ENSURING CLEAN WATER AND SANITATION: DATA DRIVEN INSIGHTS FOR SUSTAINABLE DEVELOPMENT

TEAM NAME: CODE CREW
COLLEGE NAME: DELHI TECHNICAL CAMPUS

OVERVIEW

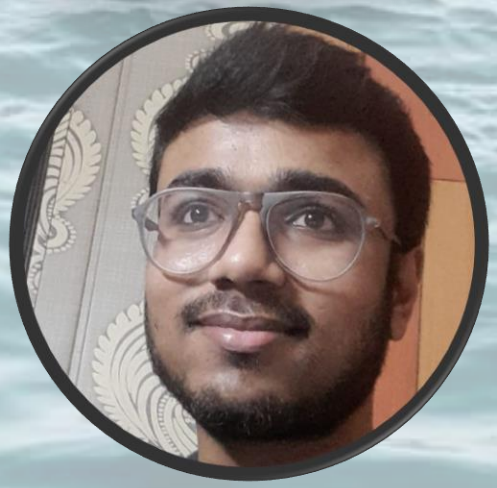
1. Team Members
2. Introduction
3. Objective
4. Problem Statement
5. Sources of data
6. Details about modules used
7. Features
8. Data before cleaning
9. Data after cleaning
10. Conclusion

TEAM MEMBERS

Karan Purohit

Kashish Srivastava

Nikhil Verma



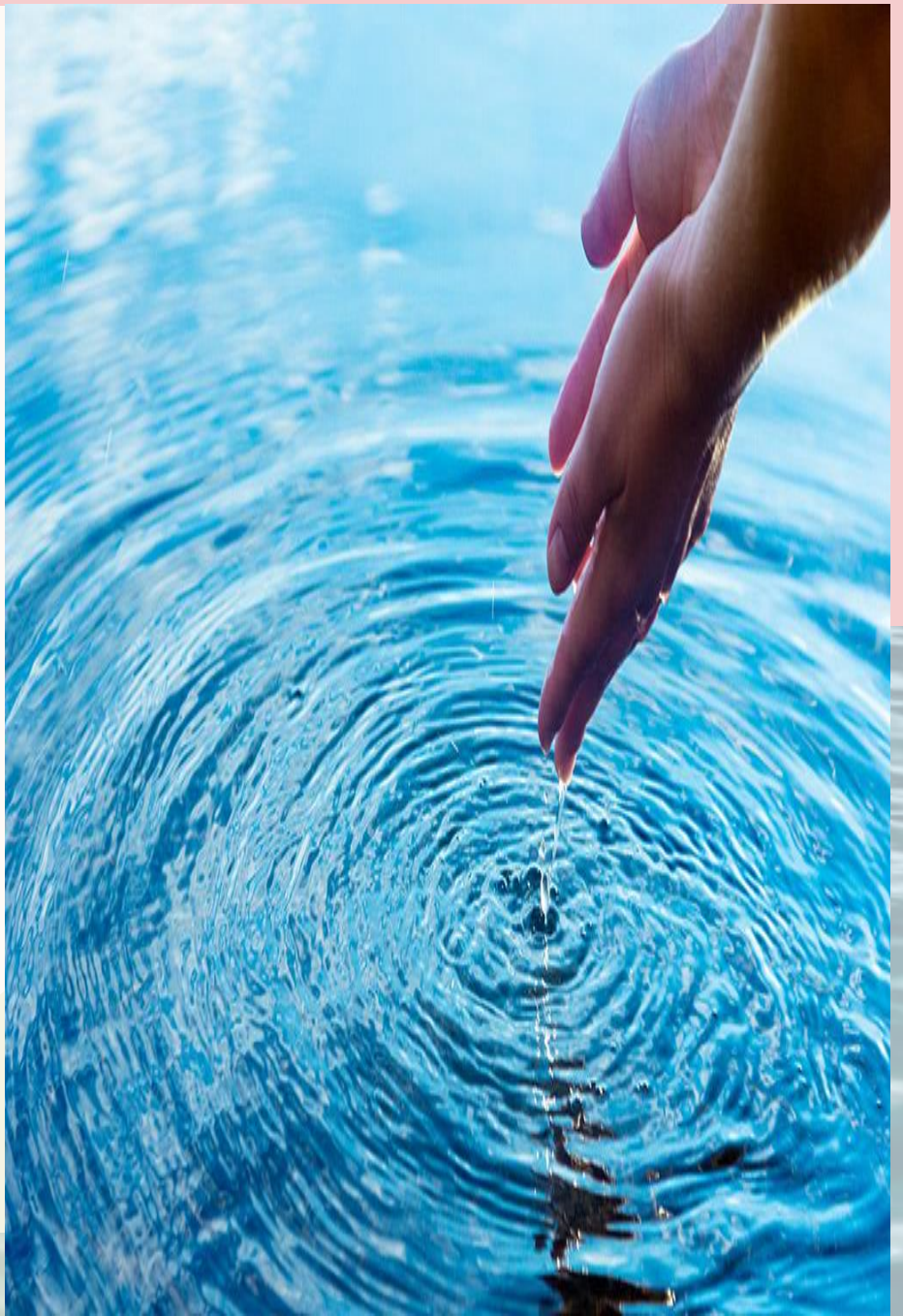
Om Singhal

INTRODUCTION

"Ensuring Clean Water and Sanitation: Data-Driven Insights for Sustainable Development" is an innovative initiative that aims to address the worldwide problem of access to sufficient sanitation and clean water by utilizing data analytics. This project aims to improve global community quality of life, support health and well-being, and help accomplish the Sustainable Development Goals of the United Nations by means of creative technical solutions and cooperative efforts.

