

Exploratory Data Analysis on

RENEWABLE ENERGY

by

Group 26



Neel Bokil
ID: 202411027
Course: MTech (ICT)



Yash Sorathiya
ID: 202411019
Course: MTech (ICT)



Divyakumar Tandel
ID: 202201469
Course:
BTech(ICT-CS)

Course Code: IT 462
Semester: Autumn 2024

Under the guidance of

Dr. Gopinath Panda



Dhirubhai Ambani Institute of Information and Communication Technology

December 2, 2024

ACKNOWLEDGMENT

This letter is an expression of my sincere appreciation for all of your help and advice during the course of my project, "Renewable Energy." Your invaluable assistance has played a pivotal role in shaping the successful completion of this endeavor.

I consider myself very fortunate to have been able to work under your guidance. Your knowledge, support, and eagerness to impart your skills have been crucial in improving the caliber and reach of my project. Your constructive feedback and insightful suggestions have helped me overcome challenges and develop a deeper understanding of the subject matter.

Additionally, I would want to thank my teammates for their support during this transition. Their valuable input and camaraderie have been a constant source of motivation.

It has been a great learning experience to finish this project, and I am sure that the information and abilities I have gained will be a strong basis for my future undertakings.

Once again, thank you for your unwavering guidance and belief in my abilities. Your mentorship has been invaluable, and I am truly grateful for the opportunity to work with you.

Sincerely,

Neel Bokil , 202411027
Yash Sorathiya , 202411019
Divyakumar Tandel , 202201469

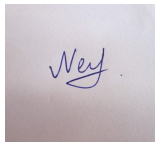
DECLARATION

We, the members of Group 26, hereby declare that the EDA project work presented in this report is our original work and has not been submitted for any other academic degree. All the sources cited in this report have been appropriately referenced.

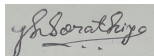
We acknowledge that the data used in this project is obtained from the www.kaggle.com site. We also declare that we have adhered to the terms and conditions mentioned in the website for using the dataset. We confirm that the dataset used in this project is true and accurate to the best of our knowledge.

We acknowledge that we have received no external help or assistance in conducting this project, except for the guidance provided by our mentor Prof. Gopinath Panda. We declare that there is no conflict of interest in conducting this EDA project.

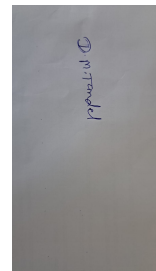
We hereby sign the declaration statement and confirm the submission of this report on December 2, 2024.



Neel Bokil
ID: 202411027
Course: MTech (ICT)



Yash Sorathiya
ID: 202411019
Course: MTech (ICT)



Divyakumar Tandel
ID: 202201469
Course: BTech(ICT)

CERTIFICATE

This is to certify that Group 26 comprising Neel Bokil, Yash Sorathiya, and Divyakumar Tandel has successfully completed an exploratory data analysis (EDA) project on the Renewable Energy, which was obtained from www.kaggle.com.

The EDA project presented by Group 26 is their original work and has been completed under the guidance of the course instructor, Prof. Gopinath Panda, who has provided support and guidance throughout the project. The project is based on a thorough analysis of the IRENA Renewable Energy Statistics dataset, and the results presented in the report are based on the data obtained from the dataset.

This certificate is issued to recognize the successful completion of the EDA project on the Renewable Energy, which demonstrates the analytical skills and knowledge of the students of Group 26 in the field of data analysis.

Signed,
Dr. Gopinath Panda,
IT 462 Course Instructor
Dhirubhai Ambani Institute of Information and Communication Technology
Gandhinagar, Gujarat, INDIA.

December 2, 2024

Contents

List of Figures	5
1 Introduction	1
1.1 Project idea	1
1.2 Data Collection	1
1.3 Dataset Description	1
1.4 Packages required	2
2 Data Cleaning	6
2.1 About Dataset	6
2.2 Missing Data Analysis	7
2.3 Features Correlation	8
2.4 Imputation	9
3 Data Visualization	11
3.1 Renewable vs Non-Renewable Electricity Generation by Region	11
3.2 Electricity Generation by Technology Type	12
3.3 Energy Generation over Time	13
3.4 Producer Type vs Region	14
3.5 Electricity Generation Trend Over Years (for a Specific Region)	14
3.6 Installed Capacity vs Electricity Generation (Scatter Plot)	15
3.7 Top Sub-regions by Electricity Generated	15
4 Feature Engineering	16
4.1 Feature extraction	16
4.2 Feature selection	16
5 Model fitting	17
5.1 Data Splitting	17
5.2 Model Parameters	17
5.3 Linear Regression	18
5.4 Decision Tree Regressor	19
5.5 Random Forest Regressor	19
5.6 Gradient Boosting Regressor	20
5.7 Support Vector Regressor	21
5.8 XGBoost Regressor	21
5.9 Ridge Regression	22

6	Conclusion & future scope	23
6.1	Findings/Observations	23
6.2	Challenges	23
6.3	Future Plan	23
6.4	Model Evaluation Metrics Comparison	24

List of Figures

1.1	List of libraries that we used in our project.	2
2.1	Loading dataset for analysis	6
2.2	Numbers of rows and column of the dataset	6
2.3	Null values in the dataset	7
2.4	Barplot using missingno library	8
2.5	Matrix plot using missingno library	8
2.6	Dendrogram plot using missingno library	9
3.1	Comparison of Renewable and Non-Renewable Electricity Generation by Region	11
3.2	Electricity Generation by Technology Type	12
3.3	Trends in Electricity Generation Over Time	13
3.4	Electricity Generation by Producer Type Across Regions	14
3.5	Electricity Generation Trend Over Years in Europe	14
3.6	Relationship Between Installed Capacity and Electricity Generation	15
3.7	Top Sub-regions by Electricity Generated	15
5.1	Linear Regression Scatter Plot	18
5.2	Decision Tree Regressor Scatter Plot	19
5.3	Random Forest Regressor Scatter Plot	20
5.4	Gradient Boosting Regressor Scatter Plot	20
5.5	Support Vector Regressor Scatter Plot	21
5.6	XGBoost Regressor Scatter Plot	22
5.7	Ridge Regression Scatter Plot	22
6.1	R^2 scores comparison	24
6.2	RMSE values comparison	25

List of Tables

6.1 Model Evaluation Metrics Comparison 24

Abstract

This report analyzes the IRENA Renewable Energy Statistics dataset, which encompasses global renewable energy data covering various energy sources such as solar, wind, hydro, geothermal, and bioenergy. The dataset includes annual data from numerous countries on metrics like installed capacity (MW) and energy generation (GWh). The study involves data cleaning, exploratory data analysis (EDA), and trend analysis to uncover patterns in renewable energy adoption and consumption over time. Key findings reveal significant variations in renewable energy capacity across regions, with substantial growth in solar and wind energy in recent years. This analysis provides valuable insights into global renewable energy development and offers a foundation for predictive modeling, particularly for forecasting energy production.

Chapter 1. Introduction

1.1 Project idea

Our project focuses on analyzing energy data and predicting Electricity Generation (GWh), specifically involving renewable and non-renewable energy sources across different regions and sub-regions from 2000 to 2022. With the global push toward sustainability, understanding energy trends is crucial for shaping policies and investments. The goal is to build regression models to predict Electricity generation based on historical data.

1.2 Data Collection

This dataset is used from kaggle, an open source website for collection of datasets, provides valuable insights into the Electricity Generation in different countries along with the specific regions and sub-regions over the years.

1.3 Dataset Description

This dataset, provided by the International Renewable Energy Agency (IRENA), offers comprehensive statistics on renewable and non-renewable energy across various regions, sub-regions, and countries from the year 2000 to 2022. The dataset includes key metrics such as Electricity Generation (in GWh) and Installed Electricity Capacity (in MW) for different energy technologies.

