Global Energy Consumption for Sustainable Development

Group Members: Frank Kuukyee, Deysy Londono, Rookayat Adedo, Yao Zhong.

1. Introduction

Our everyday lives depend on reliable and affordable energy. A well-established energy system supports all sectors: from businesses, health services, education to agriculture[1]. Electricity use is rising fast everywhere, it is estimated that its consumption will increase by 50% from 2021 to 2040 [2], and without a stable electricity supply, our society will not be able to function properly.

However, still many countries lack or have limited access to electricity; more that 800 million people live without access to this basic service. This figure corresponds approximately to one seventh of the world population. This is one reason why the United Nations have included "Affordable and Clean Energy" as one of the 17 Goals in its agenda for sustainable development provides a shared blueprint for peace and prosperity for people and the planet, now and into the future.

Understanding how energy is used globally is essential to decoding patterns, anticipating energy demands, enabling energy suppliers to optimize energy distribution, and allowing governments and agencies to focus resources towards the afformentioned goal.

Energy resources, such as oil, gas, and coal, are not evenly distributed globally, which creates disparities in energy access. Countries with abundant energy resources can significantly influence global energy markets, while regions with limited access to energy resources face challenges in meeting their energy demands, holding back economic development, and worsening socioeconomic disparities.

Expected Outcome. While this dataset holds great potential, we will focus our attention on developing a few models that that will enable us to predict if a country has mode that 99% access to electricity and determine which of these models would better serve the purpose.

2. Dataset

The "Global Data on Sustainable Energy" dataset [3] is a comprehensive collection of data spanning various countries and years, focusing on sustainable energy and its socioeconomic implications. It consists of 3,649 rows and 21 columns, covering a wide range of indicators such as access to

electricity, renewable energy capacity, financial flows to developing countries for sustainable projects, the mix of energy sources (including fossil fuels, nuclear, and renewables), and CO2 emissions. Additionally, it includes key economic metrics like GDP growth and per capita values, along with geographical data like population density, land area, and coordinates. This dataset is a valuable resource for analyzing global trends in sustainable energy, economic development, and their environmental impacts.

Column Name	Description
Entity	The name of the country or region for which the data is reported
Year	The year for which the data is reported (2000 to 2020)
Access to electricity	The percentage of population with access to electricity
Access to clean fuels for cooking	The percentage of the population with primary reliance on clean fuels
Renewable-electricity-generating-capacity-per-capita	Installed Renewable energy capacity per person
Financial flows to developing countries (US $)	Aid and assistance from developed countries for clean energy projects
Renewable energy share in total final energy consumption (%)	Percentage of renewable energy in final energy consumption
Electricity from fossil fuels (TWh)	Electricity generated from fossil fuels (coal, oil, gas) in terawatt-hours
Electricity from nuclear (TWh)	Electricity generated from nuclear power in terawatt-hours
Electricity from renewables (TWh)	Electricity generated from renewable sources (hydro, solar, wind, etc.) in terawatt-hours
Low-carbon electricity (% electricity)	Percentage of electricity from low-carbon sources (nuclear and renewables)
Primary energy consumption per capita (kWh/person)	Energy consumption per person in kilowatt-hours
Energy intensity level of primary energy	Energy use per unit of GDP at purchasing power parity
Value_co2_emissions (metric tons per capita)	Carbon dioxide emissions per person in metric tons
Renewables (% equivalent primary energy)	Equivalent primary energy that is derived from renewable sources
GDP growth (annual %)	Annual GDP growth rate based on constant local currency
GDP per capita	Gross domestic product per person

Column Name	Description
Density (P/Km2)	Population density in persons per square kilometer
Land Area (Km2)	Total land area in square kilometers
Latitude	Latitude of the country's centroid in decimal degrees
Longitude	Longitude of the country's centroid in decimal degrees

This dataset will serve as the foundation for our analysis, enabling us to dissect energy usage patterns and predict future trends.

3. Methodology.

The table below shows the first five rows of the original dataset `Global Data on Sustainable energy.csv`.

In [ ]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import plotly.express as px
import matplotlib.pyplot as plt
import warnings
import statsmodels.api as sm
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.impute import KNNImputer
from sklearn.metrics import accuracy_score,mean_squared_error,mean_absolute_error,r2_scor
from sklearn.ensemble import RandomForestRegressor

warnings.filterwarnings("ignore")

original_df = pd.read_csv('https://raw.githubusercontent.com/ddlondono/DATA606/main/Susta
original_df.head()
```

Out[ ]:

	Entity	Year	Access to electricity (% of population)	Access to clean fuels for cooking	Renewable-electricity-generating-capacity-per-capita	Financial flows to developing countries (US $)	Renewable energy share in the total final energy consumption (%)	Electricity from fossil fuels (TWh)	El
0	Afghanistan	2000	1.613591	6.2	9.22	20000.0	44.99	0.16	
1	Afghanistan	2001	4.074574	7.2	8.86	130000.0	45.60	0.09	
2	Afghanistan	2002	9.409158	8.2	8.47	3950000.0	37.83	0.13	
3	Afghanistan	2003	14.738506	9.5	8.09	25970000.0	36.66	0.31	
4	Afghanistan	2004	20.064968	10.9	7.75	NaN	44.24	0.33	

5 rows × 21 columns

3.1. Pre-Processing of the Data

To begin with, we checked for missing data in the dataset and found that some columns have a signifficant number of missing values, in some cases more that 50%. Since deleting missing values in a dataset can lead to several drawbacks, a preliminary analysis using a random sample and OLS method is performed to ensure that our analysis would be based on the most complete information possible.

In [ ]:
```
features=original_df.columns.values
missing_val=original_df.isnull().sum()
non_missing_val= original_df.notnull().sum()
total_val=original_df.shape[0]
percentage_missing=missing_val/total_val*100
percentage_missing
```

Out[ ]:
```
Entity                                                              0.000000
Year                                                                0.000000
Access to electricity (% of population)                             0.274048
Access to clean fuels for cooking                                   4.631406
Renewable-electricity-generating-capacity-per-capita               25.513839
Financial flows to developing countries (US $)                     57.248561
Renewable energy share in the total final energy consumption (%)    5.316525
Electricity from fossil fuels (TWh)                                 0.575500
Electricity from nuclear (TWh)                                      3.453001
Electricity from renewables (TWh)                                   0.575500
Low-carbon electricity (% electricity)                              1.151000
Primary energy consumption per capita (kWh/person)                  0.000000
Energy intensity level of primary energy (MJ/$2017 PPP GDP)         5.672787
Value_co2_emissions_kt_by_country                                  11.729241
Renewables (% equivalent primary energy)                           58.563990
gdp_growth                                                          8.687312
gdp_per_capita                                                      7.728145
Density(P/Km2)                                                      0.027405
Land Area(Km2)                                                      0.027405
Latitude                                                            0.027405
Longitude                                                           0.027405
dtype: float64
```

3.1.1. OLS Model

To ensure that some columns of missing data can actually be ommited form our analysis, we run a linear regression includiig all the features and using as a sample all the entries that are complete. This model will not serve our original purpose of predicting, but will be useful to know the significance of each of the variables.

In [ ]:
```
original_df['Density(P/Km2)'] = original_df['Density(P/Km2)'].replace(',', '', regex=True
df_ols = original_df.dropna()
X = sm.add_constant(df_ols.drop(['Entity', 'Year', 'Access to electricity (% of populatio
ols_model = sm.OLS(df_ols['Access to electricity (% of population)'], X)
results = ols_model.fit()                                    # Perform the regression
print(results.summary())
```

OLS Regression Results

Dep. Variable:	Access to electricity (% of population)	R-squared:	0.787
Model:	OLS	Adj. R-squared:	0.775
Method:	Least Squares	F-statistic:	66.39
Date:	Mon, 19 Feb 2024	Prob (F-statistic):	4.26e-97
Time:	16:14:48	Log-Likelihood:	-1108.9
No. Observations:	343	AIC:	2256.
Df Residuals:	324	BIC:	2329.
Df Model:	18		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	83.3384	5.404	15.422	0.000	72.707	93.969
Access to clean fuels for cooking	0.0245	0.051	0.484	0.629	-0.075	0.124
Renewable-electricity-generating-capacity-per-capita	0.0378	0.010	3.730	0.000	0.018	0.058
Financial flows to developing countries (US $)	1.527e-10	9.73e-10	0.157	0.875	-1.76e-09	2.07e-09
Renewable energy share in the total final energy consumption (%)	-0.4924	0.066	-7.406	0.000	-0.623	-0.362
Electricity from fossil fuels (TWh)	0.0101	0.009	1.180	0.239	-0.007	0.027
Electricity from nuclear (TWh)	0.0301	0.058	0.521	0.603	-0.084	0.144
Electricity from renewables (TWh)	-0.0426	0.011	-3.826	0.000	-0.065	-0.021
Low-carbon electricity (% electricity)	0.2280	0.093	2.446	0.015	0.045	0.411
Primary energy consumption per capita (kWh/person)	0.0002	9.58e-05	1.914	0.057	-5.11e-06	0.000
Energy intensity level of primary energy (MJ/$2017 PPP GDP)	-1.1552	0.255	-4.523	0.000	-1.658	-0.653
Value_co2_emissions_kt_by_country	-2.069e-06	3.4e-06	-0.609	0.543	-8.76e-06	4.62e-06
Renewables (% equivalent primary energy)	0.0840	0.198	0.424	0.672	-0.306	0.474
gdp_growth	0.1116	0.087				