OBJECTIVE

This project's main goal is to use data-driven insights to guarantee clean water and sanitation, supporting the objectives of sustainable development. The project's goal is to identify crucial regions that need intervention by analyzing many data sources, such as usage patterns, sanitation infrastructure, and indicators related to water quality. The project's ultimate goals are to advance environmental sustainability, public health, and universal access to clean water and sanitary facilities.



PROBLEM STATEMENT

Data-driven insights that are vital for sustainable development highlight how important it is to ensure access to clean water and sanitation as a major global concern. Global datasets indicate that this field is still plagued by a number of major issues. Second, water quality continues to be a major concern since untreated sewage, agricultural runoff, and industrial pollutants can contaminate it and endanger public health throughout the world. Thirdly, access to clean water is hampered by poor infrastructure, particularly in rural areas where dependable distribution networks are few. Fourthly, unpredictable rainfall patterns and droughts that put a strain on water supplies and sanitary systems are brought on by climate change, which makes these problems worse. The last barrier to efficient solution implementation is governance and budgetary constraints, which also affect the upkeep and monitoring of current infrastructure.

SOURCES OF DATA

- Organization for World Health (WHO)
- United Nations Children's Fund, or UNICEF
- World Bank Water Data
- Research Institutions and International Water Organizations

DETAILS ABOUT MODULES USED

- **ETL(Extract Transform Load)**

It is the process of combining data from multiple sources into a large, central repository called a data warehouse

- **Machine Learning**

It is a branch of AI and computer science that focuses on the using data and algorithms to enable AI to imitate the way that humans learn, gradually improving its accuracy.

- **Power BI**

Microsoft Power BI is an interactive data visualization software product developed by Microsoft with a primary focus on business intelligence.

- **Jupyter Notebook**

Project Jupyter is a project to develop open-source software, open standards, and services for interactive computing across multiple programming languages.

FEATURES

1. Country

The name of the country where the data was collected.

2. Year

The year when the data was collected or reported.

3. Usage of Improved Drinking Water Sources

The percentage of the population using improved drinking water sources.

4. Usage of Basic Drinking Water Services

The percentage of the population using basic drinking water services.

5. Usage of Limited Drinking Water Services

The percentage of the population using limited drinking water services

DATA BEFORE CLEANING

```
In [6]: #extracting

In [ ]: import pandas as pd

In [7]: input_file = 'Desktop/om(DSA)/etl/water-and-sanitation.csv'
        output_file = 'Desktop/om(DSA)/etl/transformed_water-and-sanitation.csv'

In [8]: df = pd.read_csv(input_file)

In [9]: df.head()
```

Out[9]:

Usage of limited drinking water services	Usage of unimproved drinking water sources	No usage of drinking water facilities	Usage of safely managed drinking water services	Usage of improved sanitation facilities	Usage of basic sanitation services	...	wat_pip_urban	wat_pip_number_rural	wat_pip_number	wat_pip_number_urban	wat_sm_r
3.299203	43.856777	25.402164	11.093327	26.466162	NaN	...	19.063290	0.00	822523.7	822523.7	
3.299883	43.843445	25.383093	11.105221	26.488068	NaN	...	19.063290	0.00	832069.3	832069.3	
3.607177	42.260395	24.457567	12.007733	28.414984	NaN	...	20.168760	0.00	942862.7	942862.7	
3.914072	40.677280	23.533058	12.909922	30.342781	NaN	...	21.274233	142883.52	1219756.8	1076873.1	
4.220617	39.086002	22.598950	13.818684	32.285492	NaN	...	22.379705	299784.84	1485808.1	1186023.4	

```
In [10]: #transform

In [ ]: df = df.dropna()

In [11]: df.info
```

```
Out[11]: <bound method DataFrame.info of        Country  Year  Usage of improved drinking water sources  \
76       Algeria  2007                                 96.996520
77       Algeria  2008                                 97.208450
78       Algeria  2009                                 97.415550
79       Algeria  2010                                 97.618164
80       Algeria  2011                                 97.816310
...          ...   ...                                       ...
5732    Zimbabwe  2018                                 77.055710
5733    Zimbabwe  2019                                 76.955050
5734    Zimbabwe  2020                                 76.864000
5735    Zimbabwe  2021                                 76.782080
5736    Zimbabwe  2022                                 76.810740

      Usage of basic drinking water services  \
76                                 17.888512
77                                 17.660484
78                                 17.444347
79                                 17.239166
80                                 17.045073
```

```
In [12]: df.head()
```

Out[12]:

