Predicting Vehicle CO_2 Emissions Using Machine Learning

Anjali Ashok Joshi

The Problem Statement

Road transportation is a major contributor to rising CO_2 emissions, impacting climate change and air quality.
Vehicle emissions depend on multiple factors such as engine size, cylinders, fuel type, transmission, and fuel consumption.
Accurately predicting CO_2 emissions is challenging due to the mix of numerical and categorical features.

ML Solution Goal

This project builds a machine learning model and Streamlit app to estimate vehicle CO_2 emissions for better environmental awareness.

Project Objective

1

Data Analysis

The dataset was explored using statistical summaries and visualizations to understand distributions, correlations, and outliers.
This analysis revealed key factors influencing CO_2 emissions, such as engine size, fuel consumption, and vehicle class.

2

Exploratory Data Analysis

Exploratory Data Analysis was performed to understand data distribution, patterns, and relationships between features.
It helped identify outliers, skewness, and correlations, providing insights for preprocessing and model selection.

3

Data Preprocessing

Data preprocessing involved handling missing values, removing duplicates, and treating outliers. Categorical features were encoded, and numerical features were scaled to prepare data for modeling.

1

Regression Techniques

To implement and compare different **regression techniques** such as Linear Regression, Ridge, Lasso, ElasticNet, KNN, and Random Forest.

The aim is to identify the most effective model for accurately predicting **vehicle CO_2 emissions**.

2

Model Training & Comparison

Multiple regression models such as Linear Regression, Ridge, Lasso, ElasticNet, KNN, and Random Forest were trained.

Their performance was compared using metrics like R^2 and RMSE to select the best predictive model.

3

Insights & Recommendations

Insights
Fuel consumption and engine size show the strongest correlation with CO_2 emissions.
Categorical factors like vehicle class and fuel type also significantly influence emission levels.

Recommendations
Promote smaller engine vehicles and fuel-efficient designs to reduce emissions.
Encourage adoption of hybrid/electric vehicles and stricter emission standards.

Dataset Overview

Comprehensive vehicle database

7 k

Total Records

Categorical Features x

Categorical Features Make, Model, Vehicle Class, Transmission, Fuel Type

Numerical Features Engine

Engine Size, Cylinders, Fuel Consumption (City, Highway, Combined, MPG) Target Variable

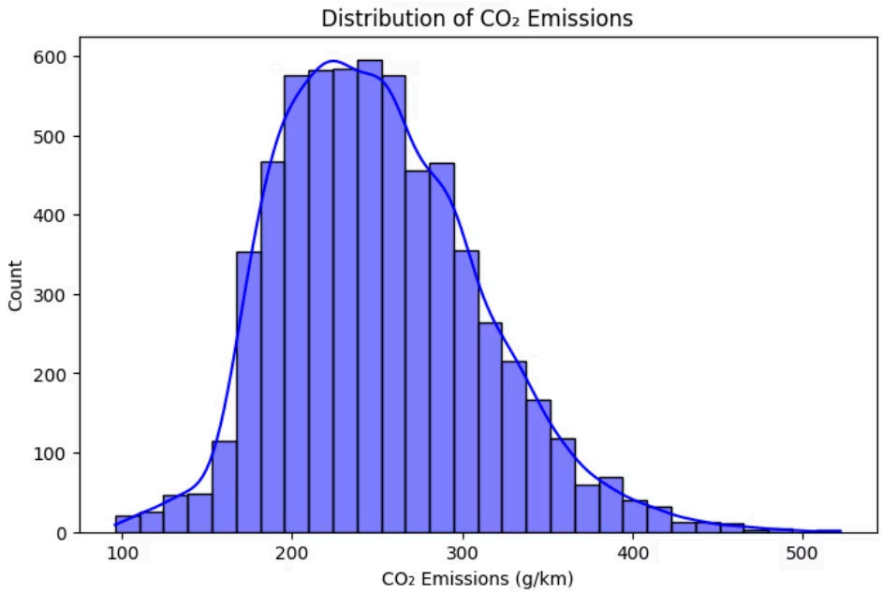
Target Variable

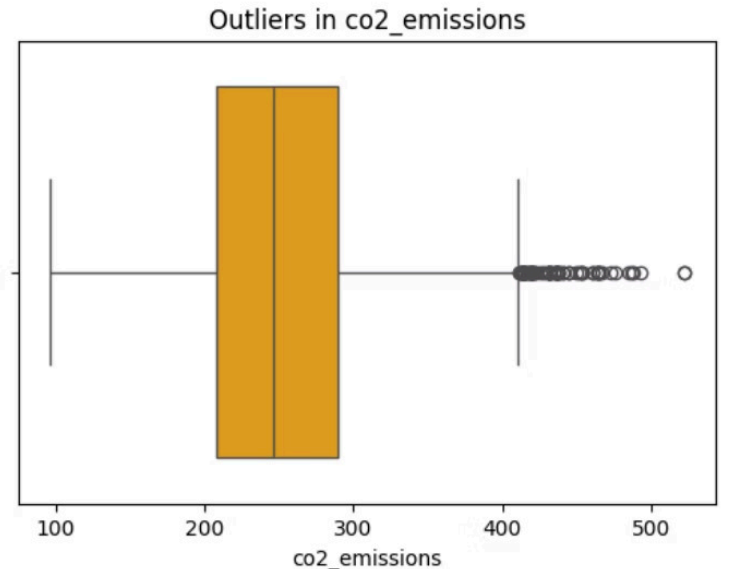
CO_2 Emissions measured in grams per kilometer (g/

Exploratory Data Analysis of Vehicle Emissions Data

A comprehensive analysis of CO_2 emissions patterns across vehicle characteristics, fuel types, and engine specifications for predictive modeling applications.

Target Variable Distribution Analysis





Histogram + KDE (Distribution of CO_2 Emissions) Shows the overall distribution, central tendency, and spread.

Boxplot (Outliers in CO_2 Emissions) Highlights presence of extreme values in the target variable.

