Course Project – Big Data Concepts

Greenhouse Gas Emissions Analysis and Prediction using Google Cloud Platform

By

Sai Sindhura Kollepara

Username: skollep

Email id: skollep@iu.edu

Indiana University Bloomington

Contents

1. Introduction

Greenhouse gas emissions are a significant factor influencing climate change and environmental sustainability. Monitoring and analyzing per capita greenhouse gas emissions across countries are crucial for policymakers to formulate strategies that balance environmental and developmental goals. This project focuses on understanding the trends and factors influencing per capita emissions of key greenhouse gases—carbon dioxide (CO_2), methane (CH_4), and nitrous oxide (N_2O)—alongside other related variables such as energy use, population, and GDP per capita. These features were carefully selected from the "Our World in Data" platform due to their relevance in measuring and interpreting emissions data.

The aim of the project is to leverage advanced data analytics tools to uncover meaningful insights from this data. By studying emission patterns over several decades in nearly 200 countries, the project seeks to provide actionable insights into the drivers of greenhouse gas emissions. The findings are intended to aid policymakers and environmental experts in implementing effective measures to mitigate climate change.

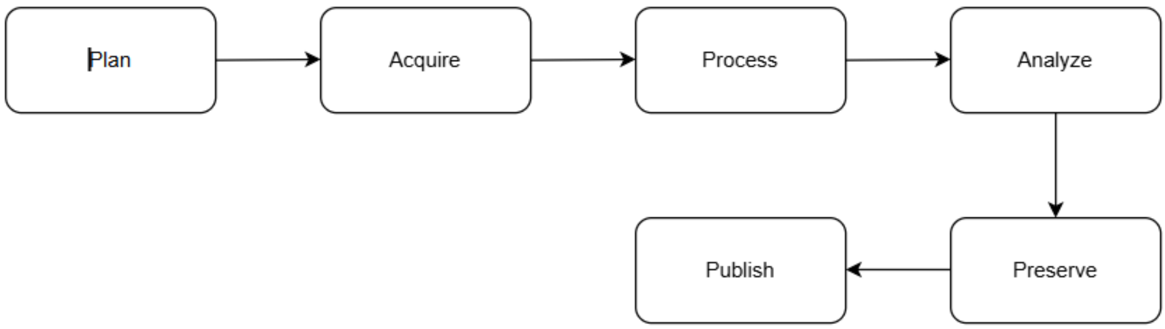
2. Background

Addressing greenhouse gas emissions is vital for ensuring long-term environmental sustainability. By understanding the relationship between emissions and factors such as population and economic growth, governments can create tailored policies that balance environmental goals with economic development. This study focuses on per capita emissions rather than total annual emissions, recognizing that population size significantly influences emission levels. Assessing emissions on a per capita basis ensures fairness in policy formulation, as it accounts for disparities in population and economic activity across nations.

The dataset used in this project captures key contributors to greenhouse gas emissions, including CO_2, CH_4, and N_2O, as well as energy usage and socioeconomic factors like population and GDP per capita. Advanced data analytics tools and Cloud technologies like Google Cloud Platform, BigQuery, Looker Studio, and Google Colab were utilized to process and analyze the data efficiently. These technologies enabled an in-depth exploration of emission trends and contributing factors, making the project highly relevant in the context of ongoing climate change discussions.

3. Methodology

Data Pipeline



1. **Plan**: Establish objectives to analyze per-capita greenhouse gas emissions globally, focusing on data aggregation, statistical analysis, and predictive modeling.
2. **Acquire**: Source datasets on GHG emissions, population, GDP, and energy use from public platforms, and store them in a GCP bucket for efficient access.
3. **Process**: Perform data preprocessing using Google Colab and GCP, including merging datasets, standardizing columns, handling missing values, and removing outliers.
4. **Analyze**: Generate insights through statistical and visual techniques, leveraging Python, BigQuery and Looker Studio for charts, heatmaps, histograms, and animated choropleth maps.
5. **Preserve**: Save the cleaned dataset, visualizations, and machine learning model outputs in the GCP bucket for secure, organized future use.
6. **Publish**: Upload the processed data and scripts to GitHub, promoting collaboration and further exploration by the data science community.

3. 1 Plan

The project aimed to analyze trends and contributing factors to per capita greenhouse gas emissions across countries. To achieve this, a structured approach was designed with the following objectives:

- **Objective 1:** Collect relevant datasets containing greenhouse gas emissions, population, GDP, and urban population share.
- **Objective 2:** Preprocess and merge datasets to create a unified dataset suitable for analysis.
- **Objective 3:** Perform statistical and visual analyses to extract meaningful insights about emission trends and relationships between features.
- **Objective 4:** Leverage cloud-based tools to enhance efficiency in processing, querying, and visualizing large datasets.

- **Objective 5:** Apply predictive modeling techniques to forecast per capita greenhouse gas emissions based on the identified contributing factors, utilizing machine learning algorithms for accurate predictions and insights.

Utilized advanced cloud technologies like **Google Cloud Platform** (GCP) for storage, **Google Colab** for preprocessing, **BigQuery** for querying, and **Looker Studio** for visualization.

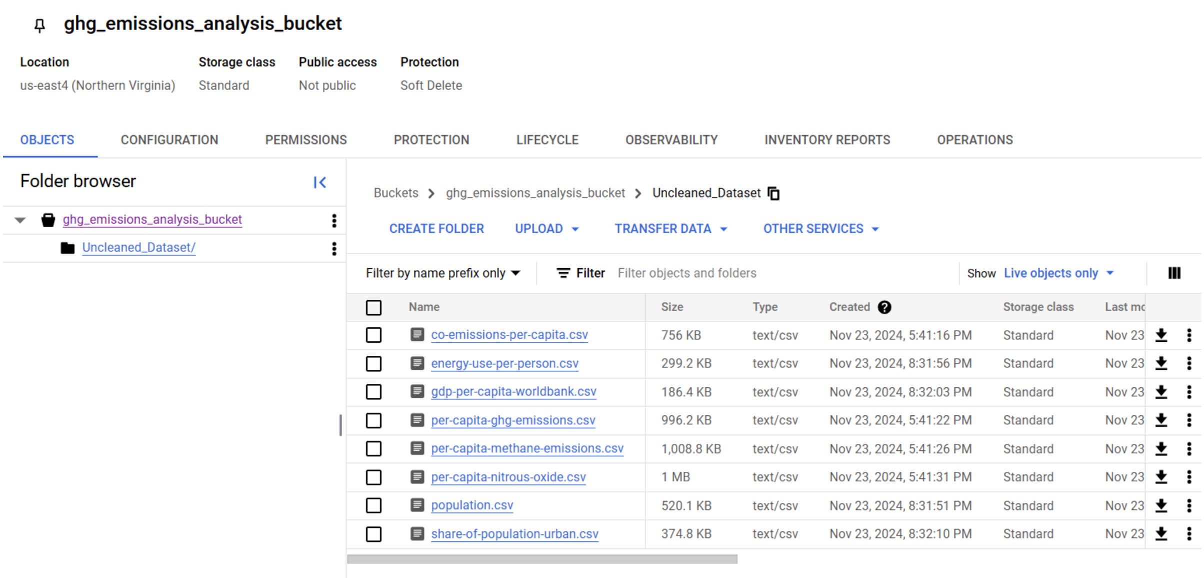
3.2 Acquire

3.2.1 Data Collection

- Data was sourced from publicly available datasets on the Our World in Data website:
    - *Greenhouse Gas Emissions, Population Growth, Economic Growth, Energy use*
- The datasets contains 42,831 records included features such as:
    - Per-Capita Greenhouse Gas Emissions: CO_2, methane (CH_4), nitrous oxide (N_2O), and overall GHG emissions per capita.
    - Demographics: Country, Population, urban population share, and GDP per capita.
    - Energy Use: Energy use per person.
    - Year
- The data was downloaded as CSV files.

3.2.2 Data Storage

- To facilitate secure and efficient storage and processing, a Google Cloud Platform (GCP) storage bucket was created.
- All downloaded CSV files were uploaded to the GCP bucket: "ghg_emissions_analysis_bucket".



3.2.3 Infrastructure - Data Loading and Cloud Access

After confirming that the data was loaded into the GCP bucket, access was established using Google Colab with the same login credentials. The data was pulled directly from the bucket to leverage the cloud's processing power. Several pre-processing steps were applied to clean and organize the data for efficient analysis.

