****

**Dynamic Analysis of Energy Consumption, Production, and Economic Growth on CO2 Emissions Across Countries**

**Technical Report, Logbook, and Reflective Discussion**

**Siddharth Kr Jaiswal**

**40425082**

**Word Count: 2532**

**Technical Report submitted in Partial Fulfillment of Degree of Masters of Science in Business Analytics**

**September 2024**

**Queen’s Business School**

Table of Contents

[1. Technical Report 2](#_Toc175992601)

[1.1 Introduction 2](#_Toc175992602)

[1.2 Solution Design and Development 4](#_Toc175992603)

[1.2.1 Choice of Programming Language and Software Packages 4](#_Toc175992604)

[1.2.2 Data Pre-processing and Encoding Methods 5](#_Toc175992605)

[1.2.3 Data Visualization 6](#_Toc175992606)

[1.2.4 Model Explanation and Evaluation Metrics 6](#_Toc175992607)

[1.3 Discussion 7](#_Toc175992608)

[1.3.1 Innovations of This Study Compared to Existing Literature 7](#_Toc175992609)

[1.3.2 Alternative Plans When Developing the Technical Solutions 8](#_Toc175992610)

[1.3.3 Improving Model Performance 9](#_Toc175992611)

[1.4 Conclusion 9](#_Toc175992612)

[2. Logbook 11](#_Toc175992613)

[Entry 1: Project Planning and Initiation 11](#_Toc175992614)

[Entry 2: Proposal Development and Submission 12](#_Toc175992615)

[Entry 3: Literature Review 12](#_Toc175992616)

[Entry 4: Data Cleaning and Preparation 13](#_Toc175992617)

[Entry 5: Methodology and Initial Analysis 13](#_Toc175992618)

[Entry 6: Data Analysis, Model Development, and Finalization 13](#_Toc175992619)

[3. Reflective Discussion 15](#_Toc175992620)

[References 16](#_Toc175992621)

[Appendix: Python Codes 18](#_Toc175992622)

# 1. Technical Report

## 1.1 Introduction

This technical report provides an in-depth examination of the technical components and decisions involved in developing the analytical solution for understanding the relationship between energy consumption, economic growth, and CO2 emissions. This report is built upon the methodology chapter of the research. It offers a detailed explanation of why specific technical choices were made and how they contributed to achieving the research objectives.

The dataset used in this research provides a detailed account of the variables influencing CO2 emissions on a global scale, with data spanning several years and encompassing many countries. The first dataset is the Global Data on Sustainable Energy (2000-2020) which contains 3649 rows with 21 variables. The second dataset is the Energy and CO2 Emissions (1980-2022) which has 55440 rows of observations with 11 variables. It captures comprehensive information on energy production, consumption, and CO2 emissions across various countries and years. These datasets were carefully chosen for their relevance to the research objectives and extensive coverage.

The analysis involved developing a robust technical solution to predict and understand CO2 emissions based on current trends in energy production, consumption, and economic variables (GDP). To achieve this, several advanced analytical techniques and machine learning models have been employed. These techniques include linear regression for exploring relationships, ARIMA modeling for forecasting future CO2 emissions, and K-Means clustering for categorizing countries based on their energy and economic profiles. Also, the study incorporated Exploratory Data Analysis to visualize and understand the distribution and relationships within the data. This ensured that the findings were both accurate and contextually relevant.

The remaining part of the report is organized into distinct sections to cover various aspects of the technical solution:

* **Solution Design and Development**: This chapter details the technical decisions made throughout the project. This includes the choice of programming language and libraries, data pre-processing steps, and the chosen models. The chapter explains why Python was selected for its extensive support in data manipulation and machine learning, and how libraries such as Pandas, Scikit-learn, and Statsmodels were utilized to develop and evaluate the models.
* **Discussion**: This section highlights the innovative aspects of the study. It compares the chosen technical solutions with alternatives considered during the development process and discusses potential improvements to enhance model performance and accuracy. It also examines how the technical approach differs from similar studies in the literature and identifies areas for future research.
* **Conclusion**: This is the final part of the report which summarizes the key findings of the report. It emphasizes the effectiveness of the technical solutions and their contributions to understanding the dynamics between energy consumption, economic growth, and CO2 emissions. It also outlines recommendations based on the study's results and suggests possible directions for further research.

## 1.2 Solution Design and Development

## 1.2.1 Choice of Programming Language and Software Packages

In this study, Python programming Language was selected as the primary programming language. This is because of its extensive support for data analysis and machine learning tasks. Python has a versatile and robust ecosystem of libraries which makes it highly suitable for handling complex datasets and implementing machine learning models(Sundaram *et al.*, 2023). Pandas library was used for data manipulation and cleaning. Pandas provides a data frame structure that simplifies the handling of large datasets and offers a suite of built-in functions to facilitate data pre-processing(Ramos-Carreño *et al.*, 2024).

The scikit-learn library was used to build and evaluate machine learning models. This library gives a comprehensive range of algorithms and tools for model training, validation, and performance assessment(Varoquaux *et al.*, 2015). This library was particularly valuable for implementing linear regression, and clustering algorithms. For time series analysis (ARIMA), Statsmodels was used. This library provides detailed statistical models and tests which are essential for understanding trends and making forecasts(Seabold and Perktold, 2010).