Categorical Features Summary Statistics

	make	model	vehicle_class	transmission	fuel_type
count	6273	6273	6273	6273	6273
unique	42	2053	16	5	5
top	FORD	F-150 FFV	SUV - SMALL	AS	X
freq	575	32	1004	2720	3030

42

Vehicle Make

FORD leads with 575 vehicles in dataset

2053

Vehicle Model

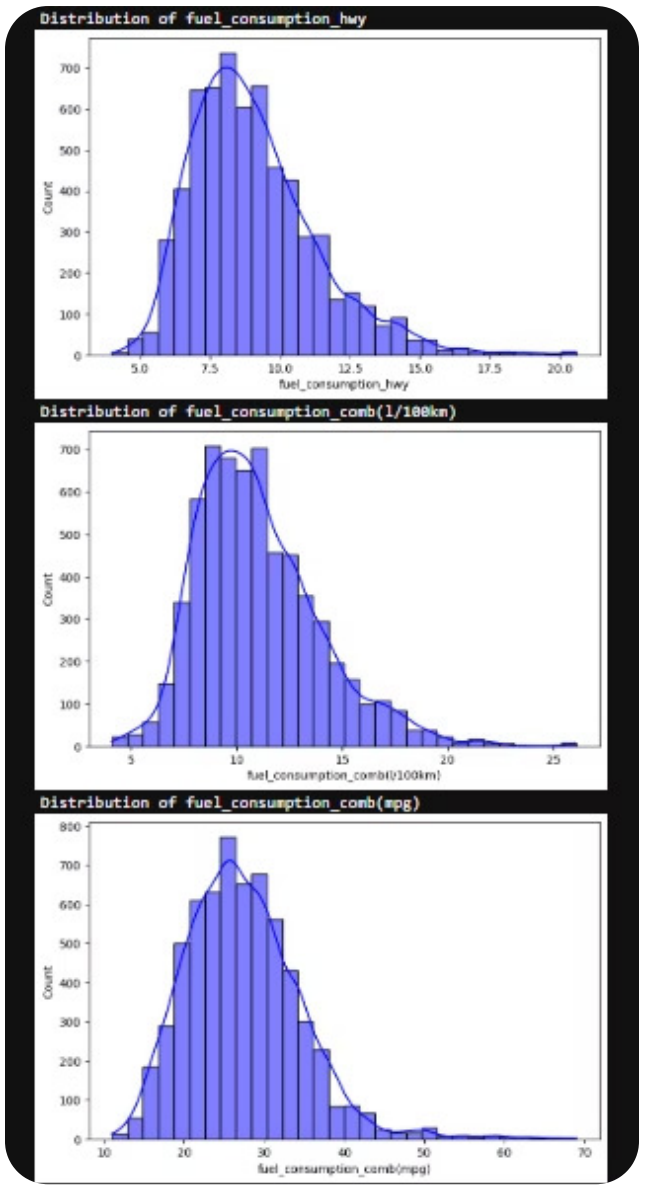
High cardinality F 150 FFV 4X4 tops at only 32 occurrences

