Sustainable World Energy Insights and Visualization Using Power-BI

CSP-571 Data Preparation Analysis Project Report

Soham Mankar
smankar1@hawk.iit.edu
A20543251

Shashank Kulkarni
skulkarni@hawk.iit.edu
A20542907

Hari Shekar Reddy
hbobbili@hawk.iit.edu
A20551047

Abstract

This project applies machine learning techniques and PowerBI visualization to analyze global sustainable energy data from 2000 to 2020. By employing algorithms such as random forest, gradient boosting, and linear regression, we seek to understand patterns and trends in sustainable energy development. While the specific findings of the study are yet to be detailed, preliminary insights suggest notable advancements in sustainable energy technologies and utilization. The insights gained from this study are expected to contribute to setting informed goals and shaping policy decisions aimed at promoting sustainable development and combating climate change. Future work will focus on further refining the models for more precise predictions and extending the analysis to emerging trends in the post-2020 period.

1.Overview

1.1 Problem Statement:

The research goal of this project is to analyze the Global Data on Sustainable Energy from 2000 to 2020 to gain insights into the trends, progress, and challenges in the global sustainable energy landscape. The main aim is to conduct a comprehensive analysis of sustainable energy data to inform policy, investment decisions, and public awareness on this critical issue.

1.2 Literature Survey:

[1]. **Towards Sustainable Energy:** A Systematic Review of Renewable Energy Sources, Technologies, and Public Opinions Atika Qazi, Fayaz Hussian, Nasrudin Abd, Khaled Shaban, and Khalid: This article tells us about Renewable energy sources, such as solar, wind, and biomass, are a sustainable way to meet our increasing energy needs. A systematic review of the literature from 2009 to 2018 found that fossil fuels still contribute 73.5% to worldwide electricity production, while renewable sources contribute only 26.5%

[2]. **Towards data-driven energy communities:** A review of open-source datasets, models and tools, Hussain Kazmi, Íngrid Munné-Collado, Fahad Mehmood, Tahir Abbas Syed, Johan Driesen: It provides an overview of a range of emerging renewable and sustainable energy technologies, including solar photovoltaics, concentrated solar power, wind turbines, biomass energy, geothermal energy, and tidal energy. The authors discuss the advantages and disadvantages of each technology, as well as their potential applications.

[3]. **Emerging renewable and sustainable energy technologies**: State of the art Akhtar Hussaina, Syed Muhammad Arifb, Muhammad Aslam: This paper compares quantitative text mining using the "Tools for Innovation Monitoring" (TIM) software and qualitative expert reviews to identify emerging technologies in solar PV, wind power, ocean and tidal energy, and hydropower. The top 300 ranked keywords from TIM Provides the best balance between technical retrieval and analyst work. Experts' identified technologies align with these keywords 65% to 25% of the time, depending on the technology and algorithm used. The study compared the frequency of occurrence of the author's keywords with the TF-IDF algorithm and found that each method is more suitable for different technical fields.

[4]. **In Introduction to Power BI for Data Analysis**: Dr. Kalpana V. Metre, Dr. Ashutosh Mathur, Dr. Ranjana Prakash Dahake, Dr. Yogita Bhapkar, Mrs.Jayashri Ghadge, Prashant Jain, Santosh Gore: Power BI would explore the existing body of knowledge related to the use, functionalities, and impact of Power BI as a data visualization and business intelligence tool. Here's an outline and key points you may consider when conducting a literature review on Power BI.

1.3 proposed methodology/approach:

• **Data Collection** : Obtain the dataset from Kaggle, which includes global energy consumption, production, and greenhouse gas emissions data from 2000 to 2020.
• **Data Cleaning and Preprocessing:** Clean the dataset by handling missing data, outliers, and ensuring consistency in data formats.

• **Descriptive Analysis**: Perform exploratory data analysis to gain insights into trends, patterns, and anomalies in the data.
• **Time Series Analysis**: Utilize time series analysis to understand how renewable Energy production and consumption have changed over the two decades.
• **Correlation Analysis**: Investigate the link between sustainable energy adoption and reductions in greenhouse gas emissions.
• **Geospatial Analysis**: Analyze regional and country-level data to identify variations in sustainable energy adoption.
• **Identifying Key Factors**: Use regression analysis and other statistical techniques to identify key factors that contribute to successful sustainable energy transitions.
• **Visualization in Power BI**: Create a series of interactive visualizations in Power BI to present the findings in a dynamic and user-friendly manner"

2. Data Processing:

Data Resource:

Dataset is Taken from Kaggle:
https://www.kaggle.com/datasets/anshtanwar/global-data-on-sustainable-energy

Data manipulation:

First we will reconstruct the dataset by replacing the old column names to a new column names as below:

```{r}

# recontruct the dataset by replacing the old column names  to a new column names

country <- energy$Entity
year <- energy$Year
E1 <- energy$Access.to.electricity....of.population.
E2 <- energy$Access.to.clean.fuels.for.cooking
E3 <- energy$Renewable.electricity.generating.capacity.per.capita
E4 <- energy$Financial.flows.to.developing.countries..US...
E5 <- energy$Renewable.energy.share.in.the.total.final.energy.consumption....
E6 <- energy$Electricity.from.fossil.fuels..TWh.
E7 <- energy$Electricity.from.nuclear..TWh.
E8 <- energy$Electricity.from.renewables..TWh.
E9 <- energy$Low.carbon.electricity....electricity.
E10 <- energy$Primary.energy.consumption.per.capita..kWh.person.
E11 <- energy$Energy.intensity.level.of.primary.energy..MJ..2017.PPP.GDP.
E12 <- energy$Value_co2_emissions_kt_by_country
E13 <- energy$Renewables....equivalent.primary.energy.
E14 <- energy$gdp_growth
E15 <- energy$gdp_per_capita
E16 <- energy$Density.n.P.Km2.
E17 <- energy$Land.Area.Km2.
E18 <- energy$Latitude
E19 <- energy$Longitude

#create a new dataframe with name energy_df and update it with new column names and data
energy_df <- data.frame(country,  year, E1, E2, E3, E4, E5, E6, E7, E8, E9, E10, E11, E12,  E13,  E14,  E15,  E16,  E17,
E18,  E19)
str(energy_df)

```

After reconstruct we will find out total No of missing values :

```r
# Counting the total number of missing values in each columns
missing_values_count <- colSums(is.na(energy_df))

# Print the missing values count for each column
print(missing_values_count)
```

```
country    year      E1      E2      E3      E4
      0       0      10     169     931    2089
     E5      E6      E7      E8      E9     E10
    194      21     126      21      42       0
    E11     E12     E13     E14     E15     E16
    207     428    2137     317     282       0
    E17     E18     E19
      1       1       1
```

The missing values can be filled with the Statistical measure like(median) and also the least missing value of particular column can be dropped.

```r
# Remove rows with missing values in the specified columns without altering other columns
energy_df <- energy_df[complete.cases(energy_df[, c("E17", "E18", "E19")]), ]

# Print the resulting 'energy_df' dataframe
head(energy_df)



#After removing the missing values of E16,E17,E18 &E19 we will see again count of missing values
# Counting the total number of missing values in each columns
missing_values_count <- colSums(is.na(energy_df))

# Print the missing values count for each column
missing_values_count
```

```r
#finding and replacing the missing values of E1 i.e Access.to.electricity....of.population.

#unique(energy_df$E1)

missing_values_count_e1 <- sum(is.na(energy_df$E1))
print(missing_values_count_e1)

# Calculate the mean of the "E1" column
median_E1 <- median(energy_df$E1, na.rm = TRUE)

# Replace missing values with the mean
energy_df$E1[is.na(energy_df$E1)] <- median_E1

missing_values_count_e1
```

Applying the above code for all the missing values in column names(E1 to E16).

