Fuel Consumption Prediction using Regression Techniques

https://linkedin.com/in/muhammad-randa-yandika

https://bit.ly/Randayandika_portofolio

About Dataset

- Datasets provide model-specific fuel consumption ratings and estimated carbon dioxide emissions for new light-duty vehicles for retail sale in Canada.
- To help you compare vehicles from different model years, the fuel consumption ratings for 2000 to 2022 vehicles have been adjusted to reflect the improved testing that is more representative of everyday driving. Note that these are approximate values that were generated from the original ratings, not from vehicle testing.

Outline

- Use Case Summary
- Data Understanding
- EDA & Visualization
- Data Preprocessing
- Modelling & Evaluation
- Conclusion

Use Case Summary

Objective

- Get an insight into how much Fuel Consumption in years,
- Knowing which car manufacturer consumes the most fuel,
- Knowing which car model consumes the most fuel,
- Knowing the relationship between emissions and fuel consumption,
- knowing what specifications of cars consume the most fuel,
- Create models to predict fuel consumption using Regression Techniques.

Outcome

- Get to know how much Fuel Consumption in years,
- Get to know which car manufacturer consumes the most fuel,
- Get to know which car model consumes the most fuel,
- Get to know the relationship between emissions and fuel consumption,
- Get to know what specifications of cars consume the most fuel,
- Making model to predict fuel consumption using Regression Techniques.

Data Understanding

DATA ATRIBUTES INFORMATION

Feature	Description
YEAR: .	The production year of the vehicle
MAKE:	The manufacturer of the vehicle.
MODEL	The specific model of the vehicle.
VEHICLE CLASS:	The category of the vehicle (e.g. compact, SUV, truck, etc.).
ENGINE SIZE:	The size of the engine in liters.
CYLINDERS:	The number of cylinders in the engine.
TRANSMISSION:	The type of transmission (manual, automatic, etc.).

Sources

- https://www.kaggle.com/datasets/ahmettyilmazz/fuel-consumption

FUEL	The type of fuel used (e.g. gasoline, diesel, etc.).
FUEL CONSUMPTION	The amount of fuel used by the vehicle.
HWY (L/100 km)	Fuel consumption on the highway, in liters per 100 km.
COMB (L/100 km)	Fuel consumption in combined city/highway driving, in liters per 100 km.
COMB (mpg)	Fuel consumption in combined city/highway driving, in miles per gallon.
EMISSIONS	The amount of emissions produced by the vehicle.

Data Information & Statistic Numerical

- dataset have 13 columns with 22556 entries and data type from each column.
- Have 5 categorical feature and 8 numerical feature
- The highest fuel consumption is 30,6

```
numerical =  df.select_dtypes(include=[np.number])
numerical.columns
```

```
Index(['YEAR', 'ENGINE SIZE', 'CYLINDERS', 'FUEL CONSUMPTION', 'HWY', 'COMB',
       'COMB_mpg', 'EMISSIONS'],
      dtype='object')
```

```
categorical = df.select_dtypes(exclude=[np.number])
categorical.columns
```

```
Index(['MAKE', 'MODEL', 'VEHICLE CLASS', 'TRANSMISSION', 'FUEL'], dtype='object')
```

df.describe()

	YEAR	ENGINE SIZE	CYLINDERS	FUEL CONSUMPTION	HWY (L/100 km)	COMB (L/100 km)	COMB (mpg)	EMISSIONS
count	22556.000000	22556.000000	22556.000000	22556.000000	22556.000000	22556.000000	22556.000000	22556.000000
mean	2011.554442	3.356646	5.854141	12.763513	8.919126	11.034341	27.374534	250.068452
std	6.298269	1.335425	1.819597	3.500999	2.274764	2.910920	7.376982	59.355276
min	2000.000000	0.800000	2.000000	3.500000	3.200000	3.600000	11.000000	83.000000
25%	2006.000000	2.300000	4.000000	10.400000	7.300000	9.100000	22.000000	209.000000
50%	2012.000000	3.000000	6.000000	12.300000	8.400000	10.600000	27.000000	243.000000
75%	2017.000000	4.200000	8.000000	14.725000	10.200000	12.700000	31.000000	288.000000
max	2022.000000	8.400000	16.000000	30.600000	20.900000	26.100000	78.000000	608.000000

df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22556 entries, 0 to 22555
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   YEAR              22556 non-null  int64
 1   MAKE              22556 non-null  object
 2   MODEL             22556 non-null  object
 3   VEHICLE CLASS     22556 non-null  object
 4   ENGINE SIZE       22556 non-null  float64
 5   CYLINDERS         22556 non-null  int64
 6   TRANSMISSION      22556 non-null  object
 7   FUEL              22556 non-null  object
 8   FUEL CONSUMPTION  22556 non-null  float64
 9   HWY (L/100 km)    22556 non-null  float64
 10  COMB (L/100 km)   22556 non-null  float64
 11  COMB (mpg)        22556 non-null  int64
 12  EMISSIONS         22556 non-null  int64
dtypes: float64(4), int64(4), object(5)
memory usage: 2.2+ MB
```

Dataset Manipulation

- Rename some column to prevent errors from occurring
- We found duplicated data in manufacturing name, from lowercase to uppercase