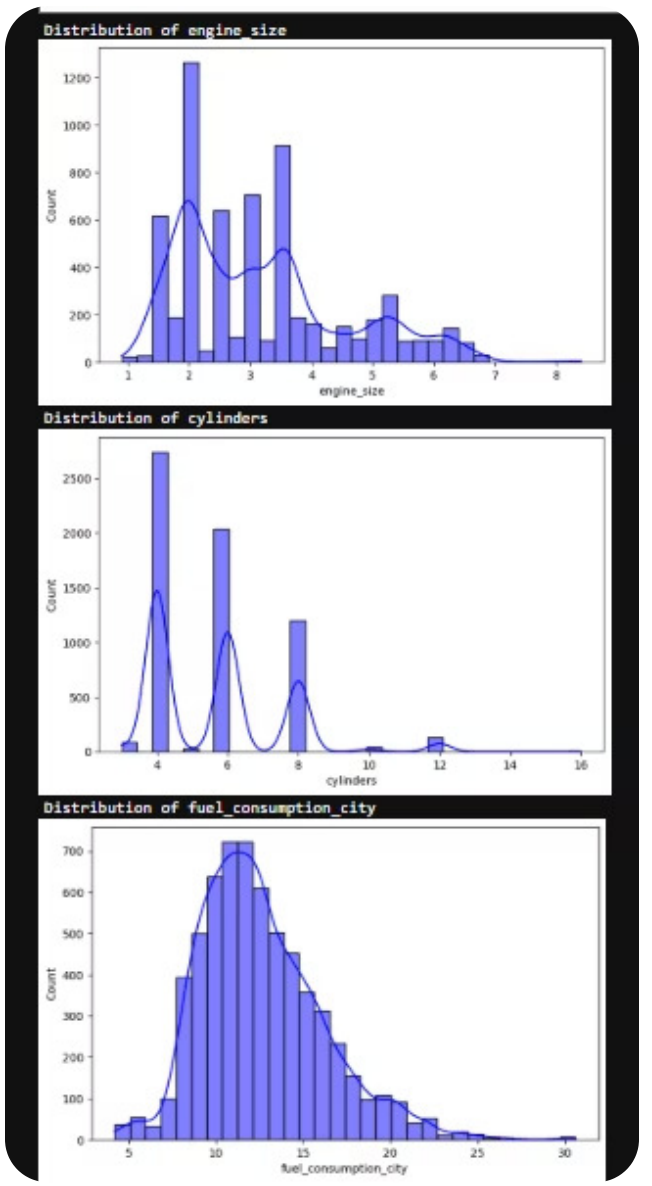
16

Vehicle Class

Dominant vehicle class with 1,004 vehicles

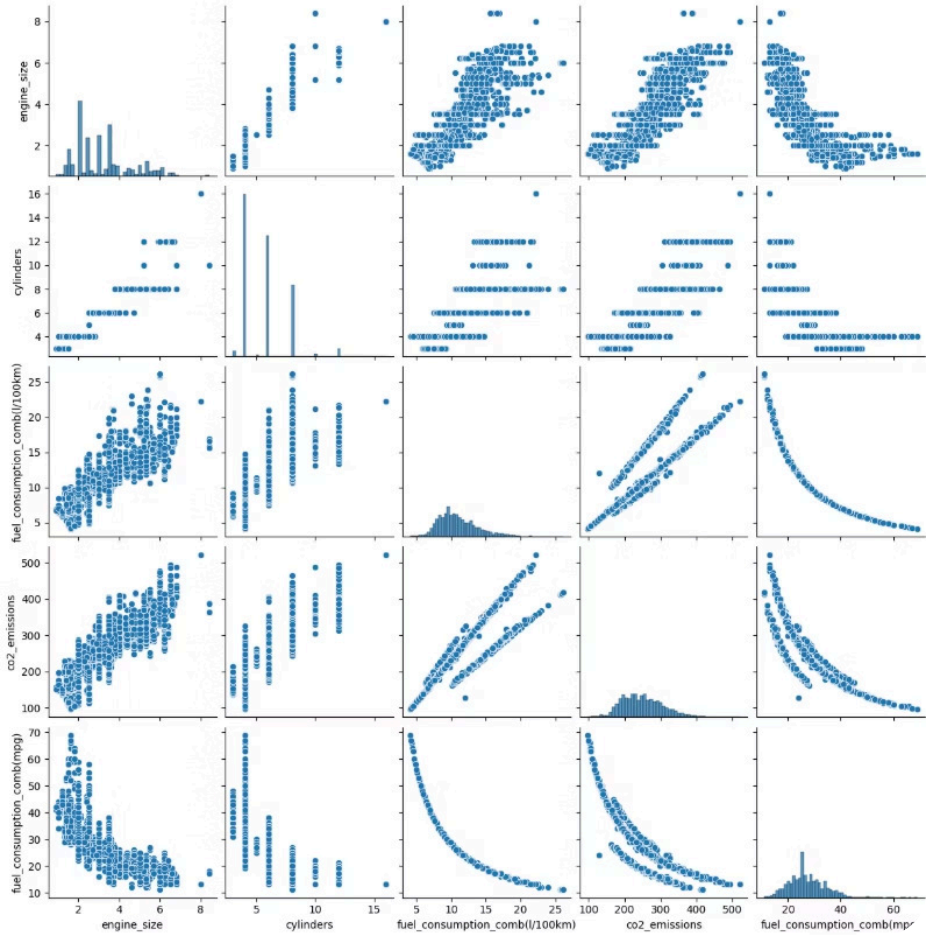
Numerical Features Distribution Analysis





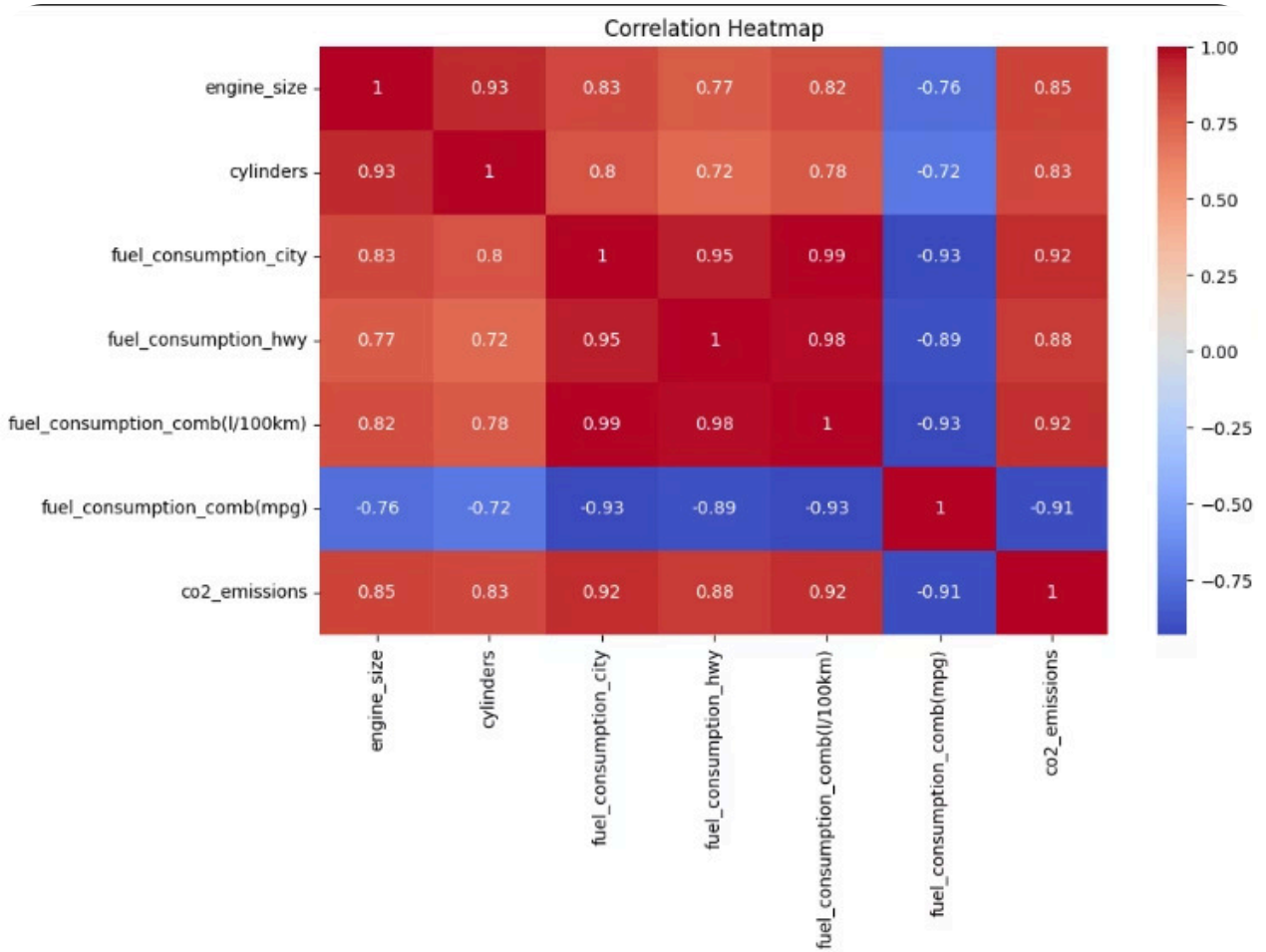
1. engine_size Distribution is right-skewed. Most cars have engine sizes between 2–4 liters, with very few above 6.
2. cylinders Clearly discrete (4, 6, 8 dominate). Peak at 4 cylinders, followed by 6 cylinders, fewer cars with 8+.
3. fuel_consumption_city Approximately normal-shaped but slightly right-skewed. Most cars consume 8–14 L/100km in the city.
4. fuel_consumption_hwy Also bell-shaped, slight right-skew. Most cars fall between 6–10 L/100km on highways.
5. fuel_consumption_comb(l/100km) Looks like a smoother mix of city & highway. Range mostly 8–13 L/100km, slight right skew. Much cleaner than analyzing city and hwy separately.
6. fuel_consumption_comb(mpg) The distribution looks close to normal but slightly right-skewed. Most cars fall in the 20–35 mpg range. A few efficient cars go above 40+ mpg, but those are rare outliers. This makes sense because high-MPG cars are usually hybrids/electric, so they’re less frequent.