Once again check for missing values:

```r
final_missing_values<- colSums(is.na(energy_df))

# Print the missing values count for each column
print(final_missing_values)
```

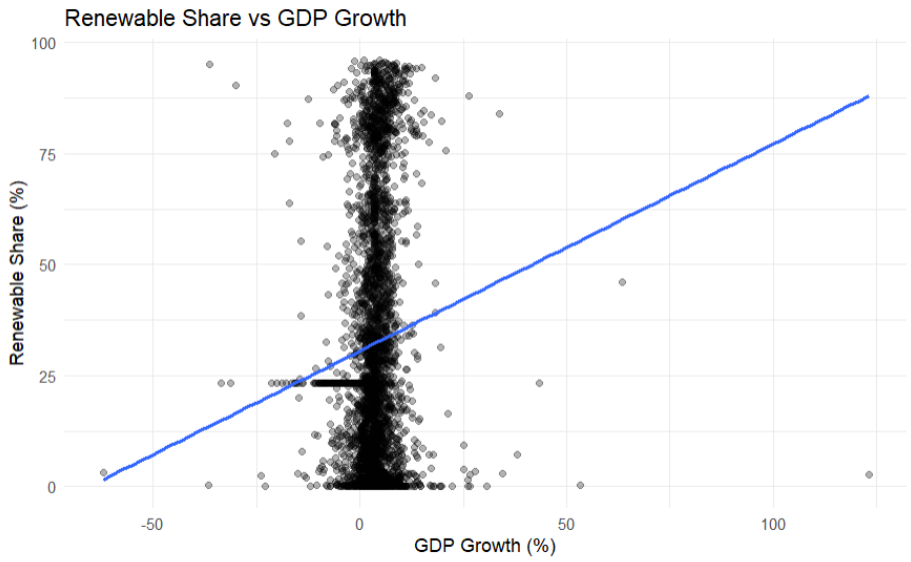
```
country    year      E1      E2      E3      E4
      0       0       0       0       0       0
     E5      E6      E7      E8      E9     E10
      0       0       0       0       0       0
    E11     E12     E13     E14     E15     E16
      0       0       0       0       0       0
    E17     E18     E19
      0       0       0
```

Now as we have no missing values further, now lets proceed to Exploratory Data Analysis.

3. Exploratory Data Analysis:

Global Energy Trends Visualization

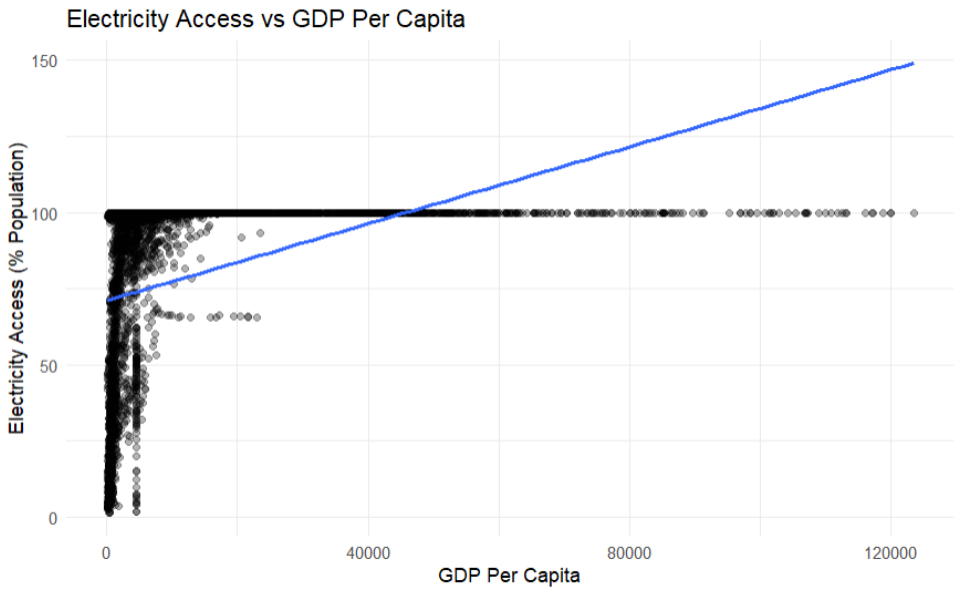
Scatterplot of Electricity Access vs GDP Per Capita



Insights:

The data suggests a positive relationship between GDP growth and the share of renewable energy, meaning that as GDP growth increases, the share of renewable energy in a country's energy mix also tends to increase.

The relationship between GDP growth and renewable share is not strictly linear. While there's a general upward trend, the wide spread of points indicates that the increase in GDP growth does not always correspond to a proportionate increase in renewable share.



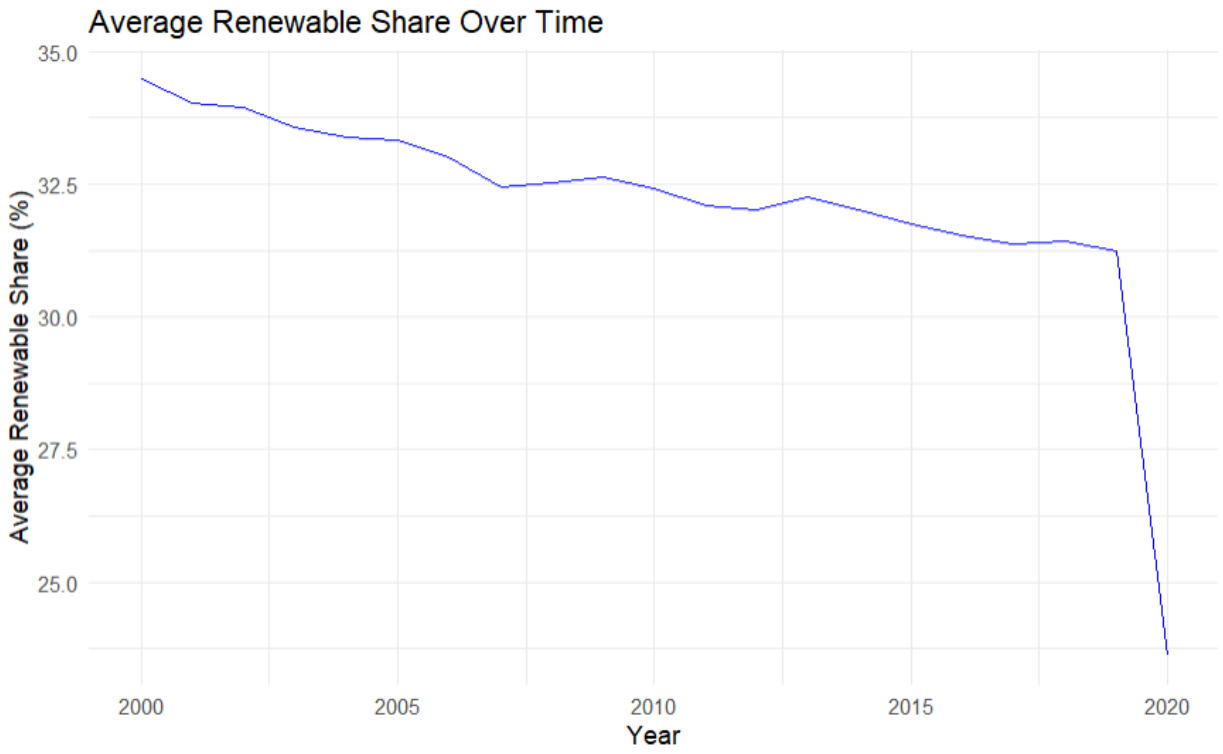
Insights:

- The trend line suggests a negative correlation between the renewable share of energy and electricity access percentage. As the renewable share increases, the percentage of the population with access to electricity seems to decrease.

- A dense cluster of data points is present where the renewable share is low, which implies that most of the sampled entities have a low percentage of renewable energy in their mix and varying levels of electricity access.

- There are a few outliers with a very high renewable share and very low electricity access, suggesting that certain regions may be investing in renewable energy without having widespread electricity access.