```
1.282      0.201      -0.060       0.283
gdp_per_capita                                                0.0012      0.000
4.537      0.000       0.001       0.002
Density(P/Km2)                                               -0.0131      0.002
-5.381      0.000      -0.018      -0.008
Land Area(Km2)                                             1.189e-06   3.74e-07
3.180      0.002    4.53e-07    1.92e-06
Latitude                                                      0.1935      0.025
7.863      0.000       0.145       0.242
Longitude                                                     0.0933      0.011
8.538      0.000       0.072       0.115
==============================================================================
Omnibus:                       28.749   Durbin-Watson:                   0.490
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               49.611
Skew:                           0.518   Prob(JB):                     1.69e-11
Kurtosis:                       4.549   Cond. No.                     6.76e+09
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specifie
d.
[2] The condition number is large, 6.76e+09. This might indicate that there are
strong multicollinearity or other numerical problems.
```

From the above results, we can infer that the following predictors with a *p-value>0.1* could be removed form the model:

- Access to Clean Cooking
- Financial flows to developing countries
- Renewable energy share in the total final energy consumption
- Electricity from fossil fuels (TWh)
- Electricity from nuclear (TWh)
- Value_co2_emissions_kt_by_country

Then we decided to drop the columns with more that 25% missing values at significance 0.1. For the retained columns, we filled in the missing values using a method called k-nearest neighbors (KNN) and using *k=5*. This technique works by looking at the 5 data points most similar to the one missing data and using their average values to fill in the gap.

This approach allowed us to keep valuable data for our analysis while ensuring that our dataset was as complete and accurate as possible.

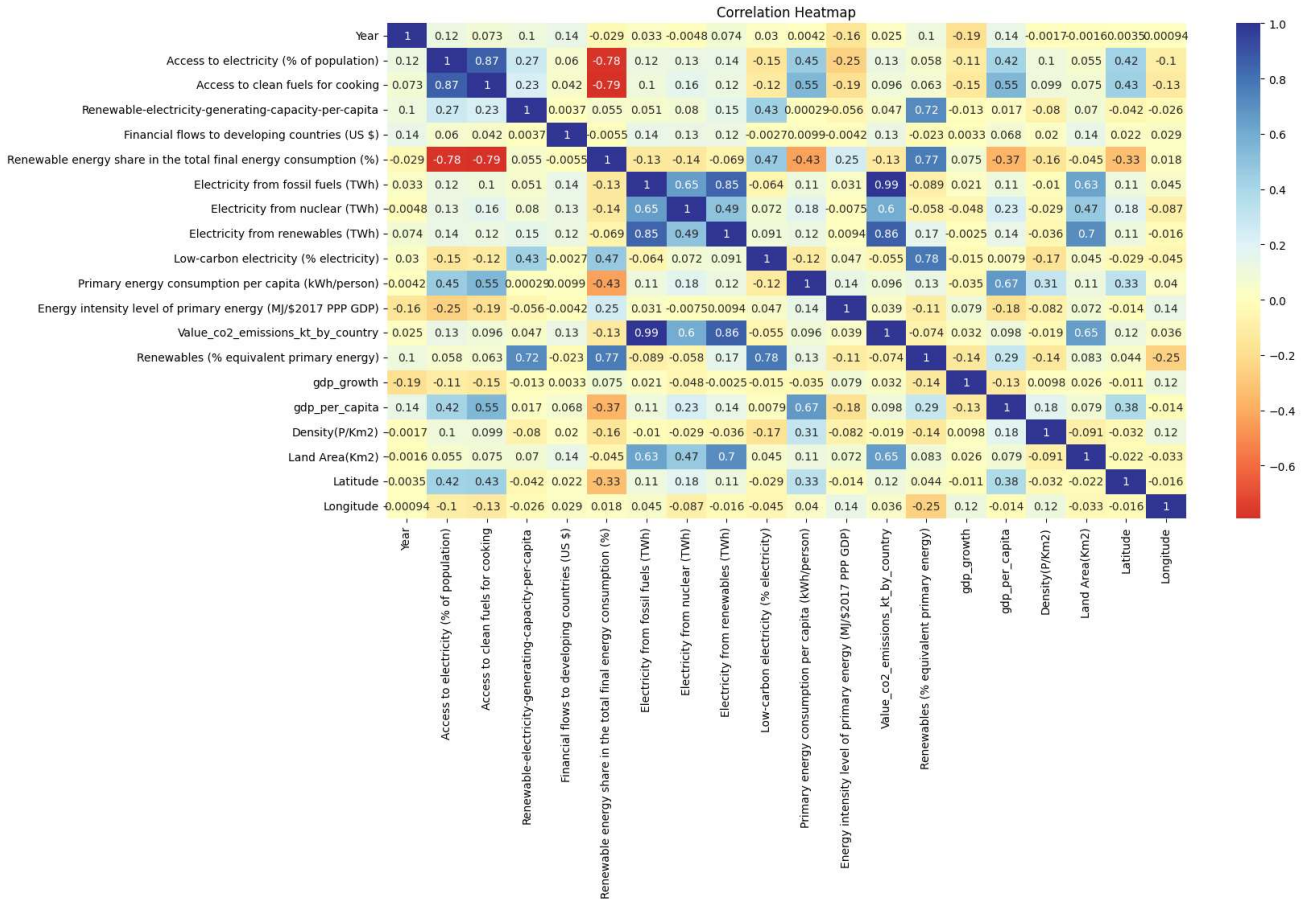
4. Exploratory analysis

This section will consist on exploring and visualizing different variables such us access to electricity by country, Access to clean fuels for cooking, relationship between compsumption of energy and

GDP, etc.

The following heatmap, shows in dark blue tones the variables that are more positively correlated.

In [ ]:
```
df=original_df.copy()
cm = df.corr()
plt.figure(figsize=(16, 8))
sns.heatmap(cm, annot=True, cmap='RdYlBu', center=0)
plt.title('Correlation Heatmap')
plt.show()
```



From the first column of this heatmap, we can conclude that the features most positively correlated to the access to electricity are:

- Access to clean fuel for cooking (74%)
- gdp per capita (58%)
- Primary Energy consumption per capita (49%)

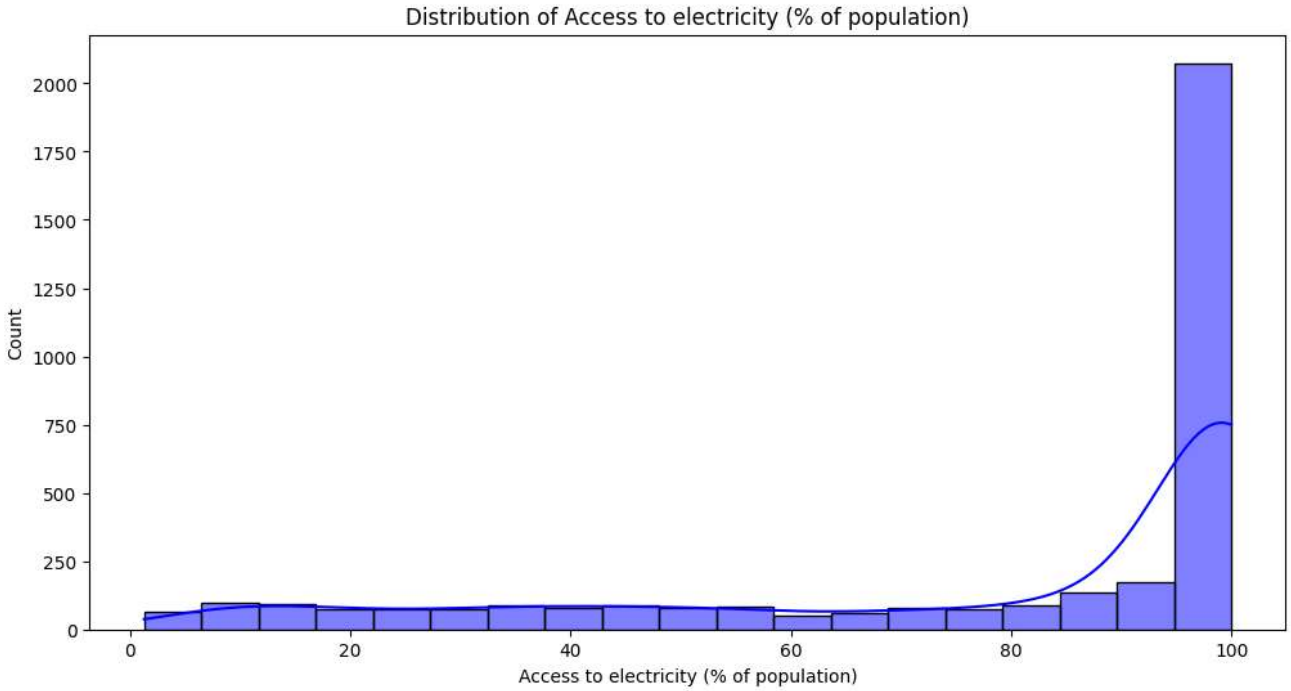
The features that have a negativecorrelation with the access to electricity are:

- Density (69%)
- Renewable Energy share in the total final energy consumption (58%)

Please note that this correlation heatmap shows only linear correlation between the variables, any non-linear relationship is not reflected in this heatmap.

The following histogram shows the distribution of access to electricity across different countries in the dataset. The curved KDE line (known as Kernel Density Estimate) gives a smooth estimate of the distribution, showing where the data points are concentrated.

In [ ]:
```
# Distribution Plot for 'Access to electricity (% of population)'
plt.figure(figsize=(12, 6))