Feature Relationships and Correlation Patterns



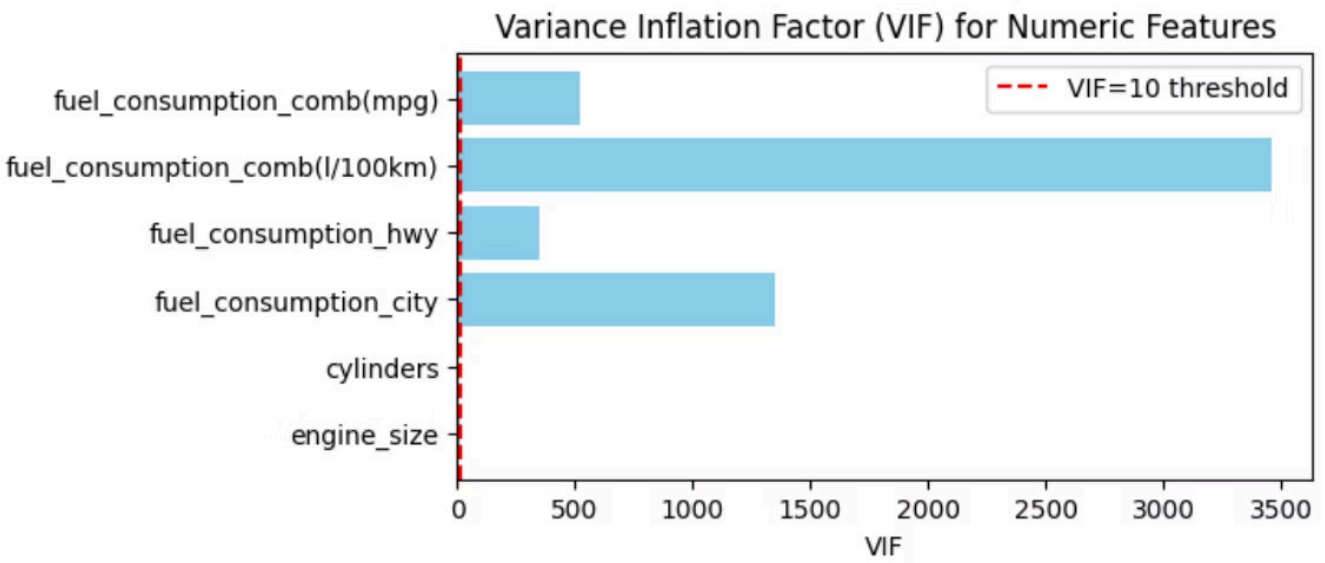
- engine_size vs co2_emissions → clear upward trend (larger engines → more emissions).
- cylinders vs co2_emissions → step-like relationship (discrete jumps since cylinders are categorical in nature, though numeric).
- fuel_consumption_comb vs co2_emissions → almost perfectly linear.
- engine_size vs cylinders → near-linear relationship (multicollinearity risk).

Correlation Heatmap: Multicollinearity Assessment



- fuel_consumption_comb (L/100km) and CO_2 emissions → very high correlation (0.92).
- engine_size and cylinders → both strongly correlated with CO_2 emissions (~0.85 and 0.84).
- fuel_consumption_city and fuel_consumption_hwy → also correlated with emissions but they overlap with fuel_consumption_comb.
- fuel_consumption_comb is essentially derived from city + highway → keeping fuel_consumption_comb alone is often enough.
- fuel_consumption_comb(mpg) vs co2_emissions → -0.91 (strong negative correlation) Since mpg is the inverse of l/100km, it's negatively correlated.

Variance Inflation Factor Analysis



- engine_size 9.131249
- cylinders 7.961396
- fuel_consumption_city 1351.020869
- fuel_consumption_hwy 347.518414
- fuel_consumption_comb(l/100km) 3459.684500
- fuel_consumption_comb(mpg) 519.832533

Data Preprocessing Steps Data

- **Data Cleaning**
- Removed duplicate records.
- Corrected inconsistent labels (e.g., “USA” vs “United States”).
- Fixed invalid values (negative ages, impossible emission levels).

- **Feature Engineering**
- Created new variables (e.g., **emissions per capita**, **BMI categories**, **yearly growth**).
- Encoded categorical features (Label Encoding / One-Hot Encoding).
- Extracted datetime features (year, month, day).

- **Outlier Treatment**
- Detected outliers using **IQR (Interquartile Range)** or **Z-score**.
- Capped extreme values or applied transformations (log/sqrt).