Renewable Share Over Time



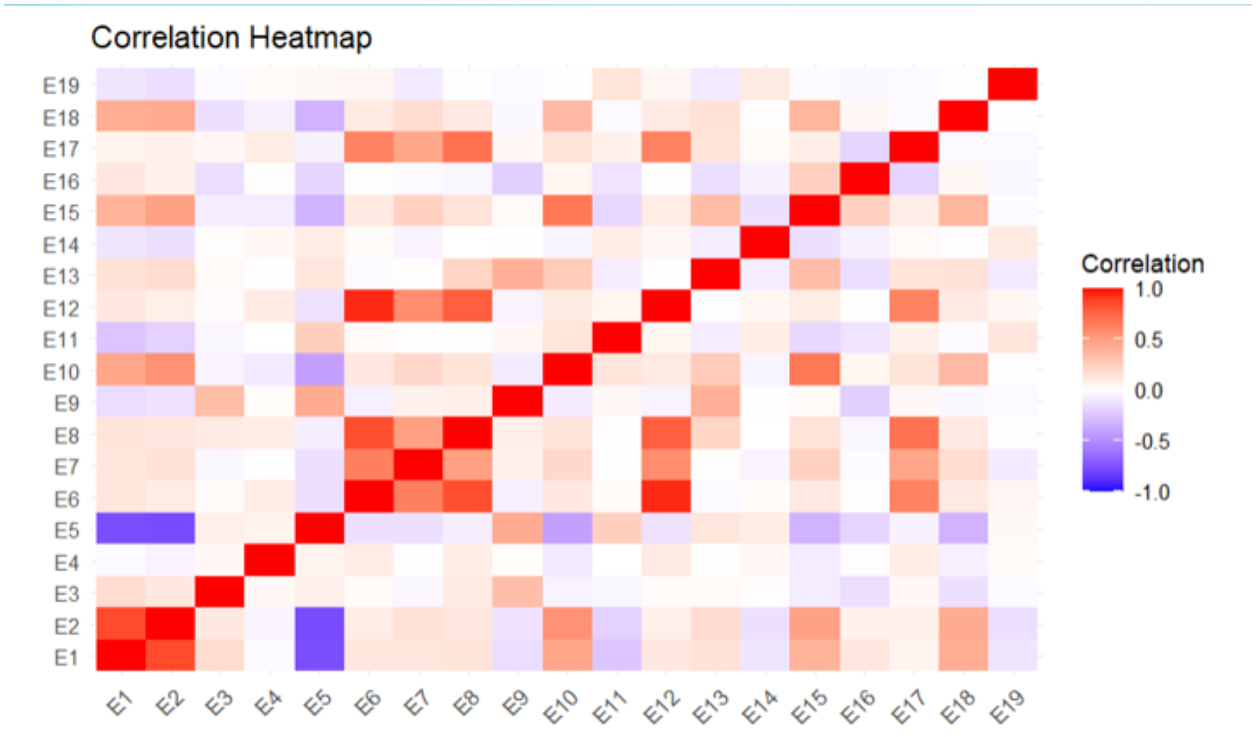
Insights:

- There is an overall decreasing trend in the average renewable share from the year 2000 until around 2015. This suggests that, on average, the reliance on renewable energy sources has diminished over this period.

- There is a sharp increase in the average renewable share starting from around 2015, indicating a recent surge in the use of renewable energy sources compared to the previous years.

- The graph shows some fluctuations, particularly noticeable between 2010 and 2015, which could be attributed to economic, policy, or market changes impacting renewable energy deployment.

Correlation Matrix:



Insights:

- There is A Strong positive correlation between E1 and E2, E6 and E8, E8 and E12, E12 and E6
- There is no correlation between E1-E4,E3TOE6-E19,E4-E11,E15TOE18-E19
- There is a Strong negative correlation between E1-E5,E2-E5

The heatmap indicates a strong relationship between GDP per capita and energy access, a negative association between renewable energy share and CO2 emissions, and a positive correlation between latitude and access to electricity.

Additionally, there's a strong interconnection among various energy consumption metrics, suggesting that higher energy consumption is associated with increased energy intensity and reliance on fossil fuels.

4. Modeling:

Feature Engineering and Model Training:

•First Select the features from the dataset for modeling and then split it into Train-Test Split

•We will split the data into 80% Train and 20% Test Split

```
# Split data into training and testing sets
set.seed(42)
split <- sample.split(y, SplitRatio = 0.8) # This should work now
X_train <- subset(X, split == TRUE)
X_test <- subset(X, split == FALSE)
y_train <- y[split == TRUE]
y_test <- y[split == FALSE]
```

•We are applying modeling on three different parts:

Primary energy consumption per capita:

Linear Regression
Random Forest
Gradient Boosting

Renewable energy share in the total final energy consumption (%):

Linear Regression

Co2 Emission:

Linear Regression
Random Forest
Gradient Boosting

1. **Primary energy consumption per capita:**

```r
# Load necessary libraries
library(tidyverse)
library(ggplot2)
library(plotly)
library(rlang)


# Set options
options(repr.plot.width=10, repr.plot.height=6)

# Read the CSV file
df <- read.csv("final_energy_df.csv")


# Rename the columns
#colnames(df) <- new_column_names

# View the first few rows of the updated dataframe
head(df)

# Display data types of columns
print(sapply(df, class))

# Convert 'Density (P/km2)' column to numeric
df$`E16` <- as.numeric(gsub(",", "", df$`E16`))

# Display data types after conversion
print(sapply(df, class))

# Display the dimensions of the data frame
print(dim(df))

# Remove duplicate rows based on 'Entity' and 'Year'
df <- df[!duplicated(df[c('country', 'year')]), ]
```

```r
# Load necessary libraries for machine learning
library(caret)
library(randomForest)
library(gbm)

# Assuming 'df' is your dataframe and it has been previously loaded into your R session

# Correct target variable name
target <- 'E10'

# Correct features variable names based on the provided list
features <- c(
  'E1',
  'E15',
  'E4',
  "E5",
  'E6'
)

# Check for missing values and create a subset of the data
# Ensure that there are no missing values in the target and feature columns
if(any(colSums(is.na(df[, c(target, features)])) > 0)){
  # Handle missing values according to your data needs, this is an example
  df[is.na(df)] <- 0
}

# Now, create a subset of the data for machine learning, making sure there are no missing values in these columns
ml_subset <- df[, c(target, features)]

# Ensure that ml_subset has at least 2 rows and no missing values
if(nrow(ml_subset) < 2) {
  stop("Not enough data points after subsetting.")
}

# Define predictors and response variable
x <- ml_subset[, features]
y <- ml_subset[, target]
```

```r
# Impute missing values and standardize the predictors
preproc <- preProcess(x, method = c('medianImpute', 'center', 'scale'))
x_imputed <- predict(preproc, x)

# Split the data into training and testing sets
set.seed(42)
index <- createDataPartition(y, p = 0.6, list = FALSE)
x_train <- x_imputed[index, ]
y_train <- y[index]
x_test <- x_imputed[-index, ]
y_test <- y[-index]

# Continue with the rest of your modeling code...

# Note: Since the rest of the code depends on a successful data partition,
# it will not execute if the above condition is not met.


# Train Gradient Boosting model with best parameters
set.seed(42) # For reproducibility when training
gradient_boosting_model <- gbm(
  formula = y_train ~ .,
  data = data.frame(y_train, x_train),
  distribution = "gaussian",
  n.trees = best_gb_params$n.trees,
  interaction.depth = best_gb_params$interaction.depth,
  n.minobsinnode = best_gb_params$n.minobsinnode,
  shrinkage = best_gb_params$shrinkage,
  verbose = FALSE # to avoid verbose output
)
```

```r
# Train Linear Regression model
linear_regression_model <- lm(y_train ~ ., data = data.frame(y_train, x_train))

# Make predictions on the test set
rf_predictions <- predict(random_forest_model, newdata = x_test)
lr_predictions <- predict(linear_regression_model, newdata = x_test)
gb_predictions <- predict(gradient_boosting_model, newdata = x_test, n.trees = best_gb_params$n.trees)

# Calculate mean squared error and R-squared for each model
rf_mse <- mean((rf_predictions - y_test)^2)
lr_mse <- mean((lr_predictions - y_test)^2)
gb_mse <- mean((gb_predictions - y_test)^2)

rf_r2 <- 1 - rf_mse / var(y_test)
lr_r2 <- 1 - lr_mse / var(y_test)
gb_r2 <- 1 - gb_mse / var(y_test)

# Create a results data frame
results <- data.frame(
  Model = c("Random Forest", "Linear Regression", "Gradient Boosting"),
  MSE = c(rf_mse, lr_mse, gb_mse),
  R_squared = c(rf_r2, lr_r2, gb_r2)
)

# Print the results
print(results)
```

The output for above code :

Description: df [3 × 3]

Model <chr>	MSE <dbl>	R_squared <dbl>
Random Forest	226876536	0.8217806
Linear Regression	677905233	0.4674820
Gradient Boosting	102878440	0.9191854

3 rows

Based on above analysis Mean Squared Error (MSE) and R-squared values for three different models, the Gradient Boosting model shows the best performance for the given data. It has the highest R-squared value of 0.9191854, indicating that it explains approximately 91.92% of the variance in the target variable. The R-squared value is a measure of how well the model's predictions approximate the actual data, with a higher value generally indicating a better fit to the data.