Code to establish connection from Google Colab to the GCP bucket "ghg_emissions_analysis_bucket":

```
# Install Google Cloud SDK
!curl https://sdk.cloud.google.com | bash
```

The necessary Google Cloud libraries were added to the Colab environment to ensure smooth integration with Google Cloud Platform features.

```
# Authenticating GCP and Colab
from google.colab import auth
auth.authenticate_user()
```

```
# Setting up GCP project
!gcloud config set project 'fa24-i535-skollep-ghgemissions'
```

```
Updated property [core/project].
```

```
#storage client
storage_client = storage.Client('fa24-i535-skollep-ghgemissions')

#bucket name and folder name
bucket_name = 'ghg_emissions_analysis_bucket'
folder_name = 'Uncleaned_Dataset'

# Creating bucket object
bucket = storage_client.get_bucket(bucket_name)

# List of files to download
file_names = [
    'co-emissions-per-capita.csv',
    'per-capita-methane-emissions.csv',
    'per-capita-nitrous-oxide.csv',
    'population.csv',
    'share-of-population-urban.csv',
    'gdp-per-capita-worldbank.csv',
    'energy-use-per-person.csv',
    'per-capita-ghg-emissions.csv'
]

# Downloading and reading the CSV files into pandas DataFrames
dfs = {}
for file_name in file_names:
    # Create a blob object
    blob = bucket.blob(f'{folder_name}/{file_name}')

    # Download the contents of the blob to a local file
    local_file_path = f'/content/{file_name}'
    blob.download_to_filename(local_file_path)

    # Read the downloaded file into a pandas DataFrame
    dfs[file_name] = pd.read_csv(local_file_path)
```

3.3 Process

3.3.1 Data Preprocessing

Data preprocessing was performed using Google Colab, integrating directly with the GCP bucket for seamless access.

Merging Datasets: The datasets were merged using an outer join based on the shared features, Country and Year. This ensured no loss of relevant information.

```
df_co2 = dfs['co-emissions-per-capita.csv']
df_methane=dfs['per-capita-nitrous-oxide.csv']
df_n2o=dfs['per-capita-methane-emissions.csv']
df_total=dfs['per-capita-ghg-emissions.csv']
df_population = pd.read_csv('population.csv')
df_urbanization = pd.read_csv('share-of-population-urban.csv')
df_gdp = pd.read_csv('gdp-per-capita-worldbank.csv')
df_energy = pd.read_csv('energy-use-per-person.csv')

# Merge the datasets using outer joins to include all combinations
dfs_to_merge = [df_gdp, df_population, df_urbanization, df_co2, df_methane, df_n2o, df_energy, df_total]
df_combined = dfs_to_merge[0]
for df in dfs_to_merge[1:]:
    df_combined = pd.merge(df_combined, df, on=["Entity", "Code", "Year"], how="outer")
```

Column Renaming: Columns were standardized with meaningful names for consistency and ease of analysis.

```
df_combined = df_combined.rename(columns={
    "Annual CO₂ emissions (per capita)": "CO₂ emissions",
    "Per capita methane emissions in CO₂ equivalents": "Methane emissions",
    "Per capita nitrous oxide emissions in CO₂ equivalents": "N2O emissions",
    "Per capita greenhouse gas emissions in CO₂ equivalents": "GHG emissions per capita",
    "Population - Sex: all - Age: all - Variant: estimates": "Population",
    "Urban population (% of total population)": "Urban share",
    "GDP per capita, PPP (constant 2017 international $)": "GDP per capita",
    "Primary energy consumption per capita (kWh/person)": "Energy use per person"
})

df_combined.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 42831 entries, 0 to 42830
Data columns (total 11 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Entity                    42831 non-null  object
 1   Code                      37838 non-null  object
 2   Year                      42831 non-null  int64
 3   GDP per capita            6562 non-null   float64
 4   Population                18944 non-null  float64
 5   Urban share               14427 non-null  float64
 6   CO₂ emissions             26182 non-null  float64
 7   N2O emissions             36320 non-null  float64
 8   Methane emissions         35813 non-null  float64
 9   Energy use per person     10694 non-null  float64
 10  GHG emissions per capita  35813 non-null  float64
dtypes: float64(8), int64(1), object(2)
memory usage: 3.6+ MB
```

3.3.2 Data Cleaning

1. **Filtering Unusable Data:**
   - Rows with data before 1980 were removed due to high levels of missing or irrelevant data.
   - Non-useful columns, such as Country Code, were eliminated.

```
#eliminating years before 1980 due to indequate data for some columns
df_combined = df_combined[df_combined["Year"] > 1980]
#elimnating code column for country as it of no use, as we have country (Entity) column
df_combined = df_combined.drop(columns=["Code"])

df_combined.head()
```

	Entity	Year	GDP per capita	Population	Urban share	CO₂ emissions	N2O emissions	Methane emissions	Energy use per person	GHG emissions per capita
131	Afghanistan	1981	NaN	11937587.0	16.562	0.165734	0.278202	0.972430	786.83690	1.687558
132	Afghanistan	1982	NaN	10991382.0	17.147	0.190566	0.306399	1.045692	926.65125	1.804140
133	Afghanistan	1983	NaN	10917986.0	17.747	0.230808	0.290531	1.009258	1149.19590	1.782830
134	Afghanistan	1984	NaN	11190220.0	18.365	0.252143	0.268575	0.900400	1121.57290	1.643149
135	Afghanistan	1985	NaN	11426855.0	18.997	0.306420	0.244525	0.817104	1067.07090	1.565640

2. **Handling Null Values:**
   - Countries with no data for any variable across all years were identified and removed.

```
#null values handling
# List of variables to check for complete missingness by country
variables = [
    "GDP per capita", "Population", "Urban share", "CO₂ emissions",
    "N2O emissions", "Methane emissions", "Energy use per person", "GHG emissions per capita"
]

# Get the number of unique years in the dataset
unique_years_count = df_combined["Year"].nunique()

# Identify countries to remove where all values for any variable are missing across all years
countries_to_remove = set()
for variable in variables:
    missing_counts = df_combined[df_combined[variable].isnull()].groupby("Entity").size()
    missing_countries = missing_counts[missing_counts == unique_years_count].index
    countries_to_remove.update(missing_countries)

# Remove the identified countries from the dataset
df_combined = df_combined[~df_combined["Entity"].isin(countries_to_remove)]
```

   - Remaining missing values in numerical columns were filled with the median of the respective column to minimize bias

```
#handling remaining null values
# Fill numerical columns
numerical_columns = [
    'GDP per capita', 'Population', 'Urban share',
    'CO₂ emissions', 'Methane emissions', 'N2O emissions',
    'Energy use per person', 'GHG emissions per capita'
]

for col in numerical_columns:
    df_combined[col] = df_combined[col].fillna(df_combined[col].median())

# Check remaining nulls (if any)
print(df_combined.isnull().sum())
```

```
Entity                      0
Year                        0
GDP per capita              0
Population                  0
Urban share                 0
CO₂ emissions               0
N2O emissions               0
Methane emissions           0
Energy use per person       0
GHG emissions per capita    0
dtype: int64
```

3. Outliers Removal

Outliers in key numerical features (e.g., CO_2 emissions, CH_4 emissions) were identified using the 0.5th and 99.5th percentiles and filtered out.