Data visualization was done with the help of Matplotlib, Seaborn, and Plotly. express. Matplotlib provides extensive customization options for creating a wide array of plots, while Seaborn simplifies the creation of attractive and informative statistical graphics(Lavanya *et al.*, 2023). Plotly express library was also used for visualization because of its interactive data visualization which offers dynamic and customizable plots that enhance data exploration and presentation. R was taken into consideration because of its robust statistical analysis features, but Python was chosen because of its integrated environment, which easily supports both data analysis and visualization.

## 1.2.2 Data Pre-processing and Encoding Methods

Data pre-processing was an important component of this study to ensure the quality and relevance of the datasets. The initial steps involved handling missing values. All the rows with missing values were excluded to avoid the introduction of bias into the models. Dropping rows with missing values is important to maintain data integrity and prevent skewed analysis to ensure that the models are trained on complete and reliable datasets(Alam *et al.*, 2023).

Normalization and scaling of data were also performed to standardize units and enhance model performance. Continuous variables, such as energy consumption, energy production, and CO2 emissions, were normalized to ensure that they were on a similar scale. Normalization and scaling are important in data analysis since they ensure that all features contribute equally to the model hence improving its accuracy and performance(Singh and Singh, 2020).

Before conducting a time series analysis, the CO2 emission variable was converted into a time series format. This is very important for performing time series forecasting since it arranges data points in chronological order which allows the model to capture temporal patterns and trends. This ensures that the model can effectively predict future values by using historical data trends and temporal relationships.

### 1.2.3 Data Visualization

Data visualization was used to both understand the dataset and communicate the results. Various visualizations were employed to explore data distributions, identify trends, and assess model performance. Bar plots were used to examine the distribution of key variables, such as CO2 emissions by low, middle, and high-income countries. Visualization provides insights into the data characteristics(Srivastava, 2023).

Line plots and scatter plots helped in identifying relationships and trends between economic growth, energy production, and CO2 emissions. These visualizations provided a deeper understanding of how these variables interact over time. Also, the performance metrics of the model were visualized using tables. Using tables to visualize data is important for presenting precise numerical information in a structured format. It enables easy comparison and analysis of specific values(Midway, 2020).

### 1.2.4 Model Explanation and Evaluation Metrics

The technical solution involved several modeling techniques to analyze and predict CO2 emissions. Linear Regression was used to quantify the relationships between continuous variables and CO2 emissions. This provided a straightforward approach to understanding these relationships. ARIMA Model was used to forecast future CO2 emissions based on historical data trends. ARIMA model can handle time series data and identify underlying patterns(Ospina *et al.*, 2023). K-Means Clustering was applied to group countries based on their energy consumption and economic profiles. This clustering technique helped in identifying patterns and similarities among countries, which could inform policy decisions and strategic planning.

Model evaluation was conducted using several metrics. R-squared was used to measure the proportion of variance explained by the model. Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) were used to assess the average magnitude of errors and the overall model accuracy respectively. Moreover, cross-validation was applied to validate model robustness and prevent overfitting by splitting the dataset into training and test sets. These methods and metrics provided a comprehensive assessment of the models' performance and helped in understanding how well the energy consumption and economic growth variables predict CO2 emissions.

## 1.3 Discussion

### 1.3.1 Innovations of This Study Compared to Existing Literature

This study on the dynamic relationship between energy consumption, production, economic growth, and CO2 emissions introduces several innovations compared to existing literature. The previous research has explored the individual impacts of these factors on CO2 emissions, but this study takes a more holistic approach by integrating these variables into a comprehensive time series analysis across multiple countries (high, low, and middle-income nations). Also, another significant innovation of this study is the 10-year forecasting of CO2 emissions from 2020 to 2029. This forward-looking approach has not been undertaken by many existing studies. This long-term forecasting provides critical knowledge of future trends and the potential long-term impacts of current energy and economic policies.

Moreover, advanced data visualization techniques, like those provided by Plotly Express, were used and this enabled a more interactive and detailed exploration of relationships between variables. This offered deeper insights than traditional static charts. Interactive charts allow users to explore data in more depth by enabling dynamic filtering, zooming, and hovering for detailed insights, which static charts cannot provide(Nguyen *et al.*, 2020).

### 1.3.2 Alternative Plans When Developing the Technical Solutions

In developing the technical solutions for this study, several alternative methods were considered to ensure a thorough analysis of CO2 emissions. Multiple linear regressions were employed to quantify the effects of predictors such as energy consumption, production, and economic growth on CO2 emissions. MLR was chosen for its simplicity and clear interpretability. However, alternative regression techniques such as Ridge Regression and Lasso Regression were also considered. These methods offer the advantage of regularization, which can mitigate issues related to multicollinearity among predictors(Melkumova and Shatskikh, 2017).