sns.histplot(df['Access to electricity (% of population)'].dropna(), kde=True, color="blu
plt.title('Distribution of Access to electricity (% of population)')
plt.xlabel('Access to electricity (% of population)')
plt.show()
```



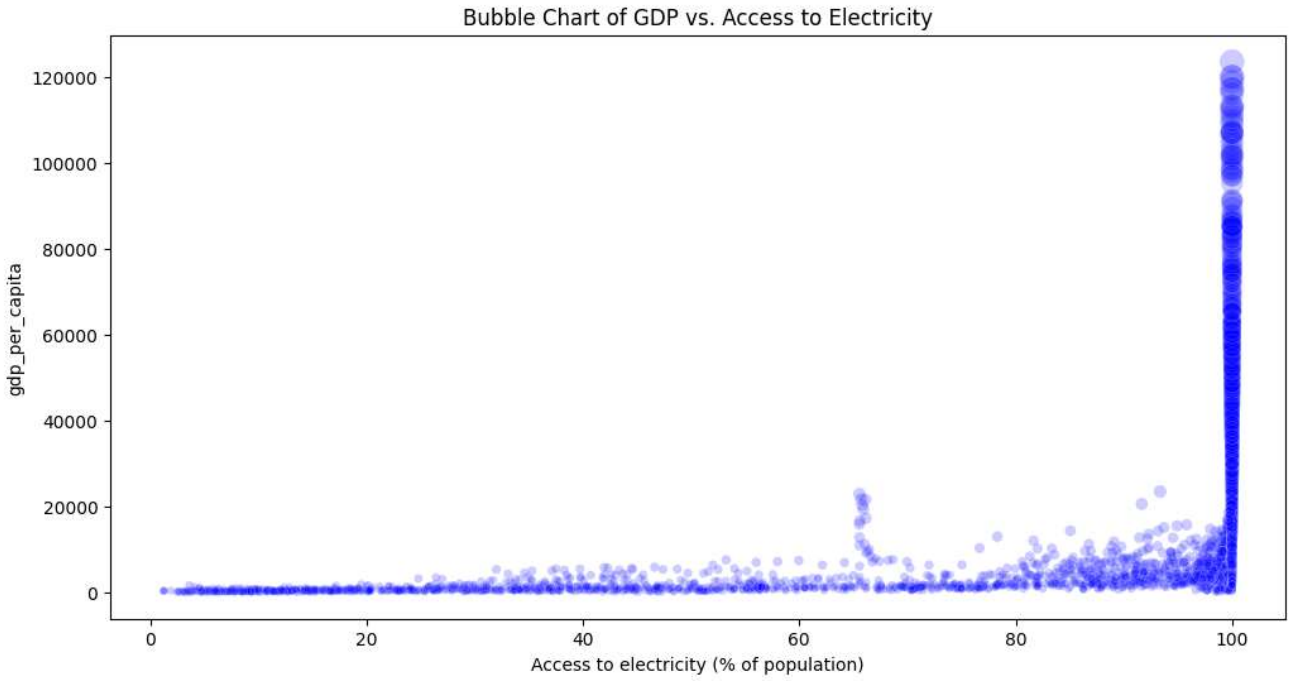
Observations:

- Skewed Right: The distribution of the data is right-skewed, which means there are fewer regions with lower electricity access and a larger number with higher access.
- Peaks: The tallest bar is at the 100% mark, indicating a significant number of observations where the entire population has access to electricity.
- Insight: This distribution suggests that while a substantial portion of the dataset has full electricity access, there are still some regions having lower access levels, pointing out the uneven distribution of electricity access among different areas.

In the following bubble chart we can visualize the relationship between access to electricity and GDP per capita, with the size of each bubble representing the GDP per capita. The bubble chart

offers a multi-dimensional view of the data and can be particularly useful for identifying patterns and correlations.

In [ ]:
```
# Bubble Chart of GDP vs. Access to Electricity
plt.figure(figsize=(12, 6))
sns.scatterplot(
    data=df,
    x='Access to electricity (% of population)',
    y='gdp_per_capita',
    size='gdp_per_capita',
    color="blue",
    legend=False,
    sizes=(20, 200),
    alpha=0.2
)
plt.title('Bubble Chart of GDP vs. Access to Electricity')
plt.show()
```

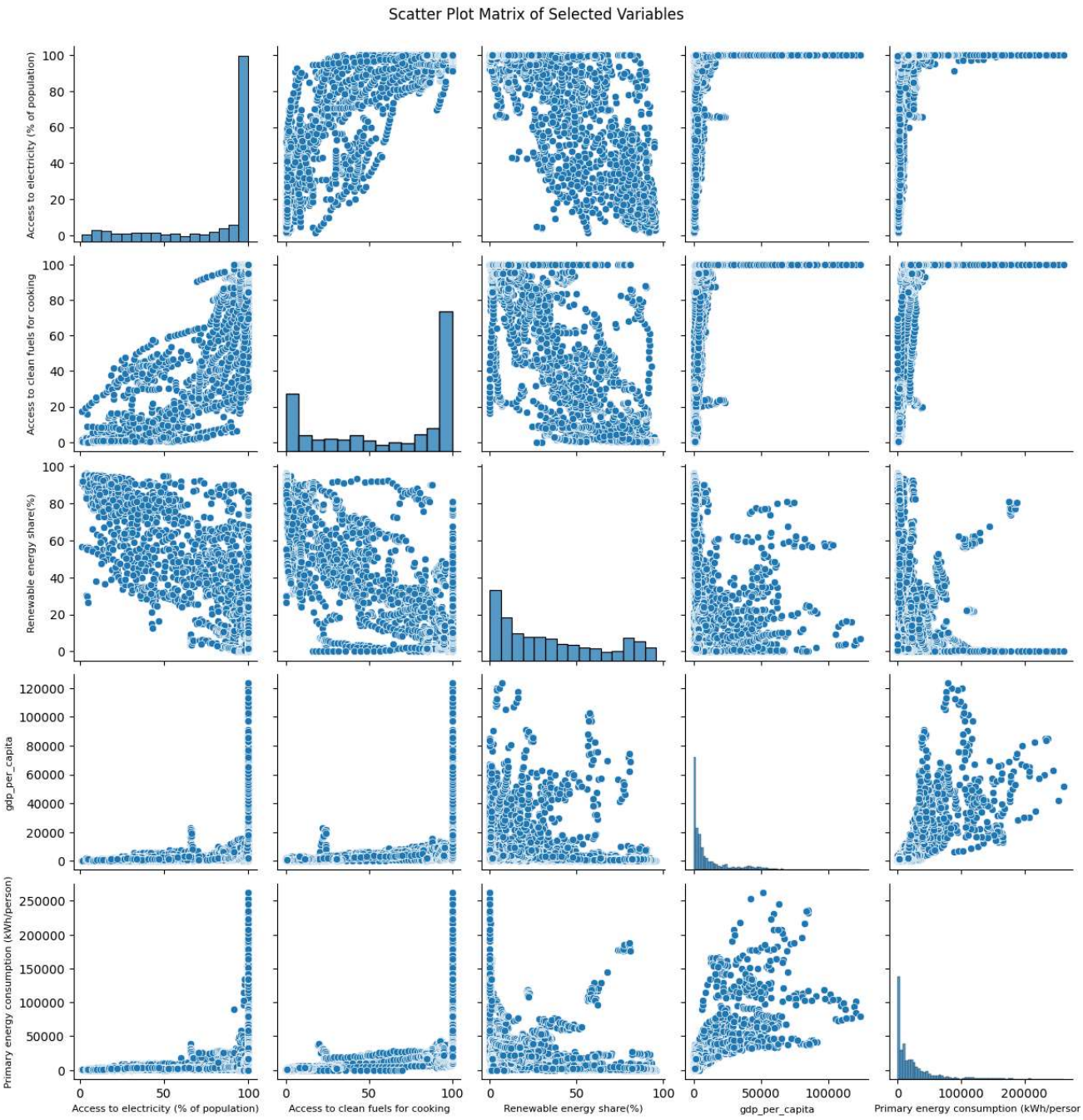


Observations:

- High GDP and Full Access to Electricity : There is a concentration of large bubbles near the 100% mark on the x-axis, suggesting that countries with high GDP per capita also have full access to electricity.
- Spread in Lower GDP: There is more spread on the y-axis at lower levels of electricity access, which could indicate that countries with lower GDP per capita have varying levels of access to electricity.
- Overall Trend: There seems to be a trend where as access to electricity increases, so does GDP per capita, although this is not a strict one-to-one relationship and there are exceptions.

The scatter plot matrix below shows the bivariate relationships between different combinations of selected variables from the dataset. The selected variables are Access to electricity (% of population), Access to clean fuels for cooking, Renewable energy share in the total final energy consumption (%), gdp_per_capita, and Primary energy consumption per capita (kWh/person). The scatter plot matrix provides a visual summary that can help identify which variables might be worth investigating further in a predictive model or other detailed analyses. The scatter plots also reveal different types of relationships: positive correlation (as one variable increases, so does the other), negative correlation (as one variable increases, the other decreases), or no correlation (variables do not show any clear relationship). These relationships help to hypothesize about potential causes and effects.

In [ ]:
```
selected_columns = ['Access to electricity (% of population)', 'Access to clean fuels for
                    'Renewable energy share in the total final energy consumption (%)',
                    'gdp_per_capita', 'Primary energy consumption per capita (kWh/person)
pairplot_df = df[selected_columns].dropna()
pairplot_df.rename(
    columns={"Renewable energy share in the total final energy consumption (%)": "Renewab
             'Primary energy consumption per capita (kWh/person)':'Primary energy consump
    inplace=True,
)