	Country	Year	Usage of improved drinking water sources	Usage of basic drinking water services	Usage of limited drinking water services	Usage of unimproved drinking water sources	No usage of drinking water facilities	Usage of safely managed drinking water services	Usage of improved sanitation facilities	Usage of basic sanitation services	...	wat_pip_urban	wat_pip_number_rural	wat_pip_
76	Algeria	2007	96.996520	17.888512	5.387287	2.697115	0.306367	73.720720	94.12311	25.189022	...	85.460900	7648207.0	266

```
In [12]: df.head()
```

Out[12]:

	Country	Year	Usage of improved drinking water sources	Usage of basic drinking water services	Usage of limited drinking water services	Usage of unimproved drinking water sources	No usage of drinking water facilities	Usage of safely managed drinking water services	Usage of improved sanitation facilities	Usage of basic sanitation services	...	wat_pip_urban	wat_pip_number_rural	wat_pip_
76	Algeria	2007	96.996520	17.888512	5.387287	2.697115	0.306367	73.720720	94.12311	25.189022	...	85.460900	7648207.0	266
77	Algeria	2008	97.208450	17.660484	5.365142	2.506071	0.285478	74.182820	94.34653	25.125036	...	84.741210	7566640.5	269
78	Algeria	2009	97.415550	17.444347	5.343516	2.319164	0.265285	74.627686	94.56176	25.060670	...	84.021520	7493047.0	272
79	Algeria	2010	97.618164	17.239166	5.322189	2.136100	0.245737	75.056810	94.76960	24.995966	...	83.301834	7424406.5	275
80	Algeria	2011	97.816310	17.045073	5.301317	1.956861	0.226832	75.469920	94.96990	24.930914	...	82.582146	7359657.0	279

5 rows × 206 columns

```
In [13]: df.isnull().sum()
```

```
Out[13]: Country                                0
Year                                   0
Usage of improved drinking water sources    0
Usage of basic drinking water services      0
Usage of limited drinking water services    0
                                      ..
wat_sm_number_without                  0
wat_sm_number_without_urban            0
wat_sm_without_rural                   0
wat_sm_without                         0
wat_sm_without_urban                   0
Length: 206, dtype: int64
```

In [14]:
```
# 2. Normalize column names: convert to lower case and replace spaces with underscores
df.columns = [col.strip().lower().replace(' ', '_') for col in df.columns]
```

In [15]:
```
df.head()
```

Out[15]:

	country	year	usage_of_improved_drinking_water_sources	usage_of_basic_drinking_water_services	usage_of_limited_drinking_water_services	usage_of_unim
76	Algeria	2007	96.996520	17.888512	5.387287	
77	Algeria	2008	97.208450	17.660484	5.365142	
78	Algeria	2009	97.415550	17.444347	5.343516	
79	Algeria	2010	97.618164	17.239166	5.322189	
80	Algeria	2011	97.816310	17.045073	5.301317	

5 rows × 206 columns

In [16]:
```
# 3. Convert 'year' column to numeric
df['year'] = pd.to_numeric(df['year'], errors='coerce')
```

In [17]:
```
# 4. Convert specific columns to numeric (e.g., percentage and numeric columns)
numeric_columns = [
    'usage_of_improved_drinking_water_sources',
    'usage_of_basic_drinking_water_services',
    'usage_of_limited_drinking_water_services',
    'usage_of_unimproved_drinking_water_sources',
    'no_usage_of_drinking_water_facilities',
    'usage_of_safely_managed_drinking_water_services',
    'usage_of_improved_sanitation_facilities',
    'usage_of_basic_sanitation_services',
    'usage_of_limited_sanitation_services',
    'usage_of_unimproved_sanitation_facilities',
    'no_usage_of_sanitation_facilities'
```

In [17]:
```
# 4. Convert specific columns to numeric (e.g., percentage and numeric columns)
numeric_columns = [
    'usage_of_improved_drinking_water_sources',
    'usage_of_basic_drinking_water_services',
    'usage_of_limited_drinking_water_services',
    'usage_of_unimproved_drinking_water_sources',
    'no_usage_of_drinking_water_facilities',
    'usage_of_safely_managed_drinking_water_services',
    'usage_of_improved_sanitation_facilities',
    'usage_of_basic_sanitation_services',
    'usage_of_limited_sanitation_services',
    'usage_of_unimproved_sanitation_facilities',
    'no_usage_of_sanitation_facilities',
```
```
    'population_total',
    'population_urban',
    'population_rural',
    'pop_den',
    'area',
    'gdp',
    'gdp_per_capita',
    'improved_san_coverage_rural',
    'improved_san_coverage_urban',
    'improved_san_coverage_total',
    'basic_san_coverage_rural',
    'basic_san_coverage_urban',
    'basic_san_coverage_total',
    'limited_san_coverage_rural',
    'limited_san_coverage_urban',
    'limited_san_coverage_total',
    'unimproved_san_coverage_rural',
    'unimproved_san_coverage_urban',
    'unimproved_san_coverage_total',
    'open_defecation_rural',
    'open_defecation_urban',
    'open_defecation_total'
]
```

