Vehicle Population Prediction

1. Project Overview

- **Objective**:

  Our goal is to predict the vehicle population for 2024 using historical data from 2019 to 2023. By forecasting the number and types of vehicles on the road, we can expect the future demand for fuel, which can support strategic decisions, such as selecting optimal locations for gas stations or charging stations.

- **Data Sources**:

  For our project, we utilize the following data that are collected and provided by the Chevron Corporation. These include:

  - **training.xlsx**: Historical dataset used for model training and validation.
  - **scoring.xlsx**: Testing dataset used for evaluating model performance.
  - **data_dictionary.xlsx**: Documentation describing the variables in the dataset.
  - **submission_format.xlsx**: Defines the required structure for model predictions.
  - **submission_file.xlsx**: Template for submitting predictions and automatically calculating the model's RMSE (Root Mean Squared Error).
  -

- **Assessment**:

  To evaluate our model's performance, we will first train multiple models on the training dataset and validate them using cross-validation techniques. The optimal model will then be applied to the scoring dataset to generate predictions. Finally, we will calculate the **Root Mean Squared Error (RMSE)** by comparing our predictions with the public testing dataset, ensuring an objective measure of model accuracy.

2. Data Understanding

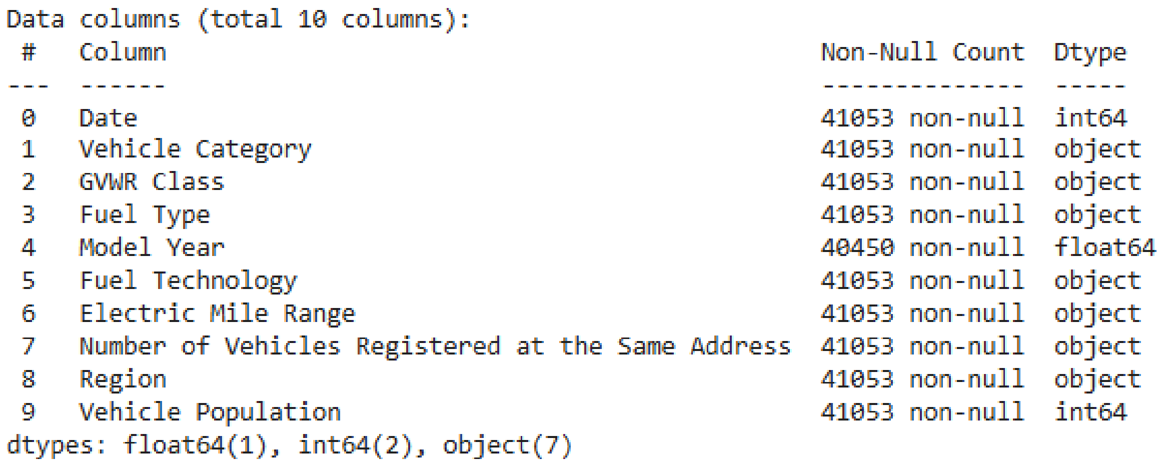
- **Examine the data**:

  By loading the dataset into a data frame, we perform an initial analysis to understand the structure, types, and distributions of the variables. Key aspects we examine include:

❖ **Data types:** Identifying numerical and categorical features.

- **Distribution:** Observing the range and central tendency of numerical variables.
- **Data size:** Assessing the volume of records available for training.

We identify that while most variables are non-null, some contain placeholder values such as **"Not Applicable"** or **"Unknown"**, which effectively represent missing data. These values require transformation into NaN (Not a Number) for proper handling. We also note a significant presence of categorical variables, which necessitates encoding before model training.



```
Data columns (total 10 columns):
 #   Column                                           Non-Null Count  Dtype  
---  ------                                           --------------  -----  
 0   Date                                             41053 non-null  int64  
 1   Vehicle Category                                 41053 non-null  object 
 2   GVWR Class                                       41053 non-null  object 
 3   Fuel Type                                        41053 non-null  object 
 4   Model Year                                       40450 non-null  float64
 5   Fuel Technology                                  41053 non-null  object 
 6   Electric Mile Range                              41053 non-null  object 
 7   Number of Vehicles Registered at the Same Address 41053 non-null  object 
 8   Region                                           41053 non-null  object 
 9   Vehicle Population                               41053 non-null  int64  
dtypes: float64(1), int64(2), object(7)
```

- **Exploratory Data Analysis (EDA)**: Conduct an EDA to explore relationships between the variables (e.g., vehicle types, model years, and fuel types) and to identify patterns or trends in the data.