pair_plot = sns.pairplot(pairplot_df)
for ax in pair_plot.axes.flatten():
    ax.set_xlabel(ax.get_xlabel(), fontsize=8)
    ax.set_ylabel(ax.get_ylabel(), fontsize=8)
plt.suptitle('Scatter Plot Matrix of Selected Variables', y=1.02)
plt.show()
```



Observations:

- Diagonal Histograms: The diagonal plots are histograms showing the distribution of each variable independently. For example, we can see the distribution of access to electricity, where most of the data points are clustered towards higher percentages, suggesting that most of the included entities have high access to electricity.
- Off-diagonal Scatter Plots: Each off-diagonal plot shows the relationship between two different variables. For instance, the scatter plot for 'Access to electricity (% of population)' vs.

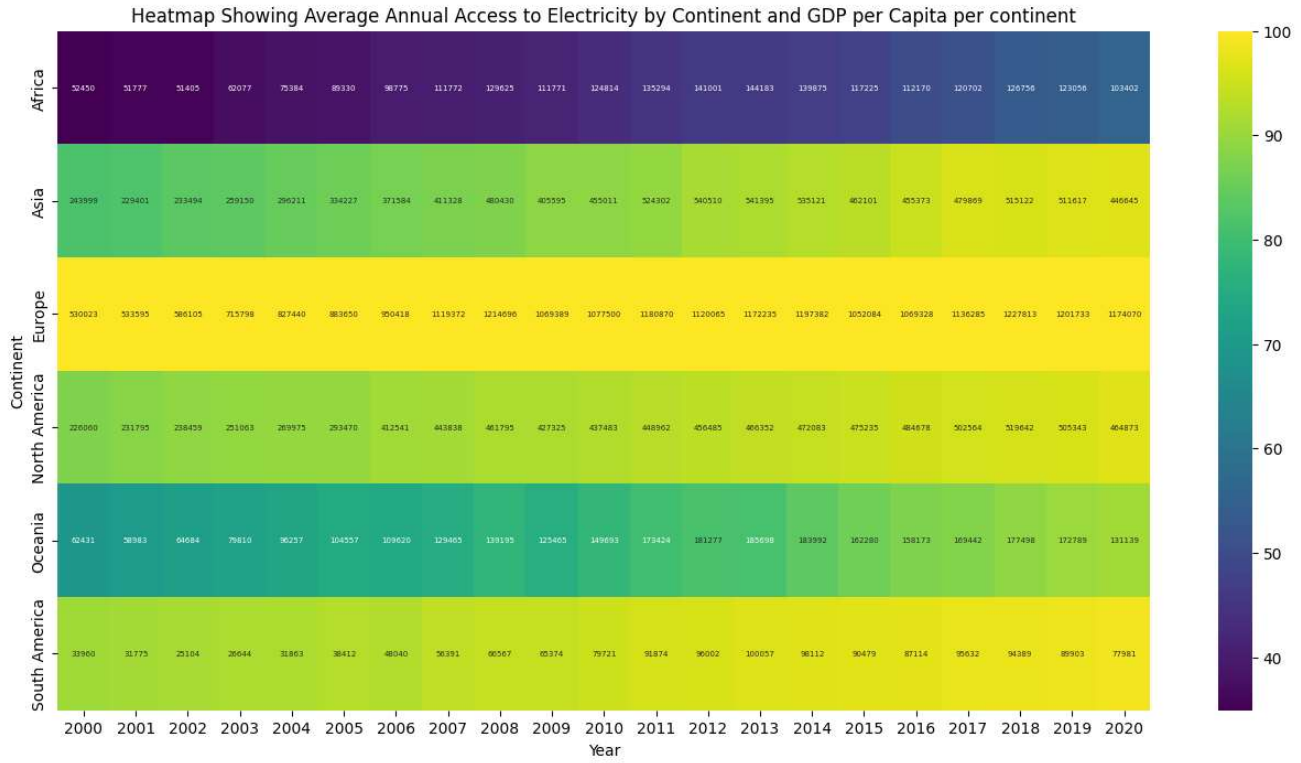
'gdp_per_capita' indicates that higher GDP per capita corresponds with better access to electricity.

- Trends and Correlations: We also notice varying degrees of correlation between different pairs of variables in the dataset. Some variables like GDP per capita and primary energy consumption per capita seem to have a positive correlation while others are more dispersed.
- Density of Points: The concentration of points within a scatter plot indicates the commonality of certain variable combinations. For example, many points are bunched together at high values of GDP and high access to electricity suggests that these two factors often occur together.

The next heatmap shows the average annual access to electricity by continent and their respective GDP per capita. It effectively illustrates the relationship between electricity access and the economic strength of the region, highlighting disparities and progress over time.

In [ ]:
```
!pip install pycountry_convert --quiet
```

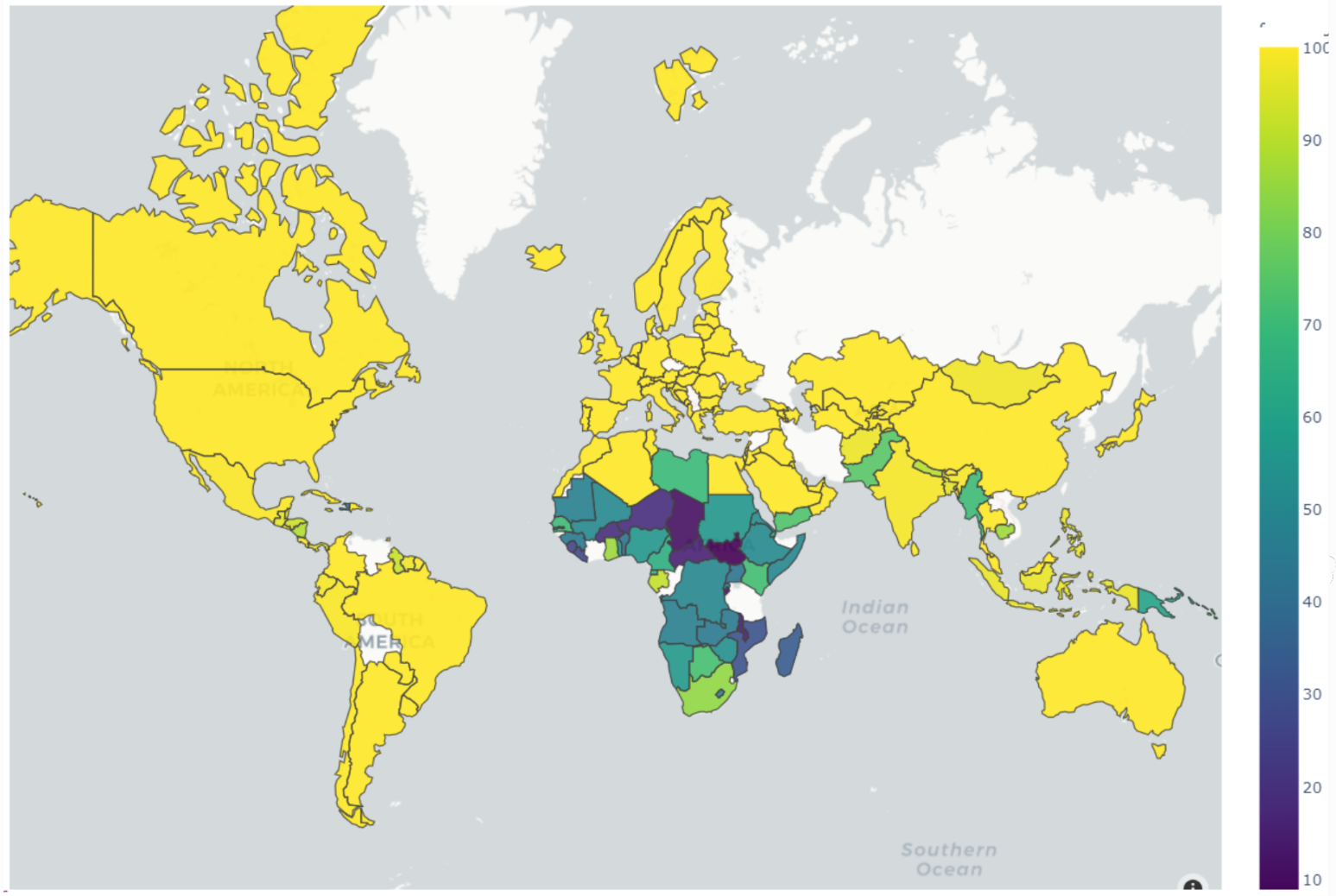
In [ ]:
```
import pycountry_convert as pc