The dataset comprises of 12 features which are as follows:

1. Region: The geographical region where the data applies (e.g., Africa, Asia).
2. Sub-region: A more specific geographical subdivision (e.g., Northern Africa).
3. Country: The name of the country.
4. ISO3 code: The ISO 3166-1 alpha-3 country code (three-letter code for each country).
5. M49 code: The United Nations M49 standard numeric code for countries and regions.
6. RE or Non-RE: Indicates whether the energy source is renewable or non-renewable.
7. Group Technology: The broad category of energy technology (e.g., Fossil fuels).
8. Technology: Specific type of energy technology (e.g., Natural gas).

9. Producer Type: Indicates whether the energy producer is on-grid or off-grid.
10. Year: The year in which the data was recorded.
11. Electricity Generation (GWh): The amount of electricity generated in gigawatt-hours.
12. Electricity Installed Capacity (MW): The installed capacity for electricity generation in megawatts.

The "Region" feature comprises of 5 unique values denoting the five regions namely,

- Africa
- America
- Asia
- Europe
- Oceania

Each region comprises of many sub-regions wise Electricity generation.

1.4 Packages required

```
import pandas as pd
import missingno as msno
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import numpy as np
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.svm import SVR
from sklearn.linear_model import Ridge
from xgboost import XGBRegressor
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
```

Figure 1.1: List of libraries that we used in our project.

- NumPy (import numpy as np):
 - NumPy is a fundamental package for numerical computing in Python.
 - It offers support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate efficiently on these arrays.
- Pandas (import pandas as pd):

- Pandas is a powerful data manipulation and analysis library in Python.
- It provides data structures like DataFrame and Series, facilitating easy handling and manipulation of structured data.
- Missingno (import missingno as msno):
 - Missingno is a Python library that provides a set of tools for visualizing and dealing with missing data in datasets.
 - It offers visualizations such as bar charts and heatmaps to visualize missing data patterns.
- Matplotlib (import matplotlib.pyplot as plt):
 - Matplotlib is a comprehensive plotting library for Python
 - It enables the creation of a wide variety of plots, including line plots, scatter plots, bar plots, histograms, and more.
- StandardScaler (from sklearn.preprocessing import StandardScaler):
 - StandardScaler is a preprocessing technique from scikit-learn used for standardizing features by removing the mean and scaling to unit variance.
 - It is commonly applied to preprocess data before feeding it into machine learning algorithms.
- LabelEncoder (from sklearn.preprocessing import LabelEncoder):
 - LabelEncoder is a utility class from scikit-learn used for encoding categorical features as integer labels.
 - It transforms categorical data into numerical form, which is required by many machine learning algorithms.
- LinearRegression (from sklearn.linear model import LinearRegression):
 - LinearRegression is a linear regression model implementation from scikitlearn.
 - It fits a linear model to the training data and makes predictions based on the relationship between independent and dependent variables.
- train test split (from sklearn.model selection import train test split):
 - train test split is a function from scikit-learn used for splitting a dataset into training and testing sets.
 - It randomly splits the data into training and testing sets according to a specified ratio, enabling evaluation of machine learning models on unseen data.
- r2 score (from sklearn.metrics import r2 score):
 - r2 score is a function from scikit-learn used for evaluating the performance of regression models.

- It calculates the coefficient of determination (R-squared), which measures the proportion of the variance in the dependent variable that is predictable from the independent variables.
- mean squared error (from `sklearn.metrics` import `mean_squared_error`):
 - mean squared error is a function from scikit-learn used for evaluating the performance of regression models.
 - It calculates the mean squared error between the predicted and true values, providing a measure of the model's accuracy.
- mean absolute error (from `sklearn.metrics` import `mean_absolute_error`):
 - mean absolute error is a function from scikit-learn used for evaluating the absolute error produce by the regression models.
 - It calculates the mean absolute error between the predicted and true values, providing a measure of the model's accuracy.
- mean absolute error (from `sklearn.metrics` import `mean_absolute_error`):
 - mean absolute error is a function from scikit-learn used for evaluating the absolute error produce by the regression models.
 - It calculates the mean absolute error between the predicted and true values, providing a measure of the model's accuracy.
- `DecisionTreeRegressor` (from `sklearn.tree` import `DecisionTreeRegressor`):
 - `DecisionTreeRegressor` is a regression algorithm from scikit-learn used for predicting continuous target values.
 - It splits the data into subsets using decision rules and minimizes the error at each leaf node to make predictions.
- `RandomForestRegressor` (from `sklearn.ensemble` import `RandomForestRegressor`):
 - `RandomForestRegressor` is a regression algorithm from scikit-learn that uses an ensemble of decision trees.
 - It improves prediction accuracy and reduces overfitting by averaging predictions from multiple trees trained on different data subsets.
- `GradientBoostingRegressor` (from `sklearn.ensemble` import `GradientBoostingRegressor`):
 - `GradientBoostingRegressor` is a boosting algorithm from scikit-learn used for predicting continuous target values.
 - It builds decision trees sequentially, minimizing errors by optimizing the gradient of the loss function.
- `SVR` (from `sklearn.svm` import `SVR`):

- SVR is a regression model from scikit-learn that uses support vector machines for predicting continuous outputs.
- It works by finding a function within a margin of tolerance and minimizes errors outside the tolerance.
- Ridge (from `sklearn.linear model import Ridge`):
 - Ridge is a regression model from scikit-learn used for linear regression with L2 regularization.
 - It reduces overfitting by penalizing large coefficients in the linear model.
- XGBRegressor (from `xgboost import XGBRegressor`):
 - XGBRegressor is a gradient boosting algorithm from the XGBoost library used for regression tasks.
 - It offers high performance and scalability, with options for regularization and custom objective functions.

Chapter 2. Data Cleaning

2.1 About Dataset

Here the dataset is loaded into the file with the help of `pd.read_csv()`. The `df.head()` command will print the five samples of the dataset.

```
df = pd.read_csv("IRENA_RenewableEnergy_Statistics_2000-2022.csv", encoding='latin1')
# first five rows of the dataset
df.head()
```

	Region	Sub-region	Country	ISO3 code	M49 code	RE or Non-RE	Group Technology	Technology	Producer Type	Year	Electricity Generation (GWh)	Electricity Installed Capacity (MW)
0	Africa	Northern Africa	Algeria	DZA	12	Total Non-Renewable	Fossil fuels	Natural gas	On-grid electricity	2000	24585.0	5459.01
1	Africa	Northern Africa	Algeria	DZA	12	Total Non-Renewable	Fossil fuels	Natural gas	On-grid electricity	2001	25781.0	5455.50
2	Africa	Northern Africa	Algeria	DZA	12	Total Non-Renewable	Fossil fuels	Natural gas	On-grid electricity	2002	26994.0	5891.01
3	Africa	Northern Africa	Algeria	DZA	12	Total Non-Renewable	Fossil fuels	Natural gas	On-grid electricity	2003	28619.4	6013.24
4	Africa	Northern Africa	Algeria	DZA	12	Total Non-Renewable	Fossil fuels	Natural gas	On-grid electricity	2004	30312.0	6305.24

Figure 2.1: Loading dataset for analysis

Here, the encoding parameter in `pd.read_csv()` specifies the character encoding used to decode the file. It guarantees that files containing unusual or non-ASCII characters—which may not be compatible with the default utf-8 encoding—are read correctly. By explicitly setting the encoding (e.g., 'latin1' for files encoded in Latin-1), it helps avoid errors like `UnicodeDecodeError` and ensures that all characters in the file are correctly interpreted and processed.