```
df = df.rename(columns = {'HWY (L/100 km)': 'HWY', 'COMB (L/100 km)': 'COMB','COMB (mpg)':'COMB_mpg'}, inplace = False)
df.head(5)
```

	YEAR	MAKE	MODEL	VEHICLE CLASS	ENGINE SIZE	CYLINDERS	TRANSMISSION	FUEL	FUEL CONSUMPTION	HWY	COMB	COMB_mpg	EMISSIONS
0	2000	ACURA	1.6EL	COMPACT	1.6	4	A4	X	9.2	6.7	8.1	35	186
1	2000	ACURA	1.6EL	COMPACT	1.6	4	M5	X	8.5	6.5	7.6	37	175
2	2000	ACURA	3.2TL	MID-SIZE	3.2	6	AS5	Z	12.2	7.4	10.0	28	230
3	2000	ACURA	3.5RL	MID-SIZE	3.5	6	A4	Z	13.4	9.2	11.5	25	264
4	2000	ACURA	INTEGRA	SUBCOMPACT	1.8	4	A4	X	10.0	7.0	8.6	33	198

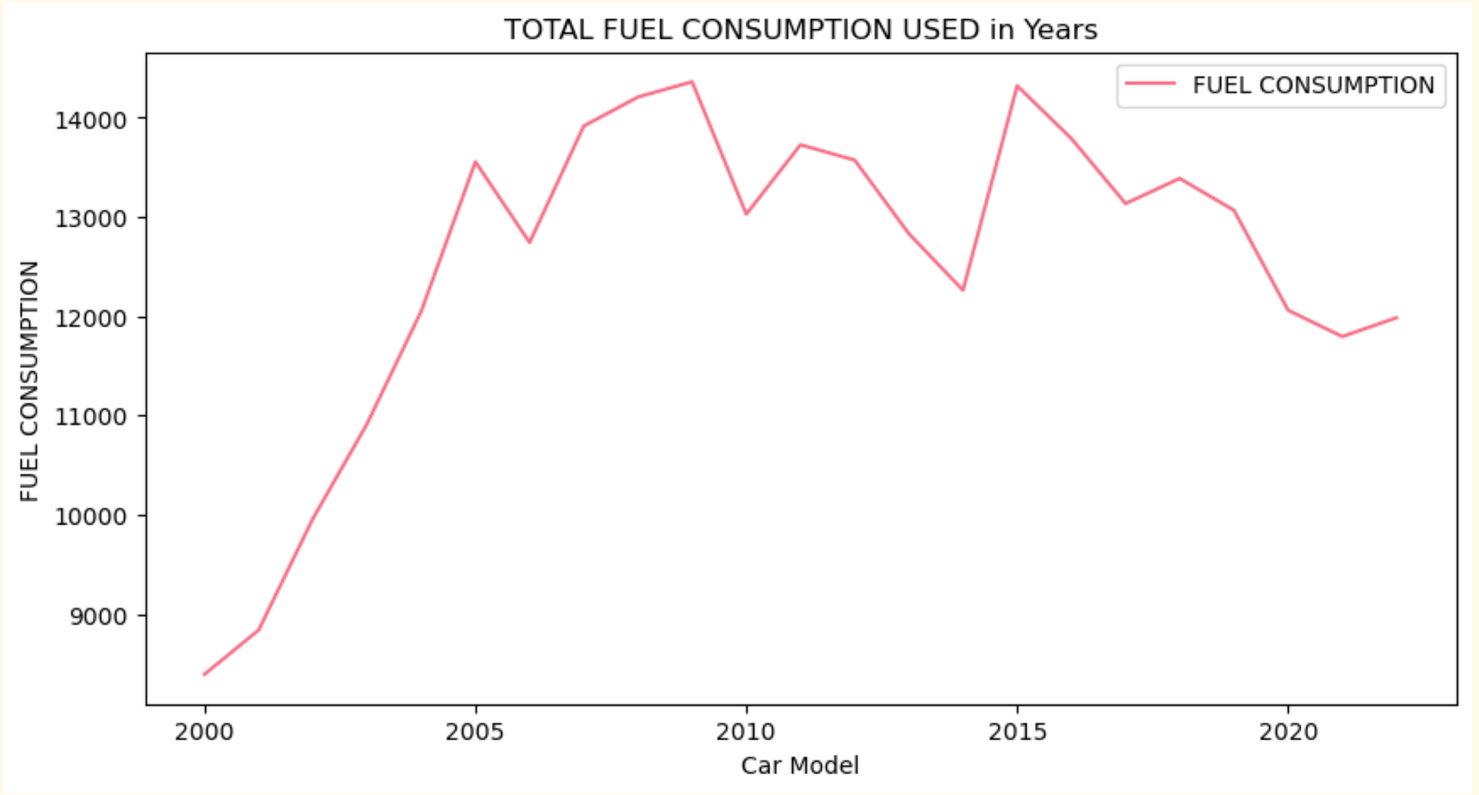
```
Found duplicate values: {'bmw', 'volvo', 'chrysler', 'acura', 'scion', 'daewoo', 'hummer', 'ferrari', 'porsche', 'volkswagen',
'lamborghini', 'pontiac', 'genesis', 'lexus', 'toyota', 'dodge', 'ford', 'jaguar', 'chevrolet', 'honda', 'plymouth', 'oldsmobil
e', 'lincoln', 'mini', 'infiniti', 'audi', 'gmc', 'hyundai', 'mazda', 'mitsubishi', 'srt', 'alfa romeo', 'suzuki', 'bugatti',
'jeep', 'kia', 'saturn', 'saab', 'bentley', 'fiat', 'maserati', 'land rover', 'buick', 'nissan', 'isuzu', 'smart', 'mercedes-be
nz', 'ram', 'subaru', 'rolls-royce', 'aston martin', 'cadillac'}
```