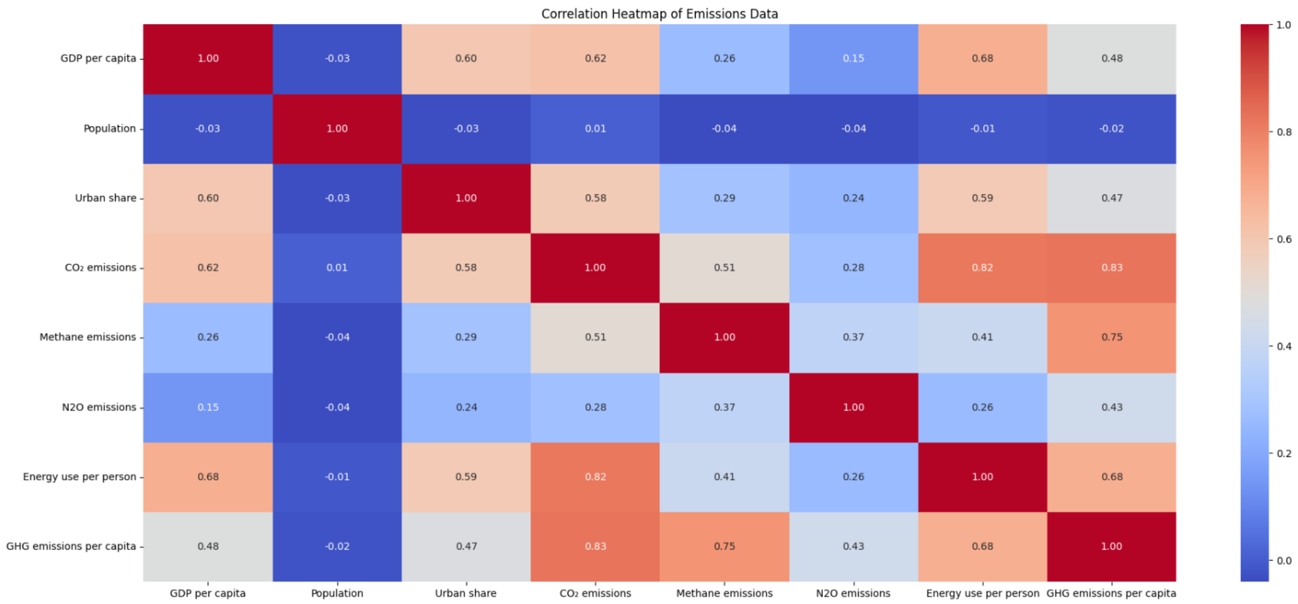
```
#Removing Outliers using Percentile Cutoffs
def remove_outliers_percentiles(df, column_list):
    for column in column_list:
        lower_percentile = df[column].quantile(0.005)
        upper_percentile = df[column].quantile(0.995)

        # Filter out outliers
        df = df[(df[column] >= lower_percentile) & (df[column] <= upper_percentile)]
    return df

# Columns to check for outliers
columns_to_check=['CO₂ emissions', 'Methane emissions', 'N2O emissions','GHG emissions per capita']

# Remove outliers using percentile cutoffs
preprocessed_data = remove_outliers_percentiles(df_combined, columns_to_check)
```

3.4 Analyze

3.4.1 Correlation Heatmap

A correlation matrix was generated to identify relationships between numerical features, such as GDP per capita and emissions.

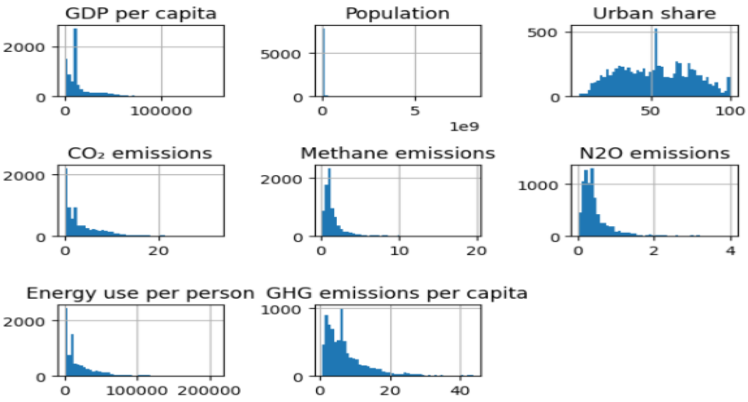
```
# Correlation heatmap
plt.figure(figsize=(23, 10))
corr = df[numerical_columns].corr()
sns.heatmap(corr, annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Correlation Heatmap of Emissions Data")
plt.show()
```



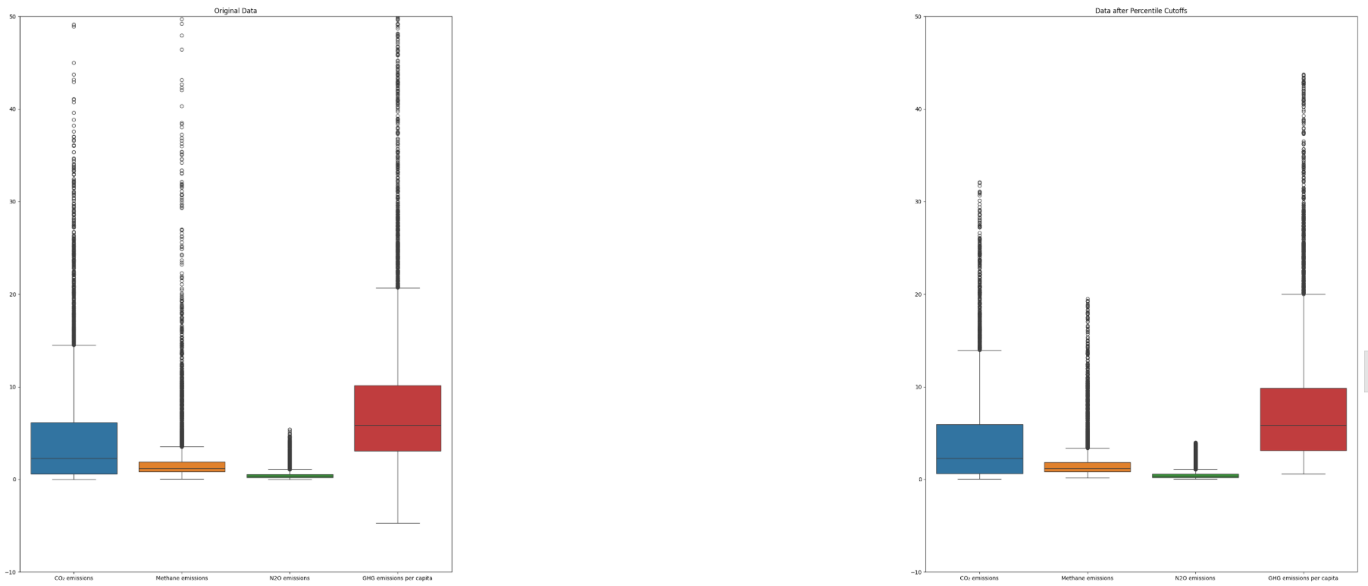
3.4.2 Visualizing Distributions

a) Histograms and boxplots were plotted for numerical columns.

```python
df=preprocessed_data

#histograms
df[numerical_columns].hist(bins=50)
plt.tight_layout()
plt.show()
```

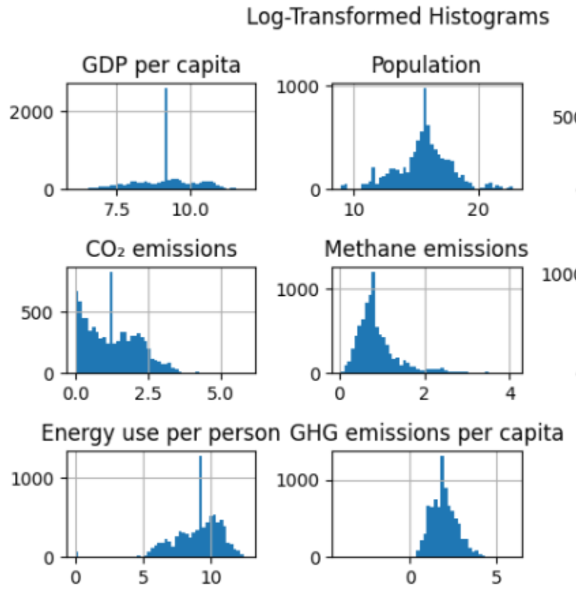



```python
# Plotting data before and after outlier removal to compare
plt.figure(figsize=(35, 15))

# Original Data Boxplot
plt.subplot(1, 3, 1)
sns.boxplot(data=df_combined[columns_to_check])
plt.title('Original Data')
#plt.xticks(rotation=45)
plt.ylim(-10, 50)

# Data after Percentile Cutoffs Boxplot
plt.subplot(1, 3, 3)
sns.boxplot(data=preprocessed_data[columns_to_check])
plt.title('Data after Percentile Cutoffs')
#plt.xticks(rotation=45)
plt.ylim(-10, 50)

plt.tight_layout()
plt.show()
```