DATA AFTER CLEANING

In [42]:
```
for col in numeric_columns:
    if col in df.columns:
        df[col] = pd.to_numeric(df[col], errors='coerce')
        # Round off the values to the nearest integer
        df[col] = df[col].round()
```

In [45]:
```
df.head()
```

Out[45]:

	country	year	usage_of_improved_drinking_water_sources	usage_of_basic_drinking_water_services	usage_of_limited_drinking_water_services	usage_of_unim
76	Algeria	2007	97.0	18.0	5.0	
77	Algeria	2008	97.0	18.0	5.0	
78	Algeria	2009	97.0	17.0	5.0	
79	Algeria	2010	98.0	17.0	5.0	
80	Algeria	2011	98.0	17.0	5.0	

5 rows × 206 columns

In [46]:
```
# Load: Write the transformed DataFrame to a new CSV file
df.to_csv(output_file, index=False)
```

In [47]:
```
print("ETL process completed successfully.")
```

```
ETL process completed successfully.
```

MACHINE LEARNING

```
[6]: data.head(10)
```

[6]:

	country	year	usage_of_improved_drinking_water_sources	usage_of_basic_drinking_water_services	usage_of_limited_drinking_water_services	usage_of_unimproved_drinki
0	Algeria	2007	97.0	18.0	5.0	
1	Algeria	2008	97.0	18.0	5.0	
2	Algeria	2009	97.0	17.0	5.0	
3	Algeria	2010	98.0	17.0	5.0	
4	Algeria	2011	98.0	17.0	5.0	
5	Algeria	2012	98.0	17.0	5.0	
6	Algeria	2013	98.0	17.0	5.0	
7	Algeria	2014	98.0	17.0	5.0	
8	Algeria	2015	99.0	16.0	5.0	
9	Algeria	2016	99.0	17.0	5.0	

10 rows × 206 columns

```
[214]: import pandas as pd
from sklearn.preprocessing import LabelEncoder

# Load data
data = pd.read_csv('transformed_water-and-sanitation.csv')

# Inspecting the column names
print(data.columns)

# Selecting features and target variable
features = [
    'usage_of_improved_drinking_water_sources', 'usage_of_basic_drinking_water_services',
    'usage_of_limited_drinking_water_services', 'usage_of_unimproved_drinking_water_sources',
    'no_usage_of_drinking_water_facilities', 'usage_of_improved_sanitation_facilities',
    'usage_of_basic_sanitation_services'
]
target = 'usage_of_safely_managed_drinking_water_services'
print(data.head())
```

```
Index(['country', 'year', 'usage_of_improved_drinking_water_sources',
       'usage_of_basic_drinking_water_services',
       'usage_of_limited_drinking_water_services',
       'usage_of_unimproved_drinking_water_sources',
       'no_usage_of_drinking_water_facilities',
       'usage_of_safely_managed_drinking_water_services',
       'usage_of_improved_sanitation_facilities',
       'usage_of_basic_sanitation_services',
       ...
       'wat_pip_urban', 'wat_pip_number_rural', 'wat_pip_number',
       'wat_pip_number_urban', 'wat_sm_number_without_rural',
       'wat_sm_number_without', 'wat_sm_number_without_urban',
       'wat_sm_without_rural', 'wat_sm_without', 'wat_sm_without_urban'],
      dtype='object', length=206)
```

[43]:
```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
# Select features and target variable
X = data[['usage_of_improved_drinking_water_sources',
          'usage_of_basic_drinking_water_services',
          'usage_of_limited_drinking_water_services',
          'usage_of_unimproved_drinking_water_sources',
          'no_usage_of_drinking_water_facilities']]
y = data['usage_of_safely_managed_drinking_water_services']
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

[18]:
```
# Initialize the model
model = RandomForestRegressor(n_estimators=100, random_state=42)

# Train the model
model.fit(X_train, y_train)
```

[18]:
```
RandomForestRegressor
RandomForestRegressor(random_state=42)
```

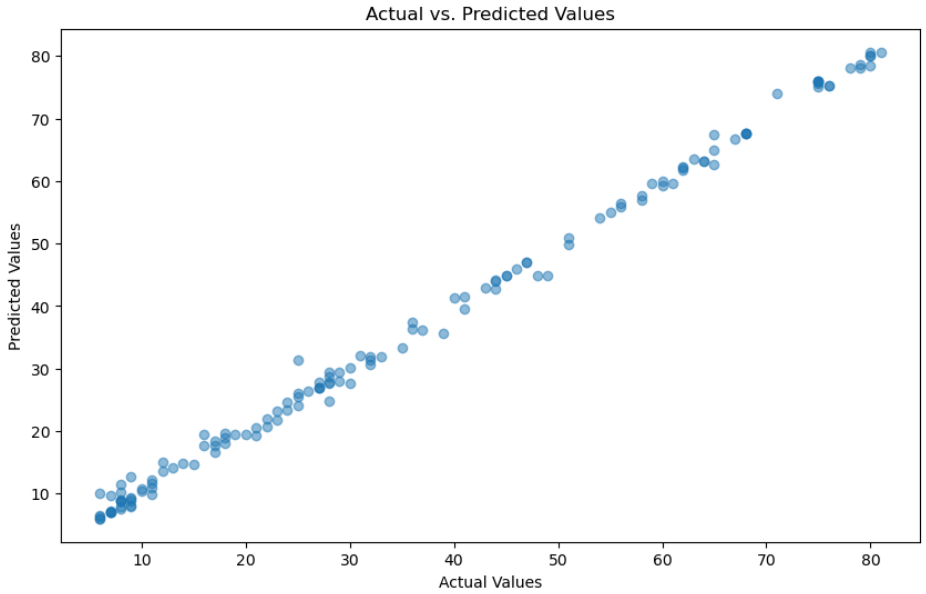
[45]:
```
# Making prediction
y_pred = model.predict(X_test)
```

[22]:
```
# Evaluating the model's performance
from sklearn.metrics import mean_squared_error, r2_score

