Car Fuel Efficiency Predictor using ML

1. Executive Summary

This project predicts a vehicle’s fuel efficiency using machine learning. The Auto MPG dataset was analyzed to study how features like weight, horsepower, engine size, and model year affect fuel consumption. After preprocessing, EDA, and feature selection, regression models were trained and evaluated. The best model was deployed as a lightweight app for real-time fuel efficiency prediction, demonstrating a complete ML workflow

🔗 **Project Links:**

- GitHub Repository
- Live Application
- Presentation

2. Problem Statement

Fuel efficiency is vital for vehicle design, the environment, and consumer decisions. This project builds a regression model to predict a car’s fuel efficiency based on its technical and physical attributes, providing data-driven insights into factors that influence automotive performance.

3. Dataset Description

Column Name	Description
mpg	Miles per gallon; represents the fuel efficiency of the vehicle (target variable)
cylinders	Number of engine cylinders
displacement	Engine displacement (cubic inches), indicating engine size
horsepower	Engine horsepower; a measure of engine power
weight	Vehicle weight in pounds
acceleration	Time taken to accelerate from 0 to 60 mph (seconds)
model_year	Model year of the vehicle
origin	Country of origin of the car (USA, Europe, or Japan)
name	Name and model of the car (identifier, not used for modeling)

4. Project Objectives

The project followed a structured machine learning pipeline consisting of :

- **Data Exploration and Preprocessing**
- **Feature Engineering**
- **Model Evaluation and Deployment**
- **Exploratory Data Analysis (EDA)**
- **Model Training**
- **Key Insights from Data Analysis**

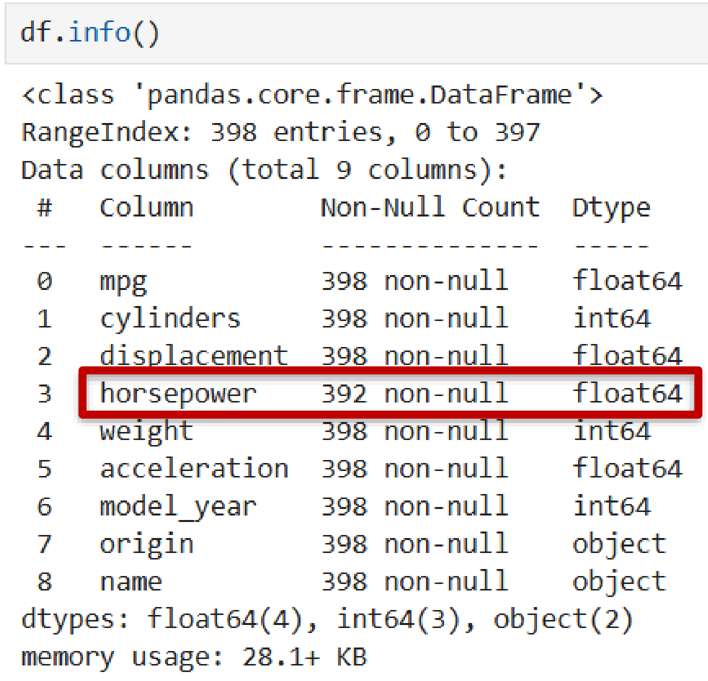
4.1 Tech Stack

- **Programming Language:** Python
- **Data Analysis & Visualization:** Pandas, NumPy, Matplotlib, Seaborn
- **Machine Learning:** Scikit-learn
- **Model Deployment:** Streamlit
- **Model Serialization:** Joblib

4.2 Data Exploration

The dataset was first examined to understand its structure, check for missing values, and review basic statistics. This step ensures a clear understanding of the variables, their types, and any potential issues before modeling.