def country_to_continent(country_name):
    try:
        country_alpha2 = pc.country_name_to_country_alpha2(country_name)
        country_continent_code = pc.country_alpha2_to_continent_code(country_alpha2)
        country_continent_name = pc.convert_continent_code_to_continent_name(country_cont
        return country_continent_name
    except:
        return 'Unknown'  # If the country name is not found

df['Continent'] = df['Entity'].apply(country_to_continent)      # Map the 'Entity' columr
continent_year_gdp_sum = df.groupby(['Continent', 'Year'])['gdp_per_capita'].sum().reset_
df_merged = df.merge(continent_year_gdp_sum, on=['Continent', 'Year'], suffixes=('', '_su

pivot_electricity = df_merged.pivot_table(index='Continent', columns='Year', values='Acce

pivot_gdp = df_merged.pivot_table(index='Continent', columns='Year', values='gdp_per_capi

plt.figure(figsize=(16, 8))
sns.heatmap(pivot_electricity, cmap='viridis', annot=pivot_gdp, fmt='.0f',annot_kws={"siz
plt.title('Heatmap Showing Average Annual Access to Electricity by Continent and GDP per
plt.xlabel('Year')
plt.ylabel('Continent')
plt.show()
```



We observe that continents like Europe and North America with higher GDP per Capita have consistently high access to electricity. However, Africa with lower GDP per Capita, has lower access to electricity compared to other continents.

Map showing the percentage of the population with Access to electricity by country.

In [ ]:
```
df_for_map =  original_df[original_df['Year'] == 2020]
df_for_map['Entity'].replace({'United States':'United States of America',
                              'Congo':'Democratic Republic of the Congo'}, inplace=True)

df_for_map = df_for_map[['Entity','Access to electricity (% of population)']]
fig = px.choropleth_mapbox(df_for_map,
                           geojson="https://raw.githubusercontent.com/ddlondono/DATA606/
                           locations='Entity',
                           featureidkey="properties.name",
                           color='Access to electricity (% of population)',
                           color_continuous_scale="Viridis",
                           mapbox_style="carto-positron",
                           zoom=1,
                           opacity=0.9,
                           center={"lat": 30, "lon": 0},
                           labels={'Access to electricity (% of population)': '% of popu
                          )
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0},height=700)
fig.show()
```


Indian
Ocean
Southern
Ocean
100
90
80
70
60
50
40
30
20
10

The map above shows that as per 2020, some countries still had limited access to electricity. The model proposed in this project will serve to answer the following questions:

- With the current data, can we predict if at leats 99% of the country's population has access to electricity?
- What is the accuracy of the prediction made with the model proposed?
- What are the best predictors in the model?

6. Formal analysis

In this section, we will be proposing several model for predicting "Access To Electricity" as a percentage of population. We will compare Regression and classification methods.

6.1. Logistic Regression

In [ ]:
```
data= original_df.copy()
knn_imputer = KNNImputer(n_neighbors=5)
data = pd.DataFrame(knn_imputer.fit_transform(data.drop('Entity',axis=1)),columns = knn_i
data['Access_encoded'] = (data['Access to electricity (% of population)'] > 99).fillna(0)
sgnf_data = data.drop(['Financial flows to developing countries (US $)'], axis=1)

cut_off_year = 2016  #(80% data for training)

train_data, test_data   = sgnf_data[sgnf_data['Year'] < cut_off_year] , sgnf_data[sgnf_da

X_train, y_train = train_data.drop(['Access to electricity (% of population)','Access_enc
X_test, y_test = test_data.drop(['Access to electricity (% of population)','Access_encode
lr_model = LogisticRegression(max_iter=1000)
lr_model.fit(X_train, y_train)                              #Fit the model on the training
y_pred = lr_model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.9062857142857143
```

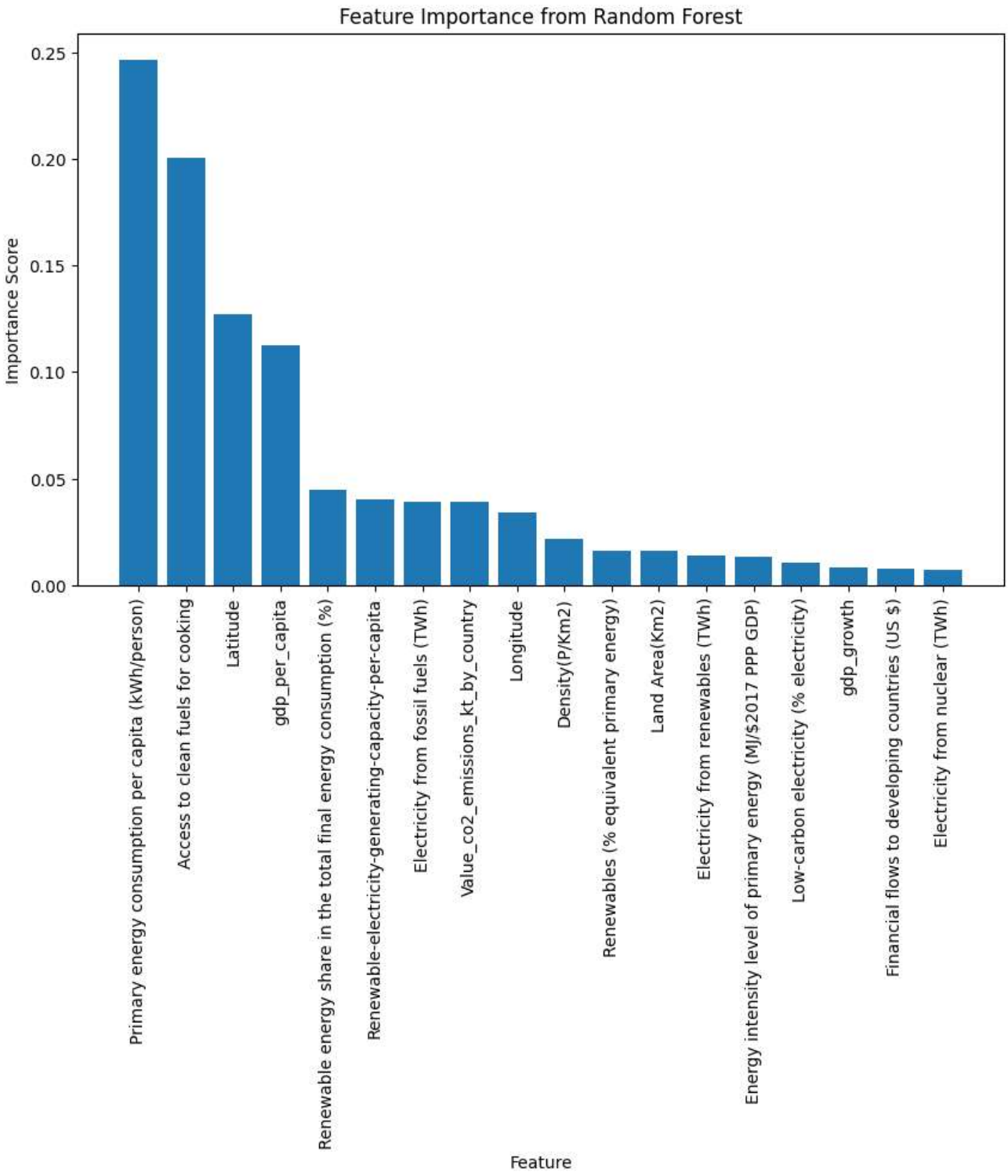
The accuracy obtained uding Logistic regression is $\approx 90.5\%$.

6.2. Random Forest Classifier

Now we use a classifier method to include possible no linearities. We will see that the most important feature in this case it is not necesarily the variable that showed the highest correlation in the preliminary analysis. Initially this seems as a discrepancy, however, we need to remember that the correlation coeficients only measure linear relationships, while level of importance using classification methods can capture non-linear relationships.

In [ ]:

```
from sklearn.ensemble import RandomForestClassifier
import matplotlib.pyplot as plt

train_data, test_data   = data[data['Year'] < cut_off_year] , data[data['Year'] >= cut_of
X_train, y_train = train_data.drop(['Year', 'Access to electricity (% of population)','Ac
X_test, y_test = test_data.drop(['Year', 'Access to electricity (% of population)','Acces

rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)       # Initial
rf_classifier.fit(X_train, y_train)                                             # Fit the

feature_importance = rf_classifier.feature_importances_                         # To extr
sorted_indices = feature_importance.argsort()[::-1]                             # To sort
sorted_features = [X_train.columns[i] for i in sorted_indices]
sorted_importance = feature_importance[sorted_indices]

plt.figure(figsize=(10, 6))
plt.bar(range(len(sorted_features)), sorted_importance, align='center')
plt.xticks(range(len(sorted_features)), sorted_features, rotation=90)
plt.xlabel('Feature')
plt.ylabel('Importance Score')
plt.title('Feature Importance from Random Forest')
plt.show()
```



Now we can predict using the model fit using the classification tree model.

In [ ]:
```
y_pred = rf_classifier.predict(X_test)                    # Predict on the test set
print("Accuracy:", accuracy_score(y_test, y_pred))        # Evaluate the model
```

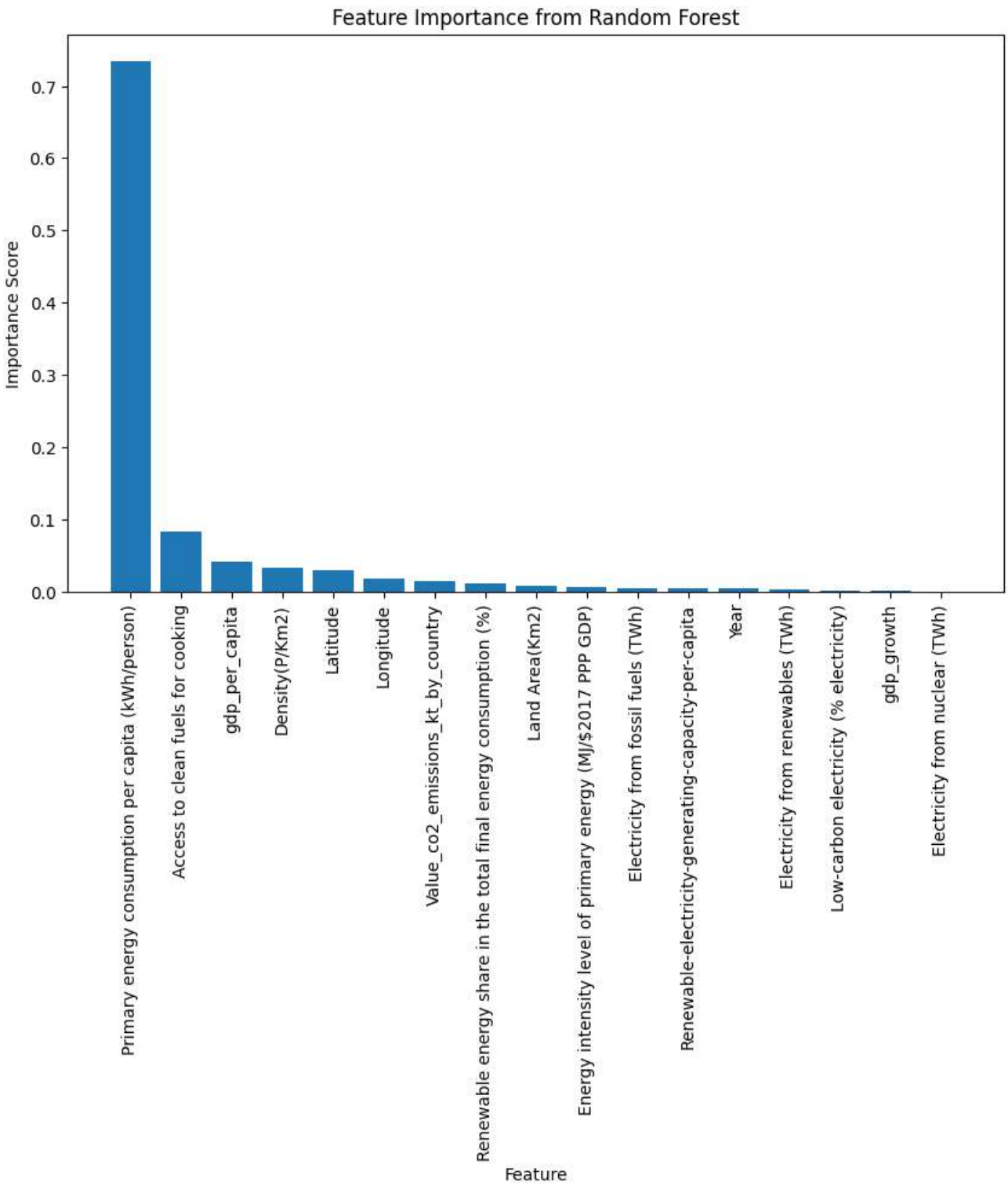
```
Accuracy: 0.9211428571428572
```

The model obtained using a classification tree method has a prediction accuracy that is approximately 92.1% that is around 2% more accurate than the predictions made using the Logistic regression model.

6.3. RandomForest Regression

Since the classification model's accuracy was not up to our expectation, we turn to regression models to discover if they work better for our dataset. In regression we don't need to encode our target variable into categories anymore. We will use the column "Access to Electricity" as our target variable.

In [ ]:
```
data= original_df.copy()
data.drop(["Financial flows to developing countries (US $)","Renewables (% equivalent pri

knn_imputer = KNNImputer(n_neighbors=5)
data = pd.DataFrame(knn_imputer.fit_transform(data.drop('Entity',axis=1)),columns = knn_i

train = data[data['Year'] <= cut_off_year]
test = data[data['Year'] > cut_off_year]
x_train = train.drop("Access to electricity (% of population)",axis=1)
y_train = train["Access to electricity (% of population)"]
x_test = test.drop("Access to electricity (% of population)",axis=1)
y_test = test["Access to electricity (% of population)"]

rf_model  = RandomForestRegressor(max_depth=10,random_state=42)
rf_model.fit(x_train,y_train)

feature_importance = rf_model.feature_importances_
sorted_indices = feature_importance.argsort()[::-1]
sorted_features = [x_train.columns[i] for i in sorted_indices]
sorted_importance = feature_importance[sorted_indices]

plt.figure(figsize=(10, 6))
plt.bar(range(len(sorted_features)), sorted_importance, align='center')
plt.xticks(range(len(sorted_features)), sorted_features, rotation=90)
plt.xlabel('Feature')
plt.ylabel('Importance Score')
plt.title('Feature Importance from Random Forest')
plt.show()
```



In [ ]:
```
pred = rf_model.predict(x_test)
r2 = r2_score(y_test,pred)
mse = mean_squared_error(y_test,pred)
mae = mean_absolute_error(y_test,pred)
print(f'Coefficient of determination: {r2:.3f}\nMean Squared Error MSE:{mse:.2f}\nMean Ab
```

```
Coefficient of determination: 0.950
Mean Squared Error MSE:31.96
Mean Absolute Error (MAE):3.078
```

In regression, the MSE measures the average difference between predicted and actual values. Smaller values of MSE are better and 0 means perfect prediction. In this particular case, the model's $MSE \approx 31.96$.

The coefficient of determination: R^2 measures the proportion of varience in dependent variables that are captured by the model. Our Random Forest Regressor yields a high R^2 of 95%.

6.4. Gradient Boosting Model

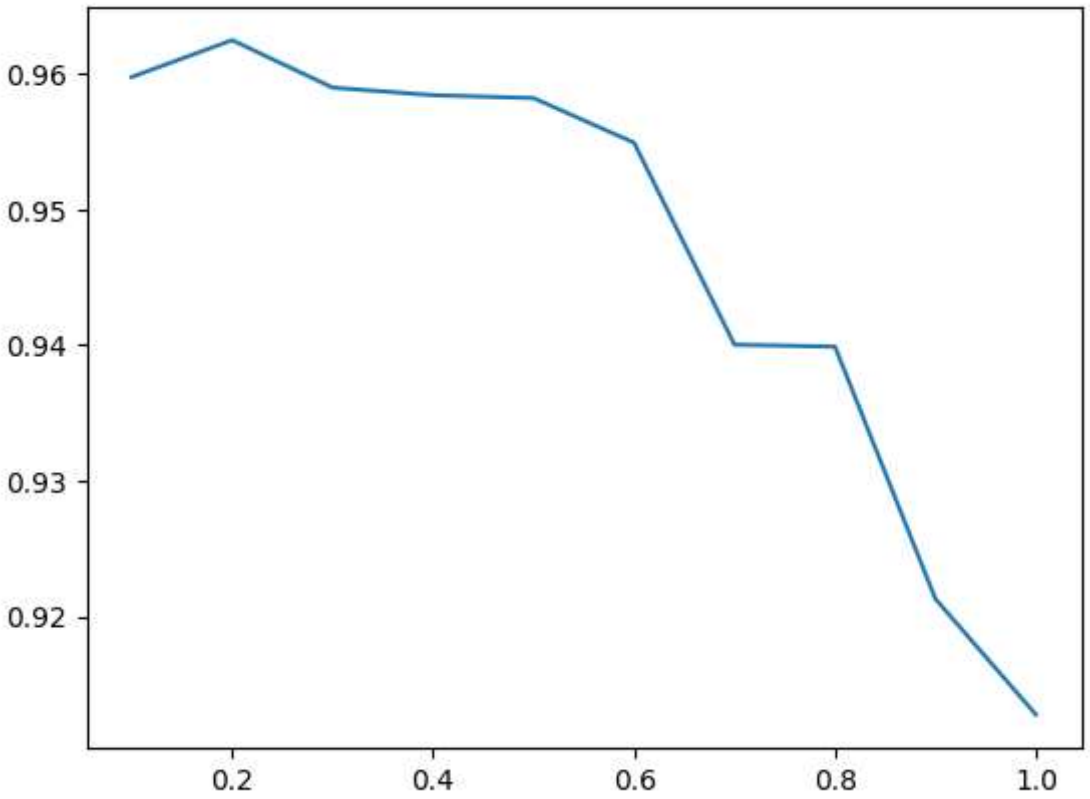
The Gradient Boosting is a tree based model that iterates new tree over the last one that minimizes the overall prediction error. In naive gradient boosting models, the model fit quickly and there's usually an over-fitting problem. To slow down the learning, learning rate is introductes as a weighting factor when new trees are added to the model. To find out the best learning rate for our model, we iterated 10 different learning rate from 0.1 - 10

In [ ]:
```
from sklearn.ensemble import HistGradientBoostingRegressor

learning_rate = []
r2 = []
mae = []

for learning in np.linspace(0.1,1,10):
    clf = HistGradientBoostingRegressor(learning_rate=learning,max_depth=10,random_state=
    clf.fit(x_train,y_train)
    pred = clf.predict(x_test)
    r2.append(r2_score(y_test,pred))
    mae.append(mean_absolute_error(y_test,pred))
    learning_rate.append(learning)
table = pd.DataFrame({"Learning_Rate":learning_rate,
                      "R2":r2,
                      "MAE":mae})
print(table)
plt.plot("Learning_Rate","R2",data = table)
```