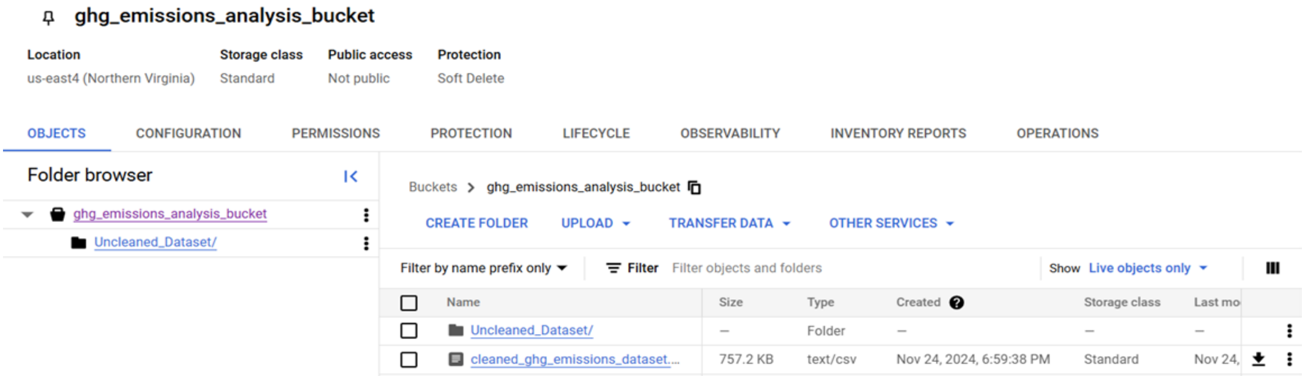
b) Due to skewness in some features, log-based histograms were created for better interpretation.

```
# Apply log transformation
df_log_transformed = df_combined[numerical_columns].apply(np.log1p)
df_log_transformed.hist(bins=50)
plt.suptitle('Log-Transformed Histograms')
plt.tight_layout()
plt.show()
```



3.4.3 Storing Processed Data

The cleaned and pre-processed dataset was saved back to the GCP bucket.

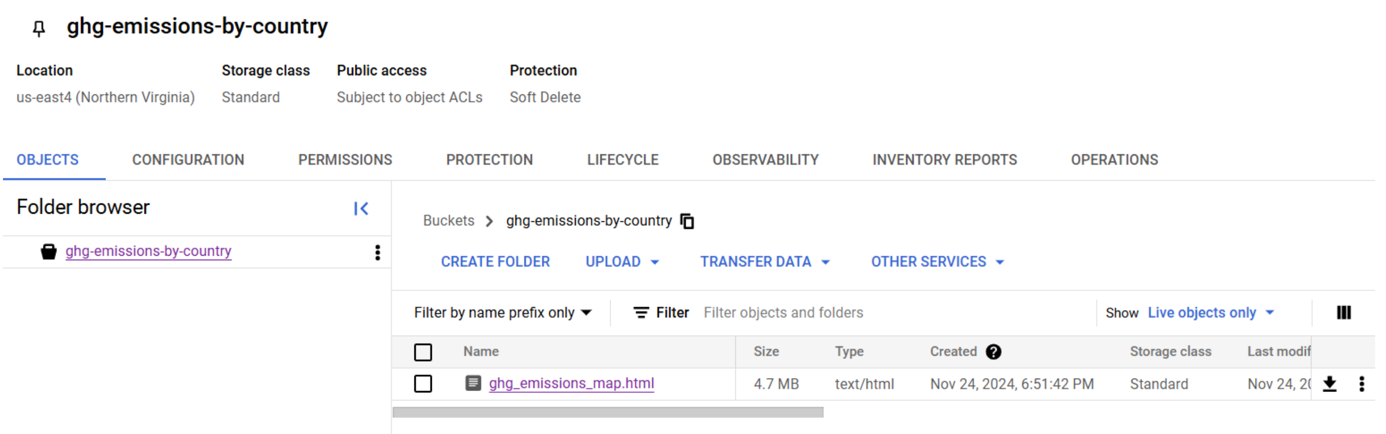


3.4.4 Choropleth Map

A choropleth map was created using Plotly to visualize the Greenhouse emissions per capita by country. The map animated over years to show the trends in GHG emissions globally. This map was saved as an HTML file and uploaded to the GCP bucket for later sharing and visualization.

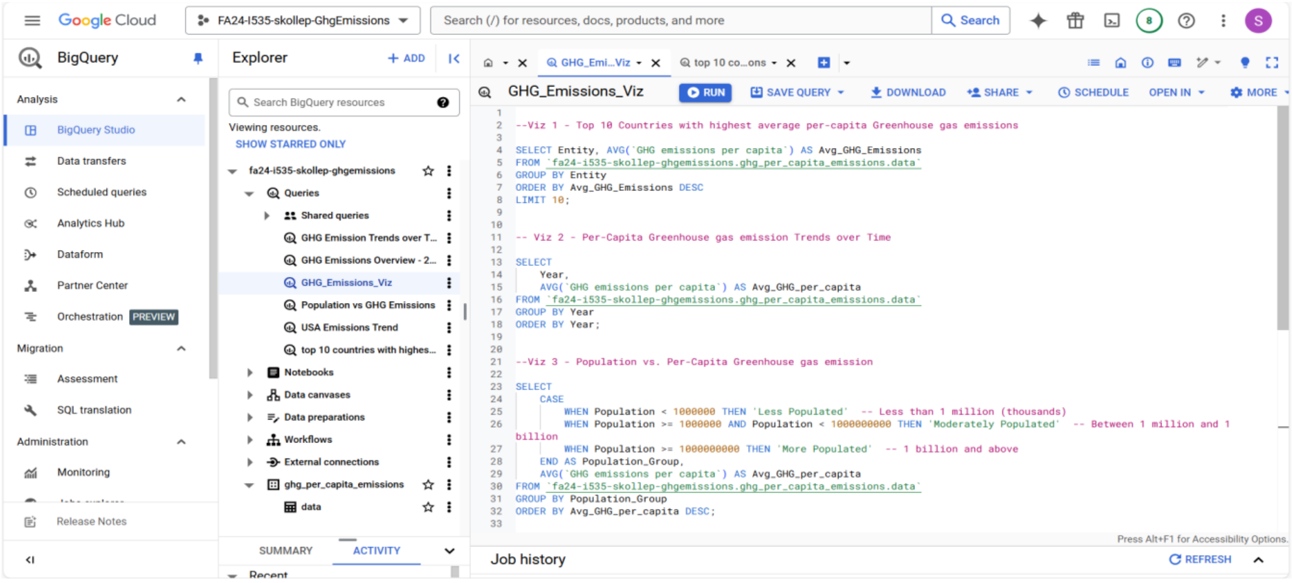
```python
# Choropleth map for GHG emissions per capita
fig = px.choropleth(df,
                    locations="Entity",
                    locationmode="country names",
                    color="GHG emissions per capita",
                    hover_name="Entity",
                    animation_frame="Year",
                    color_continuous_scale=px.colors.sequential.Cividis,
                    projection="natural earth",
                    title="Per Capita Greenhouse Gas Emissions by Country"
)
fig.update_layout(
    geo=dict(showcoastlines=True, coastlinecolor="Black", showland=True, landcolor="white"),
    coloraxis_colorbar_title="GHG Emissions per Capita",
    width=1300,
    height=800,
)
fig.show()

# Save the plot as an HTML file
fig.write_html('ghg_emissions_map.html')

# Upload the map in html format to GCP bucket
bucket = storage_client.get_bucket(bucket_name)
blob = bucket.blob('ghg_emissions_map.html')
blob.upload_from_filename('ghg_emissions_map.html')
```

The map was uploaded to a new publicly accessible GCP bucket “ghg-emissions-by-country”, enabling it to be displayed via a shared URL.



3.4.5 Querying and Insights

The processed data was imported into BigQuery, where SQL queries were used to extract insights on emissions trends, correlations, and country-specific patterns, visualizations will be shown in results



Query 1: Top 10 Countries with highest average per-capita Greenhouse gas emissions

```sql
SELECT Entity, AVG(`GHG emissions per capita`) AS Avg_GHG_Emissions
FROM `fa24-i535-skollep-ghgemissions.ghg_per_capita_emissions.data`
GROUP BY Entity
ORDER BY Avg_GHG_Emissions DESC
LIMIT 10;
```

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	Entity	Avg_GHG_Emissions
1	Bahrain	42.06315925
2	Brunei	39.08326364285...
3	United Arab Emirates	34.51756091428...
4	Kuwait	33.25580632258...
5	Australia	30.28067799999...
6	Trinidad and Tobago	26.22702306060...
7	Saudi Arabia	26.12110320930...
8	Canada	26.06492281395...
9	Luxembourg	23.88364469767...
10	United States	22.32229381395...