Identify the Null Values:

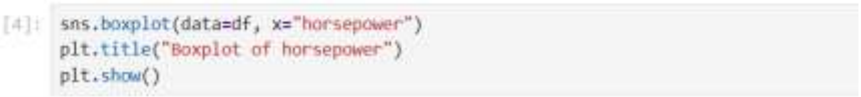


```
df.info()
```

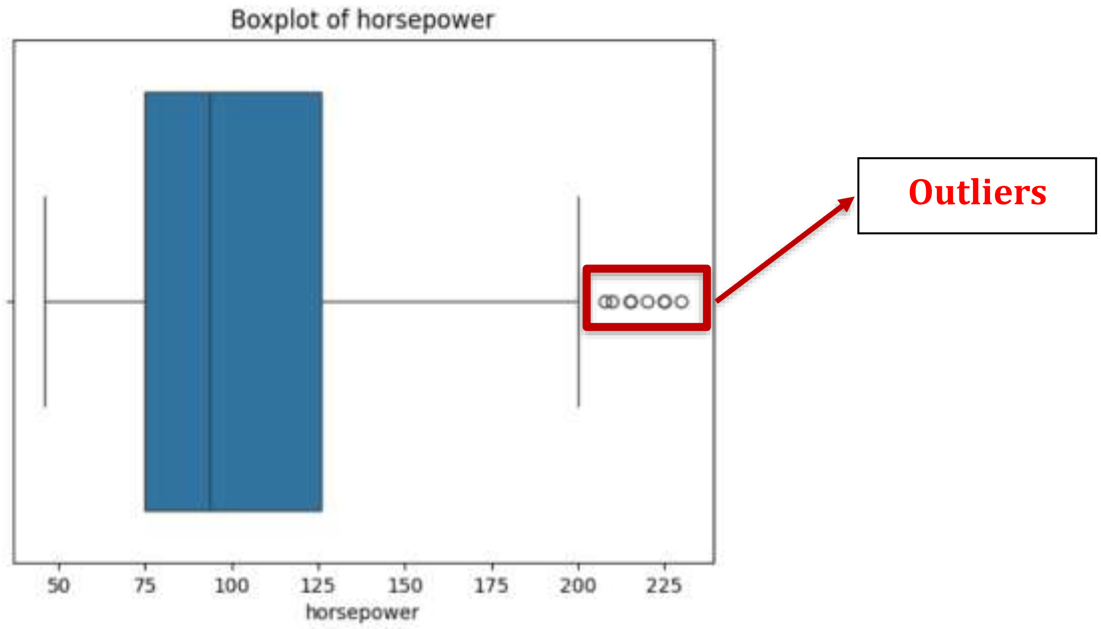
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 398 entries, 0 to 397
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   mpg           398 non-null    float64
 1   cylinders     398 non-null    int64  
 2   displacement  398 non-null    float64
 3   horsepower    392 non-null    float64
 4   weight        398 non-null    int64  
 5   acceleration  398 non-null    float64
 6   model_year    398 non-null    int64  
 7   origin        398 non-null    object 
 8   name          398 non-null    object 
dtypes: float64(4), int64(3), object(2)
memory usage: 28.1+ KB
```

Null Values

The **Horsepower** column contains **six null values.** These need to be replaced with appropriate values; however, before proceeding, it is essential to examine the column for the presence of any outliers. by using the **Boxplot** visual.