```
df['MAKE'] = df['MAKE'].apply(lambda x: x.upper() if x.lower() in duplicates else x)
df.head(5)
```

EDA
&
Data Visualization

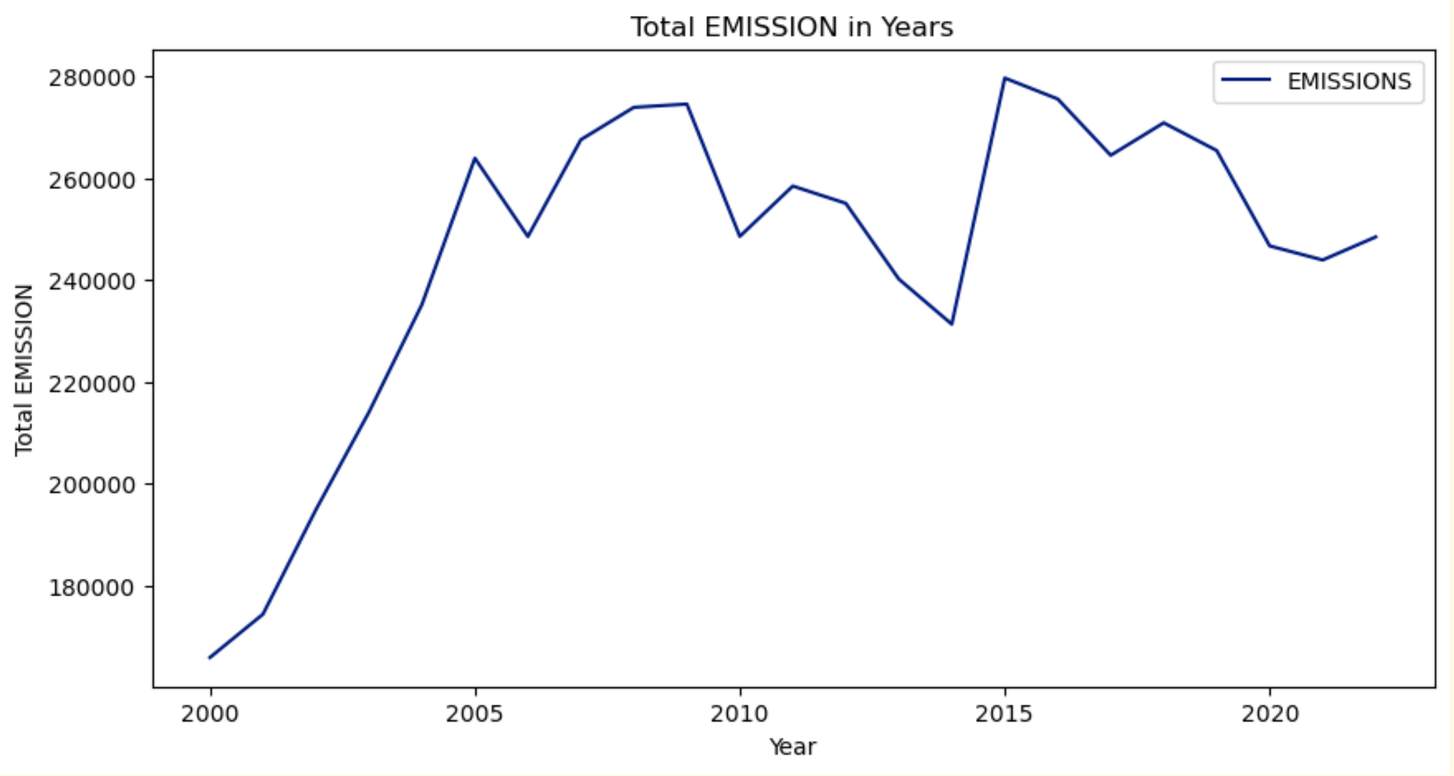
Time Series Plot for Fuel Consumption

This graph shows fuel consumption from **2000** to **2022**, where in **2009** had the highest fuel consumption at **14361.8** liters



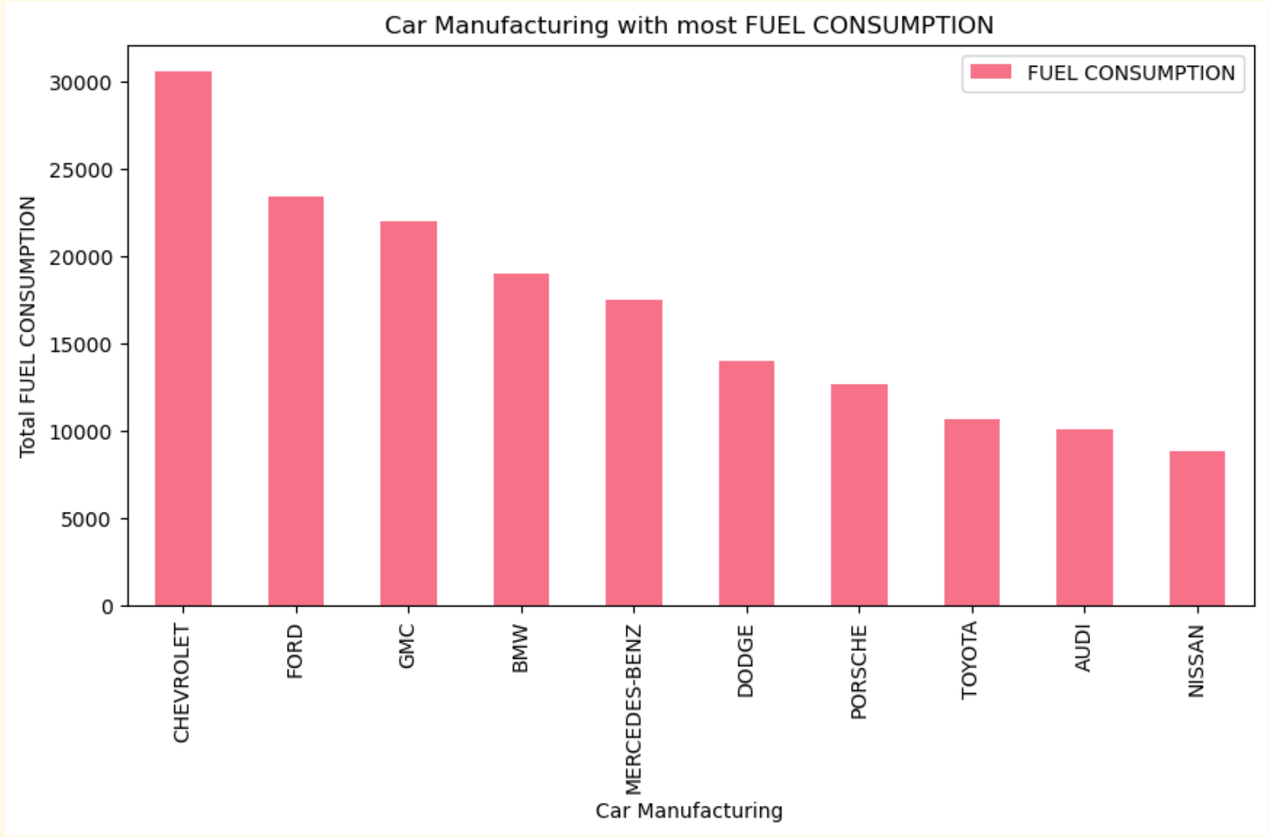
Time Series Plot for Emissions