Now let's predict next 5 years of E10 values using gradient boosting

```{r}

# Assuming you have a dataframe 'df' with historical data, and you've already trained 'gradient_boosting_model'

# Step 1: Prepare Data
# Extract the previous 10 years of data for E10 and features
historical_data <- df[11:(nrow(df)), ]  # Assuming your dataset has at least 10 years of data
features <- c('E1', 'E15', 'E4', 'E5', 'E6')

# Step 2: Trained Gradient Boosting Model (already done)

# Step 3: Prepare Data for Prediction
# Create a dataframe for the next 5 years of feature values
next_5_years <- 2021:2025  # Adjust the years as needed
future_features <- data.frame(
  E1 = historical_data$E1[(nrow(historical_data) - 9):nrow(historical_data)],
  E15 = historical_data$E15[(nrow(historical_data) - 9):nrow(historical_data)],
  E4 = historical_data$E4[(nrow(historical_data) - 9):nrow(historical_data)],
  E5 = historical_data$E5[(nrow(historical_data) - 9):nrow(historical_data)],
  E6 = historical_data$E6[(nrow(historical_data) - 9):nrow(historical_data)]
)

# Step 4: Make Predictions
# Preprocess the future data using the same preprocessing steps
future_features_imputed <- predict(preproc, future_features)

# Predict E10 for the next 5 years using the Gradient Boosting model
future_predictions <- predict(gradient_boosting_model, newdata = future_features_imputed, n.trees = best_gb_params$n.trees)

# Create a data frame to display predictions with corresponding years
predictions_df <- data.frame(Year = next_5_years, E10_Predictions = future_predictions)

# Print the predictions for the next 5 years
print(predictions_df)


```

Output for above is

Description: df [10 × 2]

Year <int>	E10_Predictions <dbl>
2021	2076.144
2022	2078.811
2023	1451.279
2024	1562.081
2025	2017.392
2021	1966.803
2022	2078.811
2023	1966.803
2024	1969.084
2025	1966.803

1-10 of 10 rows

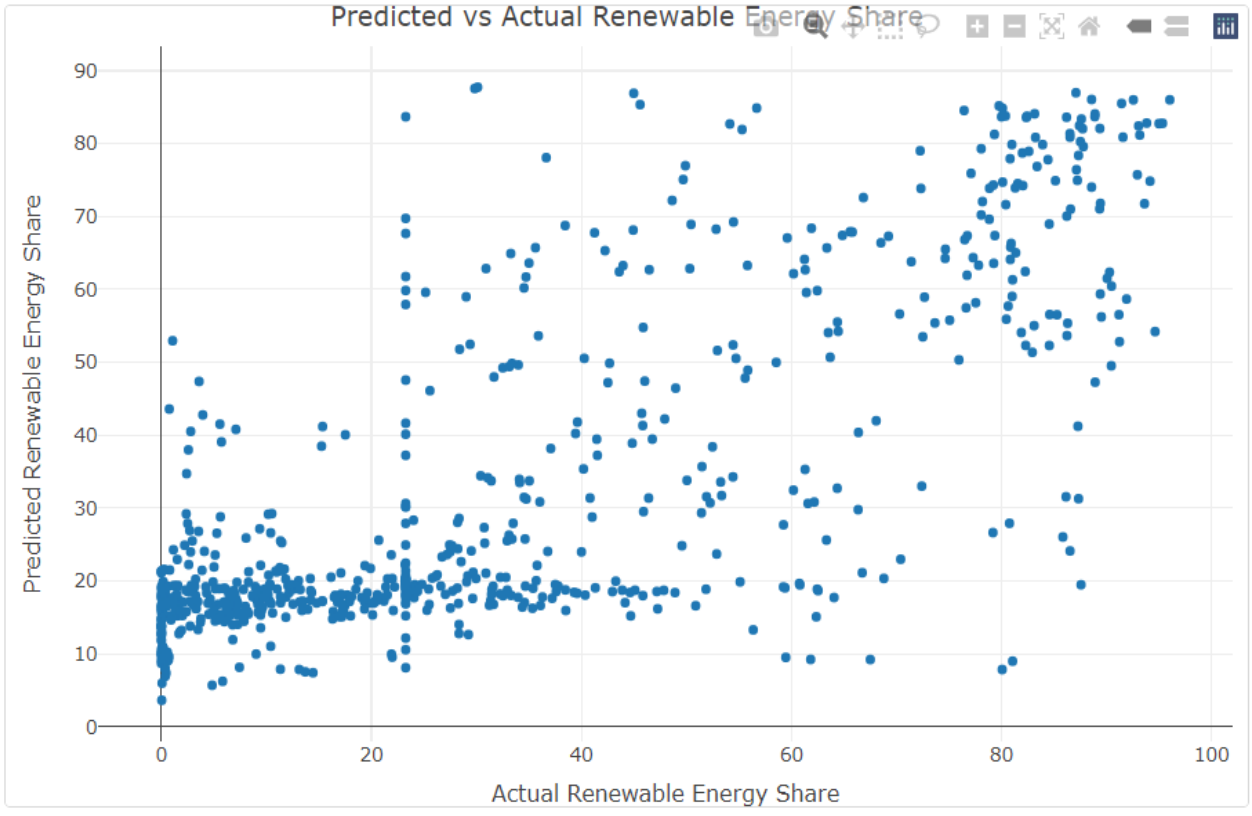
The above is next 5 year E10 Predictions using gradient Boosting

2. **Renewable energy share in the total final energy consumption (%)**

```r
# Load necessary libraries
library(tidyverse)
library(plotly)
library(leaflet)
library(caTools) # Add this line to load caTools

# Read the CSV file
global_co2 <- read.csv("final_energy_df.csv")

# Select features
selected_features <- c('year', 'E15', 'E10',
                       'E1', 'E12')

X <- global_co2[selected_features]
y <- global_co2$`E5`

# Split data into training and testing sets
set.seed(42)
split <- sample.split(y, SplitRatio = 0.8) # This should work now
X_train <- subset(X, split == TRUE)
X_test <- subset(X, split == FALSE)
y_train <- y[split == TRUE]
y_test <- y[split == FALSE]


# Fit linear regression model
model <- lm(y_train ~ ., data = X_train)

# Make predictions on the test set
y_pred <- predict(model, newdata = X_test)

# Calculate Mean Squared Error
mse <- mean((y_test - y_pred)^2)
cat("Mean Squared Error:", mse, "\n")

# Scatter plot of predicted vs actual values
plot_ly(x = y_test, y = y_pred, type = "scatter", mode = "markers") %>%
  layout(title = "Predicted vs Actual Renewable Energy Share",
         xaxis = list(title = "Actual Renewable Energy Share"),
         yaxis = list(title = "Predicted Renewable Energy Share"))
```

The output for above code is :

```
Attaching package: 'plotly'
The following object is masked from 'package:ggplot2':
    last_plot
The following object is masked from 'package:stats':
    filter
The following object is masked from 'package:graphics':
    layout
Mean Squared Error: 348.1231
```



There are some potential outliers or high-leverage points, especially in the upper region of the plot, where actual shares are high but predicted shares vary widely.