```
[4]: sns.boxplot(data=df, x="horsepower")
     plt.title("Boxplot of horsepower")
     plt.show()
```

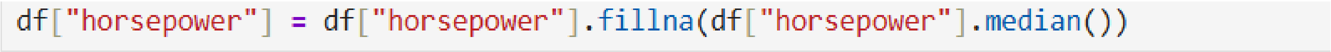


4.3 Data Preprocessing

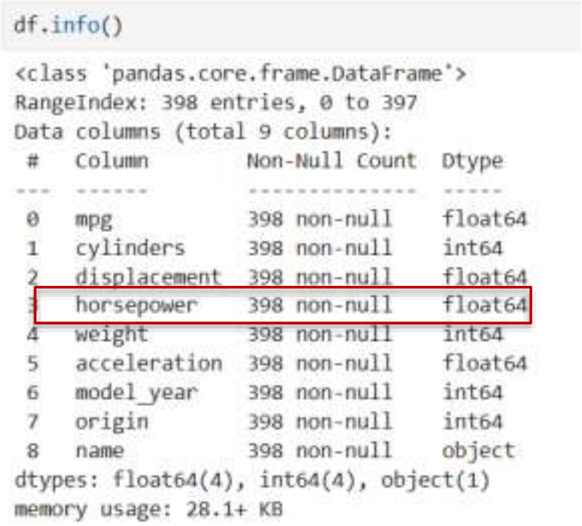
Data preprocessing involved cleaning and preparing the dataset for machine learning. Irrelevant columns like name were removed, missing values were handled, and duplicates were checked. This ensures that the data is accurate and suitable for modeling.

Null Values Filled With Median:

Since the **Horsepower** column contains outliers, the null values should be replaced using the **median** to ensure robust and reliable imputation.



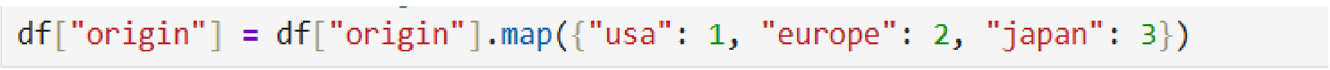
```
df["horsepower"] = df["horsepower"].fillna(df["horsepower"].median())
```



```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 398 entries, 0 to 397
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   mpg           398 non-null    float64
 1   cylinders     398 non-null    int64  
 2   displacement  398 non-null    float64
 3   horsepower    398 non-null    float64
 4   weight        398 non-null    int64  
 5   acceleration  398 non-null    float64
 6   model_year    398 non-null    int64  
 7   origin        398 non-null    int64  
 8   name          398 non-null    object 
dtypes: float64(4), int64(4), object(1)
memory usage: 28.1+ KB
```

The origin feature includes three regions—**USA, Europe, and Japan**—which have been mapped to **1, 2, and 3**, respectively, as machine learning models can interpret **numerical values only**.