This graph shows the emissions that occurred from the year 2000 to 2022,
with 2009 having the highest amount of emissions at 279.571 g/km



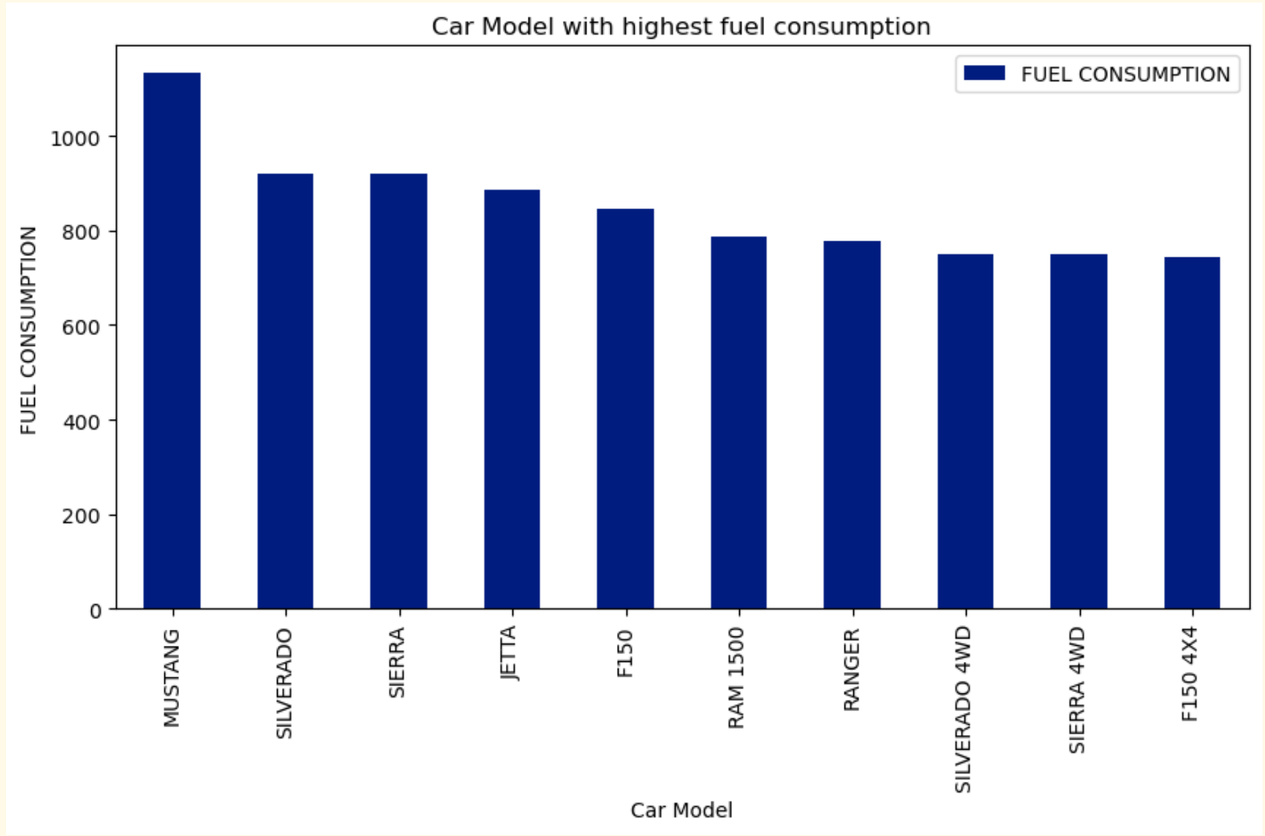
Car Manufacturing with highest consumption

This is a top 10 graph for car manufacturers with the highest fuel consumption, and Chevrolet is the manufacturer with the highest fuel consumption at 30569.8 liters



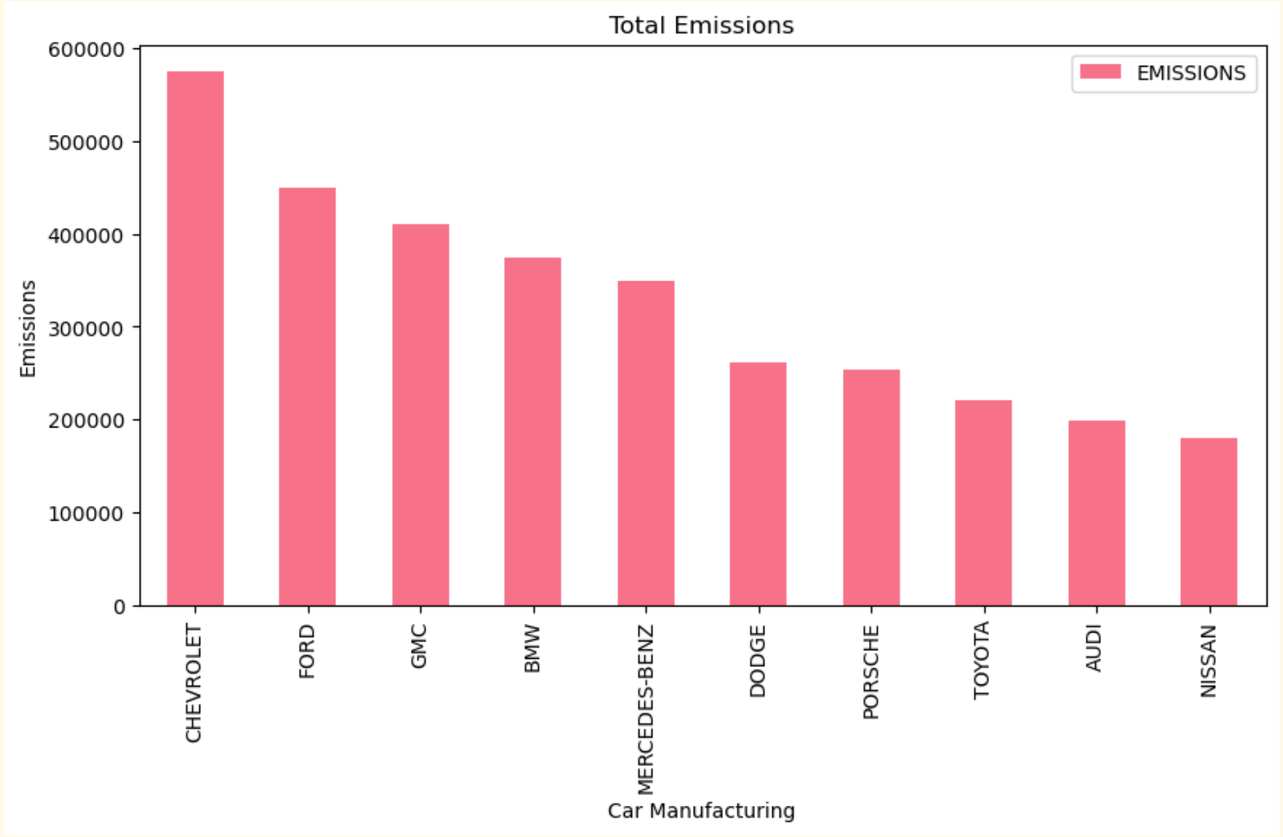
Car Model with highest consumption