Query 2: Per-Capita Greenhouse gas emission Trends over Time

```sql
SELECT
    Year,
    AVG(`GHG emissions per capita`) AS Avg_GHG_per_capita
FROM `fa24-i535-skollep-ghgemissions.ghg_per_capita_emissions.data`
GROUP BY Year
ORDER BY Year;
```

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	Year	Avg_GHG_per_capita
1	1981	8.320711099838...
2	1982	7.929220801793...
3	1983	8.250603498983...
4	1984	7.897240889893...
5	1985	7.977580439946...
6	1986	8.023816268260...
7	1987	8.282507756296...
8	1988	8.281365566842...
9	1989	7.888905471755...
10	1990	8.151115748549...
11	1991	7.863065162751...

Results per page: 50 1 – 43 of 43

Query 3: Population vs. Per-Capita Greenhouse gas emissions

```sql
SELECT
    CASE
        WHEN Population < 1000000 THEN 'Less Populated'  -- Less than 1 million (thousands)
        WHEN Population >= 1000000 AND Population < 1000000000 THEN 'Moderately Populated'  -- Between 1 million and 1 billion
        WHEN Population >= 1000000000 THEN 'More Populated'  -- 1 billion and above
    END AS Population_Group,
    AVG(`GHG emissions per capita`) AS Avg_GHG_per_capita
FROM `fa24-i535-skollep-ghgemissions.ghg_per_capita_emissions.data`
GROUP BY Population_Group
ORDER BY Avg_GHG_per_capita DESC;
```

Press Alt+F1 for Accessibi

Query results

SAVE RESULTS   EXPLORE DATA

JOB INFORMATION   RESULTS   CHART   JSON   EXECUTION DETAILS   EXECUTION GRAPH

Row	Population_Group	Avg_GHG_per_capita
1	Moderately Populated	7.955503362908...
2	More Populated	6.966289047008...
3	Less Populated	6.853009184731...

Query 4: USA Per-Capita Greenhouse gas emission Trends over Time

```sql
SELECT
    Entity AS Country,
    Year,
    `CO₂ emissions` AS CO2_Emissions,
    `Methane emissions` AS Methane_Emissions,
    `N2O emissions` AS N2O_Emissions,
    `GHG emissions per capita` AS GHG_Emissions
FROM `fa24-i535-skollep-ghgemissions.ghg_per_capita_emissions.data`
WHERE Entity="United States";
```

Press Alt+F1 for Acces:

Query results

SAVE RESULTS   EXPLORE DATA

JOB INFORMATION   RESULTS   CHART   JSON   EXECUTION DETAILS   EXECUTION GRAPH

Row	Country	Year	CO2_Emissions	Methane_Emissions	N2O_Emissions	GHG_Emissions
1	United States	1981	20.183939	3.7665575	1.2114595	25.294073
2	United States	1982	18.959597	3.7076	1.131031	23.910803
3	United States	1983	18.69184	3.5421162	1.1362886	23.519325
4	United States	1984	19.47947	3.5600061	1.1420525	24.327587
5	United States	1985	19.251644	3.491274	1.1034034	23.928703
6	United States	1986	19.11719	3.4167488	1.0690699	23.47883

Results per page: 50   1 – 43 of 43

Query 5: 2023 Global Per-Capita Greenhouse gas emissions Overview

```sql
SELECT
    Entity AS Country,
    Year AS Recent_Year,
    `GHG emissions per capita` AS GHG_Emissions
FROM `fa24-i535-skollep-ghgemissions.ghg_per_capita_emissions.data`
WHERE Year = (SELECT MAX(Year) FROM `fa24-i535-skollep-ghgemissions.ghg_per_capita_emissions.data`)
ORDER BY Country;
```

Query results

SAVE RESULTS

JOB INFORMATION RESULTS CHART JSON EXECUTION DETAILS EXECUTION GRAPH

Row	Country	Recent_Year	GHG_Emissions
1	Afghanistan	2023	0.9668713
2	Albania	2023	2.7290974
3	Algeria	2023	6.122596
4	Angola	2023	4.7541127
5	Antigua and Barbuda	2023	7.683062
6	Argentina	2023	9.267155

3.5 Predictions

Once the dataset was cleaned and preprocessed, the next step was to prepare the features for modeling. This included separating the target variable, encoding categorical variables, and standardizing numerical features.

```python
features = df.drop(columns=["GHG emissions per capita"])
target = df["GHG emissions per capita"]

# One-hot encoding for categorical features
features = pd.get_dummies(features, columns=["Entity"], drop_first=True)

# Standardize numerical features
scaler = StandardScaler()
features_scaled = scaler.fit_transform(features)
```

Two machine learning models were trained to predict GHG emissions per capita:

Linear Regression: A linear regression model was trained on the preprocessed data. The model was evaluated using metrics such as the R-squared (R^2) value and Mean Squared Error (MSE).

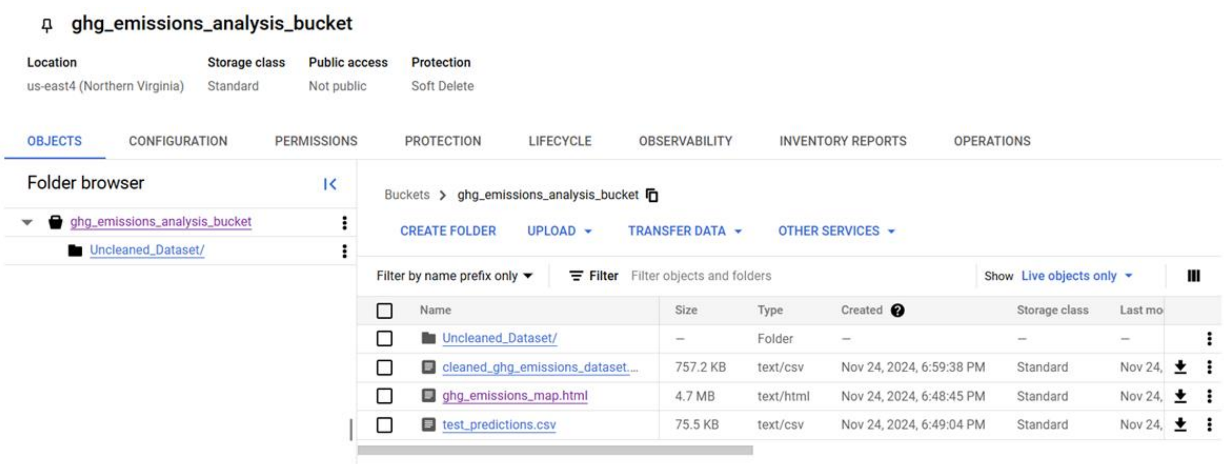
Gradient Boosting Regressor: A Gradient Boosting Regressor model was also trained on the dataset. This model generally performs better than linear regression on non-linear datasets due to its ability to capture complex relationships. Similar to the linear regression model, the Gradient Boosting model was evaluated using R^2 and MSE.

Both models were tested on a hold-out test set (20% of the data), and their performance was evaluated.

```
#train-test split
X_train, X_test, y_train, y_test = train_test_split(features_scaled, target, test_size=0.2, random_state=42)

#model - linear regression
model = LinearRegression()
model.fit(X_train, y_train)

#predictions
y_pred = model.predict(X_test)

## Evaluating performance
print("Linear Regression:")
r2 = r2_score(y_test, y_pred)
print(f"R-squared (R²): {r2}")
mse = mean_squared_error(y_test, y_pred)
print(f"MSE: {mse}")

print("\nTarget variable range:", target.min(), "-", target.max())
print("Mean of target variable:", target.mean())


#Gradient Boosting Regressor
gb = GradientBoostingRegressor(random_state=42, n_estimators=100, learning_rate=0.1)
gb.fit(X_train, y_train)
y_pred_gb = gb.predict(X_test)

# Calculating Metrics
gb_r2 = r2_score(y_test, y_pred_gb)
gb_mse = mean_squared_error(y_test, y_pred_gb)
```

3.6 Preserve

The cleaned and preprocessed dataset, as well as the model predictions, were stored in the GCP bucket for preservation and future use.



3.7 Publish

Uploaded the cleaned data and Python scripts to a GitHub repository, providing an opportunity for others to explore, improve, and derive insights from my work. This marks the completion of my project and makes the resources available for further use and development. The repository can be found here.