```
df["origin"] = df["origin"].map({"usa": 1, "europe": 2, "japan": 3})
```

4.4 Exploratory Data Analysis (EDA)

EDA was performed to uncover patterns and relationships between features and fuel efficiency (mpg). Visualizations such as histograms, correlation heatmaps, and boxplots were used to identify trends, detect outliers, and highlight features most related to the target variable.

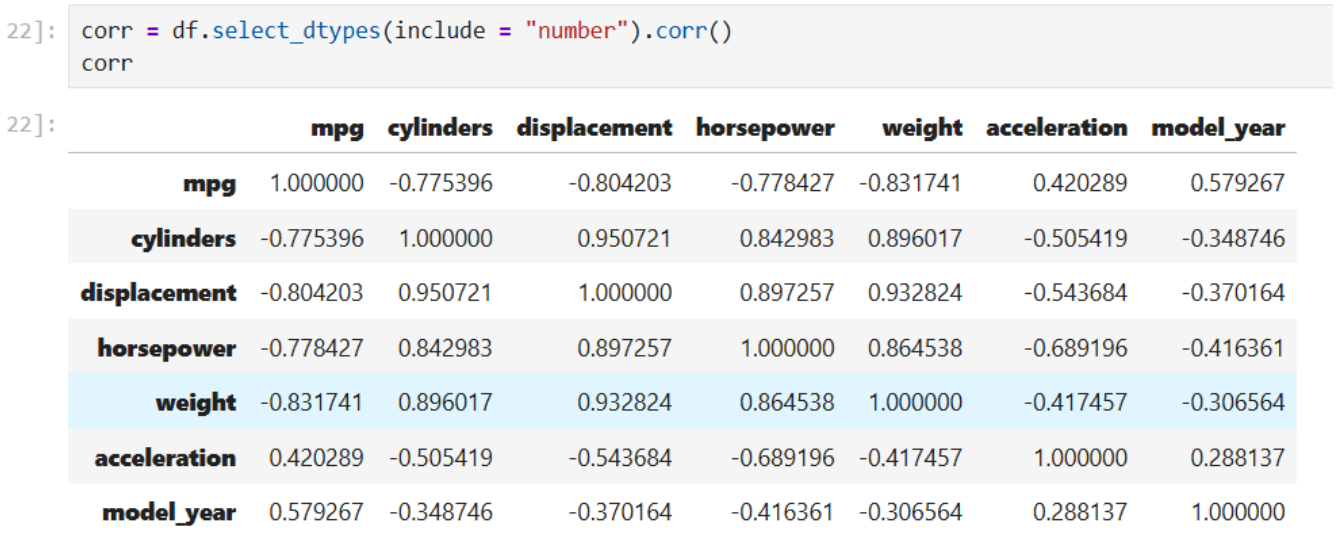
Finding the Correlation:

Before proceeding with **feature selection**, it is important to examine the **correlation between feature** This analysis helps provide an initial understanding of how strongly features are related to each other and to the target variable.

What is Correlation:
Correlation is a measure that shows how strongly two variables are related to each other.

Note:

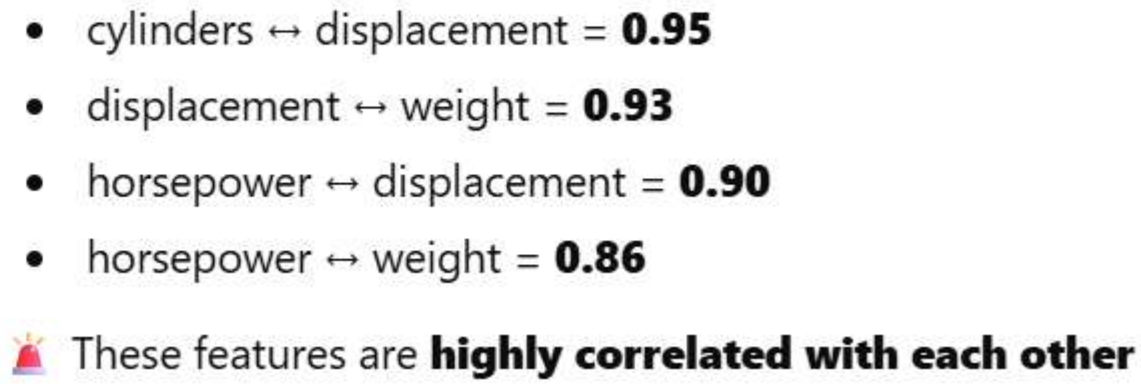
However, this is **for reference purposes only** and should not be considered a definitive method for selecting features to train the model.



```
22]: corr = df.select_dtypes(include = "number").corr()
     corr
```

22]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year
mpg	1.000000	-0.775396	-0.804203	-0.778427	-0.831741	0.420289	0.579267
cylinders	-0.775396	1.000000	0.950721	0.842983	0.896017	-0.505419	-0.348746
displacement	-0.804203	0.950721	1.000000	0.897257	0.932824	-0.543684	-0.370164
horsepower	-0.778427	0.842983	0.897257	1.000000	0.864538	-0.689196	-0.416361
weight	-0.831741	0.896017	0.932824	0.864538	1.000000	-0.417457	-0.306564
acceleration	0.420289	-0.505419	-0.543684	-0.689196	-0.417457	1.000000	0.288137
model_year	0.579267	-0.348746	-0.370164	-0.416361	-0.306564	0.288137	1.000000



- cylinders ↔ displacement = **0.95**
- displacement ↔ weight = **0.93**
- horsepower ↔ displacement = **0.90**
- horsepower ↔ weight = **0.86**

🚨 These features are **highly correlated with each other**

4.5 Key Insights from Data Analysis:

• Vehicle weight has the strongest negative impact on fuel efficiency (MPG).

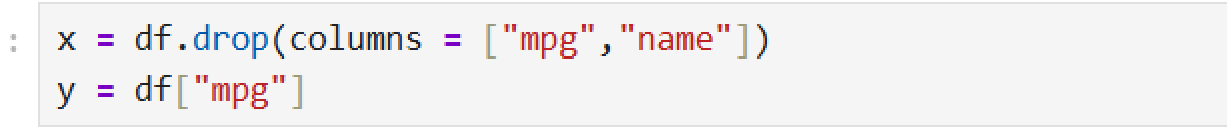
• Higher engine displacement and horsepower are associated with lower MPG.

• Newer model-year vehicles generally demonstrate better fuel efficiency due to technological advancements.

• Strong multicollinearity exists among cylinders, displacement, horsepower, and weight.

4.6 Feature Engineering

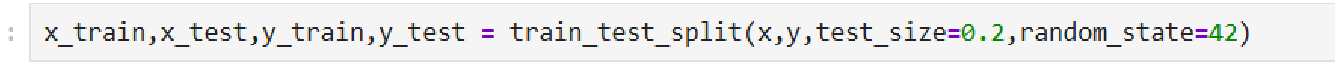
Relevant features were selected based on correlation analysis and domain knowledge.