For forecasting CO2 emissions, ARIMA was selected because effective in handling time series data. However, the study also explored alternatives such as Exponential Smoothing State Space Model (ETS) and Prophet. Both are capable of modeling complex seasonal and trend patterns(Sun, 2020). Furthermore, deep learning methods like Long Short-Term Memory (LSTM) networks were explored as potential alternatives for forecasting. Despite their potential for capturing complex temporal patterns, these methods were not used due to computational constraints and the need for easily interpretable results.

### 1.3.3 Improving Model Performance

A key area for future improvement in this study is the optimization of model parameters. This study primarily relied on default settings for time series models. In the future, optimizing hyper-parameters could significantly enhance model performance and accuracy. Also, advanced optimization methods like Bayesian optimization could be considered to automate the process and explore a wider range of potential solutions. Hyperparameter tuning could be considered to achieve more precise predictions of CO2 emissions, especially in the 10-year forecast from 2020 to 2029.

## 1.4 Conclusion

This technical report outlines the dynamic analysis of energy consumption, production, and economic growth on CO2 emissions in low, middle, and high-income countries, including the design and development of the technical solutions employed. The study demonstrated innovation compared to existing literature by incorporating a 10-year (2020 to 2029) forecasting of CO2 emissions. This provided a forward-looking perspective that many prior studies lacked. However, certain limitations exist. This includes the reliance on historical data which may not fully capture future shifts in energy production and consumption patterns. Also, the exclusion of more advanced machine learning techniques due to computational constraints leaves room for future improvement. Future research should consider integrating these advanced methods and expansion of the scope to include more diverse global data sources to enhance the robustness and applicability of the findings.

# 2. Logbook

This logbook contains six entries that summarize the key activities, decisions, and technical steps taken during the project. These entries provide a clear understanding of the development process of the work conducted.

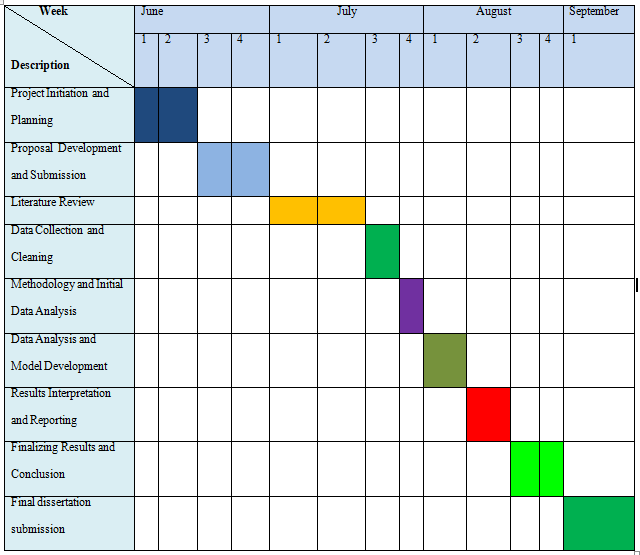


Figure : Logbook Timeline

## Entry 1: Project Planning and Initiation

**Week 1 and 2, June 2024**

The project began with the planning and initiation phase, which involved outlining the research objectives, defining the scope, and setting the timeline. The goal was to analyze the dynamic relationships between energy consumption, production, economic growth, and CO2 emissions across multiple countries. Key decisions included the identification of relevant data sources. The project's foundation was laid out by developing a comprehensive proposal that detailed the research questions, methodology, and expected outcomes.

## Entry 2: Proposal Development and Submission

**Week 3 and 4, June 2024**

During this phase, the project proposal was developed and refined. The proposal included a literature review that provided context and justification for the study, as well as a detailed research design. The proposal emphasized the need for a dynamic analysis that would incorporate both traditional econometric methods and machine learning techniques to explore the impact of energy consumption and production on CO2 emissions. The proposal was submitted for approval, marking a significant milestone in the project's early stages.

## Entry 3: Literature Review

**Week 1 and 2, July 2024**

This literature review phase focused on gathering and synthesizing existing research related to energy consumption, economic growth, and CO2 emissions. The review informed the selection of appropriate models and methodologies and this ensured that the study would contribute knowledge to the field. The literature review also guided the identification of key variables and data sources for the analysis.

## Entry 4: Data Cleaning and Preparation

**Week 3, July 2024**

Data cleaning was also an important step in ensuring the accuracy and reliability of the analysis. This phase involved handling missing values, normalizing and scaling the data, and converting relevant variables into a time series format for forecasting. The data was meticulously cleaned to remove inconsistencies.

## Entry 5: Methodology and Initial Analysis

**Week 4, July 2024**

A methodological framework was developed during this phase. It involved combining multiple linear regressions for quantifying the effects of predictors on CO2 emissions and ARIMA for forecasting future emissions. Initial data analysis involved exploratory data analysis to understand the relationships between variables. The decisions made during this phase were crucial in setting the stage for the detailed analysis and modeling work that followed.

## Entry 6: Data Analysis, Model Development, and Finalization

**Week 1 to 4, August 2024**

This phase focused on the detailed analysis and development of forecasting models. The MLR was used to quantify the effects of energy consumption, production, and economic growth on CO2 emissions, while ARIMA models were used to forecast CO2 emissions for the period 2020 to 2029. The results were interpreted and discussed in the context of existing literature. Conclusions were drawn, and the project was prepared for submission. The final step involved reflecting on the limitations of the study and identifying areas for future research, including the potential use of more advanced machine learning techniques.