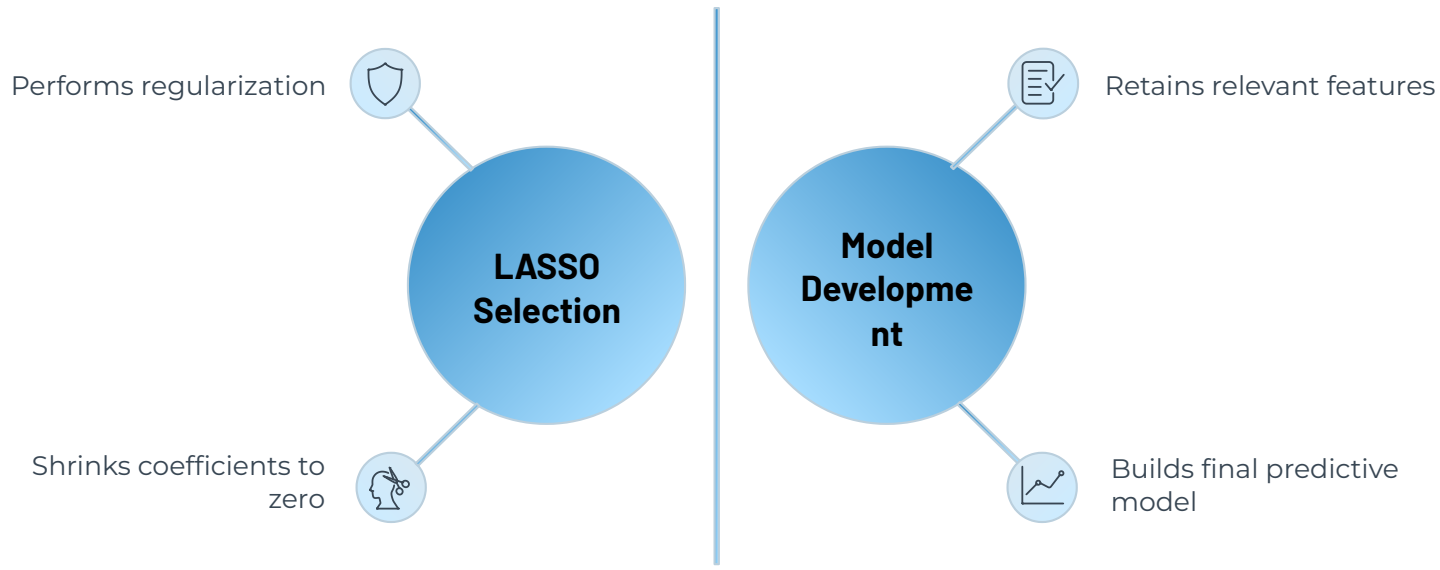
- **Data Transformation**
- Standardized or normalized features (using `StandardScaler` / `MinMaxScaler`).
- Converted skewed distributions to near-Gaussian (log transform).

- **Data Splitting**
- Divided dataset into **train / test / validation** sets.
- Ensured stratification for classification tasks.

- **Final Dataset**
- Cleaned, well-structured dataset ready for **EDA, visualization, and model training**.

Feature Selection Strategy

"We applied LASSO regression for feature selection because it performs both regularization and variable selection. By adding an L1 penalty, LASSO automatically shrinks the coefficients of less important predictors to zero, retaining only the most relevant features. This reduced dimensionality, improved model interpretability, and helped prevent overfitting. The final set of predictors obtained from LASSO were then used for model development."



Why LASSO?

- LASSO adds an **L1 penalty** to the regression coefficients.
- This penalty forces some coefficients to become **exactly zero**, effectively removing unimportant features.
- It is useful when we have many predictors and want to avoid overfitting by keeping only the most relevant ones.

How LASSO Works in Feature Selection

- Standard regression keeps all features (even weak ones).
- LASSO shrinks coefficients toward zero.
- If a feature is not strongly contributing to prediction, its coefficient is pushed to **0**.
- This makes LASSO act as both a **regularization technique** and an **automatic feature selector**.

Outcome of Using LASSO

- Only the most significant predictors remain with **non-zero coefficients**.
- Irrelevant or redundant features are eliminated.
- The final model is simpler, more interpretable, and less prone to overfitting.
- Example: If initial dataset had 20 features, LASSO may reduce it to ~7–10 meaningful predictors.

Process in the Project

- Applied **LassoCV** (cross-validated LASSO) to find the best penalty parameter (alpha).
- Extracted the features with non-zero coefficients.
- Used these selected features for final **model training and evaluation**.

Numerical

engine_size

cylinders

fuel_consumption_city

fuel_consumption_hwy

Vehicle Makes

make_ACURA
make_BENTLEY
make_BUGATTI
make_FIAT
make_INFINITI
make_LAMBORGHINI
make_LAND ROVER
make_RAM
make_SCION
make_VOLKSWAGEN

Vehicle Classes

vehicle_class_MINICOMPAT
vehicle_class_SPECIAL PURPOSE VEHICLE
vehicle_class_SUV SMALL
vehicle_class_SUV STANDARD
vehicle_class_TWO SEATER
vehicle_class_VAN CARGO

Transmission Types ·

transmission_A
transmission_AM
transmission_AS
transmission_AV
transmission_M Fuel

Fuel Types

fuel_type_E

fuel_type_X

fuel_type_Z

Model Training & Validation

📊 Data Split Strategy

- **70% Training Set (4,391 samples):**
  Used for model learning and fitting.
- **15% Validation Set (941 samples):**
  Used during model tuning and selection.
- **15% Test Set (941 samples):**
  Held out for final **unbiased evaluation** of model performance.

🔄 Cross-Validation

- Applied **5-Fold Cross Validation** on the training + validation sets.
- Ensures robust performance estimation and prevents overfitting by testing the model on multiple folds.

🔧 Hyperparameter Tuning

- Used **Optuna Framework** for **automated hyperparameter optimization**.
- Optuna efficiently explores the search space (using pruning and Bayesian optimization) to find the best parameters for each model.