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f'Mean Squared Error: {mse}')
print(f'R-squared: {r2}')
```

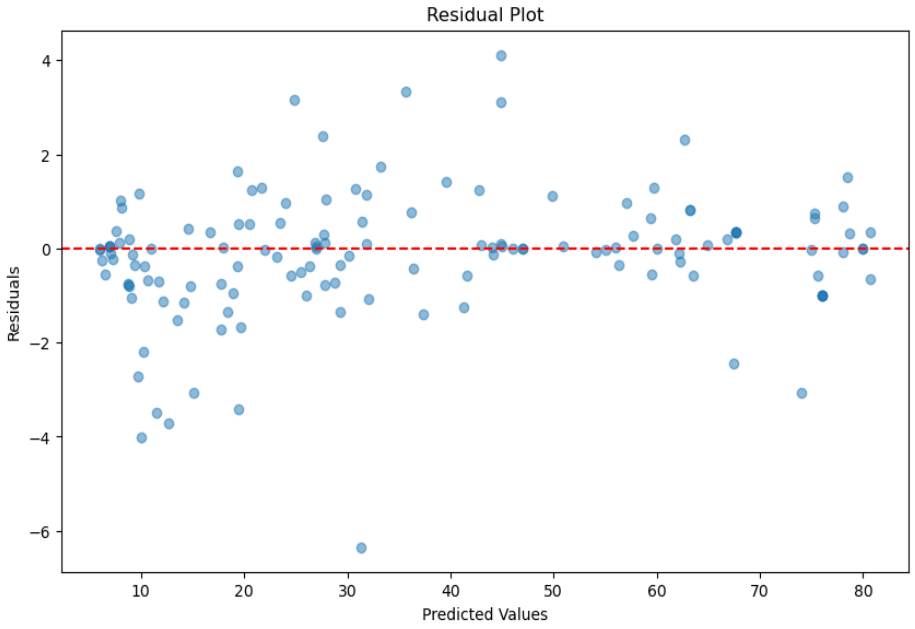
```
Mean Squared Error: 1.8625533601071518
R-squared: 0.9967647067337857
```

DATA VISUALIZATION

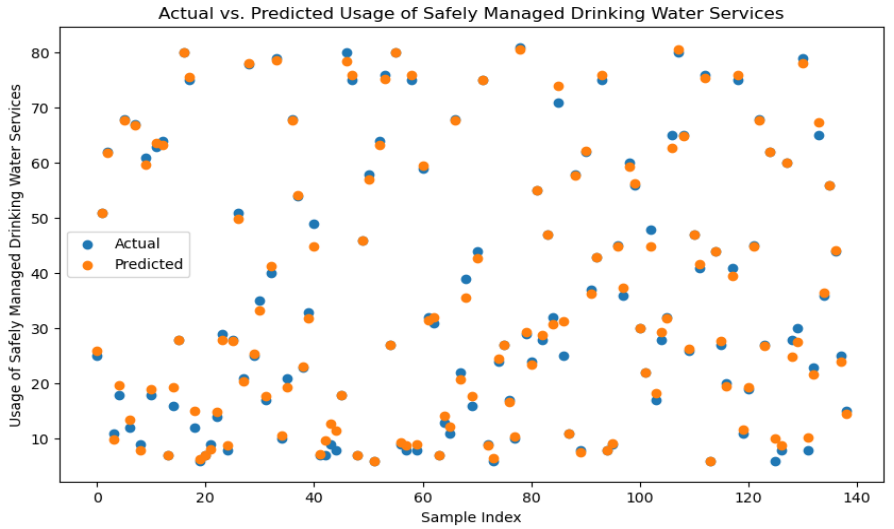
```
[24]: import matplotlib.pyplot as plt
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred, alpha=0.5)
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual vs. Predicted Values')
plt.show()
```



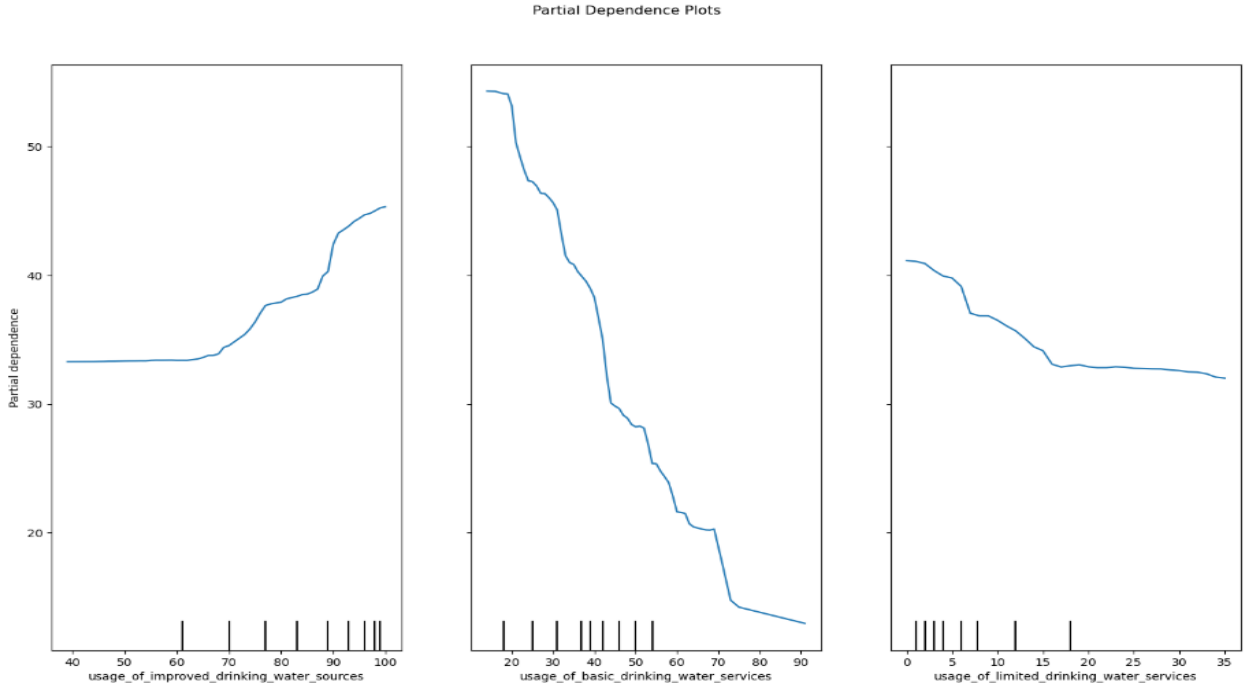

