CO_2 EMMISSIONS PREDICTION MODEL

A LINEAR REGRESSION ANALYSIS

In this project, we'll dive into a dataset about fuel consumption and the CO_2 emissions of vehicles. Specifically, we seek to develop a linear regression model that predicts the CO_2 emissions of a vehicle based on its characteristics. By understanding this relationship, we can identify key factors that influence the emissions.

DATASET

This dataset appears to provide a comprehensive view of various vehicle characteristics and their associated CO2 emissions. Here's a brief overview of the dataset:

MODELYEAR	The year the vehicle was manufactured.
MAKE	The manufacturer or brand of the vehicle.
MODEL	The specific model of the vehicle.
VEHICLECLASS	The class or category of the vehicle.
ENGINESIZE	Size of the engine (likely in liters).
CYLINDERS	Number of cylinders in the engine.
TRANSMISSION	Type of transmission in the vehicle.
FUELTYPE	Type of fuel used by the vehicle.
FUELCONSUMPTION_CITY	Fuel consumption in the city (likely in liters per 100 km).
FUELCONSUMPTION_HWY	Fuel consumption on the highway (likely in liters per 100 km).
FUELCONSUMPTION_COMB	Combined fuel consumption (likely in liters per 100 km).
FUELCONSUMPTION_COMB_MPG	Combined fuel consumption in miles per gallon.
CO2EMISSIONS	CO2 emissions (likely in grams per km).

Let's load the dataset and take a look at the first few rows to understand its structure and content.

```python
# Import Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Load Dataset
data = pd.read_csv('FuelConsumptionCo2.csv')

# show first five rows
data.head(5)
```

The dataset comprises 13 columns and 1067 rows which are about detailed specifications, fuel consumptions and CO2 emissions statistics of various vehicles.

DATA CLEANING

Before diving into any data analysis or modeling, it's essential to ensure that the data is clean and free of any errors, inconsistencies, or missing values. This is because the quality of the input data directly influences the quality of the output or predictions from any model. Let's begin by checking for any missing values in our dataset.

```python
# Check for missing values in the dataset
missing_values = data.isnull().sum()
missing_values
```

```
MODELYEAR                  0
MAKE                       0
MODEL                      0
VEHICLECLASS               0
```

ENGINESIZE	0
CYLINDERS	0
TRANSMISSION	0
FUELTYPE	0
FUELCONSUMPTION_CITY	0
FUELCONSUMPTION_HWY	0
FUELCONSUMPTION_COMB	0
FUELCONSUMPTION_COMB_MPG	0
CO2EMISSIONS	0

The dataset is free from missing values in all columns. This means we can proceed without the need for imputation or removal of missing data.

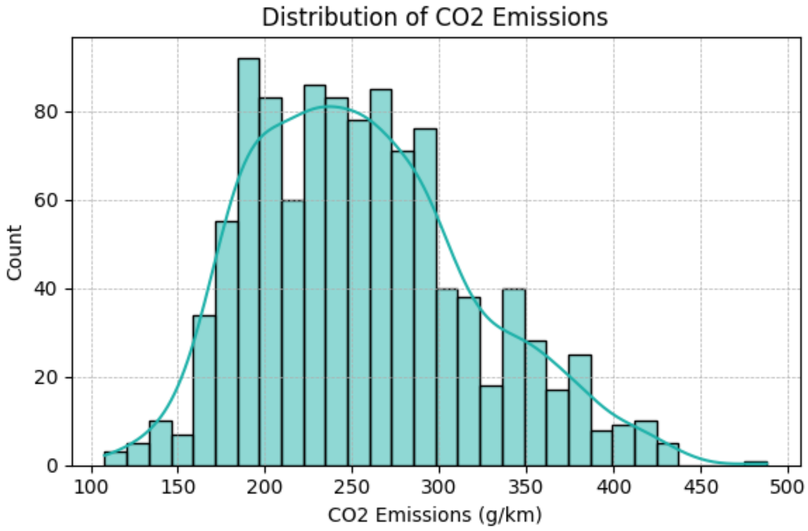
EXPLORATORY DATA ANALYSIS

Exploratory Data Analysis (EDA) is a crucial step in understanding the dataset. Through EDA, we can uncover patterns, relationships, anomalies, and more. Visualizations often form a significant part of EDA since they provide a clear and intuitive way to see these patterns and relationships. In this project we will focus on understanding the relationships between various vehicle features and CO2 emissions through EDA.