📈 Performance Metrics

- **R^2 Score (Coefficient of Determination):** Measures proportion of variance explained by the model. Higher is better.
- **Root Mean Square Error (RMSE):** Penalizes larger errors, shows model's prediction accuracy. Lower is better.
- **Mean Absolute Error (MAE):** Average magnitude of errors. Lower is better.

Model Selection Strategy

1

Linear Models

Linear Regression – baseline model to capture linear dependencies.

ElasticNet

Handles multicollinearity while performing feature selection

2

Non- Linear Models

K-Nearest Neighbors (KNN) Regressor – non-parametric model based on proximity.

Support Vector Regressor: Models complex, high dimensional patterns

3

Ensemble Methods

Random Forest: Robust to noise, captures non linear feature interactions

XGBoostRegressor : Powerful gradient boosting, highly effective on tabular data

Model Performance Results

Model	RMSE (± Std)	R^2 (± Std)	Remarks
Linear Regression	8.33 ± 0.30	8.33% ± 0.30%	Baseline linear model, moderate error.
ElasticNetCV	14.59 ± 0.44	14.59% ± 0.44%	Handles multicollinearity, higher RMSE indicates underfitting.
Random Forest Regressor	3.72 ± 0.37	3.72% ± 0.37%	Strong ensemble model, low error, robust predictions.
XGBRegressor	3.59 ± 0.35	3.59% ± 0.35%	Best performance, captures complex non-linear patterns effectively.
Support Vector Regressor (SVR)	30.78 ± 1.58	30.78% ± 1.58%	High error, struggles with large feature space.
KNN Regressor	12.89 ± 0.74	12.89% ± 0.74%	Sensitive to scaling, moderate performance.

🏆 Best Model: XGBoostRegressor

Achieved the lowest RMSE and highest R^2, making it the most accurate model for carbon emission prediction.

📈Key Finding

Random Forest also performs well, offering robustness and slightly higher error than XGB.

✅ Final Recommendation

XGBoostRegressor provides the optimal balance of accuracy and consistency for production deployment in CO_2 emission prediction.

Model Refit & Final Evaluation

Dataset	RMSE	MAE	R^2 Score
Train + Validation	1.637	1.996	99.79%
Test	1.730	2.220	99.75%

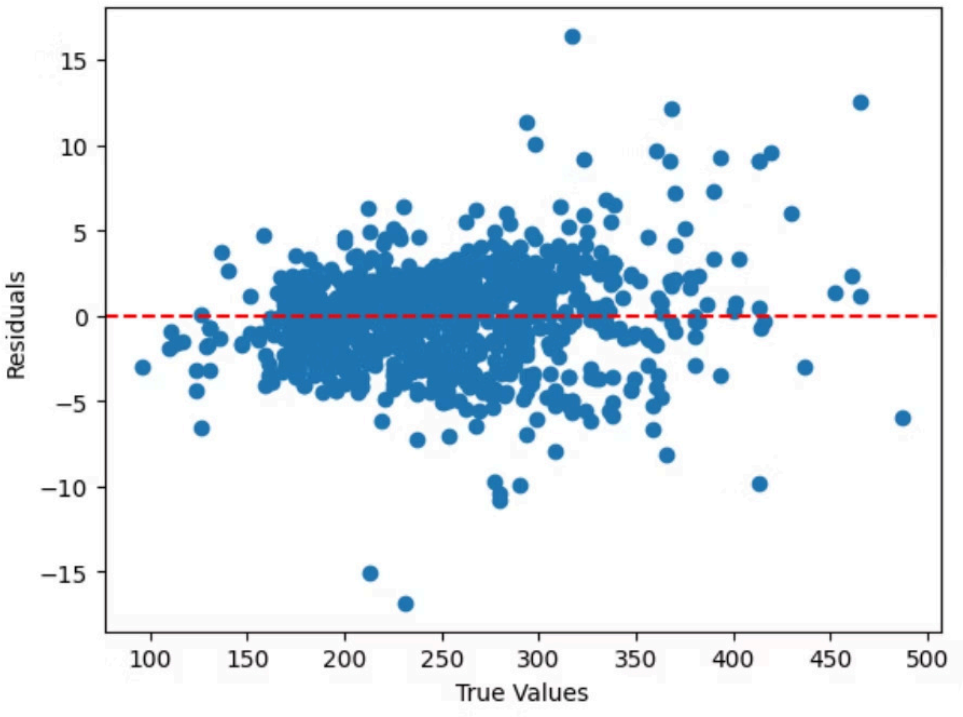
The model achieves **very high R^2 scores** on both training and test sets, indicating excellent fit and predictive power.

The small difference in RMSE and R^2 between train+val and test sets demonstrates that the model **generalizes well** without overfitting.

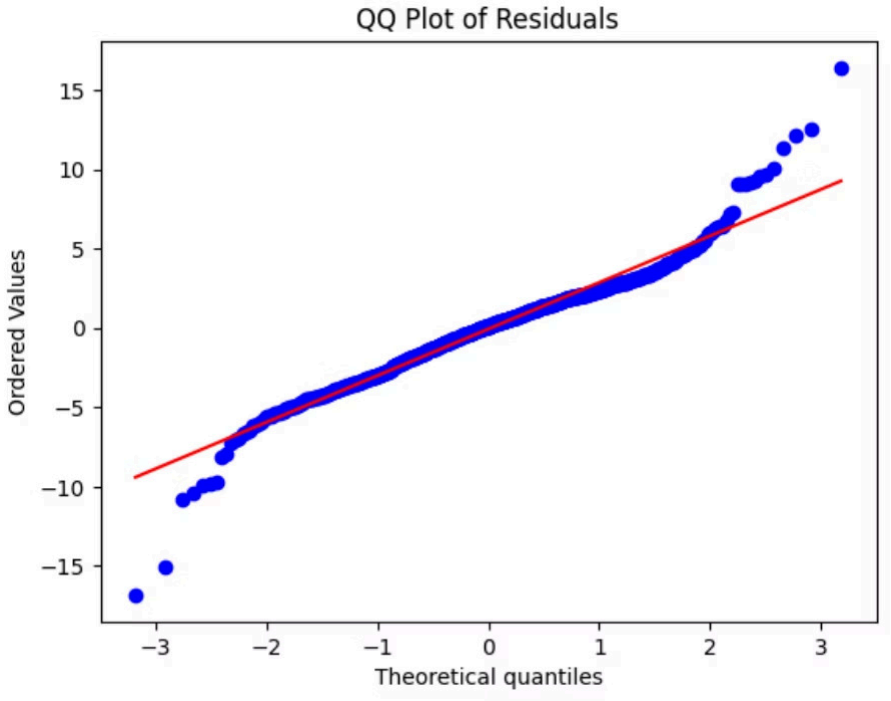
MAE values are low, confirming that the model's predictions are consistently accurate across samples.