```
[25]: #Plot Residual Plot
residuals = y_test - y_pred
plt.figure(figsize=(10, 6))
plt.scatter(y_pred, residuals, alpha=0.5)
plt.axhline(y=0, color='r', linestyle='--')
plt.xlabel('Predicted Values')
plt.ylabel('Residuals')
plt.title('Residual Plot')
plt.show()
```



[26]:
```
# Creating a dataframe for actual vs predicted values
predictions = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
# Plot actual vs. predicted values
plt.figure(figsize=(10, 6))
plt.scatter(range(len(y_test)), y_test, label='Actual')
plt.scatter(range(len(y_pred)), y_pred, label='Predicted')
plt.legend()
plt.title('Actual vs. Predicted Usage of Safely Managed Drinking Water Services')
plt.xlabel('Sample Index')
plt.ylabel('Usage of Safely Managed Drinking Water Services')
plt.show()
```

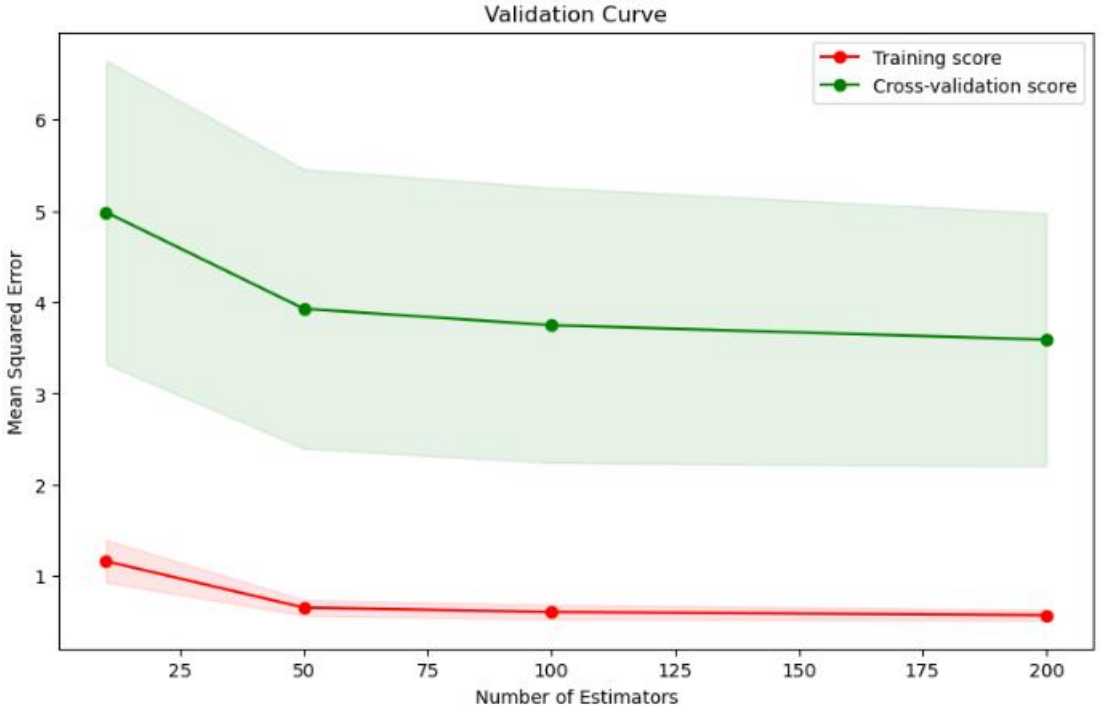


[29]:
```
from sklearn.inspection import PartialDependenceDisplay
# Initializing the PartialDependenceDisplay
fig, ax = plt.subplots(figsize=(17, 10))
display = PartialDependenceDisplay.from_estimator(model, X_train, features=[0, 1, 2], ax=ax)
plt.suptitle('Partial Dependence Plots')
plt.subplots_adjust(top=0.9)
plt.show()
```

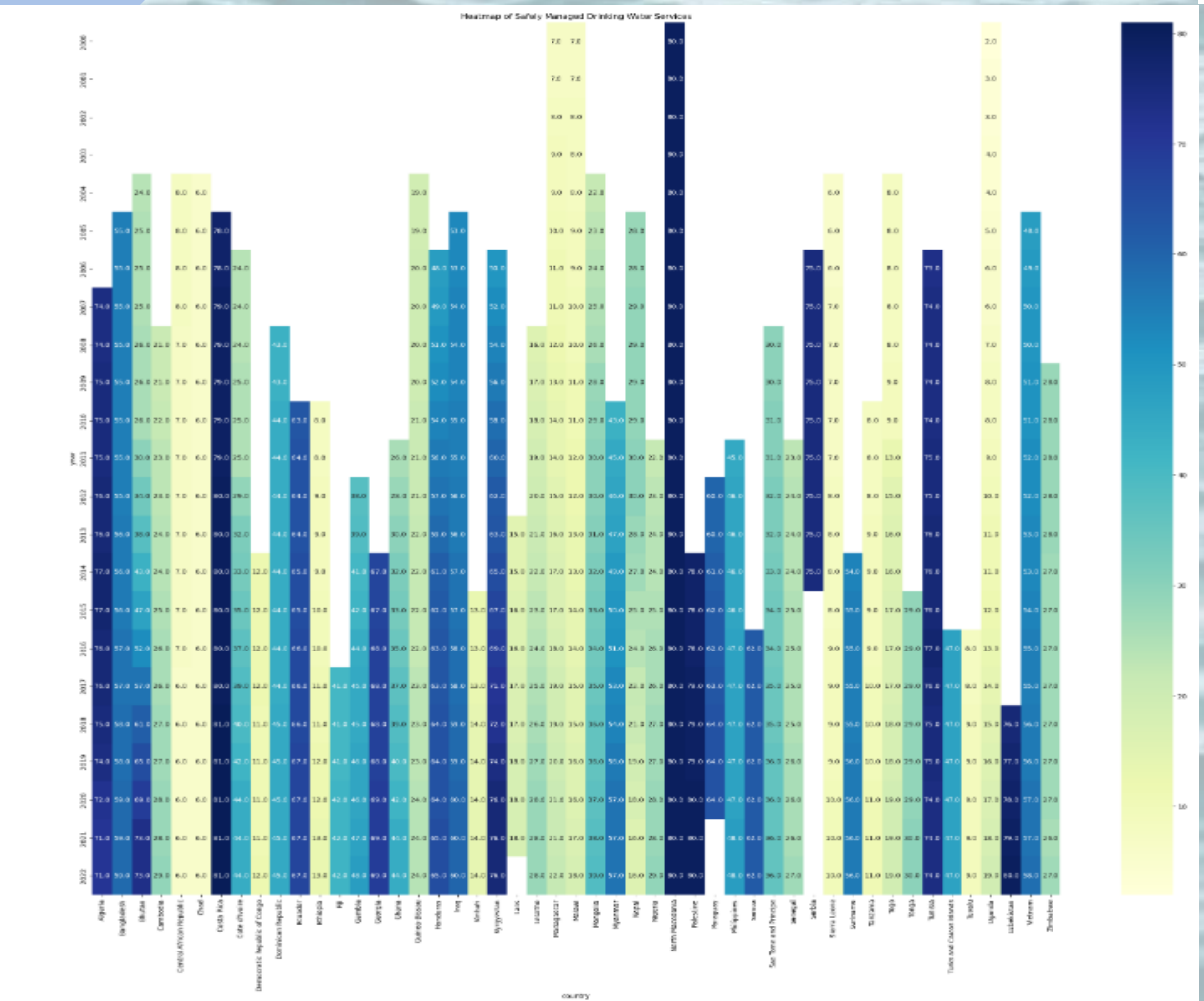


[31]:

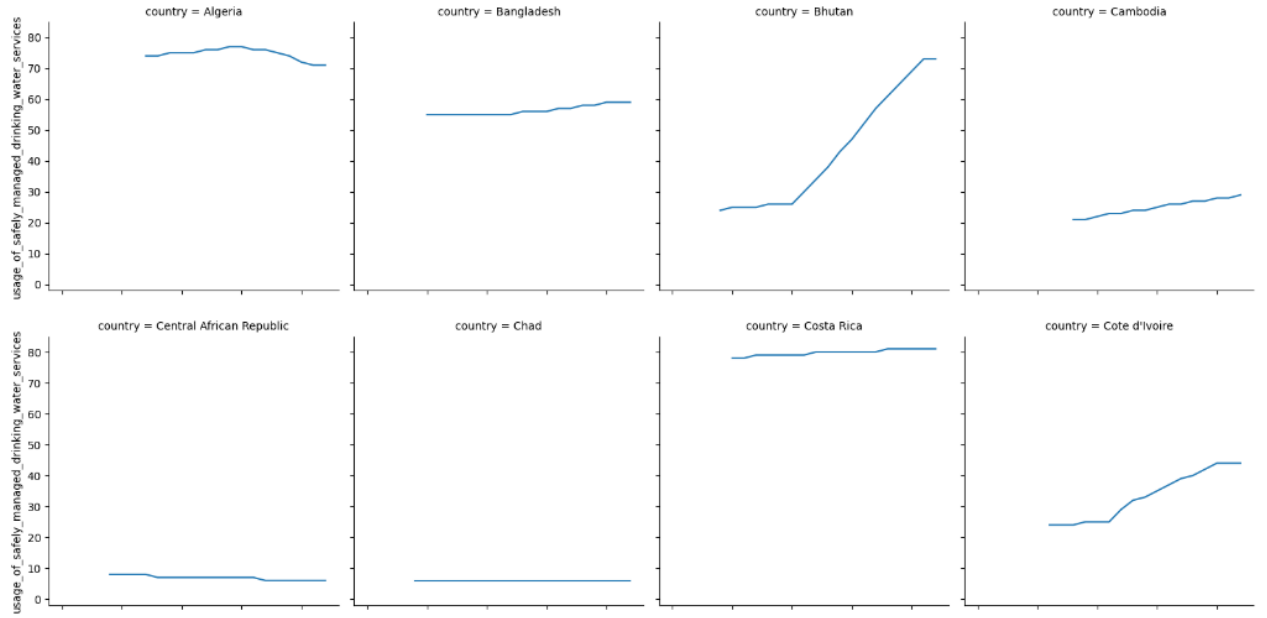
```python
from sklearn.model_selection import validation_curve