```
  File "C:\Users\Todos\AppData\Roaming\Python\Python310\site-packages\joblib\externals\lo
ky\backend\context.py", line 282, in _count_physical_cores
    raise ValueError(f"found {cpu_count_physical} physical cores < 1")
```

```
   Learning_Rate        R2       MAE
0            0.1  0.959760  2.770426
1            0.2  0.962478  2.653323
2            0.3  0.958981  2.819748
3            0.4  0.958421  2.763903
4            0.5  0.958210  2.903726
5            0.6  0.954920  3.065945
6            0.7  0.940033  3.408695
7            0.8  0.939889  3.411608
8            0.9  0.921292  3.805705
9            1.0  0.912770  4.011159
```

Out[ ]: [<matplotlib.lines.Line2D at 0x1b2d0390220>]



From our experiment we can see when learning rate is 0.2 the model gives the highest R^2. We will use $learningrate = 0.2$ to train our model.

In [ ]:
```
clf = HistGradientBoostingRegressor(learning_rate=0.2,max_depth=10)
clf.fit(x_train,y_train)
pred = clf.predict(x_test)
r2 = r2_score(y_test,pred)
mse = mean_squared_error(y_test,pred)
mae = mean_absolute_error(y_test,pred)
print(f'Coefficient of determination: {r2:.3f}\nMean Squared Error MSE:{mse:.2f}\nMean Ab
```

```
Coefficient of determination: 0.962
Mean Squared Error MSE:24.04
Mean Absolute Error (MAE):2.653
```

Here we are glad to see a slight improvement in R^2, MSE and MAE over our RandomForest Model.

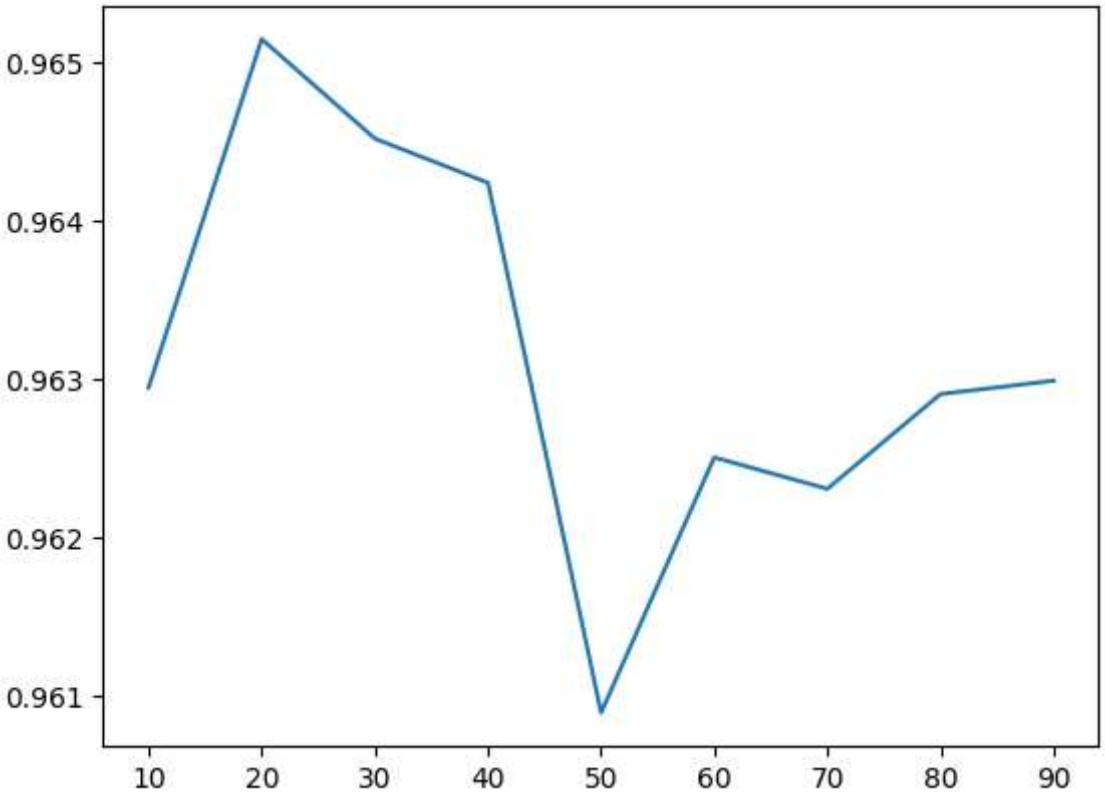
6.5. LightGBM Model

LightGBM is another tree baed model that utilizes gradient boosting. It carries out leaf-wise growth instead of level wise growth and is supposed to yield more loss reduction and higher accuracy. Again we will try to iterate over different num_leaves to find the best number to yield the best performance.

In [ ]:
```
from lightgbm import LGBMRegressor
nleaves = []
r2 = []
mae = []

for n in range(10,100,10):
    lgbm = LGBMRegressor(num_leaves=n, max_depth=20, n_estimators=300, n_jobs=-1,random_s
    lgbm.fit(x_train,y_train)
    pred = lgbm.predict(x_test)
    r2.append(r2_score(y_test,pred))
    mae.append(mean_absolute_error(y_test,pred))
    nleaves.append(n)
table = pd.DataFrame({"num_leaves":nleaves,
                      "R2":r2,
                      "MAE":mae})
print(table)
plt.plot("num_leaves","R2",data = table)
```

```
   num_leaves        R2       MAE
0          10  0.962947  2.699375
1          20  0.965147  2.605231
2          30  0.964519  2.590047
3          40  0.964240  2.553186
4          50  0.960892  2.597609
5          60  0.962505  2.524063
6          70  0.962307  2.567445
7          80  0.962905  2.562136
8          90  0.962989  2.542465
```

Out[ ]: `[<matplotlib.lines.Line2D at 0x1b2d046be80>]`



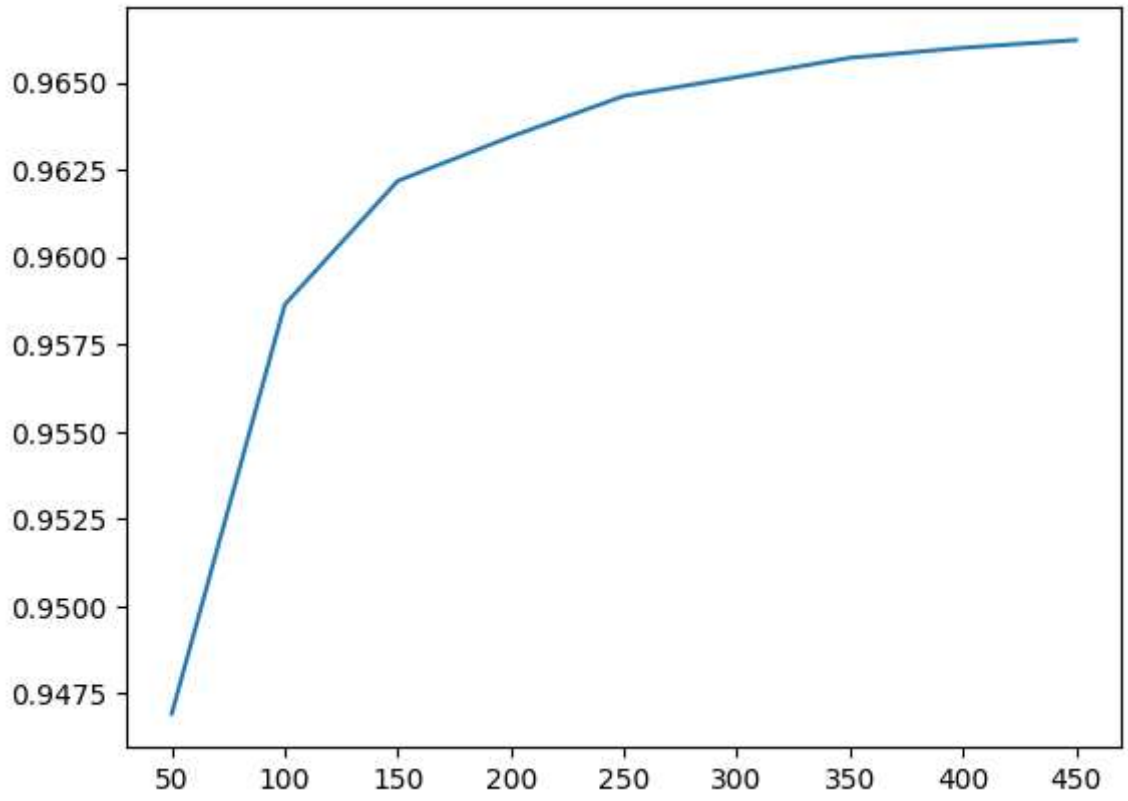
It looks like the 20 leaves gives the best result but the difference is minimal.

Yet again we will use the same procedure to determine the best number of trees.

In [ ]:
```
ntree = []
r2 = []
mae = []