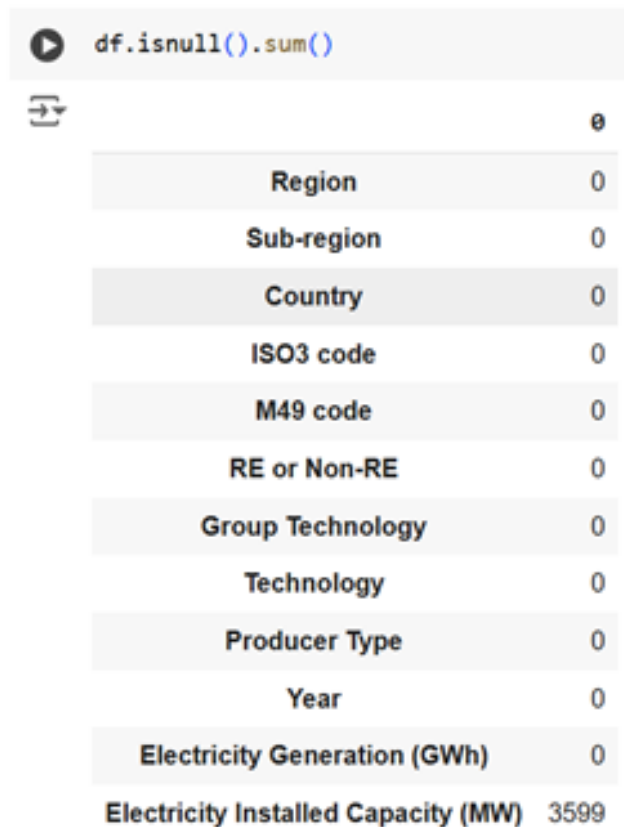
```
print("The shape of our dataset is: ", df.shape)
print("Total number of Rows in our dataset is: ", df.shape[0])
print("Total number of Columns in our dataset is: ", df.shape[1])
```

The shape of our dataset is: (35193, 12)
Total number of Rows in our dataset is: 35193
Total number of Columns in our dataset is: 12

Figure 2.2: Numbers of rows and column of the dataset

2.2 Missing Data Analysis

The first step in data-preprocessing is to check the null values present in the columns of the dataset. For that we have used `df.isnull().sum()`.



	0
Region	0
Sub-region	0
Country	0
ISO3 code	0
M49 code	0
RE or Non-RE	0
Group Technology	0
Technology	0
Producer Type	0
Year	0
Electricity Generation (GWh)	0
Electricity Installed Capacity (MW)	3599

Figure 2.3: Null values in the dataset

As we can observe, there are some missing values in our dataset. But all of these are present only in the last feature of the dataset, i.e; Electricity Installed Capacity (MW).

To visualize the null values in better way lets use different plot of missingno library.

```
msno.bar(df)
plt.show()
```

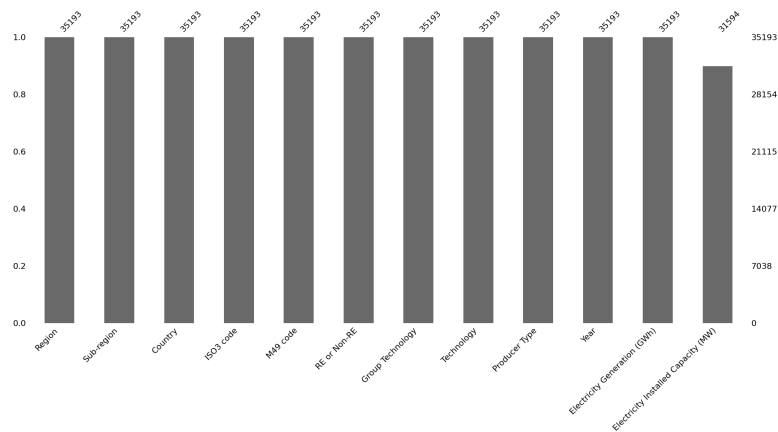



Figure 2.4: Barplot using missingno library

As we can clearly see the data is only missing in the "Electricity installed capacity". By plotting the matrix plot we can see where the exact missing values are there.



```
msno.matrix(df)
plt.show()
```

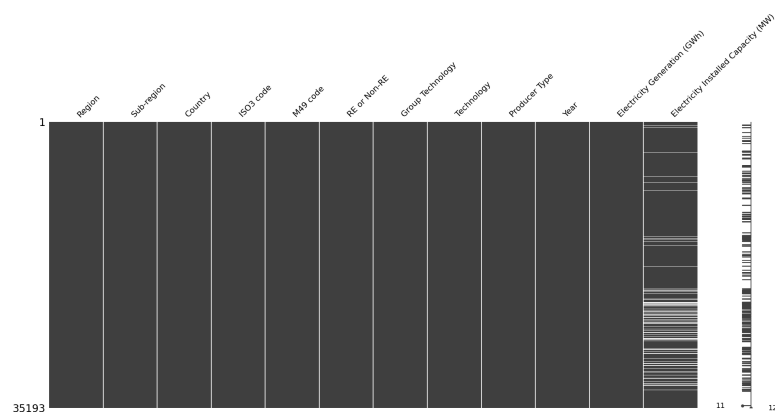


Figure 2.5: Matrix plot using missingno library

2.3 Features Correlation

In order to see which features are related to each other we use dendrogram plot of missingno library.



```
msno.dendrogram(df)
plt.show()
```

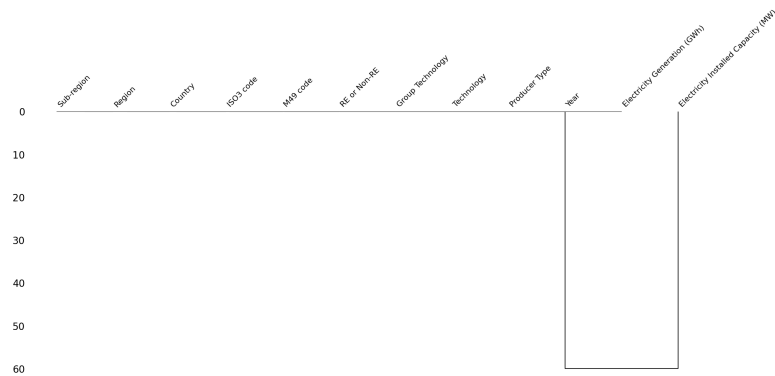


Figure 2.6: Dendrogram plot using missingno library

As there is only one feature with missing values in them. The dendrogram will appear sparse or show a trivial pattern.

So, in order to understand the nature of missingness. We observe our dataset and understand what each feature represents.

In this dataset, Electricity Installed Capacity (MW) refers to the maximum amount of electricity that power generation facilities (like power plants) can produce at full operational capacity under ideal conditions. It is measured in megawatts (MW) and represents the total capacity of all electricity-generating equipment installed within a specific system, region, or facility.

So, this means that the missing values in this feature simply represent that the data was not available to be recorded for some countries for some years.

Hence, the data is missing due to intrinsic reasons because it might be related to specific features like country, technology, or year.

Therefore, the distribution of these missing values is **Missing Not at Random (MNAR)**.

2.4 Imputation

So, our plan to handle these missing values is as follows:

1. Analyzing patterns of missingness:

- Checking if missing values are concentrated to some specific groups like regions or years.

2. Handling missing values:

- Adding a binary indicator column to retain information about missingness.
- Using group-wise imputation based on the relevant features such as Region, Technology etc.
- Imputing the remaining missing data with the overall median of the feature Electricity Installed Capacity (MW).

Chapter 3. Data Visualization

This chapter presents 17 visualizations providing insights into electricity generation data, including trends, distributions, and patterns across various dimensions.