Distributions of CO2 Emissions:
To examine the distributions of CO2 Emissions in the dataset, we will create histogram. It is an estimate of the probability distribution of a continuous variable.

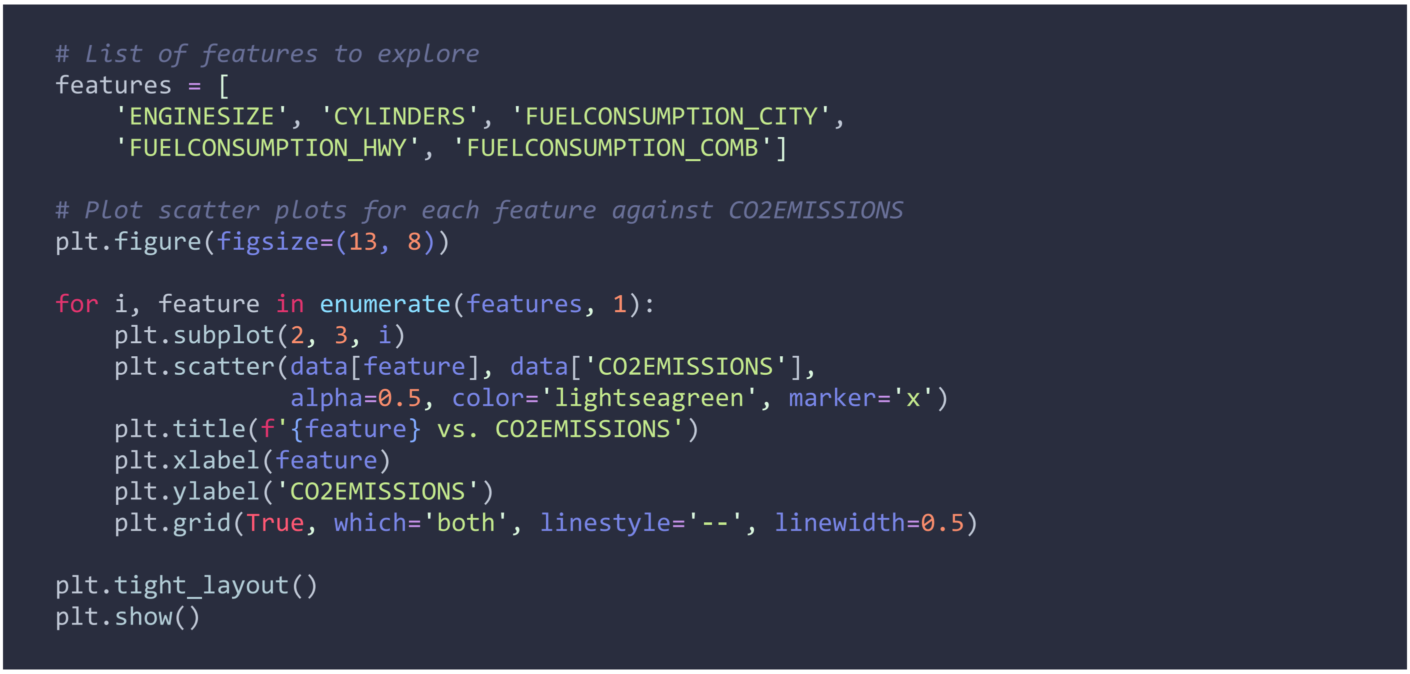
```
# Plot Distribution of CO2 Emissions

plt.figure(figsize=(6, 4))
sns.histplot(data['CO2EMISSIONS'], kde=True, bins=30, color='lightseagreen')
plt.grid(True, which='both', linestyle='--', linewidth=0.5)
plt.title('Distribution of CO2 Emissions')
plt.xlabel('CO2 Emissions (g/km)')
plt.tight_layout()
plt.show()
```



The histogram presents the distribution of CO2 emissions across all vehicles in the dataset. We observe that the majority of vehicles have CO2 emissions in the range of approximately 170-300 g/km.

Relationships Between Key Variables and CO2 Emissions:
We will focus on a few numerical features that might have a strong relationship with CO2 emissions. These features are 'ENGINESIZE', 'CYLINDERS', 'FUELCONSUMPTION_CITY', 'FUELCONSUMPTION_HWY', 'FUELCONSUMPTION_COMB'. Let's create scatter plots for each of the specified features against CO2EMISSIONS.