# 3. Reflective Discussion

This section allows me to evaluate my personal growth and the broader implications of my work. Throughout this project, I have significantly developed my skills in data analysis and modeling, particularly in the context of energy consumption and CO2 emissions. One of the key takeaways was my ability to effectively clean and prepare data, ensuring the accuracy and reliability of subsequent analyses. This project also deepened my understanding of applying both traditional econometric methods and modern machine learning techniques which allowed me to explore the dynamic relationships between various factors influencing CO2 emissions.

The challenges of integrating my methodologies to produce reliable forecasts strengthen my problem-solving abilities, especially when addressing issues like model selection. Moving forward, I am eager to apply these skills in real-world scenarios, in fields where data-driven insights can drive meaningful environmental and economic outcomes. This project has not only equipped me with technical expertise but also instilled a sense of responsibility to approach future work with both analytical rigor and ethical consideration.

# References

Alam, S. *et al.* (2023) ‘An investigation of the imputation techniques for missing values in ordinal data enhancing clustering and classification analysis validity’, *Decision Analytics Journal*, 9, p. 100341. Available at: https://doi.org/10.1016/j.dajour.2023.100341.

Lavanya, A. *et al.* (2023) ‘Assessing the Performance of Python Data Visualization Libraries: A Review’, *International Journal of Computer Engineering in Research Trends*, 10(1), pp. 28–39. Available at: https://doi.org/10.22362/ijcert/2023/v10/i01/v10i0104.

Melkumova, L.E. and Shatskikh, S.Ya. (2017) ‘Comparing Ridge and LASSO estimators for data analysis’, *Procedia Engineering*, 201, pp. 746–755. Available at: https://doi.org/10.1016/j.proeng.2017.09.615.

Midway, S.R. (2020) ‘Principles of Effective Data Visualization’, *Patterns*, 1(9), p. 100141. Available at: https://doi.org/10.1016/j.patter.2020.100141.

Nguyen, Q.V. *et al.* (2020) ‘Evaluation on interactive visualization data with scatterplots’, *Visual Informatics*, 4(4), pp. 1–10. Available at: https://doi.org/10.1016/j.visinf.2020.09.004.

Ospina, R. *et al.* (2023) ‘An Overview of Forecast Analysis with ARIMA Models during the COVID-19 Pandemic: Methodology and Case Study in Brazil’, *Mathematics*, 11(14), p. 3069. Available at: https://doi.org/10.3390/math11143069.

Ramos-Carreño, C. *et al.* (2024) ‘**scikit-fda** : A *Python* Package for Functional Data Analysis’, *Journal of Statistical Software*, 109(2). Available at: https://doi.org/10.18637/jss.v109.i02.

Seabold, S. and Perktold, J. (2010) ‘Statsmodels: Econometric and Statistical Modeling with Python’, in. *Python in Science Conference*, Austin, Texas, pp. 92–96. Available at: https://doi.org/10.25080/Majora-92bf1922-011.

Singh, D. and Singh, B. (2020) ‘Investigating the impact of data normalization on classification performance’, *Applied Soft Computing*, 97, p. 105524. Available at: https://doi.org/10.1016/j.asoc.2019.105524.

Srivastava, D. (2023) ‘An Introduction to Data Visualization Tools and Techniques in Various Domains’, *International Journal of Computer Trends and Technology*, 71(4), pp. 125–130. Available at: https://doi.org/10.14445/22312803/IJCTT-V71I4P116.

Sun, Z. (2020) ‘Comparison of Trend Forecast Using ARIMA and ETS Models for S&P500 Close Price’, in *2020 The 4th International Conference on E-Business and Internet*. *ICEBI 2020: 2020 The 4th International Conference on E-Business and Internet*, Singapore Singapore: ACM, pp. 57–60. Available at: https://doi.org/10.1145/3436209.3436894.

Sundaram, J. *et al.* (2023) ‘An Exploration of Python Libraries in Machine Learning Models for Data Science’:, in S.M. Biju, A. Mishra, and M. Kumar (eds) *Advances in Computational Intelligence and Robotics*. IGI Global, pp. 1–31. Available at: https://doi.org/10.4018/978-1-6684-8696-2.ch001.

Varoquaux, G. *et al.* (2015) ‘Scikit-learn: Machine Learning Without Learning the Machinery’, *GetMobile: Mobile Computing and Communications*, 19(1), pp. 29–33. Available at: https://doi.org/10.1145/2786984.2786995.