This is a top 10 graph for car models with the highest fuel consumption, and Mustang is the car model with the highest fuel consumption at 1134.3 liters



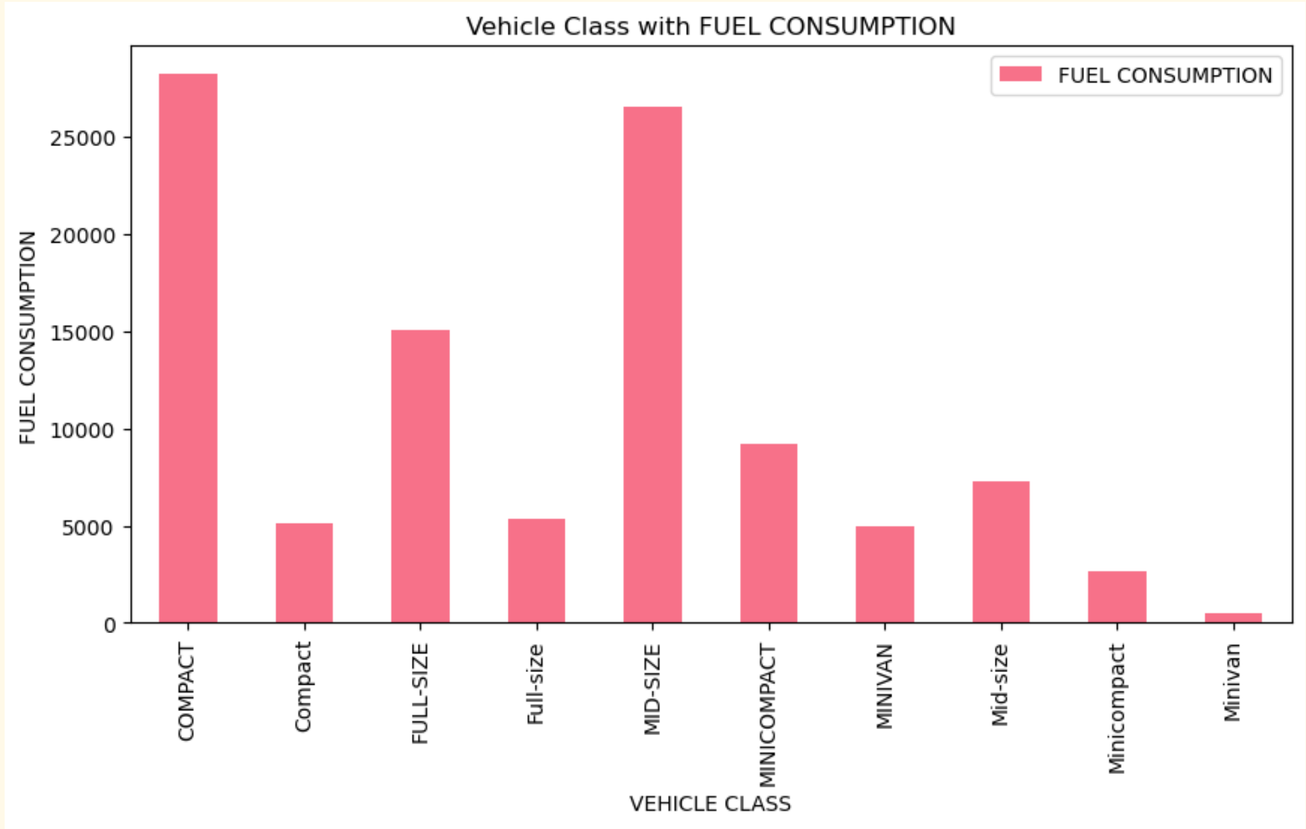
Car Manufacturing with highest Emissions

This is a top 10 graph for car manufacturers with the highest emissions, and
Chevrolet is the manufacturer with the highest emissions at 575099 g/km



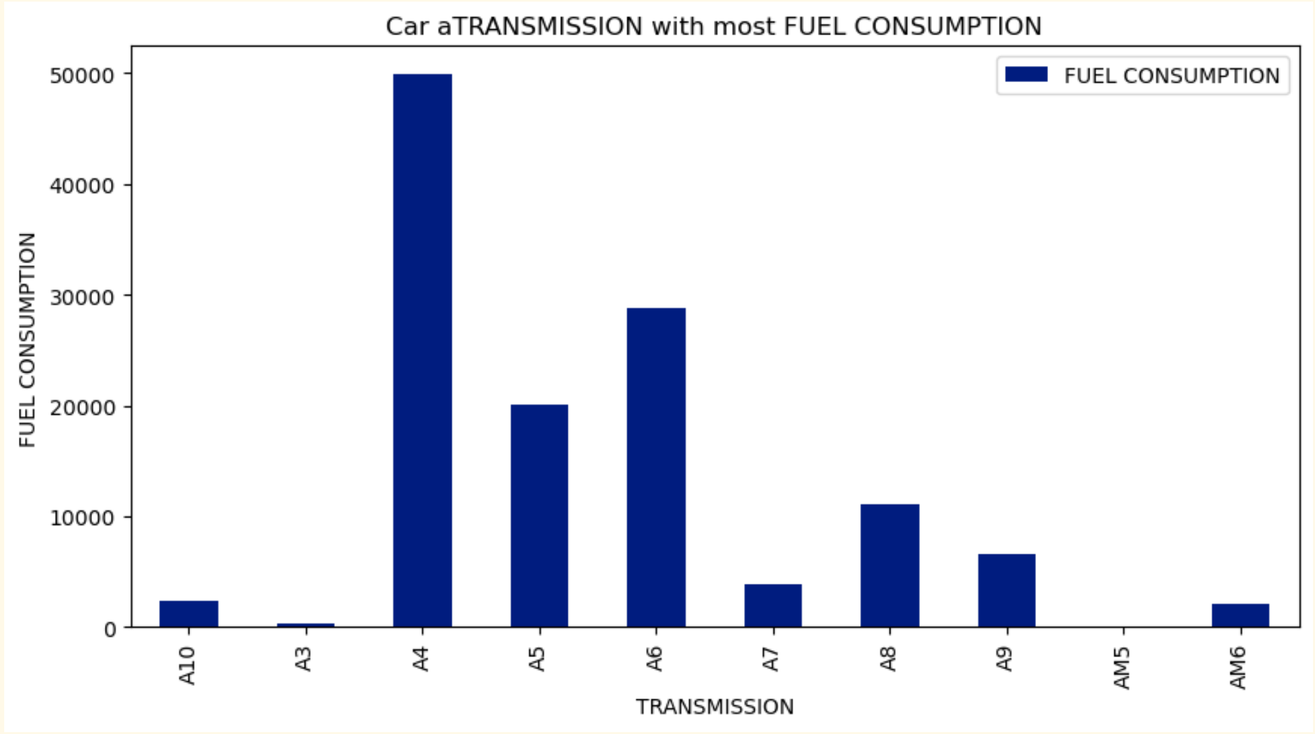
Vehicle Class Consumption