3.1 Renewable vs Non-Renewable Electricity Generation by Region

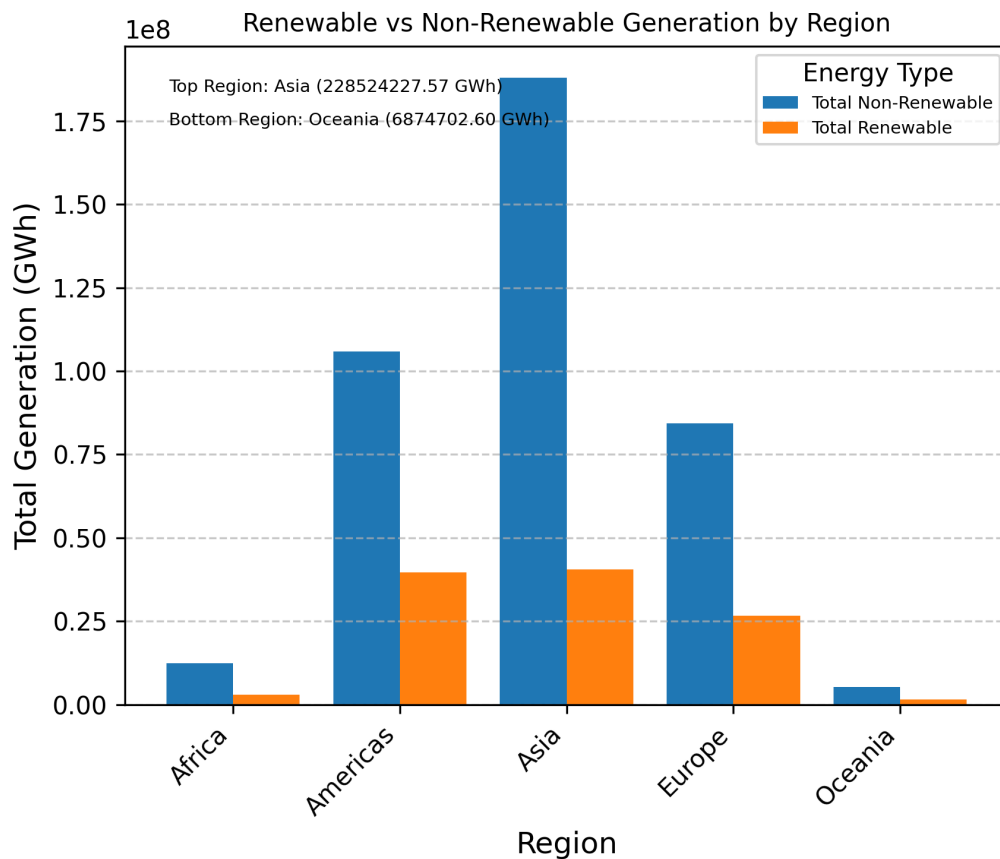


Figure 3.1: Comparison of Renewable and Non-Renewable Electricity Generation by Region

Observation:Non-Renewable energy dominates in all regions and renewable sources are less prevalent.

3.2 Electricity Generation by Technology Type

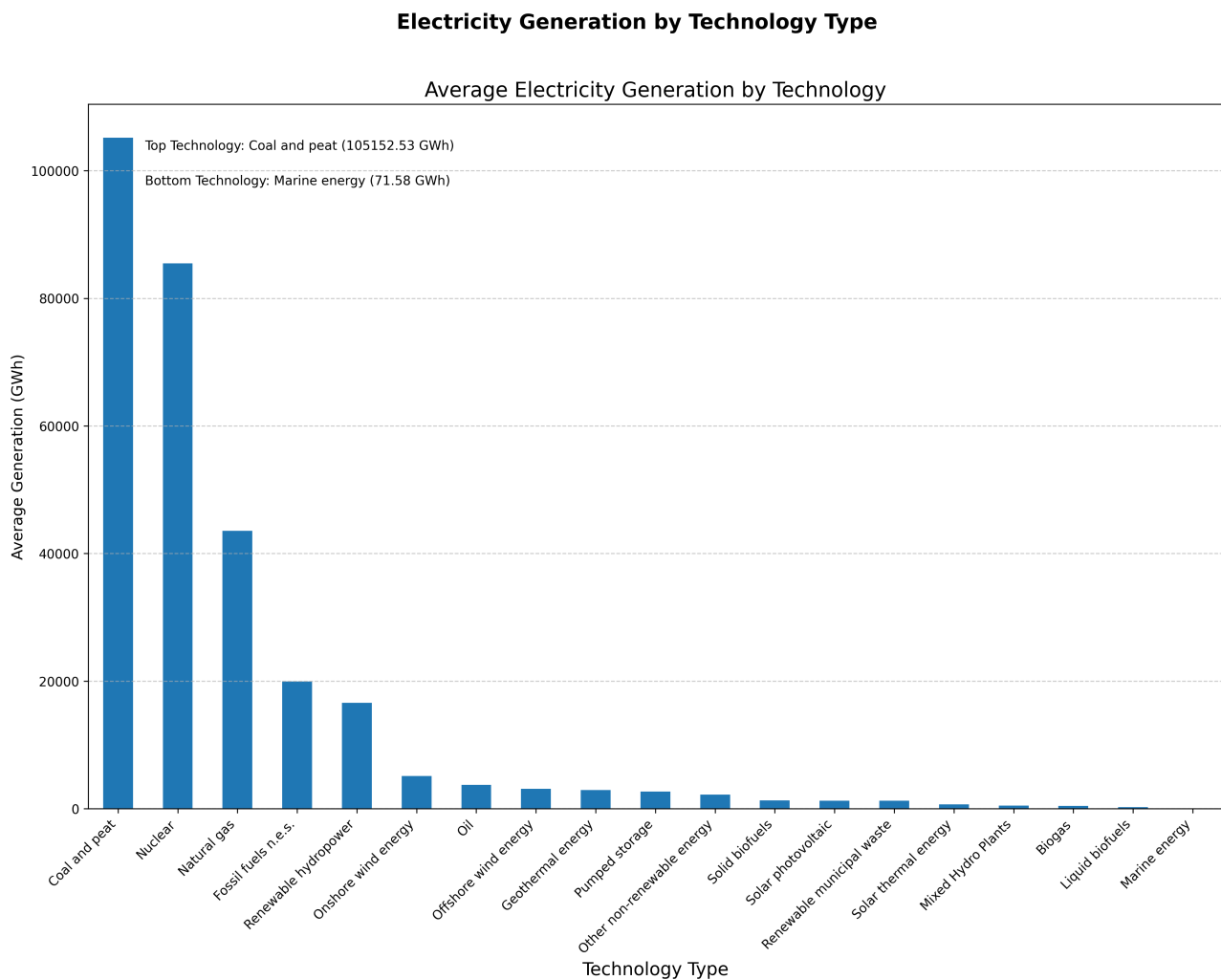


Figure 3.2: Electricity Generation by Technology Type

Observation: Fossil fuels remain dominant globally, but renewable technologies like wind and solar are gaining ground.

3.3 Energy Generation over Time

Renewable vs Non-Renewable Energy Generation Trend (2000-2022)

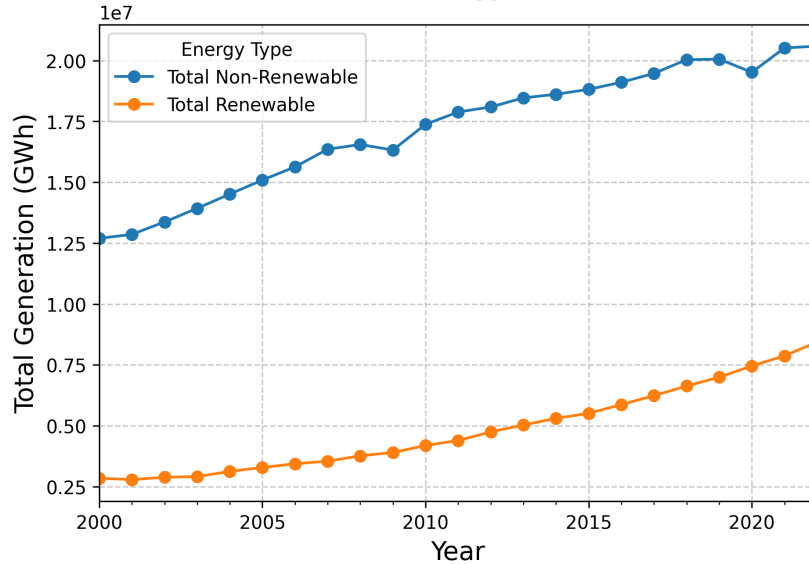


Figure 3.3: Trends in Electricity Generation Over Time

Observation: Renewable energy shows consistent growth over the years, while non-renewable sources exhibit variability.