# Appendix: Python Codes

# Import necessary libraries

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.linear\_model import LinearRegression

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, r2\_score

import statsmodels.api as sm

from scipy import stats

import plotly.express as px

from statsmodels.tsa.seasonal import seasonal\_decompose

from statsmodels.tsa.arima.model import ARIMA

from sklearn.preprocessing import StandardScaler

from sklearn.cluster import KMeans

import warnings

from pandas.tseries.offsets import DateOffset

from statsmodels.tsa.stattools import adfuller

warnings.filterwarnings("ignore")

# Load the data

global\_data = pd.read\_csv("global-data-on-sustainable-energy.csv")

co2\_data = pd.read\_csv("energy data.csv")

# Clean Global Data

# Check for missing values

print("Missing values in Global Data")

print(global\_data.isnull().sum())

# Drop rows with missing values

global\_data.dropna(inplace=True)

print(global\_data.isnull().sum())

# Check for duplicate rows

print("Duplicate rows in Global Data")

print(global\_data.duplicated().sum())

# Clean CO2 Data

# Check for missing values

print("Missing values in CO2 Data")

print(co2\_data.isnull().sum())

# Drop rows with missing values

co2\_data.dropna(inplace=True)

# Check for missing values

print("Missing values in CO2 Data")

print(co2\_data.isnull().sum())

# Convert 'Year' to datetime

global\_data['Year'] = pd.to\_datetime(global\_data['Year'], format='%Y')

co2\_data['Year'] = pd.to\_datetime(co2\_data['Year'], format='%Y')

# Exclude the "World" data

co2\_data = co2\_data[co2\_data['Country'] != 'World']

### Data Visualization

#### Descriptive Statistics

*# Descriptive statistics for co2\_data*

co2\_desc\_stats **=** co2\_data.drop(columns**=**['ID']).describe()

print("\nDescriptive Statistics for co2\_data:")

co2\_desc\_stats.head(9)

#### Visualization

*# Renaming for explainability*

global\_data.rename(columns**=**{"Value\_co2\_emissions\_kt\_by\_country":"CO2", "Entity": "Country"} , inplace**=True**)

*# Main Sources of Electricity in Different Countries*

df\_aggregated **=** global\_data.groupby('Country').agg({

'Electricity from fossil fuels (TWh)': 'sum',

'Electricity from nuclear (TWh)': 'sum',

'Electricity from renewables (TWh)': 'sum'

}).reset\_index()

​

df\_melted **=** df\_aggregated.melt(id\_vars**=**'Country', var\_name**=**'Source', value\_name**=**'Electricity (TWh)')

fig **=** px.pie(df\_melted, names**=**'Source', values**=**'Electricity (TWh)', color**=**'Source')

​

fig.update\_layout(title**=**{'text': 'Main Sources of Electricity Globally', 'x': 0.5, 'xanchor': 'center'})

​

fig.show()

# Access to elecricity changed over the years

df\_global\_electricity = global\_data.groupby('Year')['Access to electricity (% of population)'].mean().reset\_index()

fig = px.line(df\_global\_electricity, x='Year', y='Access to electricity (% of population)',

labels={'Access to electricity (% of population)': 'Access to Electricity (%)'})

# Centering the title

fig.update\_layout(title={'text': 'Global Change in Access to Electricity (2000-2020)', 'x': 0.5, 'xanchor': 'center'})

# Displaying the chart

fig.show()

# Countries with the highest CO2 emissions

maxco2 = co2\_data.groupby('Country')['CO2\_emission'].max().reset\_index()

maxco2 = maxco2.sort\_values(by='CO2\_emission', ascending=False)

top10co2 = maxco2.head(10)

fig = px.bar(

top10co2,

x='Country',

y='CO2\_emission',

color='CO2\_emission',

labels={'Country': 'Country', 'CO2\_emission': 'CO2 Emissions (MMtonnes)'}

)

fig.update\_layout(

height=500,

title={'text': 'Top 10 Countries with Highest CO2 Emissions', 'x': 0.5, 'xanchor': 'center'}

)

fig.show()

# Countries with highest fossil fuel energy

fossil\_fuel\_sum = global\_data.groupby('Country')['Electricity from fossil fuels (TWh)'].sum().sort\_values(ascending=False).head(10).reset\_index()

fig = px.bar(fossil\_fuel\_sum, color='Country', x='Country', y='Electricity from fossil fuels (TWh)')

fig.update\_layout(title={'text': 'Top 10 Countries by Electricity Consumption from Fossil Fuels',

'x': 0.5, 'xanchor': 'center'})

fig.show()

# CO2 Emissions Over Time

df\_global\_CO2 = global\_data.groupby('Year')['CO2'].mean().reset\_index()

fig = px.line(df\_global\_CO2, x='Year', y='CO2',

labels={'CO2': 'CO2 Emissions (kt)'})

# Centering the title

fig.update\_layout(title={'text': 'CO2 Emissions Over Time', 'x': 0.5, 'xanchor': 'center'})

# Displaying the chart

fig.show()

# Define income categories

income\_categories = {

'High Income': [

'Australia', 'Austria', 'Belgium', 'Canada', 'Denmark', 'Finland', 'France',

'Germany', 'Hong Kong', 'Ireland', 'Israel', 'Italy', 'Japan', 'Luxembourg',

'Netherlands', 'New Zealand', 'Norway', 'Singapore', 'South Korea', 'Sweden',

'Switzerland', 'Taiwan', 'United Arab Emirates', 'United Kingdom', 'United States'