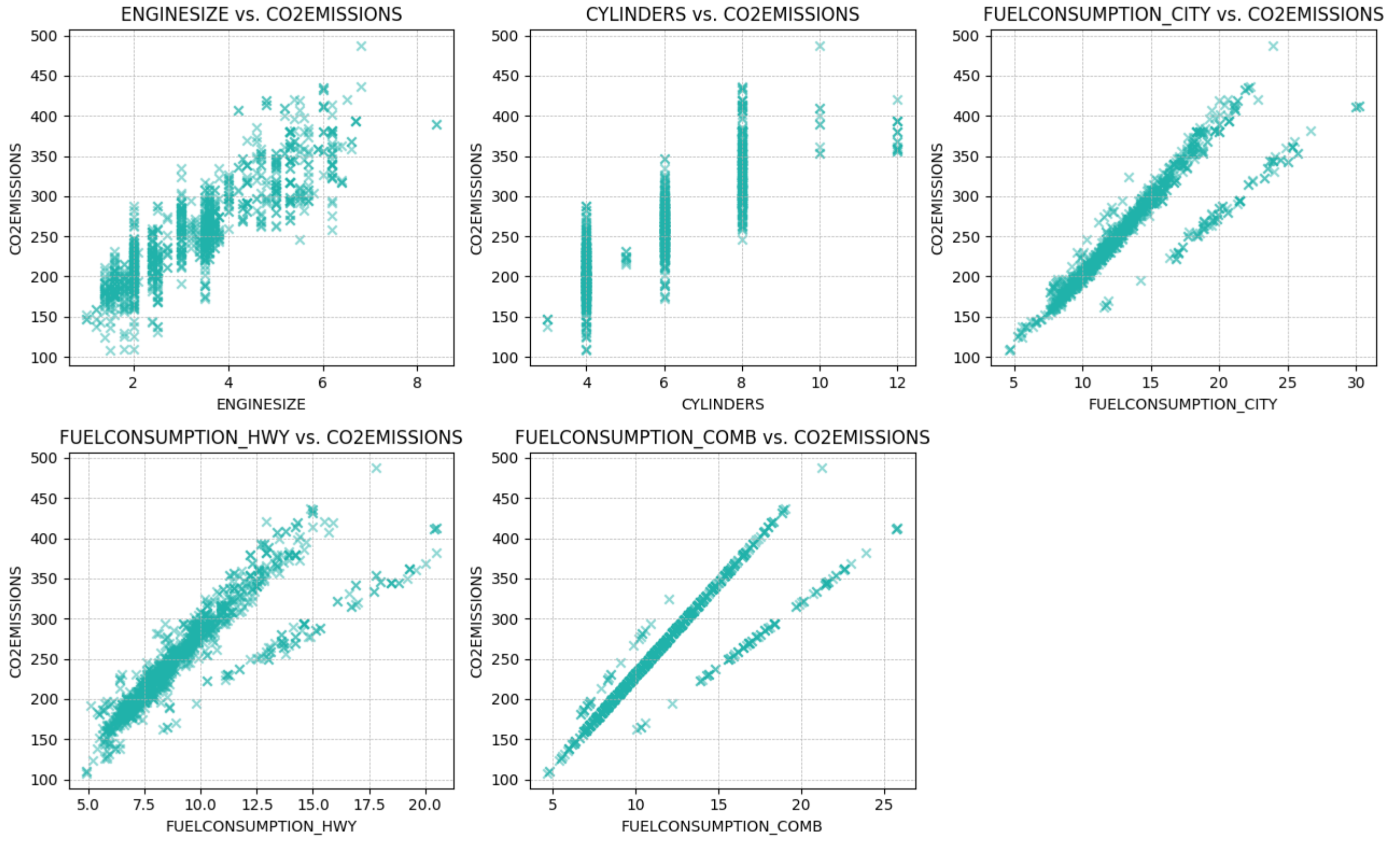


```python
# List of features to explore
features = [
    'ENGINESIZE', 'CYLINDERS', 'FUELCONSUMPTION_CITY',
    'FUELCONSUMPTION_HWY', 'FUELCONSUMPTION_COMB']

# Plot scatter plots for each feature against CO2EMISSIONS
plt.figure(figsize=(13, 8))

for i, feature in enumerate(features, 1):
    plt.subplot(2, 3, i)
    plt.scatter(data[feature], data['CO2EMISSIONS'],
                alpha=0.5, color='lightseagreen', marker='x')
    plt.title(f'{feature} vs. CO2EMISSIONS')
    plt.xlabel(feature)
    plt.ylabel('CO2EMISSIONS')
    plt.grid(True, which='both', linestyle='--', linewidth=0.5)

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



These scatter plots provide a visual representation of how various vehicle features relate to CO2 emissions:

- Engine Size vs. CO2 Emissions: There seems to be a positive correlation between engine size and CO2 emissions. As the engine size increases, CO2 emissions also tend to increase. This relationship is expected, as larger engines often consume more fuel, leading to higher emissions.
- Cylinders vs. CO2 Emissions: While there is some variability, a general trend indicates that vehicles with a higher number of cylinders tend to have higher CO2 emissions. This is consistent with the fact that vehicles with more cylinders often have larger engines and, therefore, might consume more fuel.
- Fuel Consumption (City) vs. CO2 Emissions: City fuel consumption presents a clear positive relationship with CO2 emissions.
- Fuel Consumption (Highway) vs. CO2 Emissions: Similarly, highway fuel consumption also shows a positive correlation with CO2 emissions.

- Combined Fuel Consumption vs. CO2 Emissions: As expected, there's a positive relationship between combined fuel consumption and CO2 emissions. Vehicles that consume more fuel (higher L/100km) tend to emit more CO2.

LINEAR REGRESSION MODEL

Linear regression is a statistical method used to model and analyze the relationships between a dependent variable and one or more independent variables. In our context, we want to predict the CO2 emissions (dependent variable) based on various vehicle features (independent variables). The main goal of linear regression is to find the best fit straight line that accurately predict the output values within a range.

Feature Selection: The 'FUELCONSUMPTION_COMB' is the average of 'FUELCONSUMPTION_CITY' and 'FUELCONSUMPTION_HWY', making it a holistic measure of a vehicle's fuel consumption. We will focus on three features for our linear regression model. Let's select ENGINE_SIZE, CYLINDERS and FUELCONSUMPTION_COMB as independent variable (X) and 'CO2EMISSIONS' as dependent variable (y).