4. Results

4.1 BigQuery Integration and Looker Studio Visualizations

Query 1: Top 10 Countries with highest average per-capita Greenhouse gas emissions

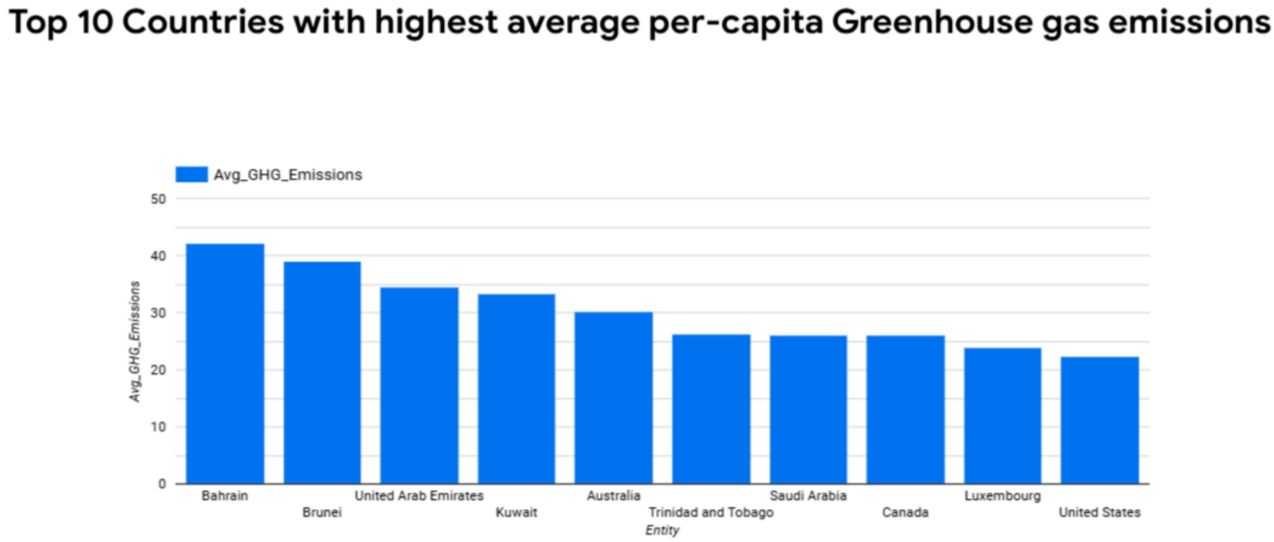


Fig 1

Query 2: Per-Capita Greenhouse gas emission Trends over Time

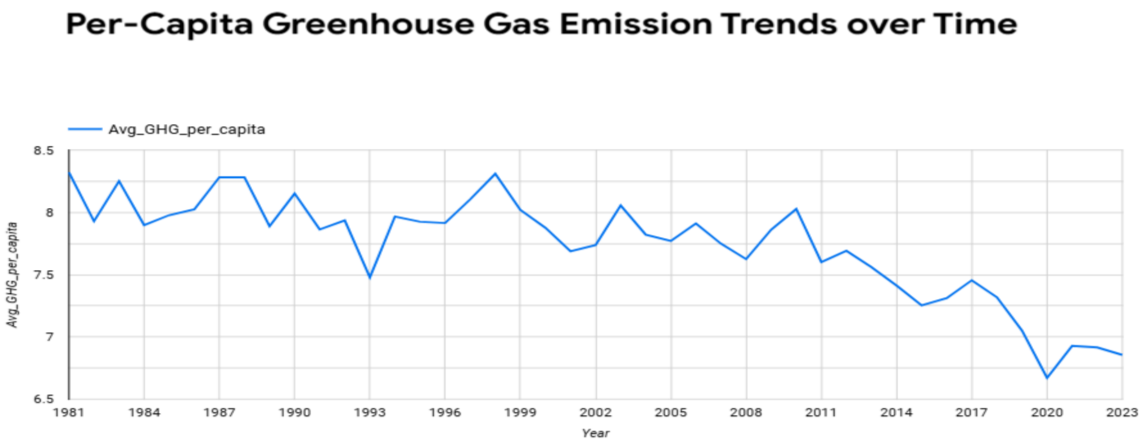


Fig 2

Query 3: Population vs. Per-Capita Greenhouse gas emission

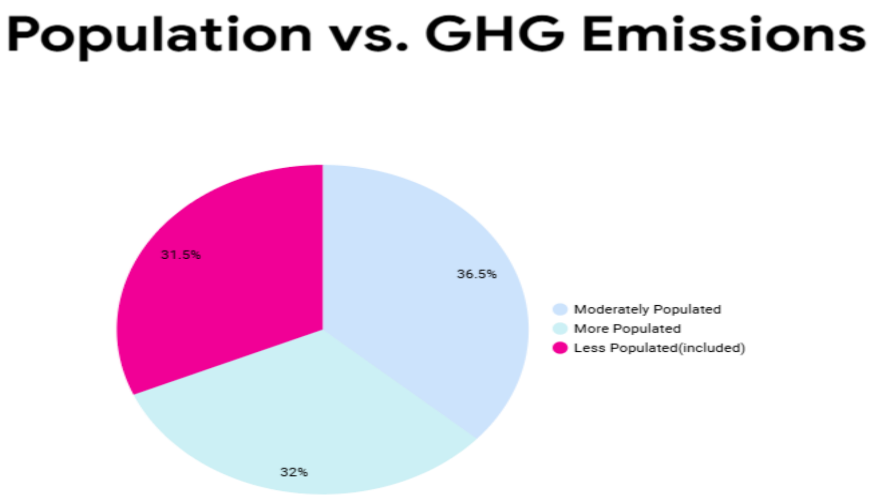


Fig 3

Query 4: USA Per-Capita Greenhouse gas emission Trends over Time

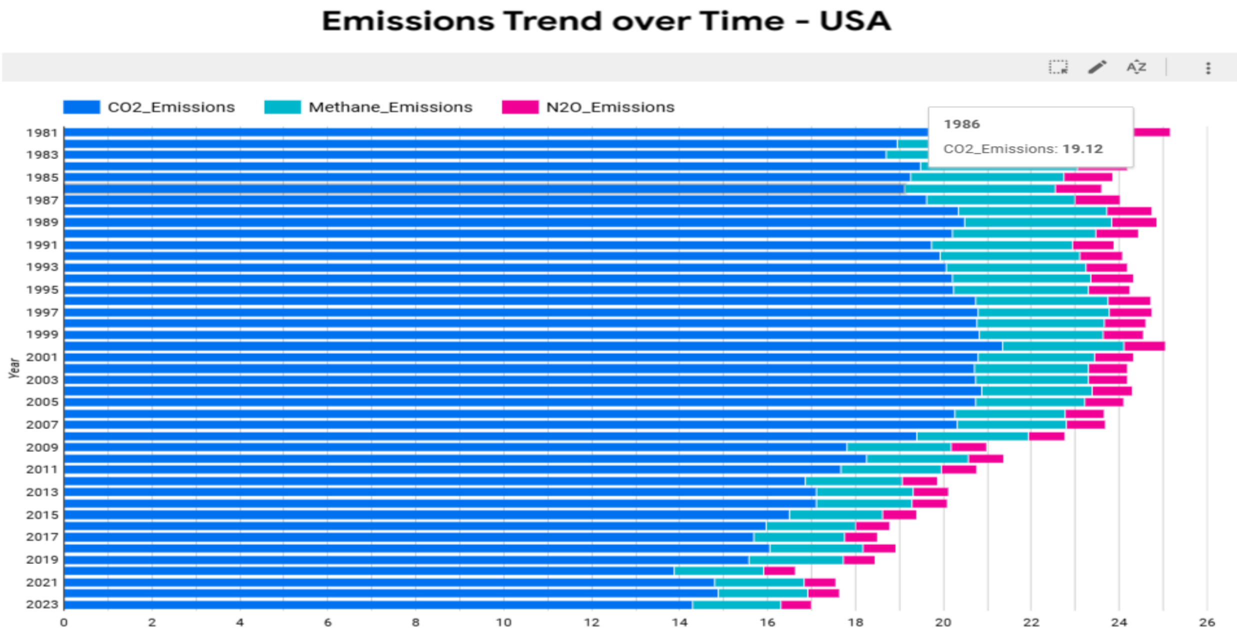


Fig 4

Query 5: 2023 Global Per-Capita Greenhouse gas emissions Overview

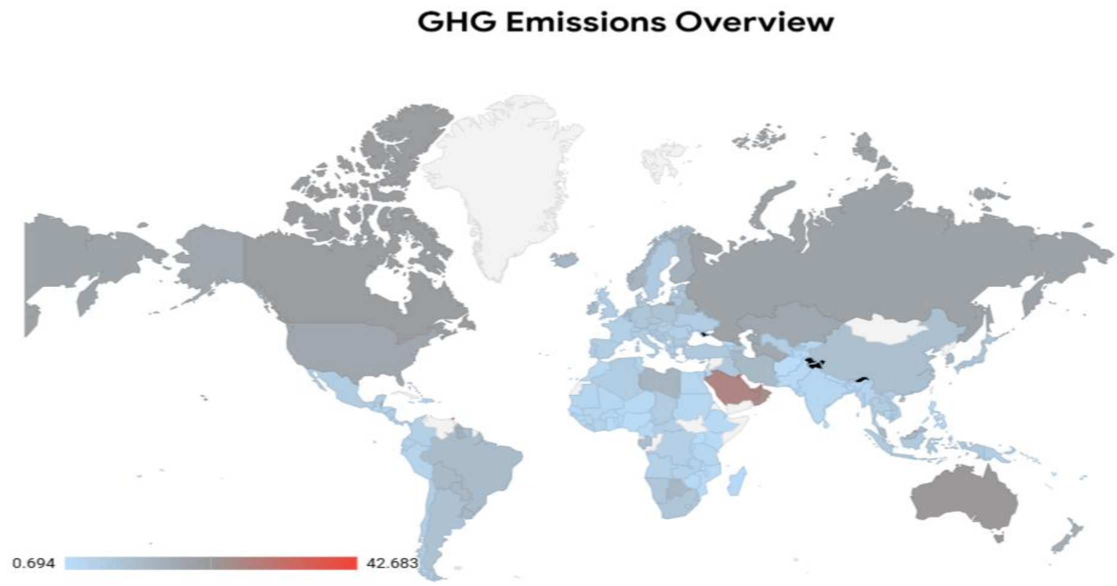
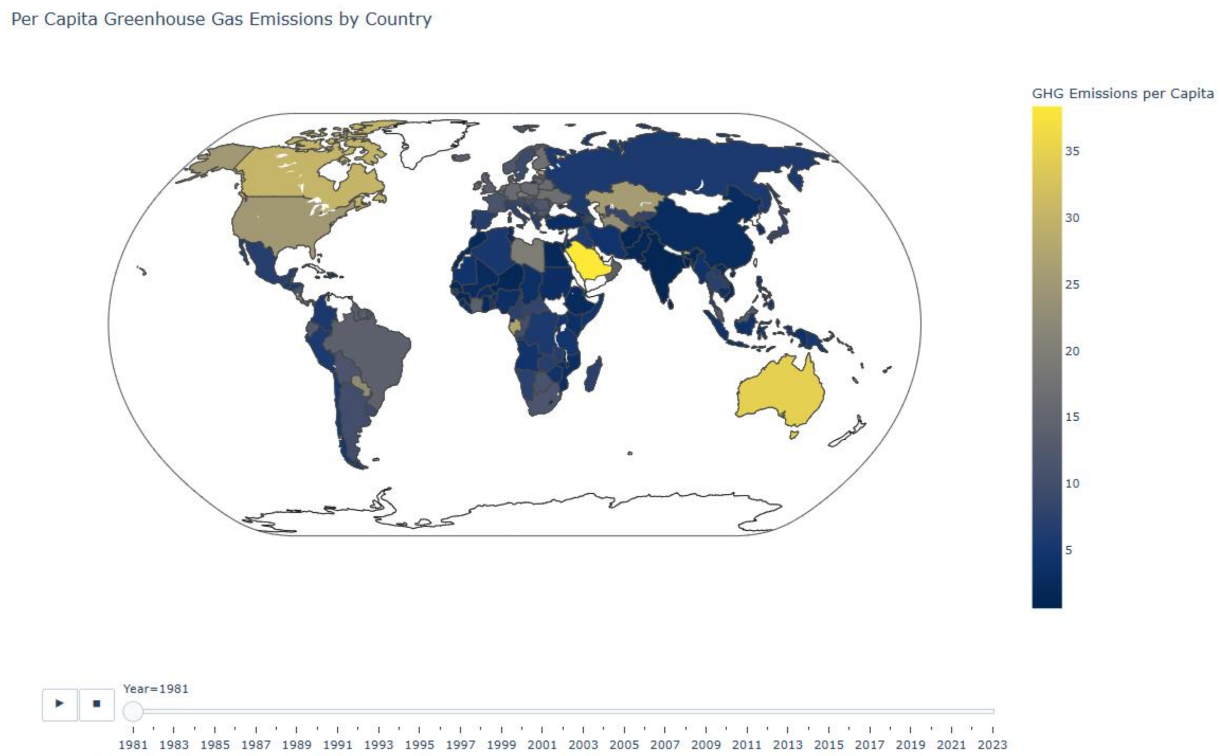


Fig 5

4.2 Choropleth Map

Per Capita Greenhouse Gas Emissions by Country over Years



Static website launched using GCP

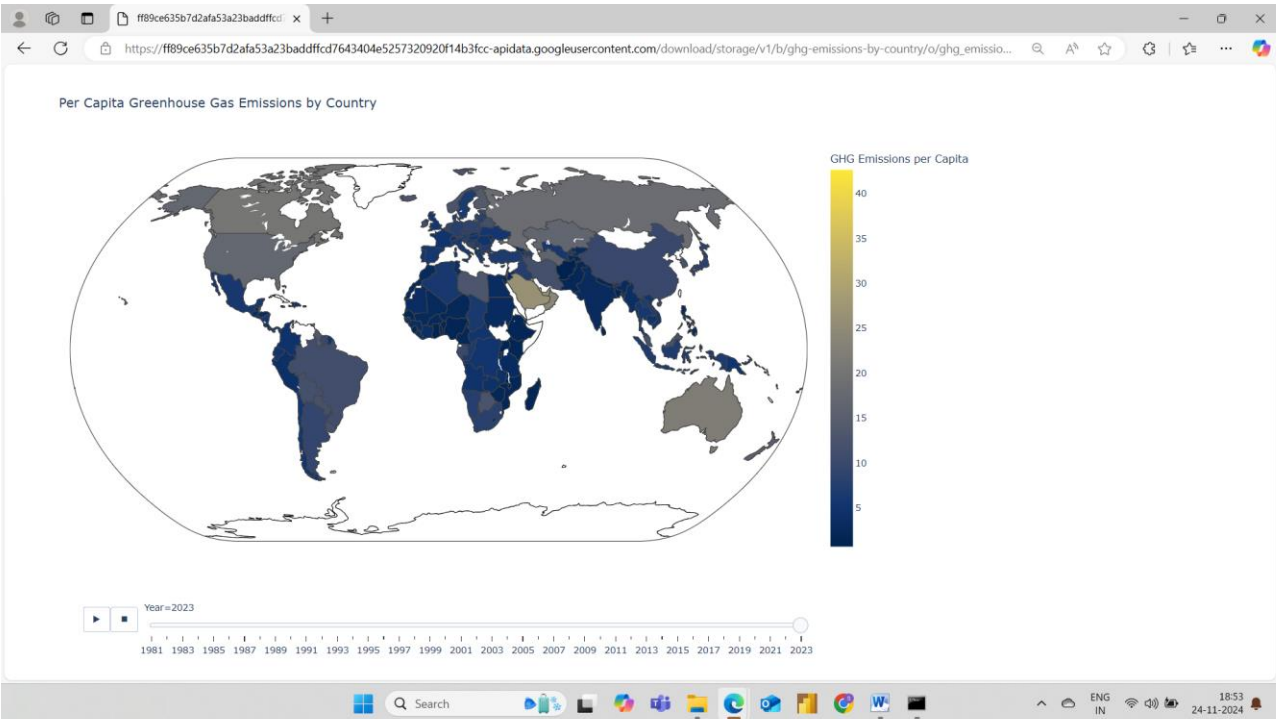


Fig 6

4.3 Linear Regression And Gradient Boost Regression

```
#model - linear regression
model = LinearRegression()
model.fit(X_train, y_train)

#predictions
y_pred = model.predict(X_test)

## Evaluating performance
print("Linear Regression:")
r2 = r2_score(y_test, y_pred)
print(f"R-squared (R²): {r2}")
mse = mean_squared_error(y_test, y_pred)
print(f"MSE: {mse}")

print("\nTarget variable range:", target.min(), "-", target.max())
print("Mean of target variable:", target.mean())


#Gradient Boosting Regressor
gb = GradientBoostingRegressor(random_state=42, n_estimators=100, learning_rate=0.1)
gb.fit(X_train, y_train)
y_pred_gb = gb.predict(X_test)

# Calculating Metrics
gb_r2 = r2_score(y_test, y_pred_gb)
gb_mse = mean_squared_error(y_test, y_pred_gb)

# Print Metrics
print("\nGradient Boosting:")
print(f"R² Score: {gb_r2}")
print(f"MSE: {gb_mse}")
```

```
Linear Regression:
R-squared (R²): 0.9619176642883988
MSE: 1.747533475027548

Target variable range: 0.6038687 - 43.737896
Mean of target variable: 7.735305173357646

Gradient Boosting:
R² Score: 0.9505955465095186
MSE: 2.2670861615181033
```