],

'Middle Income': [

'Argentina', 'Brazil', 'Chile', 'China', 'Colombia', 'Costa Rica', 'Czech Republic',

'Egypt', 'India', 'Indonesia', 'Malaysia', 'Mexico', 'Peru', 'Philippines',

'Poland', 'South Africa', 'Thailand', 'Turkey', 'Venezuela', 'Romania', 'Bulgaria',

'Mauritius', 'Panama', 'Qatar', 'Saudi Arabia', 'Puerto Rico'

],

'Lower Income': [

'Afghanistan', 'Albania', 'Algeria', 'Angola', 'Antigua and Barbuda', 'Aruba',

'Bahrain', 'Bangladesh', 'Barbados', 'Belize', 'Benin', 'Bermuda', 'Bolivia',

'British Virgin Islands', 'Brunei', 'Burkina Faso', 'Burma', 'Burundi', 'Cabo Verde',

'Cambodia', 'Cameroon', 'Cayman Islands', 'Chad', 'Comoros', 'Congo-Brazzaville',

'Congo-Kinshasa', 'Cuba', 'Côte d’Ivoire', 'Djibouti', 'Ethiopia', 'Fiji', 'Gabon',

'Gambia', 'Ghana', 'Grenada', 'Guatemala', 'Guinea', 'Guyana', 'Haiti', 'Honduras',

'Iran', 'Iraq', 'Jamaica', 'Kiribati', 'Kuwait', 'Laos', 'Lebanon', 'Lesotho', 'Libya',

'Madagascar', 'Malawi', 'Mali', 'Malta', 'Mongolia', 'Morocco', 'Nepal', 'Niger',

'Nigeria', 'North Korea', 'Pakistan', 'Papua New Guinea', 'Paraguay', 'Rwanda',

'Sao Tome and Principe', 'Senegal', 'Seychelles', 'Sierra Leone', 'Solomon Islands',

'Somalia', 'Sri Lanka', 'Sudan', 'Suriname', 'Syria', 'Tanzania', 'Togo', 'Uganda',

'Zambia', 'Zimbabwe', 'Central African Republic', 'Djibouti', 'Yemen', 'Armenia',

'Azerbaijan', 'Belarus', 'Bosnia and Herzegovina', 'Croatia', 'Estonia', 'Georgia',

'Jordan', 'Kazakhstan', 'Kyrgyzstan', 'Latvia', 'Lithuania', 'North Macedonia',

'Russia', 'Slovenia', 'Tajikistan', 'Turkmenistan', 'Ukraine', 'Uzbekistan', 'Moldova',

'Palestinian Territories', 'American Samoa', 'Guam', 'Northern Mariana Islands',

'Timor-Leste', 'Montenegro', 'Serbia', 'Kosovo', 'South Sudan'

]

}

# Function to categorize countries

def categorize\_country(country):

for category, countries in income\_categories.items():

if country in countries:

return category

return 'Unknown'

# Apply the categorization

co2\_data['Income Category'] = co2\_data['Country'].apply(categorize\_country)

# Exclude rows with 'Unknown' income category

co2\_data = co2\_data[co2\_data['Income Category'] != 'Unknown']

# Create a scatter plot using Plotly Express

fig = px.scatter(

co2\_data,

x='Energy\_consumption',

y='CO2\_emission',

color='Income Category',

labels={'Energy\_consumption': 'Energy Consumption (quad Btu)', 'CO2\_emission': 'CO2 Emissions (MMtonnes)'}

)

fig.update\_layout(title={'text': 'Energy Consumption vs. CO2 Emissions by Income Category', 'x': 0.5, 'xanchor': 'center'})

# Show the plot

fig.show()

# Create a scatter plot of Economic Growth vs Energy Production by Income Category

fig = px.scatter(

co2\_data,

x='GDP',

y='Energy\_production',

color='Income Category',

labels={

'GDP': 'GDP (USD)',

'Energy\_production': 'Energy Production (quad Btu)',

'Income Category': 'Income Category'

},

height=500,

width=1000,

facet\_col='Income Category',

trendline='ols'

)

fig.show()

# Drop rows with missing values for 'GDP' and 'Income category'

co2\_data\_cleaned = co2\_data.dropna(subset=['GDP', 'Income Category'])

# Group the data by 'Income category' and calculate the total GDP for each category

gdp\_distribution = co2\_data\_cleaned.groupby('Income Category')['GDP'].sum().reset\_index()

# Create a bar plot

fig = px.bar(

gdp\_distribution,

x='Income Category',

y='GDP',

color = 'Income Category',

title='Distribution of GDP by Income Category',

labels={

'Income Category': 'Income Category',

'GDP': 'Total GDP (USD)'

},

height=500,

width=800

)

fig.update\_layout(title={'text': 'Distribution of GDP by Income Category', 'x': 0.5, 'xanchor': 'center'})

fig.show()

# Create a scatter plot of GDP and CO2

fig = px.scatter(

co2\_data,

x='GDP',

y='CO2\_emission',

color='Income Category',

labels={

'GDP': 'GDP (USD)',

'CO2\_emission': 'CO2 Emissions (MMtonnes)',

'Income Category': 'Income Category'