Now considering last values of Previous data we will be predicting next 5 years values using linear regression model

```r
```{r}

# Get the last known values for each predictor
last_values <- tail(global_co2, 1)

# Create a dataframe for future years
future_years <- data.frame(
  year = 2021:2025,
  E15 = rep(last_values$E15, 5),
  E10 = rep(last_values$E10, 5),
  E1 = rep(last_values$E1, 5),
  E12 = rep(last_values$E12, 5)
)

# Predict renewable energy share for 2021-2025
predicted_values <- predict(model, newdata = future_years)

# Display the predicted values
predicted_values


```
```

The output is :

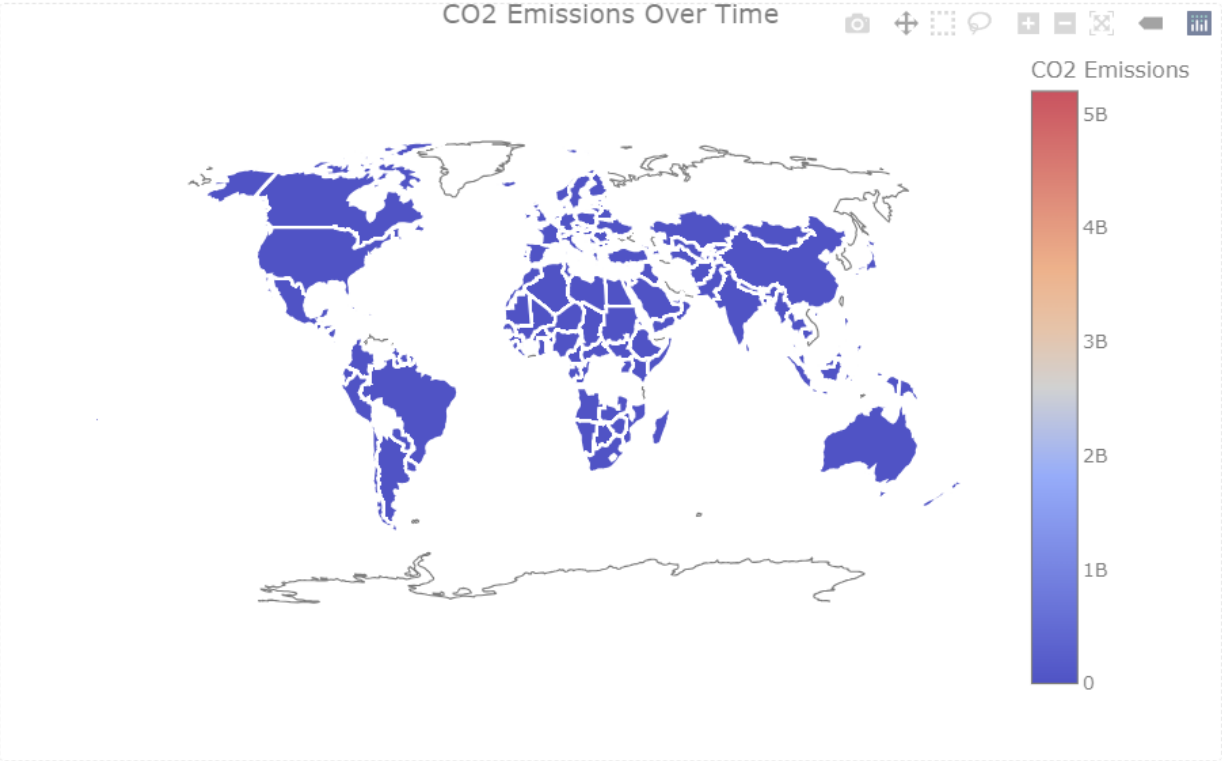
```
       1        2        3        4        5
54.25737 54.43583 54.61428 54.79274 54.97120
```

3. Global Co2 Emission

```{r}


# Load necessary libraries
library(tidyverse)
library(ggplot2)
library(leaflet)
library(plotly)
library(dplyr)

# Read the CSV file
global_co2 <- read.csv("final_energy_df.csv")


head(global_co2)
colnames(global_co2)
dim(global_co2)
str(global_co2)
summary(global_co2)


```

```{r}

library(plotly)

# Assuming global_co2 is defined elsewhere and has 'country', 'year', and 'E12' columns

# Check country codes and convert them to ISO-3 if necessary
# You can use the countrycode or rworldmap package to convert country names to ISO-3 codes

# Transform the data to a long format for plotting
co2_emission_long <- reshape2::melt(global_co2, id.vars = 'country', variable.name = 'year', value.name = 'co2_emissions')

# Make sure that the year column is a factor, as required for the animation_frame argument in plot_ly
co2_emission_long$year <- as.factor(co2_emission_long$year)

# Create the choropleth map
plot_map <- function(data, title) {
  p <- plot_ly(data = data,
               locations = ~country,
               locationmode = "country names",
               z = ~co2_emissions, # Use 'z' instead of 'color' for the choropleth map
               text = ~paste(country, co2_emissions), # Set hover text
               type = "choropleth",
               colors = "RdYlGn",
               marker = list(line = list(color = "rgb(255,255,255)", width = 2)), # Set country borders
               colorbar = list(title = "CO2 Emissions"), # Set colorbar title
               animation_frame = ~year) %>% # Set animation frame to year
    layout(
      title = title,
      geo = list(
        showframe = FALSE,
        showcoastlines = TRUE,
        projection = list(type = 'natural earth')
      )
    )
  return(p)
}

# Run the plotting function with the co2_emission_long data frame
fig <- plot_map(co2_emission_long, "CO2 Emissions Over Time")
fig # This will display the plot in your R environment


```

For above code we get plot as :



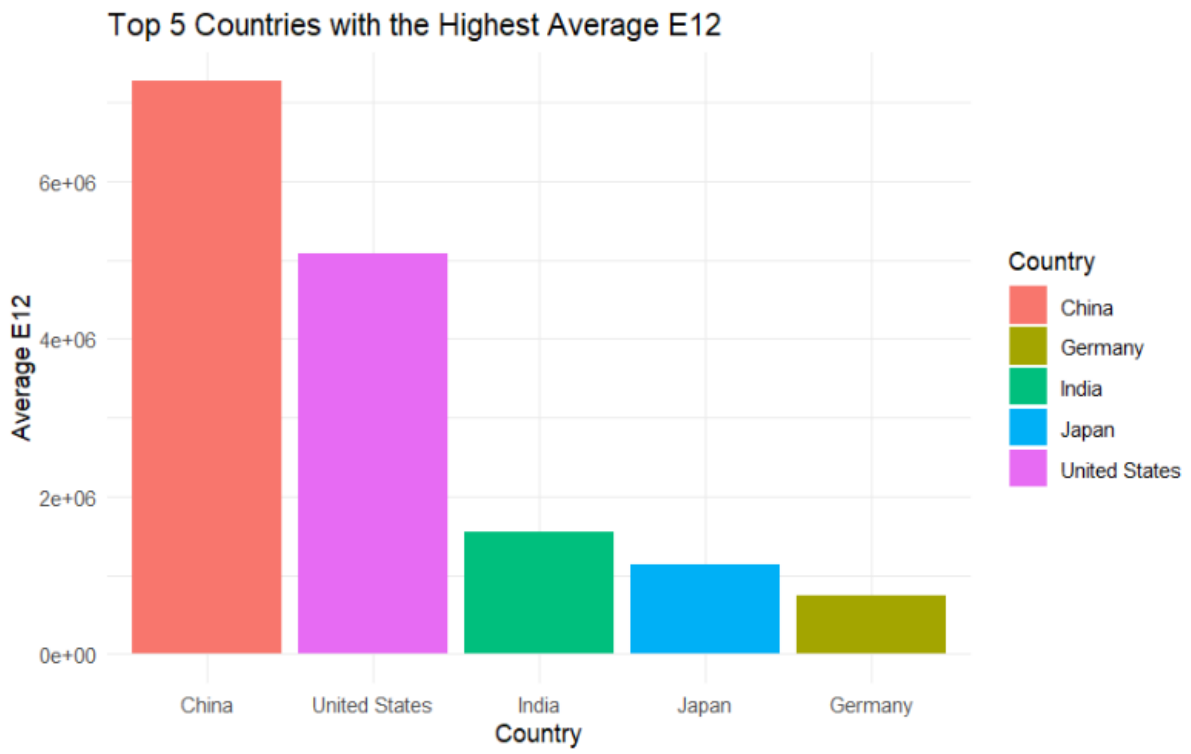
There are a few analysis based on Co2 Emission:

```r
```{r}
library(dplyr)
library(ggplot2)

# Select the necessary columns and calculate the average of E12 for each country
country_avg_E12 <- global_co2 %>%
  select(country, E12) %>%
  group_by(country) %>%
  summarize(average_E12 = mean(E12, na.rm = TRUE)) %>%
  ungroup() %>%
  arrange(desc(average_E12))