Compart is vehicle class with the most fuel consumption than other class at 28218.8 liters



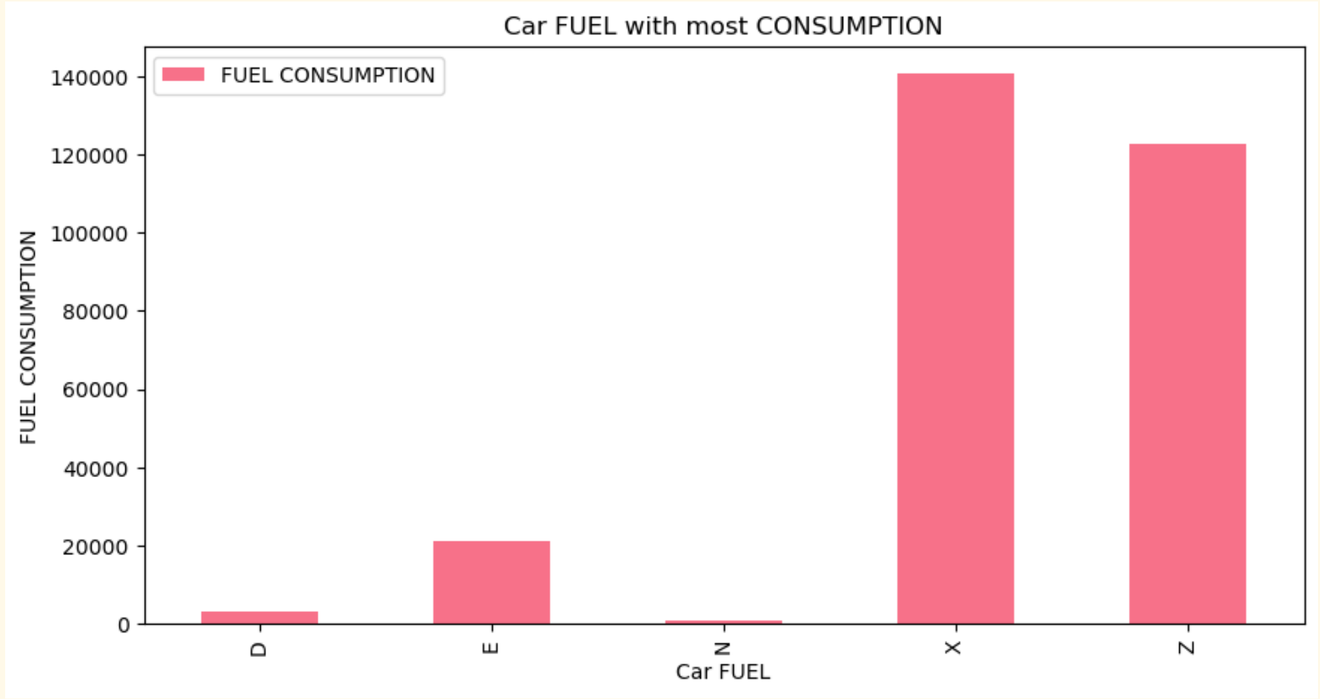
Car Transmission with fuel

This is a top 10 graph for car transmission with the highest fuel consumption, and A4 is the car transmission with the highest fuel consumption 49958.1 liters



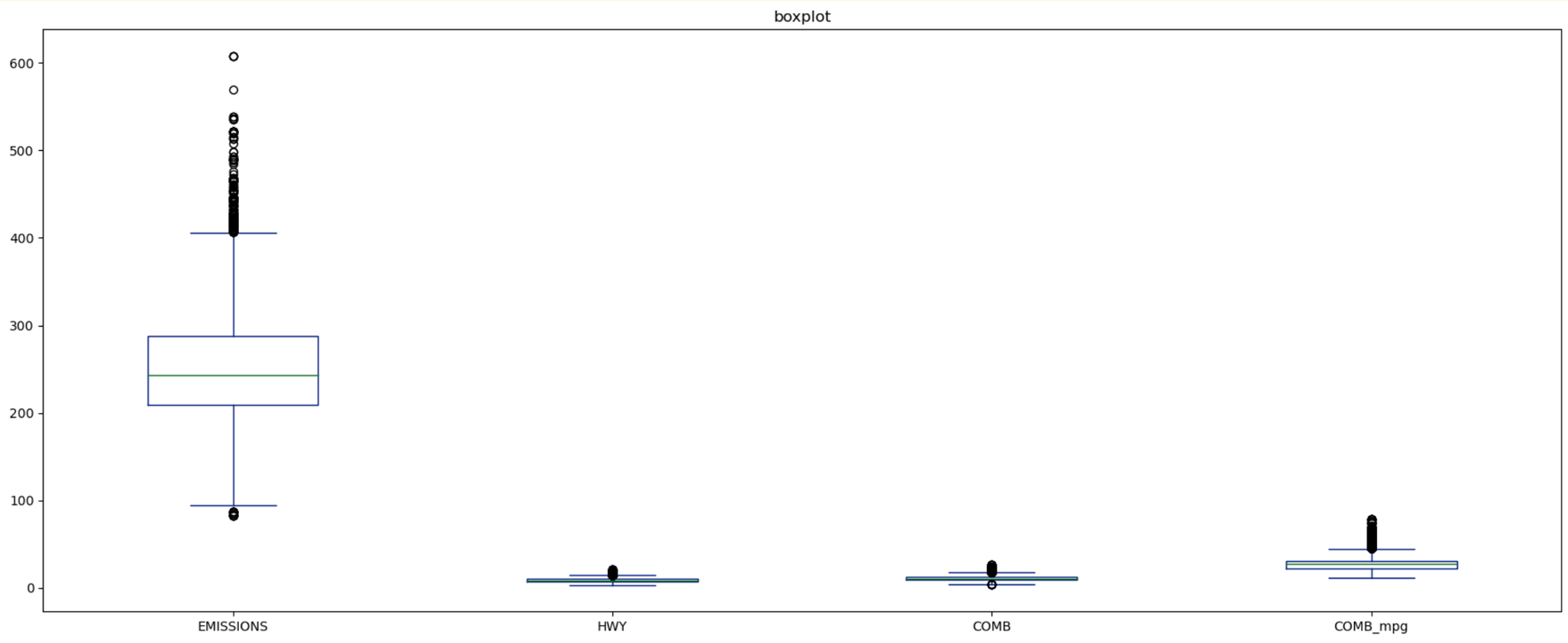
Car Fuel with Consumption

X is regular gasoline with the most consumption than other fuel at 140663.5 liters