3.4 Producer Type vs Region

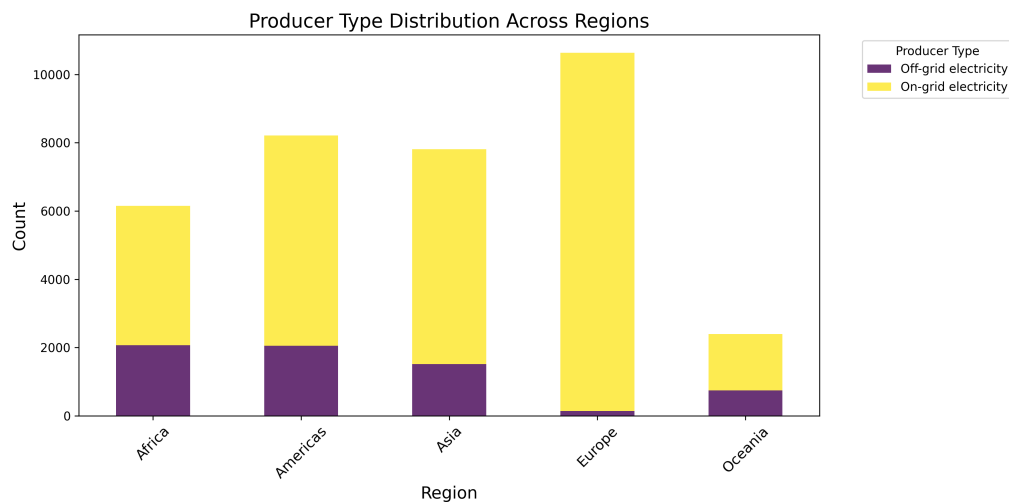


Figure 3.4: Electricity Generation by Producer Type Across Regions

Observation: Public and private producers contribute variably across regions, reflecting policy and market differences.

3.5 Electricity Generation Trend Over Years (for a Specific Region)

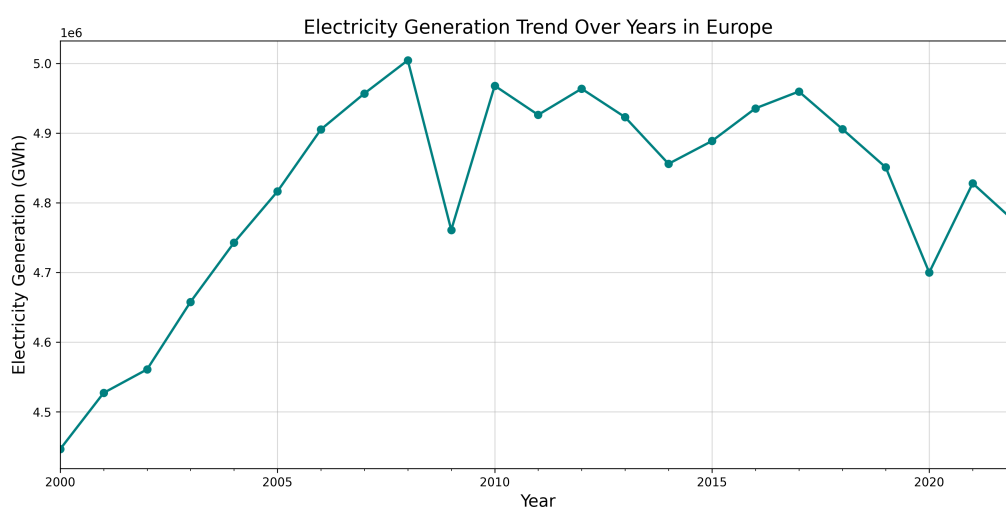


Figure 3.5: Electricity Generation Trend Over Years in Europe

Observation: The generation trend reflects policy-driven shifts towards renewables in recent years.

3.6 Installed Capacity vs Electricity Generation (Scatter Plot)

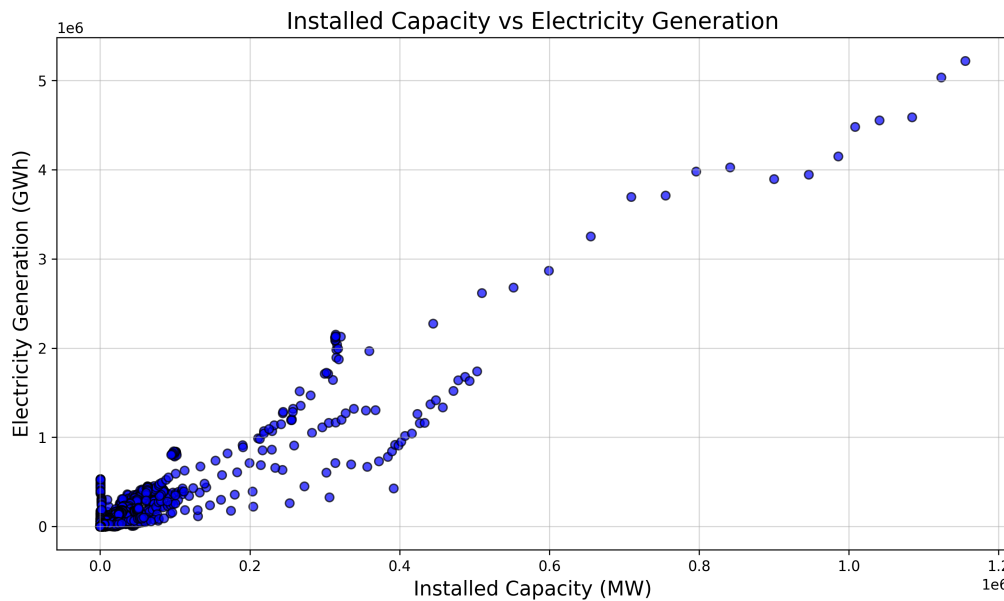


Figure 3.6: Relationship Between Installed Capacity and Electricity Generation

Observation: Installed capacity correlates strongly with electricity generation, with some outliers.

3.7 Top Sub-regions by Electricity Generated

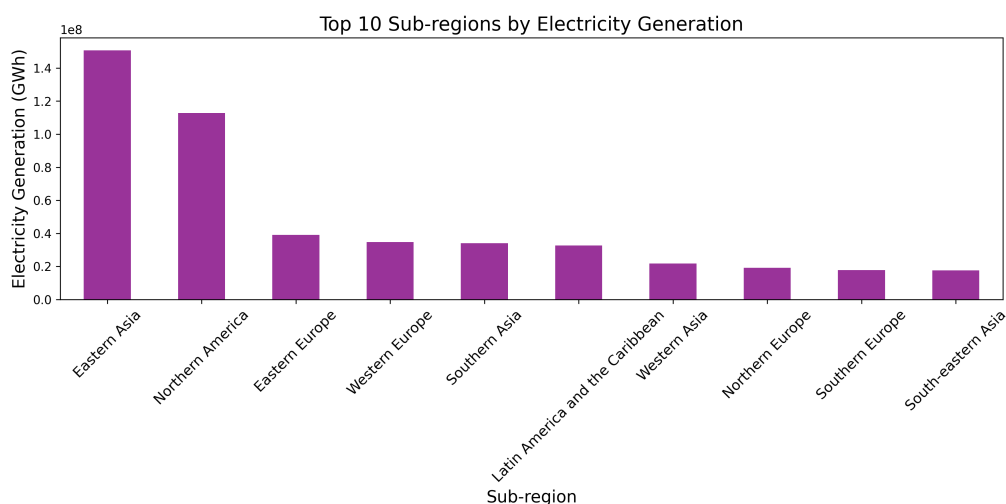


Figure 3.7: Top Sub-regions by Electricity Generated

Observation: The top sub-regions collectively contribute a significant majority of global electricity.