```
# Redefining the list of selected features and target variable
selected_features = ['ENGINESIZE', 'CYLINDERS', 'FUELCONSUMPTION_COMB']
X_selected = data[selected_features]
y = data['CO2EMISSIONS']
```

Spliting Dataset: We'll divide our dataset into training and testing sets. The training set will be used to train our linear regression model, while the testing set will help us evaluate the model's performance.

```
from sklearn.model_selection import train_test_split

# Split the dataset into training and testing sets (80% train, 20% test)
X_train, X_test, y_train, y_test = train_test_split(
    X_selected, y,
    test_size=0.2, random_state=42)
```

Creating and Fitting the Model: Using the training set, we'll train our linear regression model. This involves the model learning the relationships between our selected features and CO2 emissions.

```
from sklearn.linear_model import LinearRegression

# Initialize the linear regression model
lr_model_selected = LinearRegression()

# Fit the model to the training data
lr_model_selected.fit(X_train, y_train)
```

Prediction and Evaluation: Once trained, we'll use the model to predict CO2 emissions on the testing set. We'll then compare these predictions to the actual emissions to evaluate the model's performance. Key metrics include the Mean Squared Error (MSE) and R^2 value.

```
from sklearn.metrics import mean_squared_error, r2_score

# Predict the CO2 emissions on the test set
y_pred_selected = lr_model_selected.predict(X_test)

# Calculate the model's performance metrics for the selected features
```

```
    mse_selected = mean_squared_error(y_test, y_pred_selected)
    r2_selected = r2_score(y_test, y_pred_selected)