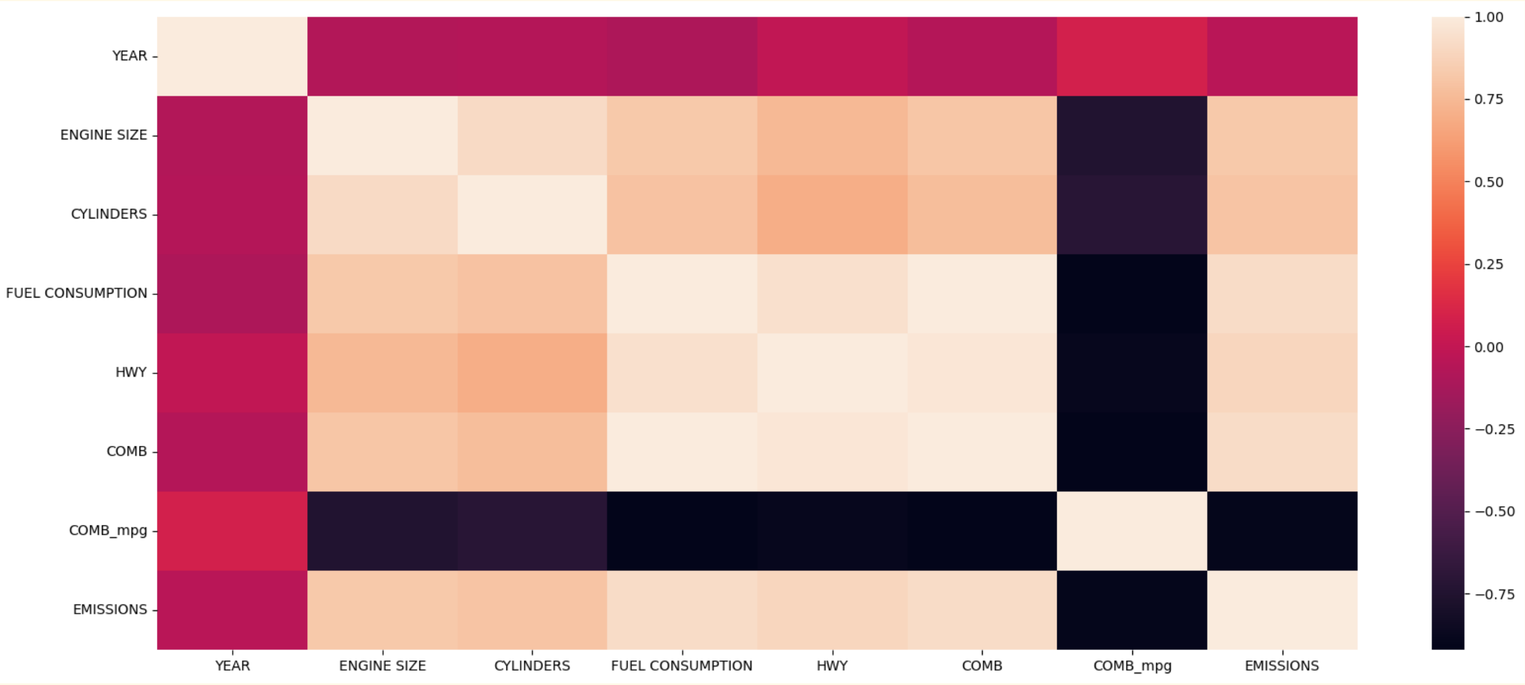
Boxplot

This boxplot to detect outliers data from some column we choose



Heatmap Correlation

This graph show column correlation from dataset. where HWY, COMB, and EMISSIONS have a high correlation with the target, which is fuel consumption



Data Preprocessing

Data Preprocessing

In this process, it is ensured that there are no missing and duplicate data, and the unique values in the dataset are checked.