Chapter 4. Feature Engineering

Feature engineering is a vital part of improving machine learning model performance. In this section, we explore the methods used for data scaling, encoding, and splitting.

4.1 Feature extraction

The primary feature used in this analysis was "Electricity Installed Capacity (MW)," which is closely related to the target variable, "Electricity Generation (GWh)." Additionally, categorical variables like "Technology" and "Producer Type" were encoded into numerical values for model compatibility. These features were encoded with the use of `LabelEncoding`.

4.2 Feature selection

The selected features for the model training included:

- RE or Non-RE
- Group Technology
- Technology
- Producer Type
- Electricity Installed Capacity (MW)

Chapter 5. Model fitting

In this chapter, we discuss how various machine learning algorithms were trained on the data. We used multiple regression models and evaluated their performance using metrics such as R^2 , Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE).

5.1 Data Splitting

The data was split into training and testing sets using an 80-20 split. We used the following code to perform the data splitting:

- Features X consisted of the transformed and encoded data, including the columns 'RE or Non-RE', 'Group Technology', 'Technology', 'Producer Type', and 'Electricity Installed Capacity (MW)'.
- The target variable y was 'Electricity Generation (GWh)'.

The data was then split into training (80%) and test (20%) sets using the following code:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

5.2 Model Parameters

Each model used in this project has specific hyperparameters that can be tuned to optimize performance. Below are the key parameters for the models used:

- **Linear Regression:**
 - `fit_intercept`: Whether to calculate the intercept for this model. Default is True.
 - `normalize`: Whether to normalize the input features. Default is False.
- **Decision Tree Regressor:**
 - `max_depth`: The maximum depth of the tree. Controls overfitting.
 - `random_state`: Ensures reproducibility.
- **Random Forest Regressor:**
 - `n_estimators`: The number of trees in the forest. Default is 100.

- `random_state`: Ensures reproducibility.
- **Gradient Boosting Regressor:**
 - `n_estimators`: The number of boosting stages to perform. Default is 100.
 - `learning_rate`: The rate at which the model learns. Default is 0.1.
- **Support Vector Regressor:**
 - `kernel`: Specifies the kernel type to be used in the model. Default is 'rbf'.
 - `C`: Regularization parameter. The strength of regularization is inversely proportional to this value.
- **XGBoost Regressor:**
 - `n_estimators`: The number of boosting rounds. Default is 100.
 - `learning_rate`: Step size shrinking. Default is 0.1.
- **Ridge Regression:**
 - `alpha`: Regularization strength. Larger values indicate stronger regularization.

5.3 Linear Regression

Linear Regression model helps us make prediction about a dependent variable based on an independent variable. We assume that the relationship between what we're predicting and what we're using to make the prediction is a straight line.

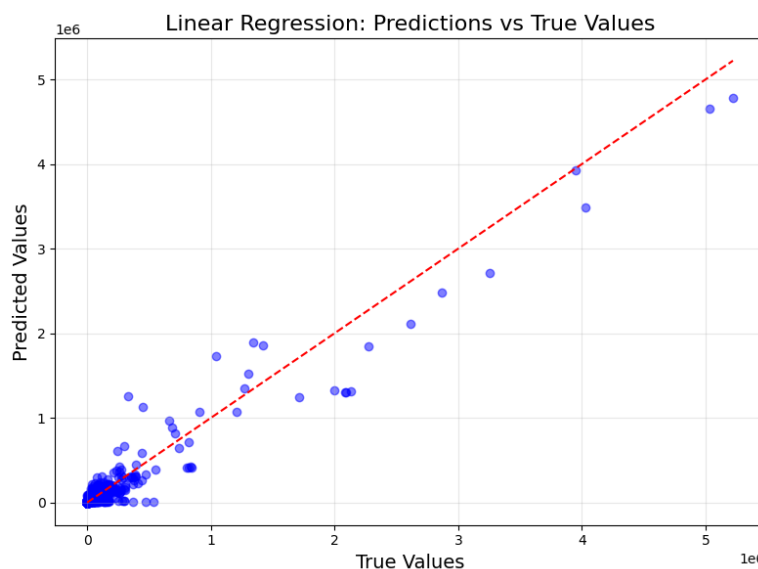


Figure 5.1: Linear Regression Scatter Plot

As we can infer from the above plot, the predicted values align reasonably well with the original values, though deviations increase for higher values.

5.4 Decision Tree Regressor

It is a non-linear model that splits data recursively into regions, making predictions based on the mean of the target variable in each region. It is prone to overfitting but works well for datasets with clear splits.

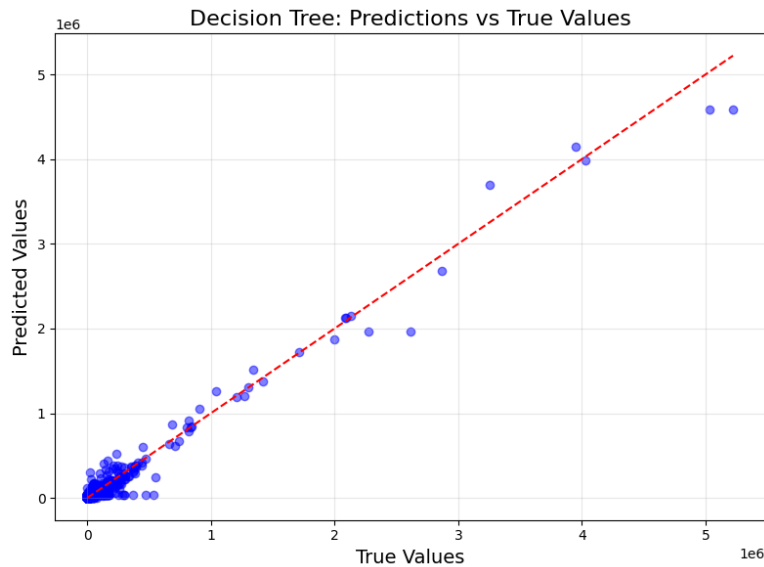


Figure 5.2: Decision Tree Regressor Scatter Plot

This model captures the true values reasonably well as compared to linear regression but shows slightly higher variance compared to the ensemble models.

5.5 Random Forest Regressor

This is an ensemble model that combines multiple decision trees through bagging, reducing overfitting and improving generalization, making it robust for most regression tasks.

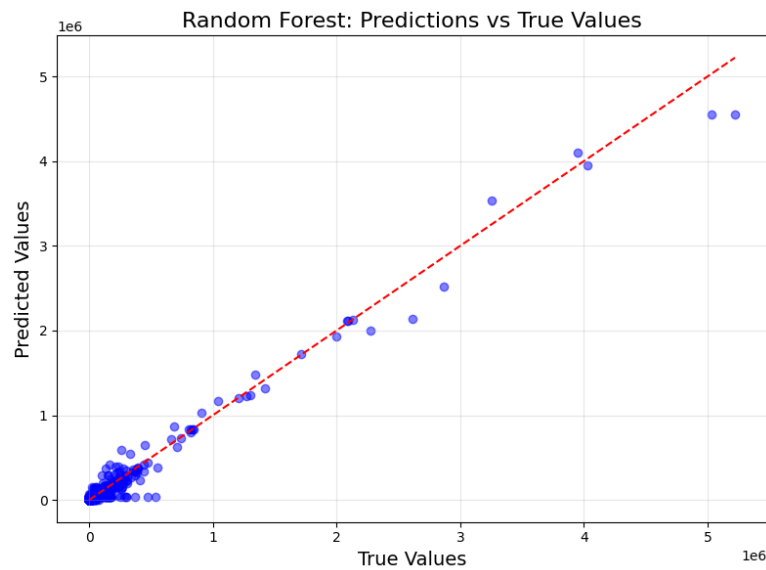


Figure 5.3: Random Forest Regressor Scatter Plot

The plot given above explains that the random forest model also demonstrates good alignment with the true values, with minor deviations for higher values.