Analysis after the Final Evaluation Residual

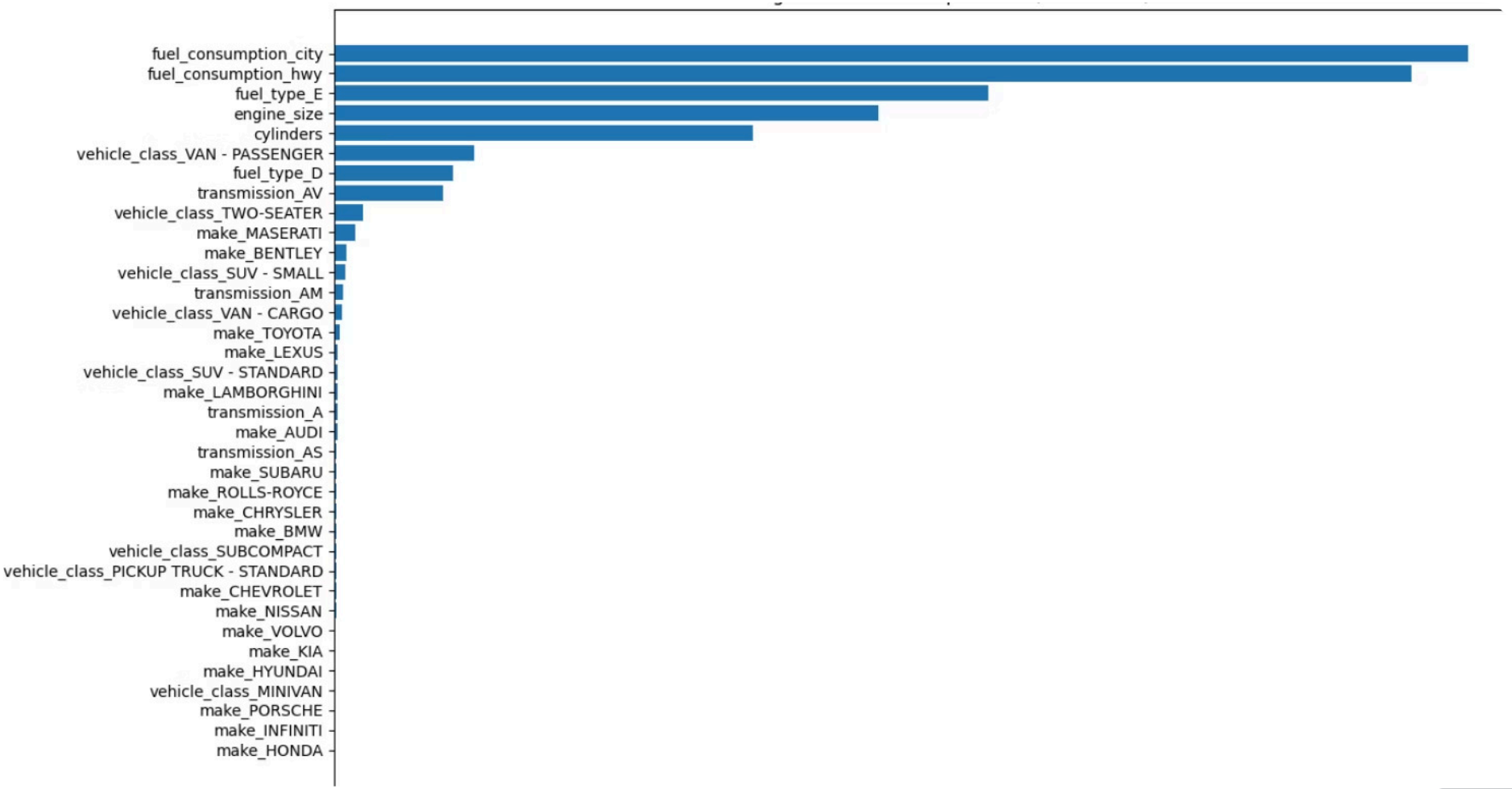
Residual Distribution QQ Plot

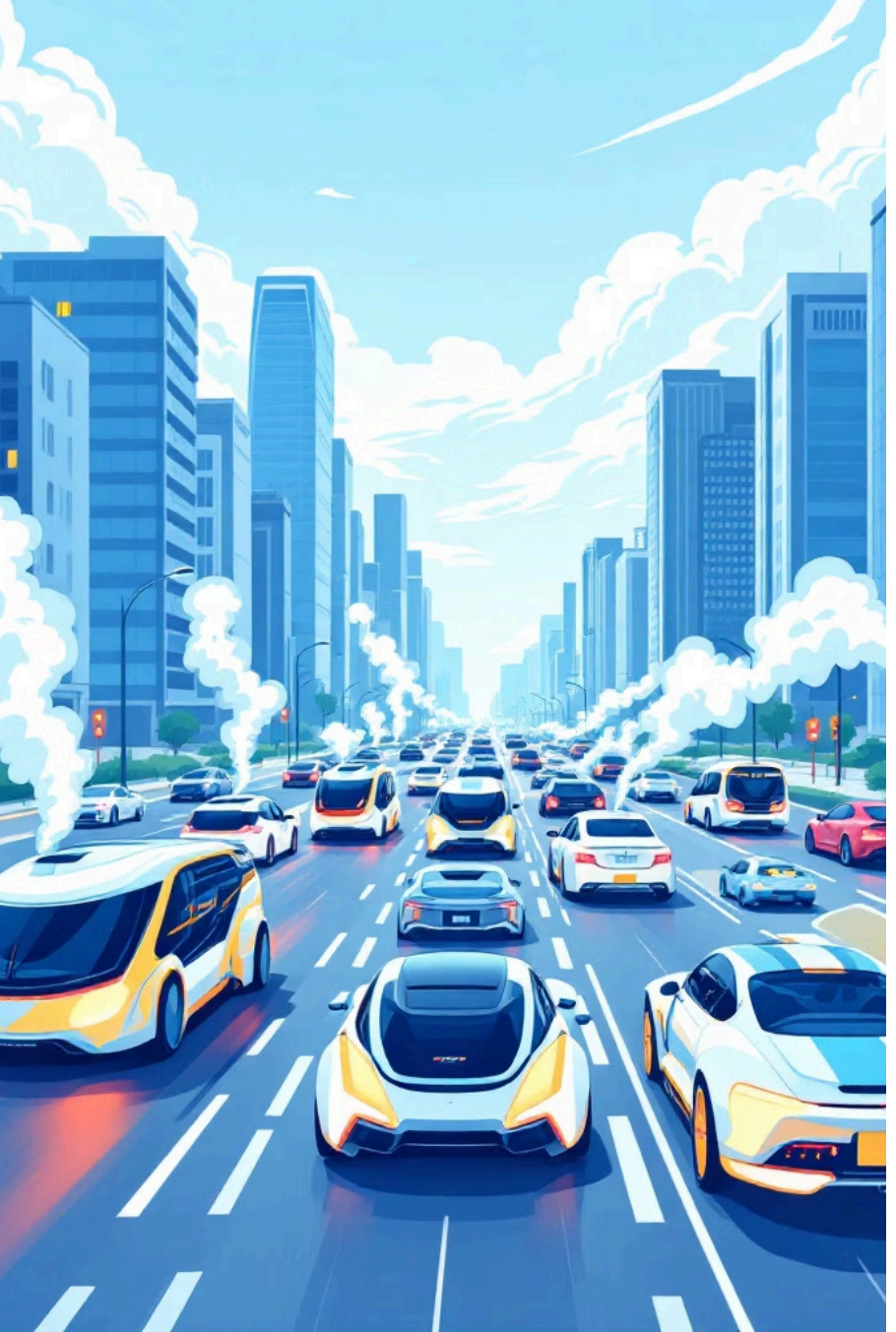


QQ Plot Normality



Feature Importance Hierarchy





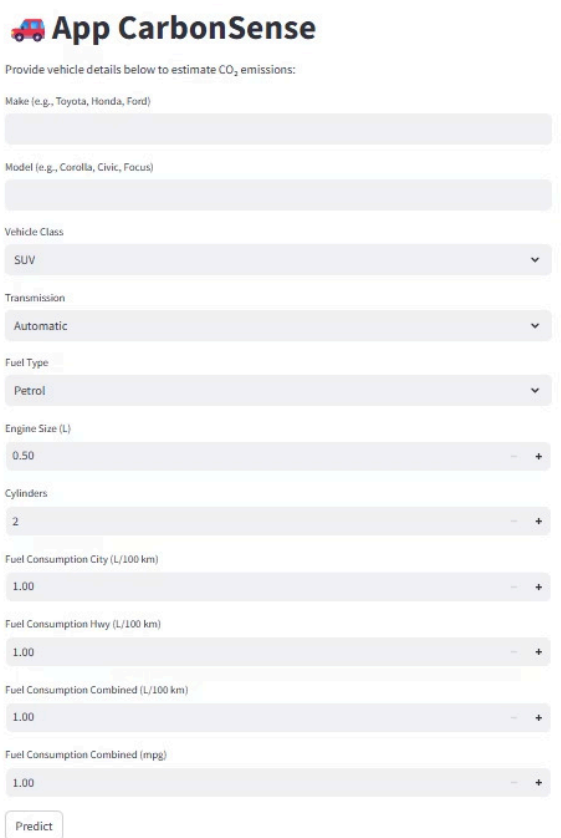
Deployment

🚗App CarbonSense

CO_2 Emission Predictor App

A web-based application that predicts carbon emissions based on input features like energy usage, industrial activity, and population. It provides real-time, accurate forecasts using the trained XGBoost model.

Home Tab User Input Interface



The Home Tab serves as the main interaction point for users. It provides **interactive input fields** such as sliders, dropdowns, and text boxes for entering features like Make, Model, Number of Cylinders, and Engine size etc. Users can **submit their inputs** to the app, which then processes the data using the trained XGBoost model to generate **instant CO_2 emission predictions**. The interface is designed to be **intuitive and user-friendly**, ensuring accessibility for both technical and non-technical users.

Final Conclusion & Future Enhancements

The project successfully developed a **robust predictive model** for carbon emissions using XGBoost, achieving **high accuracy ($R^2 \approx 99.75\%$)** and low error metrics. The model generalizes well to unseen data, and the **CO_2 Emission Predictor App** provides an intuitive interface for real-time forecasting based on user inputs. This enables stakeholders to make informed decisions for emission reduction and sustainability planning.

Future Enhancement Roadmap

1. **Real-Time Integration**
   - Integrate live data from vehicles and sensors to provide **instant carbon emission predictions**.
2. **Dataset Expansion**
   - Include broader vehicle and industrial datasets to improve **model coverage and accuracy**.
3. **Model Explainability**
   - Add interpretability features (e.g., **SHAP/LIME**) to provide transparency on **feature contributions**.
4. **Eco Recommendations**
   - Generate actionable insights for users to adopt **sustainable practices** and reduce emissions.

Thank You!

We appreciate your time and attention to our project on predicting vehicle CO_2 emissions and our path towards a sustainable future.

Should you have any questions or wish to explore collaboration, please feel free to reach out.