    mse_selected, r2_selected
```

Result: (512.8551370148301, 0.875970520691407)

Mean Squared Error (MSE): For our refined model, the MSE is approximately 512.86, this value represents the average squared difference between the observed actual outturn values and the values predicted by the model. A lower MSE indicates a better fit to the data.

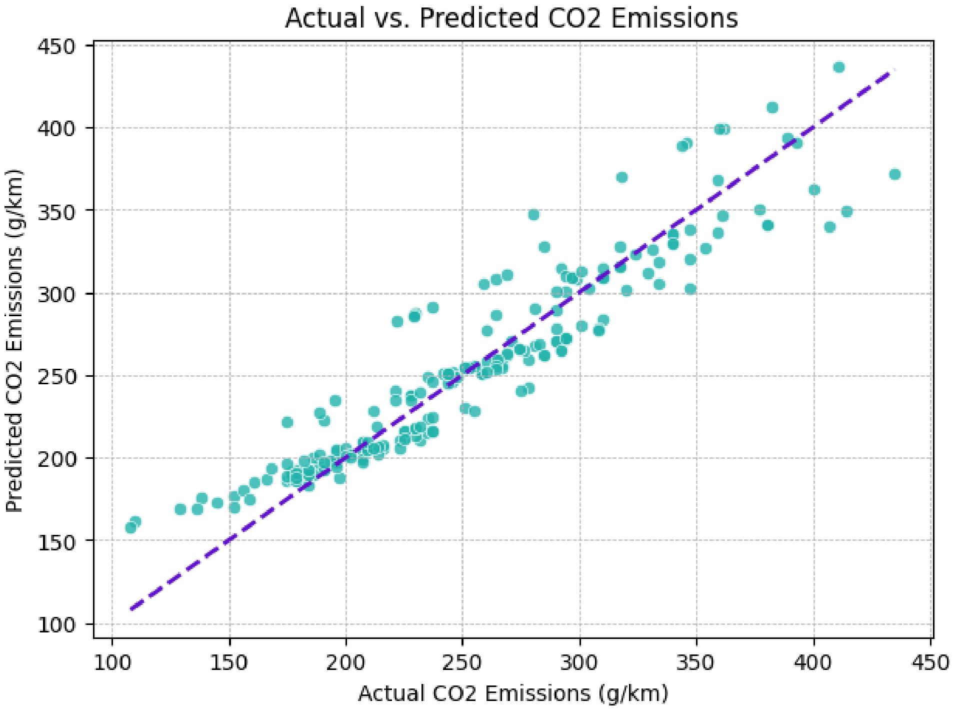
R-squared (R^2): The R^2 value is approximately 0.8760, suggesting that our model explains about 87.60% of the variance in CO2 emissions based on the three selected features. An R^2 value closer to 1 indicates that the model has a better fit to the observed data.

In summary, by focusing on the three features (ENGINESIZE, CYLINDERS, and FUELCONSUMPTION_COMB), we've built a model that provides a comprehensive understanding of vehicles' CO$_2$ emissions without overcomplicating the model. The performance metrics indicate that the model is quite accurate, explaining a significant portion of the variance in CO$_2$ emissions with these selected features.

VISUALIZING THE PREDICTION

Visualizing the actual values against the predicted values in a scatter plot provides an understanding of the model's accuracy. Let's visualize the model's prediction with a scatter plot for the actual CO$_2$ emissions vs. the predicted CO$_2$ emissions to get a better understanding of its performance.

```
    # Plotting the actual vs. predicted values
    plt.figure(figsize=(7, 4))
    sns.scatterplot(x=y_test, y=y_pred_selected, alpha=0.8, color='lightseagreen')
    plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--', lw=2,
color='#5F14CB')  # Diagonal line
    plt.xlabel('Actual CO2 Emissions (g/km)')
    plt.ylabel('Predicted CO2 Emissions (g/km)')
    plt.title('Actual vs. Predicted CO2 Emissions')
    plt.grid(True, which='both', linestyle='--', linewidth=0.5)
    plt.show()
```



The scatter plot illustrates the actual CO_2 emissions (x-axis) against the predicted CO_2 emissions (y-axis) for our linear regression model based on the selected features. Here are some observations from the plot:

- Diagonal Line: The dashed diagonal line represents where the predicted values would lie if they were perfectly accurate. Points close to this line indicate accurate predictions by the model.
- Distribution of Points: Most of the data points are clustered around the diagonal line, indicating that our model's predictions are generally accurate and closely match the actual values.
- Variability: While many predictions are close to the diagonal, there's some variability, especially for vehicles with higher CO2 emissions. This suggests areas where the model might benefit from further refinement or additional features.

Our linear regression model serves as an instrumental tool for comprehending and forecasting a vehicle's CO_2 emissions, leveraging a select set of significant features. With an R^2 value of 87.60%, the model exemplifies commendable accuracy using the current features.