```
# Checking if any rows are missing any data.
df.isnull().sum()
```

```
YEAR                0
MAKE                0
MODEL               0
VEHICLE CLASS       0
ENGINE SIZE         0
CYLINDERS           0
TRANSMISSION        0
FUEL                0
FUEL CONSUMPTION    0
HWY                 0
COMB                0
COMB_mpg            0
EMISSIONS           0
dtype: int64
```

```
df.duplicated()
```

```
0        False
1        False
2        False
3        False
4        False
         ...
22551    False
22552    False
22553    False
22554    False
22555    False
Length: 22556, dtype: bool
```

```
# Determine count of unique values for each
df.nunique()
```

```
YEAR                 23
MAKE                 52
MODEL              4242
VEHICLE CLASS        32
ENGINE SIZE          63
CYLINDERS             9
TRANSMISSION         30
FUEL                  5
FUEL CONSUMPTION    228
HWY                 152
COMB                192
COMB_mpg             59
EMISSIONS           358
dtype: int64
```

Outliers Handling

Deleting ouliers in every column using IQR, form **22556** to **21347** column

```
print(f'Jumlah Baris Sebelum Outlier Dihapus: {len(df)}')
filtered_entries = np.array([True] * len(df))
for col in['EMISSIONS','HWY','COMB','COMB_mpg']:

    q1=df[col].quantile(0.25)
    q3=df[col].quantile(0.75)
    iqr=q3-q1

    min_IQR = q1 - (1.5 * iqr)
    max_IQR = q3 + (1.5 * iqr)

    filtered_entries=((df[col]>=min_IQR) & (df[col]<=max_IQR)) & filtered_entries
    df=df[filtered_entries]

print(f'Jumlah Baris Sebelum Outlier Dihapus: {len(df)}')
```

```
Jumlah Baris Sebelum Outlier Dihapus: 22556
Jumlah Baris Sebelum Outlier Dihapus: 21347
```

Encoding

Encoding process to change the categorical feature to numerical feature using One Hot Encoding and Label Encoder

```
from sklearn.preprocessing import LabelEncoder

encoder = LabelEncoder()
df_encoded = df
df_encoded["MAKE"] = encoder.fit_transform(df["MAKE"])
```

```
categorical1 = ['MODEL', 'VEHICLE CLASS', 'TRANSMISSION', 'FUEL']
```

```
for cat in categorical1:
    onehots = pd.get_dummies(df[cat], prefix=cat)
    df = df.join(onehots)
```

```
df_clean = df.drop(['MODEL', 'VEHICLE CLASS', 'TRANSMISSION', 'FUEL'],axis=1)

df_clean.head(10)
```

Data Modelling
&
Model Evaluation

Split Data & Data Shape For Modelling

```
X = df_clean.drop(columns='FUEL CONSUMPTION')
y = df_clean['FUEL CONSUMPTION']
```

```
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=1)
```

```
print(f'X_train Shape: {(X_train.shape)}')
print(f'y_train Shape: {(y_train.shape)}')
print(f'X_test Shape: {(X_test.shape)}')
print(f'y_test Shape: {(y_test.shape)}')
```

```
X_train Shape: (17077, 4161)
y_train Shape: (17077,)
X_test Shape: (4270, 4161)
y_test Shape: (4270,)
```

mean_absolute_error result

Linear Regression Model

0.05533

Support Vector Regression Model

0.49352

Xgboost Model

0.06420

Conclusion

Conclusion

- Based on the results of the evaluation that has been carried out, we can see that the Linear Regression Model MAE value is 0.05533, the smaller the MAE value, the more accurate the model used.
- Emissions and fuel consumption are closely related, the higher the fuel consumption the higher the emissions produced. 2009 is year with highest fuel consumption and emissions
- Mustang is Car with most fuel consumptions than other model.
- A car with the specifications Engine Size=3.0, Cylinder=6, Transmission=A4, and Fuel=regular gasoline will consume more fuel.
- A car with the specifications Engine Size=4.1, Cylinder=2, Transmission=AM5, and Fuel=Natural Gas will consume less fuel.

Thank You!

https://linkedin.com/in/muhammad-randa-yandika

https://bit.ly/Randayandika_portofolio