Fig 7

5. Discussion

1. Figure 1 – This bar chart displays the top 10 countries with the highest average per-capita greenhouse gas emissions over all years in the dataset. Countries such as the UAE, Australia, Canada, and the USA are among the highest emitters, offering a clear view of the countries most contributing to per-capita emissions.

2. Figure 2 – This line chart shows the global average per-capita greenhouse gas emissions from 1981 to 2023. By averaging the emissions for all countries over the years, it reveals a noticeable downward trend, indicating a reduction in per-capita emissions in more recent years, suggesting global progress in reducing emissions.

3. Figure 3 – In this pie chart, the global population is divided into three categories—countries with populations in the thousands, millions, and billions. It illustrates that, regardless of population size, the per-capita greenhouse gas emissions are relatively similar across these different population groups, showing that population size does not significantly alter the per-capita emissions.

4. Figure 4 – This stacked bar chart depicts the trends in per-capita emissions for three main greenhouse gases—CO_2, Methane, and N_2O—in the USA from 1981 to 2023. The chart shows a significant decrease in emissions over the years, with CO_2 consistently being the largest contributor to total emissions, followed by Methane and N_2O.

5. Figure 5 – This filled geo map visualizes the per-capita greenhouse gas emissions for all countries in 2023. The map uses a color gradient with sky blue indicating low emissions, grey for moderate emissions, and red for high emissions. Most countries fall into the moderate emissions category, with only one country in the high emissions range. Countries with missing data are represented in white.

6. Figure 6 – This choropleth map, using a natural earth projection, shows the evolution of per-capita greenhouse gas emissions by country over the years, from 1981 to 2023. The map plays like a video, allowing users to pause or select specific years to analyze emissions trends. The Cividis colormap is used for enhanced visualization, and hovering over individual countries reveals precise emission values, offering an interactive and detailed look at emissions data across time. The map was uploaded to a publicly accessible GCP bucket, enabling it to be displayed via a *shared URL*.

7. Figure 7 - The predictive modeling results show how well the models predict per capita greenhouse gas emissions, with the target variable (GHG emissions per capita) ranging from 0.60 to 43.74, and a mean of 7.74.
   - Linear Regression:

- R^2: 0.9619: This indicates that 96.19% of the variance in the target variable is explained by the model, demonstrating a strong fit.
- MSE: 1.7475: A relatively low MSE shows good prediction accuracy. The range of the target variable suggests that the model can handle a wide range of emission values effectively.

- Gradient Boosting:
  - R^2: 0.9506: While slightly lower than Linear Regression, this value still indicates a strong model performance, explaining 95.06% of the variance in the data.
  - MSE: 2.2671: This is higher than Linear Regression, indicating slightly less precision but still within an acceptable range.
- Significance: The target variable's range (0.60 - 43.74) and mean (7.74) reflect the variability in emissions across countries. Both models show strong predictive performance, with Linear Regression performing slightly better in terms of R^2 and MSE. The results highlight that the models effectively capture the emission trends, though Gradient Boosting shows a slightly higher error margin.

Overall, several key insights can be drawn from the analysis:

- **Decline in Per-Capita Greenhouse Gas Emissions:** Across nearly all countries, there has been a consistent year-on-year decrease in per-capita greenhouse gas emissions. This trend reflects global efforts and improvements in reducing emissions, likely driven by changes in technology, policy, and shifts towards greener practices.
- **Inverse Relationship Between Population and Per-Capita Emissions**: The population feature exhibits a nearly perfect negative correlation with per-capita greenhouse gas emissions. This is expected, as per-capita emissions are calculated by dividing total emissions by population. As the population increases, the per-capita emissions decrease, indicating an inverse relationship between these two variables.
- **Dominance of CO_2 Emissions:** Among the various greenhouse gases considered, CO_2 emissions are consistently the largest contributors to total emissions in every country. This is in line with global patterns where fossil fuel consumption, particularly for energy and transportation, is the primary source of CO_2 emissions, highlighting the critical role of reducing CO_2 in mitigating climate change.

5.1 Skills Implemented

Throughout the course of this project, the concepts learned from the course played a pivotal role in shaping my approach and guiding the execution. The modules on 'Lifecycles and Pipelines' and 'Ingest and Storage' were particularly helpful in structuring my data preprocessing workflow. These modules provided valuable insights on how to efficiently handle raw data, ensuring a seamless transition to clean, structured datasets that could be analyzed effectively.

The knowledge gained from the 'Modeling' module proved essential for performing statistical analysis and developing predictive models. It also informed the creation of meaningful visualizations that illustrated trends and relationships within the data. This was key in deriving actionable insights from the greenhouse gas emissions data.

Additionally, the 'Computing Principles and System Design' module provided the foundation for setting up my work environment in Google Colab, enabling me to work systematically and code more efficiently. It helped me understand how to structure my project for scalability and ease of use, while also ensuring optimal performance in executing tasks.

This project was an ideal opportunity to apply the academic knowledge gained from the course in a real-world context. It underscored the practical value of the skills and concepts learned, highlighting how theoretical knowledge can be transformed into impactful outcomes when applied effectively.

5.2 Challenges Encountered

One of the main challenges faced was the lack of a pre-existing dataset for the project topic. After extensive searching, no suitable dataset was found, which led to the decision to create one from scratch. Initially, there was uncertainty about which features to include, as the data for greenhouse gas emissions can vary widely. However, by focusing on key greenhouse gases along with features like country, year, population growth, and economic indicators, a meaningful dataset and a robust model were developed for analysis.

Another challenge encountered was connecting Google Colab to Google Cloud Platform (GCP), an unfamiliar process. The integration presented several errors, which were resolved through troubleshooting and experimentation. While it required significant time and effort, overcoming these obstacles provided valuable learning experiences and contributed to the success of the project.

6. Conclusion

In conclusion, this project successfully analyzed per capita greenhouse gas emissions across countries by a comprehensive dataset. Statistical analysis and predictive modelling revealed key trends, such as the inverse relationship between population size and per capita emissions and the dominance of CO2 emissions. The project leveraged Google Colab for seamless coding and integration with Google Cloud Platform (GCP), where data storage, processing, and analysis were efficiently handled. BigQuery was also used for managing large datasets, enabling quicker querying and processing of complex data. Despite challenges like technical hurdles, the project provided valuable insights, demonstrating the practical application of course concepts in addressing real-world issues.

This project highlighted the value of adaptability and persistence in overcoming technical challenges. It also reinforced how the skills learned from the course, especially in using tools like Google Cloud Platform and Google Colab, can be directly applied to real-world data analysis and visualization.

7. References

1. Hannah Ritchie, Pablo Rosado and Max Roser (2020) - “Greenhouse gas emissions” Published online at OurWorldinData.org.

   Retrieved from: 'https://ourworldindata.org/greenhouse-gas-emissions' [Online Resource]

2. https://ourworldindata.org
3. https://pub.towardsai.net/connect-colab-to-gcs-bucket-using-gcsfuse-29f4f844d074
4. https://www.cloudskillsboost.google/focuses/3692?catalog_rank=%7B"rank"%3A1,"num_filters"%3A0,"has_search"%3Atrue%7D&parent=catalog&search_id=26980459
5. https://www.geeksforgeeks.org/hosting-a-static-website-on-google-cloud-storage-step-by-step-tutorial/