for n in range(50,500,50):
    lgbm = LGBMRegressor(num_leaves=20, max_depth=20, n_estimators=n, n_jobs=-1,random_st
    lgbm.fit(x_train,y_train)
    pred = lgbm.predict(x_test)
    r2.append(r2_score(y_test,pred))
    mae.append(mean_absolute_error(y_test,pred))
    ntree.append(n)
table = pd.DataFrame({"n_estimators":ntree,
                      "R2":r2,
                      "MAE":mae})
print(table)
plt.plot("n_estimators","R2",data = table)
```

	n_estimators	R2	MAE
0	50	0.946926	3.382463
1	100	0.958637	2.877914
2	150	0.962180	2.733461
3	200	0.963445	2.687651
4	250	0.964609	2.636792
5	300	0.965147	2.605231
6	350	0.965702	2.584276
7	400	0.965989	2.572568
8	450	0.966210	2.569462

Out[ ]: [<matplotlib.lines.Line2D at 0x1b2d04ee620>]



Looks like dinimishing return occurs upward of 250 trees. We will use 250 trees to avoid potential overfitting.

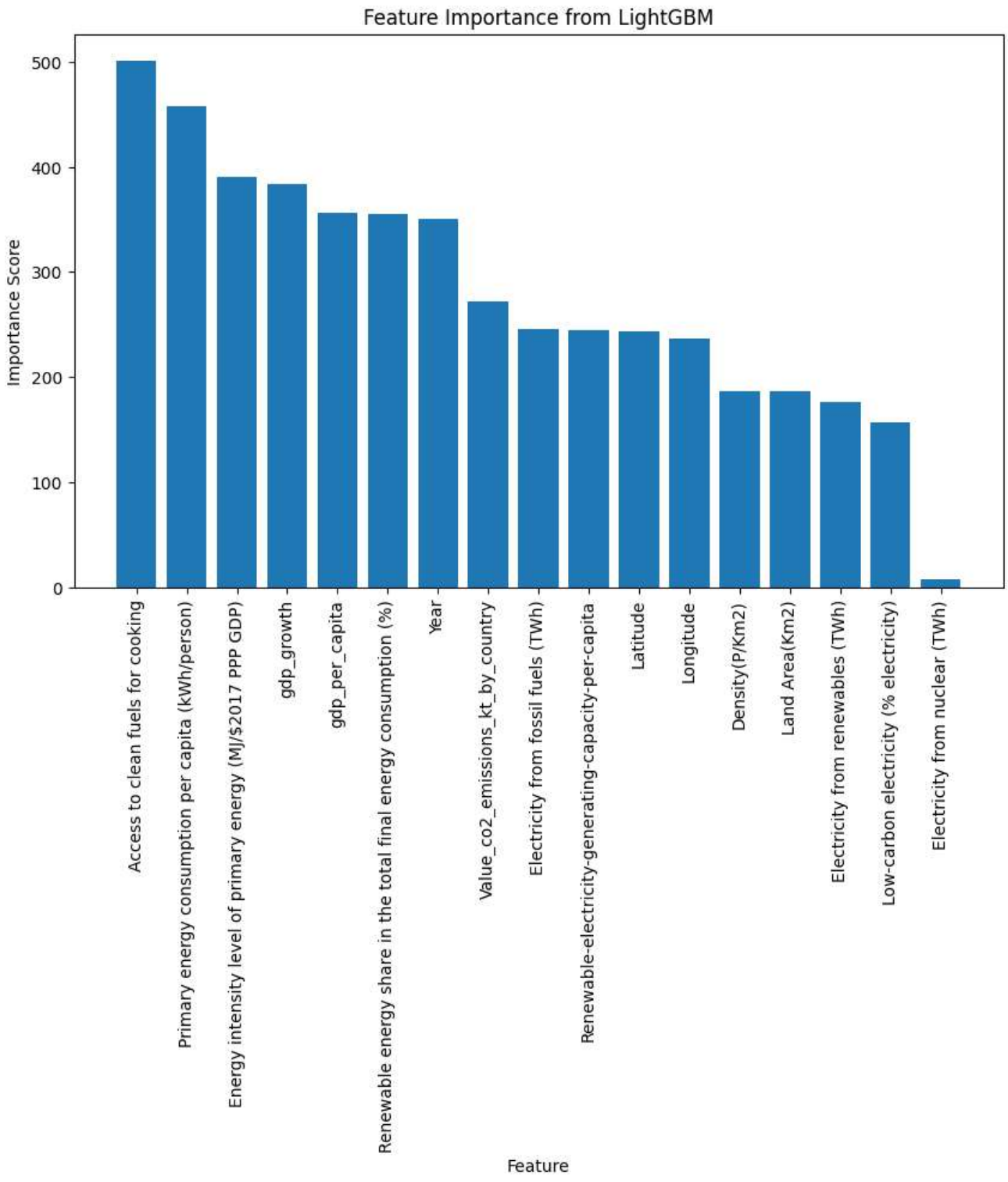
In [ ]:
```
model = LGBMRegressor(num_leaves=20,max_depth=20, n_estimators=250, n_jobs=-1,random_stat
model.fit(x_train,y_train)
pred = model.predict(x_test)
r2 = r2_score(y_test,pred)
mse = mean_squared_error(y_test,pred)
mae = mean_absolute_error(y_test,pred)
print(f'Coefficient of determination: {r2:.3f}\nMean Squared Error MSE:{mse:.2f}\nMean Ab
```

```
Coefficient of determination: 0.965
Mean Squared Error MSE:22.68
Mean Absolute Error (MAE):2.637
```

The performance improvement over gradient boosting model is neglectable considering the randomness.

Now we will look at the feature importance of the LightGBM Model.

In [ ]:
```
feature_importance = model.feature_importances_
sorted_indices = feature_importance.argsort()[::-1]
sorted_features = [x_train.columns[i] for i in sorted_indices]
sorted_importance = feature_importance[sorted_indices]

plt.figure(figsize=(10, 6))
plt.bar(range(len(sorted_features)), sorted_importance, align='center')
plt.xticks(range(len(sorted_features)), sorted_features, rotation=90)
plt.xlabel('Feature')
plt.ylabel('Importance Score')
plt.title('Feature Importance from LightGBM')
plt.show()
```



The number one feature is 'Access to clean fuels for cooking', which if you recall, is insignificant in OLS. The importance score is a lot more evenly distributed comparing to the Random Forest Model.

7. Conclussions

We explored five different models to predict the "Access to Electricity" as a percentage of the population:

- *Logistic Regression*. It was used to predict the categorical outcome (1 for countries whose more thatn 99% of its population has access to electricity or 0 othewise). Using this method, it can predict with accuracy of 90.5%
- *Random Forest Classifier*. It was used for the same purpose as the logistic regression method. With this method we achieved accuracy of 92.1% which was an improvement from the results obtained using Logistic regression.
- *Random Forest Regressor*: The model yields a R^2 of 95% and $MAE = 3.078$. The number one important feature is 'Primary energy consumption per capita" and far exceeds other features.
- *Gradient Boosting Regressor*: We iterated several models to find the best learning rate is 0.2. The model yields a R^2 of 96.2%, and $MAE = 2.653$. A slight improvement over Random Forest Regressor.
- *LightGBm Regressor* We iterated several models to find the best number of trees to be 250 and number of leaves to be 20. The model yields a R^2 of 96.5%, and $MAE = 2.637$. The number one important feature is 'Access to clean fuels for cooking' and the importance of features are more evenly distributed compared to Random Forest.

Note that we cannot apply cross-validation techniques since our dataset is a time-series. It doesn't make sense to predict the past using future data. Our training-testing split is one time and final.

It's interesting that the highest importance for our Forest classifier method is '*Primary energy consumption per capita*' while in the correlation matrix the correlation between this variable and the target variable is not the highest, only 0.45. Here's why: Random forest classifiers are capable of capturing complex non-linear relationships between features and the target variable. They can identify interactions and combinations of features that contribute significantly to the predictive power of the model. On the other hand, correlation coefficients measure linear relationships between variables. In this case the Primary energy consumption has a non-linear relationship with the target variable, which can be observed in the scatter plot.

8. References.

[1] Martin, “Energy,” United Nations Sustainable Development. Accessed: Feb. 02, 2024. [Online]. Available: https://www.un.org/sustainabledevelopment/energy/

[2] “World Energy Outlook 2022 – Analysis,” IEA. Accessed: Feb. 02, 2024. [Online]. Available: https://www.iea.org/reports/world-energy-outlook-2022

[3] “Global Data on Sustainable Energy (2000-2020).” Accessed: Feb. 02, 2024. [Online]. Available: https://www.kaggle.com/datasets/anshtanwar/global-data-on-sustainable-energy

[4] "A random forest classifier.". https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html