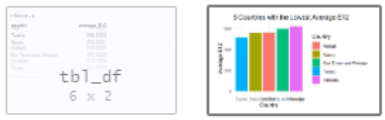
# Display the first 6 countries and their average E12 values
top_countries_avg_E12 <- head(country_avg_E12, 6)
print(top_countries_avg_E12)

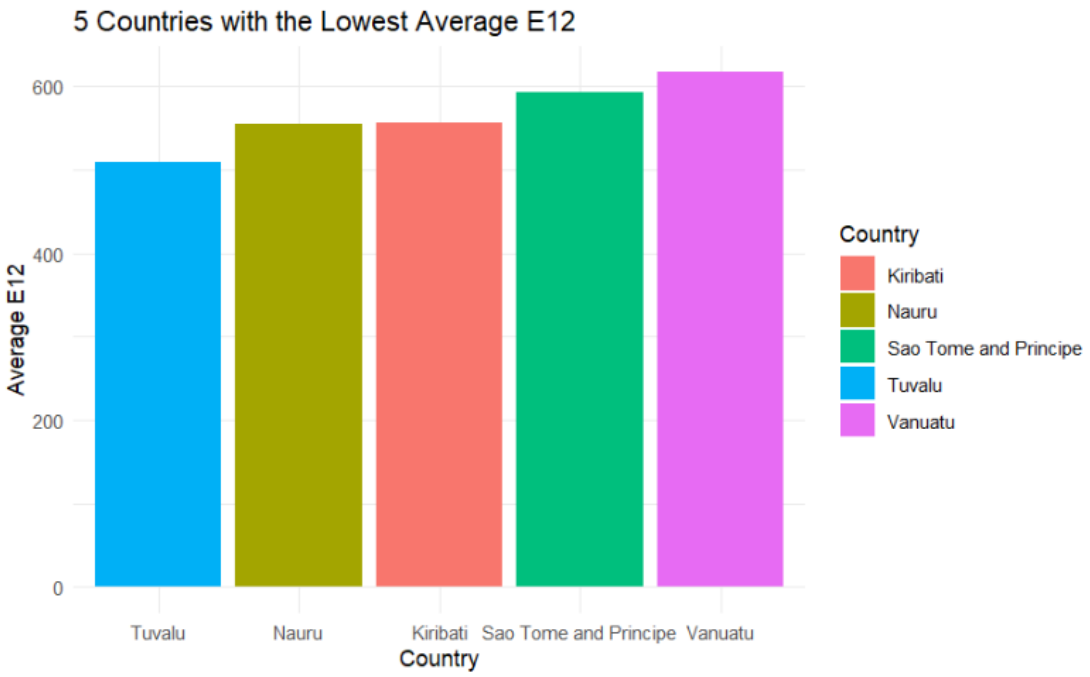
# Plot a bar plot for the top 5 countries based on their average E12 value
# Map the fill aesthetic to the country to get different colors for each bar
ggplot(head(country_avg_E12, 5), aes(x = reorder(country, -average_E12), y = average_E12, fill = country)) +
  geom_bar(stat = "identity") +
  labs(title = "Top 5 Countries with the Highest Average E12",
       x = "Country",
       y = "Average E12") +
  theme_minimal() +
  scale_fill_discrete(name = "Country") # Optional: to add a legend title
```
```



Top lowest avaerage co2 emsiion vs contriees

```{r}

library(dplyr)
library(ggplot2)

# Select the necessary columns and calculate the average of E12 for each country
country_avg_E12 <- global_co2 %>%
  select(country, E12) %>%
  group_by(country) %>%
  summarize(average_E12 = mean(E12, na.rm = TRUE)) %>%
  ungroup() %>%
  arrange(average_E12)

# Display the first 6 countries and their average E12 values
bottom_countries_avg_E12 <- head(country_avg_E12, 6)
print(bottom_countries_avg_E12)

# Plot a bar plot for the 5 countries with the lowest average E12 value
# Map the fill aesthetic to the country to get different colors for each bar
ggplot(head(country_avg_E12, 5), aes(x = reorder(country, average_E12), y = average_E12, fill = country)) +
  geom_bar(stat = "identity") +
  labs(title = "5 Countries with the Lowest Average E12",
       x = "Country",
       y = "Average E12") +
  theme_minimal() +
  scale_fill_discrete(name = "Country") # Optional: to add a legend title

```





Now before go to modeling for co2 emission once we will find out Average growth of CO2 emission over the years

```r
library(dplyr)

# Assuming your data frame is named global_co2 and it has a column named "year" and "co2_emission"
# Calculate the total CO2 emissions for each year
yearly_total_co2 <- global_co2 %>%
  group_by(year) %>%
  summarize(total_co2_emission = sum(E12, na.rm = TRUE))

# Calculate the growth rate for each year
yearly_total_co2 <- yearly_total_co2 %>%
  mutate(growth_rate = (total_co2_emission - lag(total_co2_emission)) / lag(total_co2_emission) * 100)

# Calculate the average growth rate
average_growth_rate <- mean(yearly_total_co2$growth_rate, na.rm = TRUE)


yearly_total_co2

# Print the result
cat("Average Growth Rate of CO2 Emissions:", round(average_growth_rate, 2), "%\n")
```

year <int>	total_co2_emission <dbl>	growth_rate <dbl>
2000	20126246	*NA*
2001	20480401	1.759670640
2002	20762145	1.375675542
2003	21775568	4.881107009
2004	22825063	4.819601414
2005	23696161	3.816406433
2006	24440605	3.141624924
2007	25460840	4.174343136
2008	25627008	0.652642937
2009	25372815	-0.991894935

```
Average Growth Rate of CO2 Emissions: -2.69 %
```

Applying three models:

Linear regression
Gradient Boosting
Random Forest

```r
# Load required libraries
library(caret)
library(ranger) # ranger is used instead of randomForest
library(xgboost)

# Assuming you have a data frame named 'global_co2' with columns 'country' and 'E12'

# Convert 'country' to a numeric variable
global_co2$country_code <- as.numeric(as.factor(global_co2$country))

# Prepare the data
X <- global_co2[, !(colnames(global_co2) %in% c("E12", "country"))] # Remove 'country' and use 'country_code' instead
y <- global_co2$E12

# Split the data into training and testing sets
set.seed(42)  # Set random seed for reproducibility
train_indices <- createDataPartition(y, p = 0.8, list = FALSE)
X_train <- X[train_indices, ]
y_train <- y[train_indices]
X_test <- X[-train_indices, ]
y_test <- y[-train_indices]

# Create a list of regression models
set.seed(42) # Set the seed before training the models
models <- list(
  'Linear Regression' = lm(E12 ~ ., data = data.frame(E12 = y_train, X_train)),
  'Random Forest' = ranger(E12 ~ ., data = data.frame(E12 = y_train, X_train), num.trees = 500),
  'Gradient Boosting' = train(E12 ~ ., data = data.frame(E12 = y_train, X_train), method = 'gbm', verbose = FALSE)
)

best_model <- NULL
best_r2 <- -Inf # Start with negative infinity for comparison

# Load Metrics library for R2 and MAE functions
library(Metrics)

for (model_name in names(models)) {
  model <- models[[model_name]]

```

```r
  # Predict on the test set
  if (model_name == "Random Forest") {
    y_pred <- predict(model, data = X_test)$predictions # Use ranger's predict method
  } else {
    y_pred <- predict(model, newdata = X_test)
  }

  # Evaluate the model
  r2 <- R2(y_test, y_pred)
  mae <- mae(y_test, y_pred)
  mse <- mean((y_test - y_pred)^2)
  rmse <- sqrt(mse)

  # Create a dataframe for comparison
  submit <- data.frame('Actual E12' = y_test, 'Predicted_E12' = y_pred)
  submit <- cbind(index = as.numeric(rownames(submit)), submit)

  if (r2 > best_r2) {
    best_r2 <- r2
    best_model <- model_name
  }

  cat(paste(model_name, ":\n"))
  cat(paste("R2 Score:", format(r2, digits = 2), "\n"))
  cat(paste("Mean Absolute Error (MAE):", format(mae, big.mark = ","), "\n"))
  cat(paste("Mean Squared Error (MSE):", format(mse, scientific = TRUE), "\n"))
  cat(paste("Root Mean Squared Error (RMSE):", format(rmse, big.mark = ","), "\n"))
  print(head(submit, 5))  # Print only the first 5 rows for brevity
  cat("-------------------------------------------------\n")
}

cat(paste("The best performing model is:", best_model, "with R2 score:", format(best_r2, digits = 2), "\n"))