},

height=600,

width=1000,

facet\_col='Income Category',

trendline='ols'

)

fig.show()

# Group the data by income category and calculate the total CO2 emissions for each category

co2\_by\_income = co2\_data.groupby('Income Category')['CO2\_emission'].sum().reset\_index()

# Create the bar plot

fig = px.bar(

co2\_by\_income,

x='Income Category',

y='CO2\_emission',

color='Income Category',

labels={

'CO2\_emission': 'Total CO2 Emissions (MMtonnes)',

'Income Category': 'Income Category'

},

title='Total CO2 Emissions by Income Category',

height=600,

width=800

)

fig.update\_layout(title={'text': 'Distribution of Total CO2 Emissions by Income Category', 'x': 0.5, 'xanchor': 'center'})

# Show the plot

fig.show()

# Create a scatter plot

fig = px.scatter(

co2\_data,

x='Energy\_production',

y='CO2\_emission',

color='Income Category',

labels={

'Energy\_production': 'Energy Production (quad Btu)',

'CO2\_emission': 'CO2 Emissions (MMtonnes)',

'Income Category': 'Income Category'

},

height=600,

width=1000

)

fig.update\_layout(title={'text': 'Energy Production vs. CO2 Emissions by Income Category', 'x': 0.5, 'xanchor': 'center'})

# Show the plot

fig.show()

# Select relevant columns for correlation

relevant\_columns = ['Energy\_consumption', 'Energy\_production', 'GDP', 'Population', 'Energy\_intensity\_per\_capita', 'Energy\_intensity\_by\_GDP', 'CO2\_emission']

correlation\_matrix = co2\_data[relevant\_columns].corr()

# Plot the correlation heatmap

plt.figure(figsize=(6, 5))

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm', fmt='.2f')

plt.title('Correlation Heatmap')

plt.show()

### Modeling

#### Multiple Linear Regression

*# Define the predictors and target variable*

X **=** co2\_data[['Energy\_consumption', 'Energy\_production', 'GDP']]

y **=** co2\_data['CO2\_emission']

​

*# 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)

​

*# Initialize the model*

model **=** LinearRegression()

​

*# Fit the model on the training data*

model.fit(X\_train, y\_train)

# Make predictions on the testing set

y\_pred = model.predict(X\_test)

# Add a constant to the predictors

X\_train\_sm = sm.add\_constant(X\_train)

# Fit the model

model\_sm = sm.OLS(y\_train, X\_train\_sm).fit()

# Print the summary

print(model\_sm.summary())

# Mean Absolute Error (MAE)

mae = mean\_absolute\_error(y\_test, y\_pred)

# Root Mean Squared Error (RMSE)

rmse = np.sqrt(mean\_squared\_error(y\_test, y\_pred))

# R-squared

r2 = r2\_score(y\_test, y\_pred)

# Adjusted R-squared calculation

n = X\_test.shape[0]

p = X\_test.shape[1]

adjusted\_r2 = 1 - (1 - r2) \* (n - 1) / (n - p - 1)

# Print the results

print(f'Mean Absolute Error (MAE): {mae}')

print(f'Root Mean Squared Error (RMSE): {rmse}')

print(f'R-squared: {r2}')

print(f'Adjusted R-squared: {adjusted\_r2}')

**Cluster Analysis**

# Select relevant features for clustering

features = co2\_data[['Energy\_consumption', 'Energy\_production', 'GDP', 'CO2\_emission']]

# Handle missing data

features = features.dropna()

# Standardize the features

scaler = StandardScaler()

features\_scaled = scaler.fit\_transform(features)

# Use the Elbow Method to determine the optimal number of clusters

inertia = []

for k in range(1, 11):

kmeans = KMeans(n\_clusters=k, random\_state=42)

kmeans.fit(features\_scaled)

inertia.append(kmeans.inertia\_)

# Plot the Elbow Method graph

plt.figure(figsize=(8, 6))

plt.plot(range(1, 11), inertia, 'bo-', markersize=8)

plt.xlabel('Number of Clusters (k)')

plt.ylabel('Inertia')

plt.title('Elbow Method for Optimal k')

plt.show()

# Fit the K-Means model

kmeans = KMeans(n\_clusters=3, random\_state=42)

co2\_data['Cluster'] = kmeans.fit\_predict(features\_scaled)

# Scatter plot of Energy Consumption vs CO2 Emissions colored by cluster

fig = px.scatter(

co2\_data,

x='Energy\_consumption',

y='CO2\_emission',

color='Cluster',

labels={'Energy\_consumption': 'Energy Consumption (quad Btu)',

'CO2\_emission': 'CO2 Emissions (MMtonnes)', 'Cluster': 'Cluster'},

height=600,

width=1000

)

fig.update\_layout(title={'text': 'Clusters of Countries by Energy Consumption and CO2 Emissions',

'x': 0.5, 'xanchor': 'center'})

fig.show()