5.6 Gradient Boosting Regressor

This is a powerful ensemble method that builds trees sequentially, optimizing for the residuals of previous models to improve accuracy, often excelling in predictive tasks with fine-tuned hyperparameters.

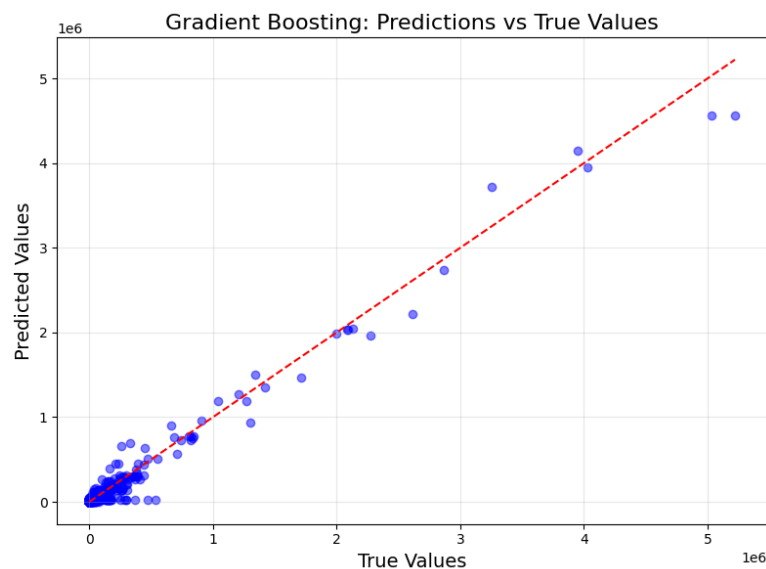


Figure 5.4: Gradient Boosting Regressor Scatter Plot

As we can infer from the above plot, this model's predictions closely align with the true or original values, showcasing strong predictive accuracy.

5.7 Support Vector Regressor

It is a regression method that fits the data within a defined margin, making it effective for small datasets with clear patterns, though sensitive to hyperparameter tuning.

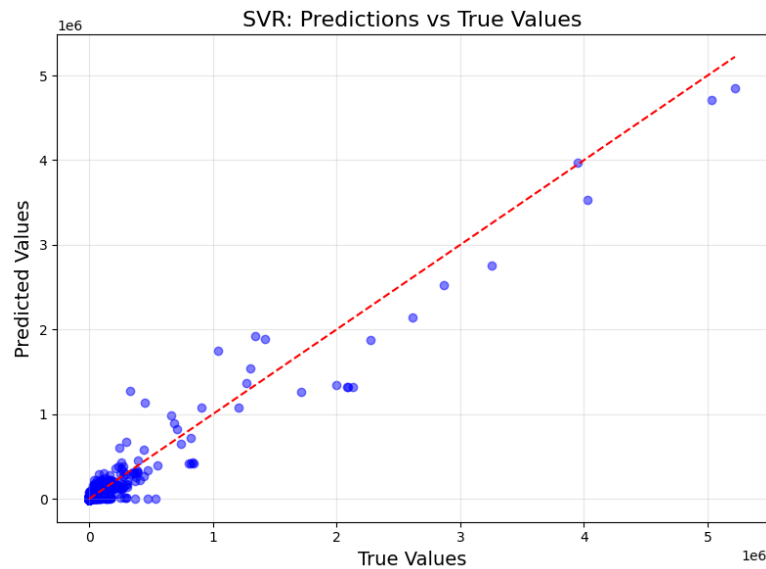


Figure 5.5: Support Vector Regressor Scatter Plot

As we can see, it performs well overall, maintaining alignment with the ideal line, though some deviations are visible for extreme values.

5.8 XGBoost Regressor

It is an advanced implementation of gradient boosting with optimizations like regularization and parallelization, delivering high accuracy and efficiency for complex datasets.

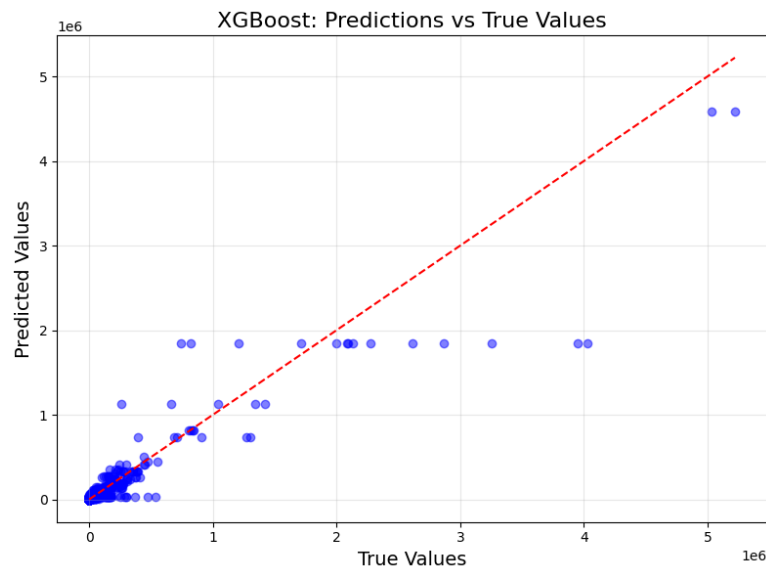


Figure 5.6: XGBoost Regressor Scatter Plot

It shows stronger predictive power with better alignment for higher value ranges but slight under-fitting at lower ranges.

5.9 Ridge Regression

Its a linear regression model with L2 regularization, effective in handling multicollinearity by penalizing large coefficients, ensuring a balance between simplicity and performance.

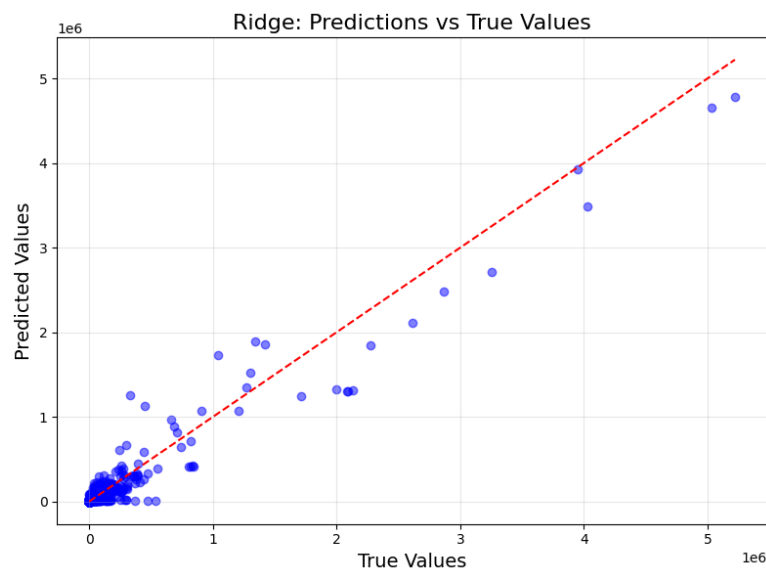


Figure 5.7: Ridge Regression Scatter Plot

Similar to Linear Regression, with slightly better regularization handling, particularly for extreme values.