```
x = df.drop(columns = ["mpg","name"])
y = df["mpg"]
```

Train_Test_Split:

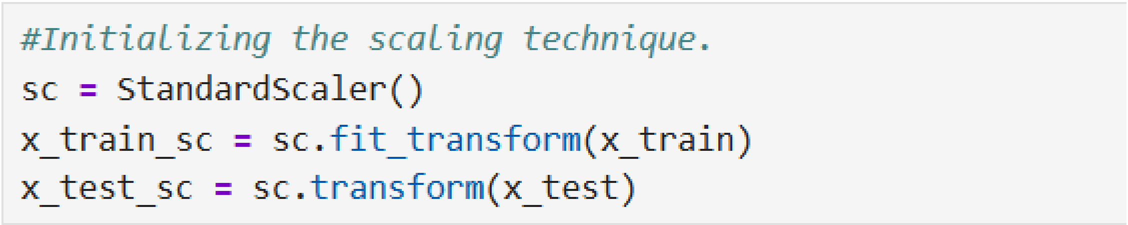
Finally, the dataset was split into training (80%) and testing (20%) sets for model evaluation.



```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=42)
```

Scaling Techniques:

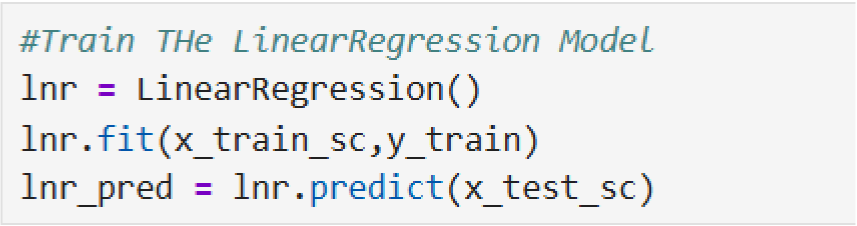
Numerical features were scaled using StandardScaler to standardize their ranges, improving model performance