```

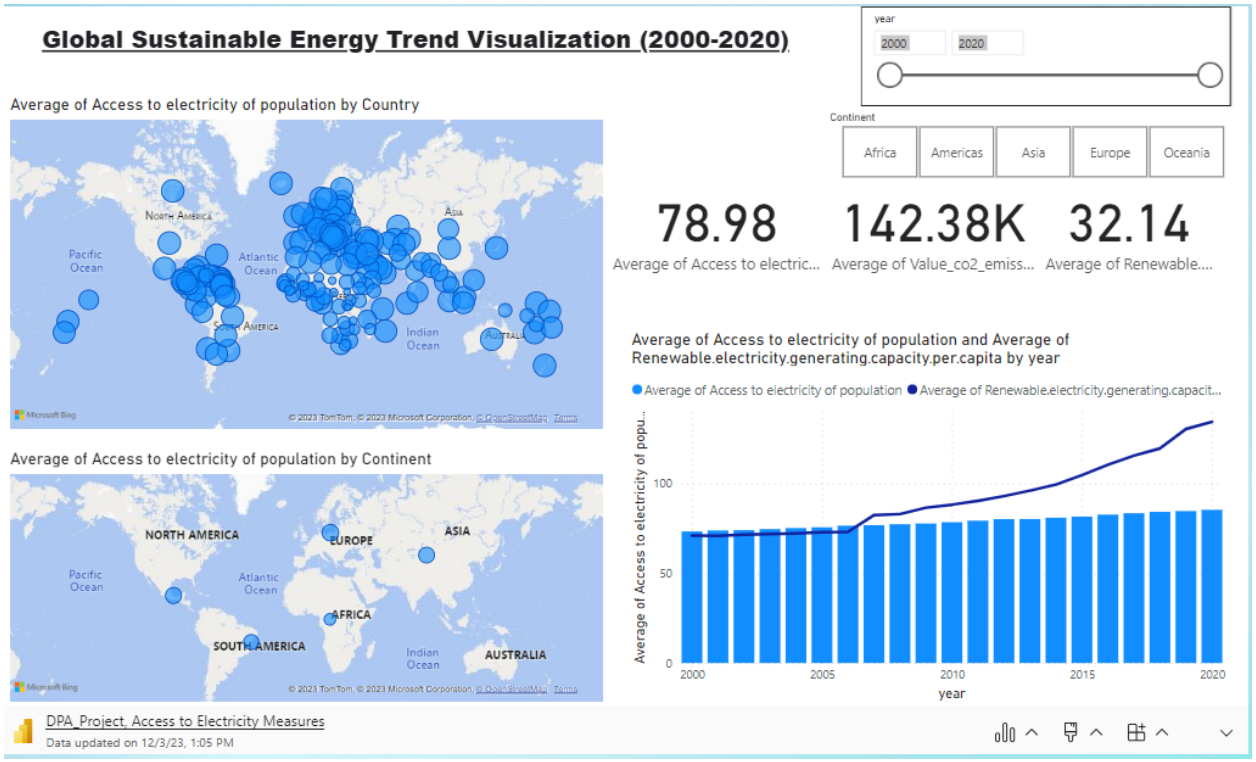
The output obtained from above code is:

```
Linear Regression :
R2 Score: 0.97
Mean Absolute Error (MAE): 53,083.37
Mean Squared Error (MSE): 1.756714e+10
Root Mean Squared Error (RMSE): 132,541.1
-------------------------------------------------
Random Forest :
R2 Score: 0.99
Mean Absolute Error (MAE): 12,963.49
Mean Squared Error (MSE): 6.818058e+09
Root Mean Squared Error (RMSE): 82,571.53
-------------------------------------------------
Gradient Boosting :
R2 Score: 0.99
Mean Absolute Error (MAE): 33,424.19
Mean Squared Error (MSE): 1.015382e+10
Root Mean Squared Error (RMSE): 100,766.2
-------------------------------------------------
The best performing model is: Random Forest with R2 score: 0.99
```

So we can conclude that Random forest is best model in performance with r2 score: 0.99

5. PowerBI Visualizations:

Page 1



Insights/Data Story for above:

Given the visual information provided, here are potential stories and insights:

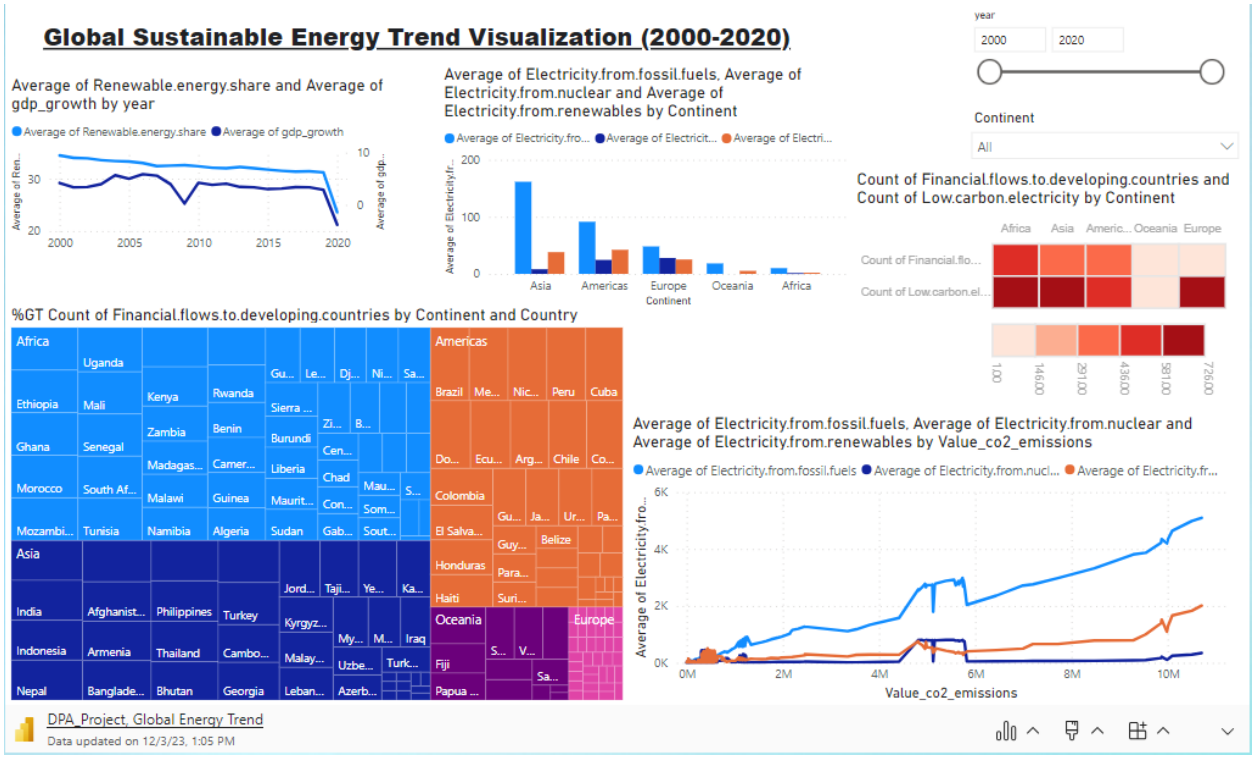
- There has been a consistent rise in the average values of E1 and E3 from 2000 to 2020, indicating a positive trend in the sustainable energy metrics being tracked.

- Europe and Asia seem to be significant contributors to the average of E1, suggesting these regions may have a high focus on sustainable energy initiatives.

- The considerable difference in magnitude between E1 and E12 might imply different scales of measurement or different types of energy-related metrics, with E12 being substantially larger.

- The dashboard's interactivity implies that users can drill down into specific years or continents to see how sustainable energy trends have evolved over time and across different regions.

Page 2



Insights and Data Stories:

- From these visualizations, one could draw the following data stories and insights:

- Fluctuating Investments or Outputs: The varying trends in the average of E5 and E14 suggest that there may be fluctuating investments in sustainable energy or variable outputs from year to year, requiring deeper investigation into the causes.