Chapter 6. Conclusion & future scope

In conclusion, this project demonstrates the effectiveness of various regression models in predicting electricity generation based on features like installed capacity and technology type. The models' performances, as measured by R^2 , MSE, RMSE, and MAE, show that the Random Forest Regressor achieved the best results.

6.1 Findings/Observations

The key findings from the models' performance are as follows:

- The Random Forest Regressor achieved the highest accuracy, with an R^2 score of 0.98.
- The Decision Tree and Gradient Boosting models also performed well, with R^2 values above 0.97.
- Linear Regression and Ridge Regression provided a useful baseline, but did not outperform more complex models like Random Forest and Decision Tree.
- Support Vector Regression and XGBoost had lower performance compared to the others.

6.2 Challenges

The main challenges encountered during this project included:

- Handling outliers in the dataset, despite scaling and encoding.
- The need for domain expertise to select relevant features and tune models effectively.
- Models like XGBoost and Support Vector Regression performed suboptimally in this context, possibly due to hyperparameter settings or data characteristics.

6.3 Future Plan

Future work will focus on:

- Integrating additional features such as geographic location or renewable energy type to enhance model accuracy.
- Experimenting with deep learning models for further improvements in prediction.

- Exploring the use of ensemble methods to combine the strengths of multiple models.
- Fine-tuning models like XGBoost and Support Vector Regression to improve performance.

6.4 Model Evaluation Metrics Comparison

The table below summarizes the evaluation metrics for each of the models used in this study:

Model	R^2	RMSE	MAE
Linear Regression	0.93	38,550.46	9,331.67
Decision Tree	0.97	24,087.56	4,570.90
Random Forest	0.98	22,788.08	4,359.28
Gradient Boosting	0.97	23,812.67	4,986.97
Support Vector	0.93	38,345.44	11,223.93
XGBoost	0.87	53,777.41	6,103.03
Ridge Regression	0.93	38,551.76	9,331.82

Table 6.1: Model Evaluation Metrics Comparison

Here, we present bar charts comparing the R^2 scores and RMSE values of all the models used in this study:

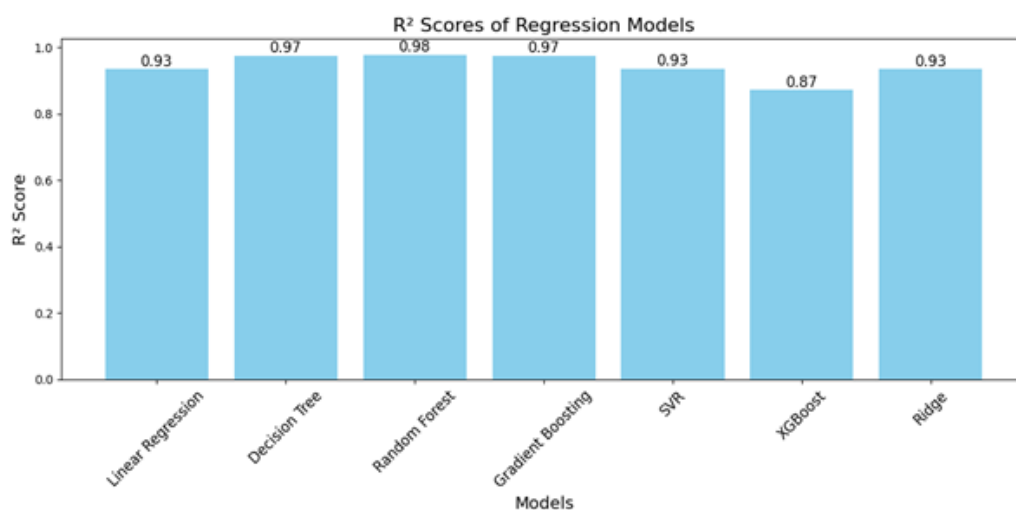


Figure 6.1: R^2 scores comparison

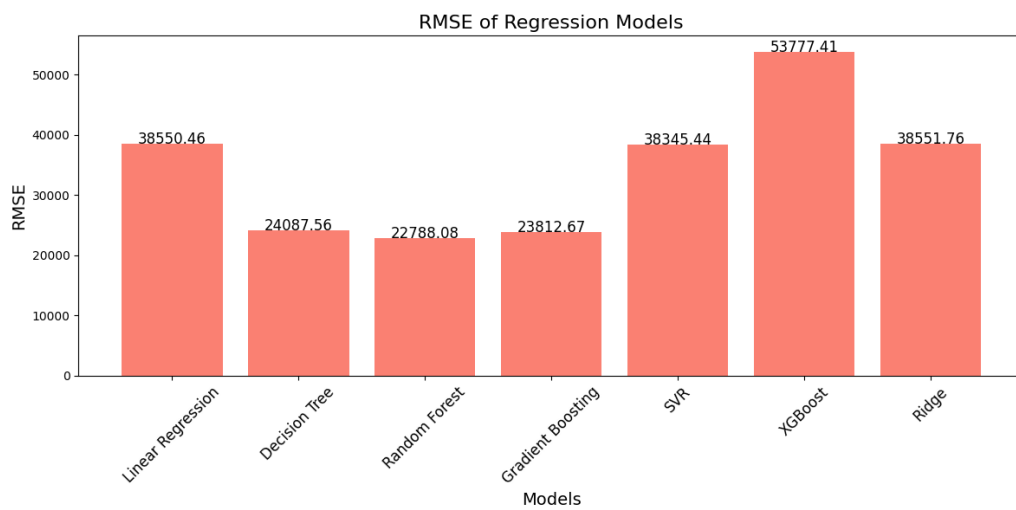


Figure 6.2: RMSE values comparison

Group Contribution

Neel Bokil

Contributions: Found the dataset. Created the end to end ML pipeline and conducted analysis of the data, pre-processing the data and model creation. Helped in report.

Yash Sorathiya

Contribution: Helped in visualizing the data to gain better and deeper insights of the data. Helped in model predictions. Helped in report.

Divyakumar Tandel

Contribution: Helped in adding some more content in the plots of the model comparisons. Created the presentation. Helped in report.

Short Bio

1. **Neel Bokil:** I am devoted to pursue my dream to excel in the field of data science and machine learning. I have the capacity to do more. I most certainly appreciate the challenges that I face in solving complex problems or dealing with complex situations. All of these transform me to a better self. With a strong sense in current technological developments and a keen enthusiasm for innovation, I aim to make a meaningful impact in every project I pursue.

I focus on being flexible, cooperating well with the others, and paying attention to details, which help me succeed in changing environments.

My goal is to make a difference and keep improving both personally and professionally.

2. **Yash Sorathiya:** I am a passionate and motivated individual with a strong drive to learn and grow. I enjoy taking on challenges that require creativity, problem-solving, and dedication to achieve meaningful results. With a background in technology and a deep interest in innovative solutions, I strive to contribute positively to every endeavor I undertake.

I value adaptability, teamwork, and attention to detail, which enable me to excel in dynamic environments. In my free time, I enjoy exploring new ideas, staying curious about advancements, and engaging in activities that inspire creativity and growth. My goal is to make a positive impact and continuously evolve both personally and professionally.

3. **Divyakumar Tandel:** I am deeply passionate about machine learning and computational science, constantly driven by a desire to learn new things. Collaboration and problem-solving are my strengths, and I excel in computational analysis, which fuels my ability to tackle complex challenges.

I approach every obstacle with determination, putting in my full effort to find effective solutions. My goal is to grow both personally and professionally, striving to become a hardworking and outstanding individual in my field. With a keen enthusiasm for innovation and continuous improvement, I aim to make a meaningful impact in every project I undertake.

References

- [1] IRENA Dataset URL: <https://www.kaggle.com/datasets/shakaal/global-renewable-energy-production-2000-2022>
- [2] Geeks for geeks URL: <https://www.geeksforgeeks.org/>
- [3] W3Schools URL: <https://www.w3schools.com/>