# Computing validation curve
param_range = [10, 50, 100, 200]
train_scores, test_scores = validation_curve(model, X_train, y_train, param_name='n_estimators', param_range=param_range, cv=5, scoring='neg_

# Computing mean and standard deviation of training and testing scores
train_mean = -train_scores.mean(axis=1)
test_mean = -test_scores.mean(axis=1)
train_std = train_scores.std(axis=1)
test_std = test_scores.std(axis=1)

# Plots validation curve
plt.figure(figsize=(10, 6))
plt.plot(param_range, train_mean, 'o-', color='r', label='Training score')
plt.plot(param_range, test_mean, 'o-', color='g', label='Cross-validation score')
plt.fill_between(param_range, train_mean - train_std, train_mean + train_std, alpha=0.1, color='r')
plt.fill_between(param_range, test_mean - test_std, test_mean + test_std, alpha=0.1, color='g')
plt.xlabel('Number of Estimators')
plt.ylabel('Mean Squared Error')
plt.title('Validation Curve')
plt.legend()
plt.show()
```



[37]:
```
#Plot Pviot table
import seaborn as sns
plt.figure(figsize=(30, 26))
pivot_table = data.pivot_table(values='usage_of_safely_managed_drinking_water_services', index='year', columns='country')
sns.heatmap(pivot_table, cmap='YlGnBu', annot=True, fmt='.1f')
plt.title('Heatmap of Safely Managed Drinking Water Services')
plt.show()
```



[39]:
```
#Plot Facet Grid
g = sns.FacetGrid(data, col="country", col_wrap=4, height=4)
g.map(sns.lineplot, "year", "usage_of_safely_managed_drinking_water_services")
g.add_legend()
plt.show()
```



SAVING PREDICTION AND TRAINED MODEL

[41]:
```python
import joblib

# Saving predictions to a CSV file
predictions = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
predictions.to_csv('predictions.csv', index=False)

# Saving the trained model to a file
joblib.dump(model, 'safely_managed_drinking_water_model.pkl')
```

[41]:
```
['safely_managed_drinking_water_model.pkl']
```

CONCLUSION

"Ensuring Clean Water and Sanitation: Data-Driven Insights for Sustainable Development" is a project aimed at furthering Sustainable Development Goal 6 (SDG 6), which states that water and sanitation should be available and managed sustainably for all people. This project uses innovative data analytics to evaluate and resolve the world's problems with water availability, quality, and sanitation infrastructure. It also creates predictive models for future water scarcity and sanitation concerns, allowing policymakers and stakeholders to make educated decisions.

THANK YOU

By:- Code Crew