```
#Initializing the scaling technique.
sc = StandardScaler()
x_train_sc = sc.fit_transform(x_train)
x_test_sc = sc.transform(x_test)
```

5. Model Development

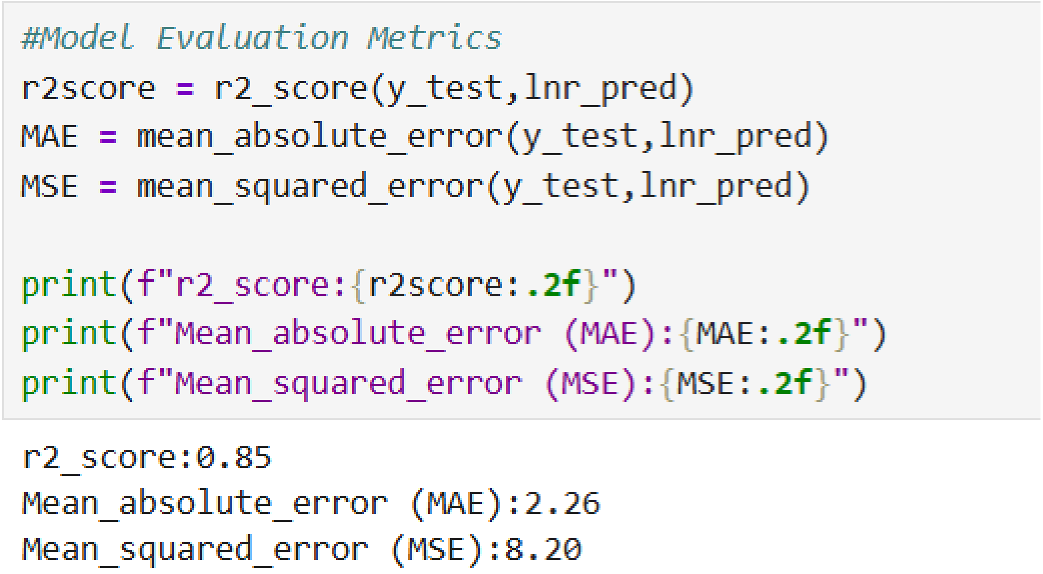
For this project, multiple regression models were explored to predict MPG. Among them, the Linear Regression model was selected and implemented.



```
#Train THe LinearRegression Model
lnr = LinearRegression()
lnr.fit(x_train_sc,y_train)
lnr_pred = lnr.predict(x_test_sc)
```

6. Model Evaluation

Models were evaluated using metrics such as R² score, Mean Absolute Error (MAE), and Mean Squared Error (MSE). This step allowed for a quantitative comparison to identify the model that best predicts fuel efficiency.



```
#Model Evaluation Metrics
r2score = r2_score(y_test,lnr_pred)
MAE = mean_absolute_error(y_test,lnr_pred)
MSE = mean_squared_error(y_test,lnr_pred)

print(f"r2_score:{r2score:.2f}")
print(f"Mean_absolute_error (MAE):{MAE:.2f}")
print(f"Mean_squared_error (MSE):{MSE:.2f}")
```

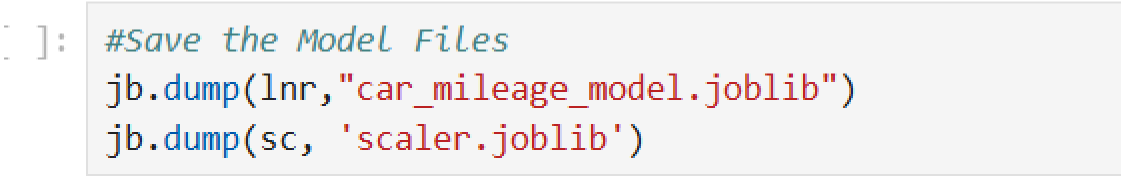
```
r2_score:0.85
Mean_absolute_error (MAE):2.26
Mean_squared_error (MSE):8.20
```

7. Model Selection & Interpretation

The final model was selected based on evaluation metrics and its ability to generalize to unseen data. This analysis identifies which vehicle characteristics most strongly influence fuel efficiency

8. Save The Model and Deployment

The selected model was saved by using Joblib and deployed using Streamlit. The application accepts vehicle attributes such as weight, horsepower, and engine size to predict fuel efficiency.



```
[ ]: #Save the Model Files
     jb.dump(lnr,"car_mileage_model.joblib")
     jb.dump(sc, 'scaler.joblib')
```

9. Conclusion

Developed an end-to-end machine learning pipeline to predict vehicle fuel efficiency (MPG) using automotive features. Linear Regression model achieved the best performance, and feature importance analysis highlighted key factors like weight, horsepower, and displacement. This project demonstrates practical skills in data analysis, modeling, and ML deployment.