# Scatter plot of Energy Consumption vs CO2 Emissions colored by cluster

fig = px.scatter(

co2\_data,

x='Energy\_production',

y='CO2\_emission',

color='Cluster',

labels={'Energy\_production': 'Energy Production (quad Btu)',

'CO2\_emission': 'CO2 Emissions (MMtonnes)', 'Cluster': 'Cluster'},

height=600,

width=1000

)

fig.update\_layout(title={'text': 'Clusters of Countries by Energy Production and CO2 Emissions',

'x': 0.5, 'xanchor': 'center'})

fig.show()

#### Time Series Forecasting (ARIMA)

*# Select only the 'Year' and 'CO2\_emission' columns*

forecasting\_data **=** co2\_data[['Year', 'CO2\_emission']]

*# Aggregate CO2 emissions by year*

forecasting\_data **=** forecasting\_data.groupby('Year')['CO2\_emission'].sum().reset\_index()

*# Set 'Date' as the index*

forecasting\_data.set\_index('Year', inplace**=True**)

*# Testing for Stationarity*

adf\_test **=** adfuller( forecasting\_data['CO2\_emission'])

*# Output the results*

print('ADF Statistic: %f' **%** adf\_test[0])

print('p-value: %f' **%** adf\_test[1])

*# Differencing the data*

forecasting\_data['CO2\_emission\_diff'] **=** forecasting\_data['CO2\_emission'].diff().dropna()

​

*# Split the data into training and testing sets*

train, test **=** train\_test\_split(forecasting\_data['CO2\_emission'], test\_size**=**0.2, random\_state**=**13, shuffle**=False**)

​

*# Parameters for ARIMA model*

p, d, q **=** 1, 1, 0

​

*# Initialize a list to store the forecasts*

fcst **=** []

​

*# Loop through the test set to forecast each step and update the training data*

**for** step **in** range(test.shape[0]):

**try**:

*# Fit ARIMA model on the current training data*

arima **=** ARIMA(train, order**=**(p, d, q))

arima\_final **=** arima.fit()

​

*# Forecast the next step*

prediction **=** arima\_final.forecast(steps**=**1)

fcst.append(prediction.iloc[0])

​

*# Update the training data with the actual value from the test set*

train **=** train.append(pd.Series(test.iloc[step], index**=**[test.index[step]]))

**except** Exception **as** e:

*# Handle any errors during the process*

error **=** **-**99999

print(f"Error at step {step}: {e}")

fcst.append(error)

​

*# Convert the forecast list to a Series*

fcst **=** pd.Series(fcst, index**=**test.index)

​

*# Display the first few forecasted values*

print(fcst.head())

​

*# Plot the actual vs. forecasted values using Plotly*

fig **=** px.line(title**=**"ARIMA CO2 Emissions Forecast")

fig.add\_scatter(x**=**train.index, y**=**train, mode**=**'lines', name**=**'Train Data')

fig.add\_scatter(x**=**test.index, y**=**test, mode**=**'lines', name**=**'Actual Test Data')

fig.add\_scatter(x**=**fcst.index, y**=**fcst, mode**=**'lines', name**=**'Forecast')

fig.update\_layout(xaxis\_title**=**"Year", yaxis\_title**=**"CO2 Emissions (MMtonnes)")

fig.show()

# Calculate MAE

mae = mean\_absolute\_error(test, fcst)

print(f"MAE: {mae}")

# Calculate RMSE

rmse = np.sqrt(mean\_squared\_error(test, fcst))

print(f"RMSE: {rmse}")

# Calculate R-squared

r\_squared = r2\_score(test, fcst)

print(f"R-squared: {r\_squared}")

# Calculate Adjusted R-squared

n = len(test)

k = 1

adjusted\_r\_squared = 1 - (1 - r\_squared) \* (n - 1) / (n - k - 1)

print(f"Adjusted R-squared: {adjusted\_r\_squared}")

# Fit the ARIMA model

model = ARIMA(forecasting\_data['CO2\_emission'], order=(1, 1, 0))

model\_fit = model.fit()

# Create future dates (10 years into the future)

future\_dates = [forecasting\_data.index[-1] + DateOffset(years=x) for x in range(1, 11)]

future\_dates\_df = pd.DataFrame(index=future\_dates, columns=forecasting\_data.columns)

# Combine historical and future data

future\_df = pd.concat([forecasting\_data, future\_dates\_df])

# Forecast the next 10 years

forecast\_steps = len(future\_dates)

forecast = model\_fit.forecast(steps=forecast\_steps)

# Add forecasted values to the future DataFrame

future\_df['forecast'] = np.nan

future\_df.loc[future\_dates, 'forecast'] = forecast

# Prepare data for Plotting

plot\_data = future\_df.reset\_index()

plot\_data.rename(columns={'index': 'Year'}, inplace=True)

# Plot

fig = px.line(plot\_data, x='Year', y=['CO2\_emission', 'forecast'],

labels={'value': 'CO2 Emissions (MMtonnes)', 'variable': 'Data Type'},

title='CO2 Emissions Forecast',

line\_dash\_sequence=['solid', 'dash'])

# Update layout for better visualization

fig.update\_layout(

xaxis\_title='Year',

yaxis\_title='CO2 Emissions (MMtonnes)',

grid=dict(rows=1, columns=1),

title=dict(x=0.5)

)

fig.show()

​