- Asia's Prominence: Asia's lead in the average of E6, E7, and E8 could indicate a strong emphasis on certain sustainable energy initiatives or outcomes, which could be due to increased investments, policy changes, or technological advancements in this region.

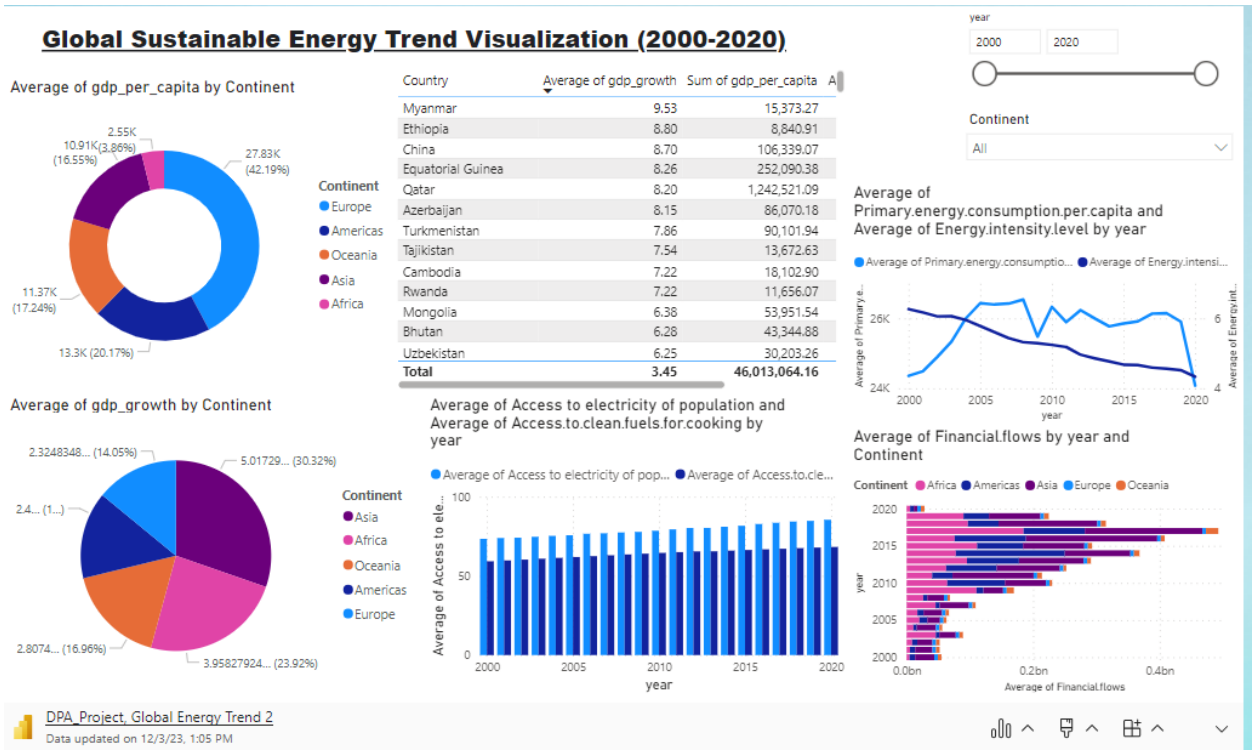
- Resource Allocation: The heat map and tree map hint at how resources or outputs (E4) are distributed across countries and continents. Countries with larger and darker tiles may

be investing more heavily or may be more resource-rich when it comes to sustainable energy.

- Diverse Metrics: The different trends and distributions of E4, E6, E7, E8, and E9 across continents suggest that various regions may be focusing on different aspects of sustainable energy, which could be influenced by regional priorities, availability of natural resources, or socioeconomic factors.

- Growth and Potential: The sharp upward trend of E8 in the second line chart might represent a growing area of sustainable energy, such as renewable energy generation or efficiency improvements, indicating potential areas for future investment and policy focus.

Page 3



Insights and Data Stories:

- Asia's Leadership in Sustainable Energy: Across multiple metrics (E14, E15), Asia is shown as a leader in sustainable energy, which may be a result of the continent's large

population, economic growth, and possibly concerted efforts in sustainable energy investments.

- Varying Contributions by Continent: The data suggests that different continents are contributing to sustainable energy trends at varying degrees, with Africa and Oceania contributing less in some metrics compared to Asia, the Americas, and Europe.

- Country-Specific Differences: There are significant differences in sustainable energy metrics at the country level, highlighting the importance of local policies, resources, and economic conditions.

- Trends Over Time: The line charts suggest changing trends in sustainable energy metrics over the years. The reasons for these changes could be multifaceted, including technological advancements, policy changes, economic factors, or shifts in energy consumption patterns.

- Growth in Sustainable Energy Utilization:The increasing trend in the average of E4 over time suggests a positive growth in sustainable energy utilization or development across all continents, particularly in Asia and Europe.

6. Conclusion:

- From the project we find that out of all models, gradient boosting provides a good fit for our dataset.
- This project demonstrates the harmonious integration of data analysis, visualization, and machine learning to untangle the complexities surrounding global energy consumption.
- By combining insights derived from historical data with predictive capabilities, it strives to contribute to a sustainable energy future and a world that is better informed.
- Key findings showcase progress in improving access to electricity and cleaner cooking fuels, alongside the ongoing challenge of escalating CO2 emissions.
- The predictive models developed in this project serve as valuable tools for decision-makers, facilitating resource allocation and targeted energy initiatives.
- Opportunities for future refinement exist through the incorporation of additional variables, the application of advanced modeling techniques, and more comprehensive regional analyses.
- In essence, this project underscores the critical role of data-driven approaches in addressing global energy challenges and advancing sustainability efforts.

7. Future Works:

1. **Data Expansion**: Suggest expanding the dataset beyond 2020 to include the most recent trends and developments in sustainable energy. This could include the impact of recent global events such as the COVID-19 pandemic on energy usage patterns.
2. **Advanced Analytical Techniques**: Propose the use of more advanced data analysis techniques, like machine learning and predictive modeling, to forecast future energy trends and model the impact of different energy policies.

3. **Interactive Dashboard Enhancements**: For the dashboard, consider incorporating more interactive elements or real-time data updates. This would make it a more dynamic tool for tracking ongoing changes in global energy trends.

4. **Collaborative Research Opportunities**: Identify opportunities for collaboration with other researchers or institutions. This could lead to more comprehensive studies or the development of a global sustainable energy database.

5. **Policy Impact Assessment:** Suggest conducting an analysis of how different countries' energy policies have influenced sustainable energy trends and what lessons can be learned from these policies.

6. **Public Engagement and Education**: Highlight the importance of using the project's findings to educate the public and engage them in discussions about sustainable energy.

8. Data Source:

https://www.kaggle.com/datasets/anshtanwar/global-data-on-sustainable-energy

9. Source Code:

https://github.com/theshashankkulkarni/Data-Preparation-and-Analysis-Project/tree/main

Bibliography:

[1] Towards Sustainable Energy: A Systematic Review of Renewable Energy Sources, Technologies, and Public Opinions Atika Qazi, Fayaz Hussian, Nasrudin Abd, KhaledShaban, and Khalid, Volume 7, 2019 IEEE.

[2] Towards data-driven energy communities: A review of open-source datasets, models and tools, Hussain Kazmi, Íngrid Munné-Collado, Fahad Mehmood, Tahir AbbasSyed, Johan Driesen, Published by Elsevier Ltd.

[3] Emerging renewable and sustainable energy technologies: State of the art Akhtar Hussaina, Syed Muhammad Arifb, Muhammad Aslam, 2016 Published by ElsevierLtd.

[4] An Introduction to Power BI for Data Analysis: Dr. Kalpana V. Metre, Dr. AshutoshMathur, Dr. Ranjana Prakash Dahake, Dr. Yogita Bhapkar, Mrs.Jayashri Ghadge,Prashant Jain, Santosh Gore, Published by IJISAE

[5] "R for Data Science" by Hadley Wickham and Garrett Grolemund: This book provides a comprehensive introduction to data manipulation, visualization, and analysis using R.

[6] "The Art of R Programming" by Norman Matloff: This book is a practical guide to R-programming and covers various aspects of R, from data structures to debugging.

[7] "The Art of Data Science" by Roger D. Peng: Provides insights into the data analysis process and the art of interpreting results.