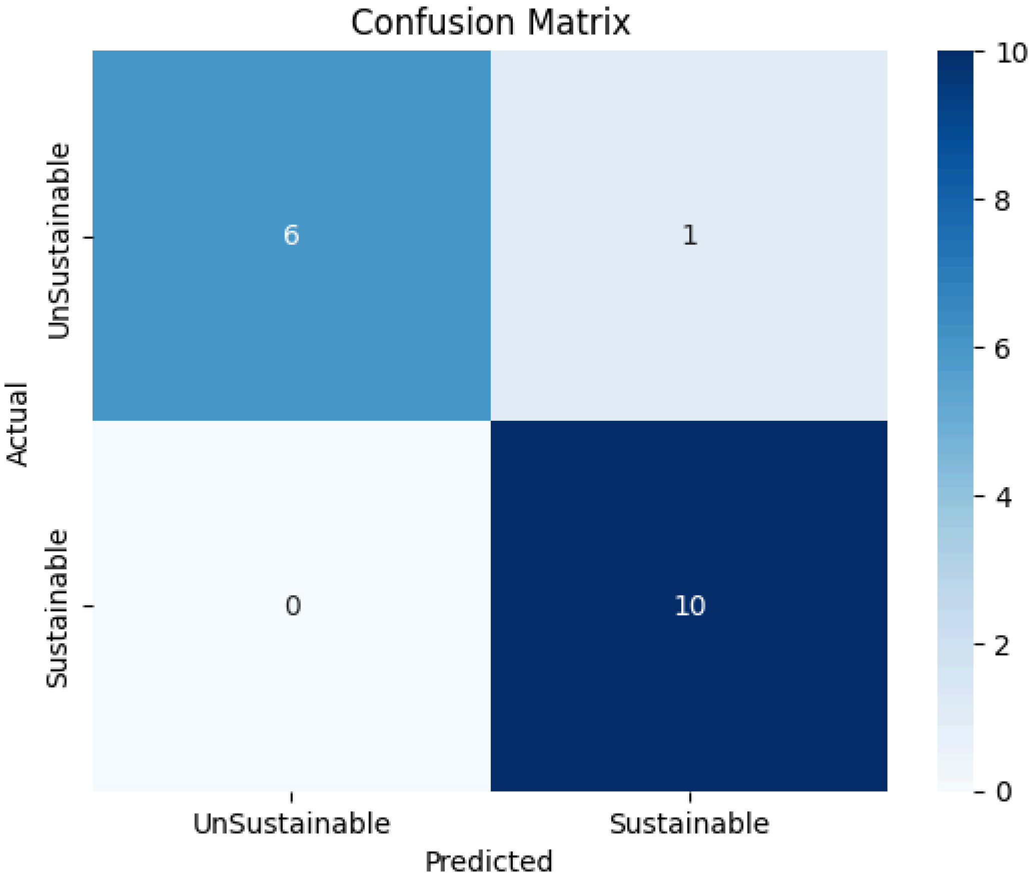
In [12]:

```
print("\nAccuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

```
Accuracy: 0.9411764705882353

Classification Report:
               precision    recall  f1-score   support

           0       1.00      0.86      0.92         7
           1       0.91      1.00      0.95        10

    accuracy                           0.94        17
   macro avg       0.95      0.93      0.94        17
weighted avg       0.95      0.94      0.94        17
```

In [13]:

```
conf_matrix = confusion_matrix(y_test, y_pred)
plt.figure()
sns.heatmap(conf_matrix, annot=True, fmt="d", cmap="Blues",
            xticklabels=["UnSustainable", "Sustainable"],
            yticklabels=["UnSustainable", "Sustainable"])
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()
```


Confusion Matrix
Actual
UnSustainable
Sustainable
6
1
0
10
Predicted
UnSustainable
Sustainable
10
8
6
4
2
0

```
In [ ]:
```