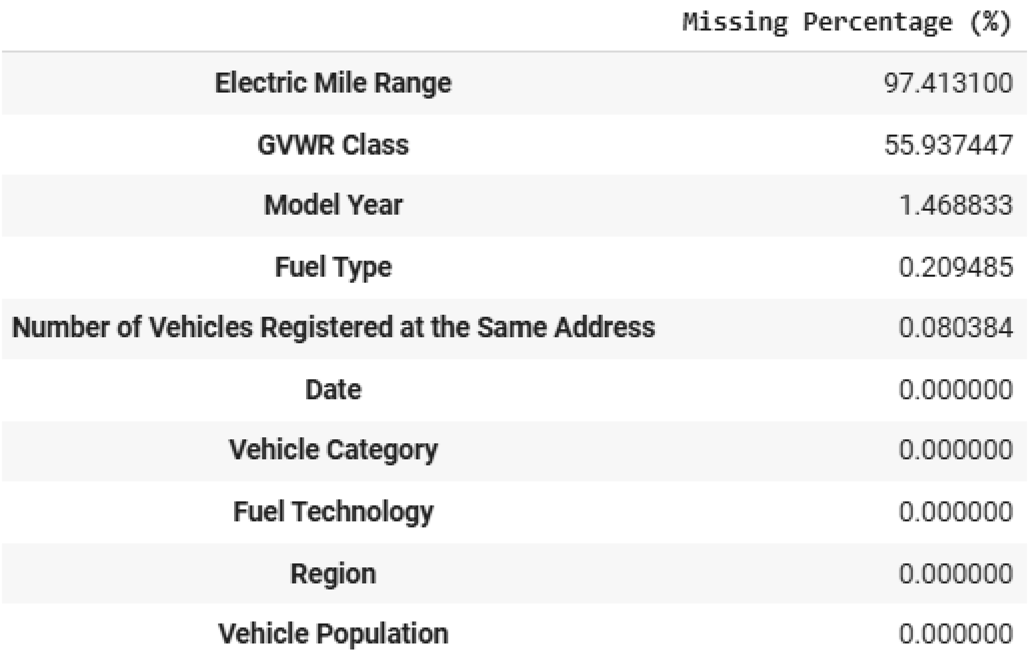
3. Data Preprocessing

- **Handling Missing Data**:

After replacing unclear placeholder values with NaN, we assess the proportion of missing values in each column. Our analysis reveals:

  - **"Electric Mile Range"** has an extremely high percentage of missing values and is removed from the dataset due to its lack of predictive utility.
  - **"GVWR Class"** contains approximately 50% missing data, requiring imputation.
  - **"Region"** is removed, as all values are identical, contributing no additional information.

To handle missing values in other columns, we employ **K-Nearest Neighbors (KNN) imputation**, which estimates missing values based on the most similar records in the dataset. This technique ensures the preservation of meaningful patterns within the data.



	Missing Percentage (%)
Electric Mile Range	97.413100
GVWR Class	55.937447
Model Year	1.468833
Fuel Type	0.209485
Number of Vehicles Registered at the Same Address	0.080384
Date	0.000000
Vehicle Category	0.000000
Fuel Technology	0.000000
Region	0.000000
Vehicle Population	0.000000

- **Feature Engineering**:

  To enhance predictive performance, we add a new column:**"Year Diff"** = **(Current Year - Model Year)**: This serves as an indicator for vehicle age, which we hypothesize to be a stronger predictor of vehicle retention and usage trends than the raw model year alone.

- **Scaling and Encoding:** We standardize the numerical features with z-score normalization and use one-hot encoding to convert categorical variables into numerical format.

4. Model Selection

I initially started with Decision Tree and Random Forest, and finally, I moved to XGBoost. XGBoost operates as an ensemble of decision trees, where each tree sequentially corrects the errors of the previous ones. Instead of making direct predictions, each tree learns to predict the residual errors (the difference between actual and predicted values) and updates the model iteratively. The final prediction is the sum of all tree contributions, weighted by a learning rate, ensuring both accuracy and efficiency.

5. Model Performance

After the model was changed to XGBoost, the RMSE and CV of the training dataset decreased significantly. To fine-tune the model, I used Grid Search and selected the following hyperparameters: n_estimators=100, learning_rate=0.1, max_depth=8, random_state=42, colsample_bytree=0.7, and subsample=0.7.

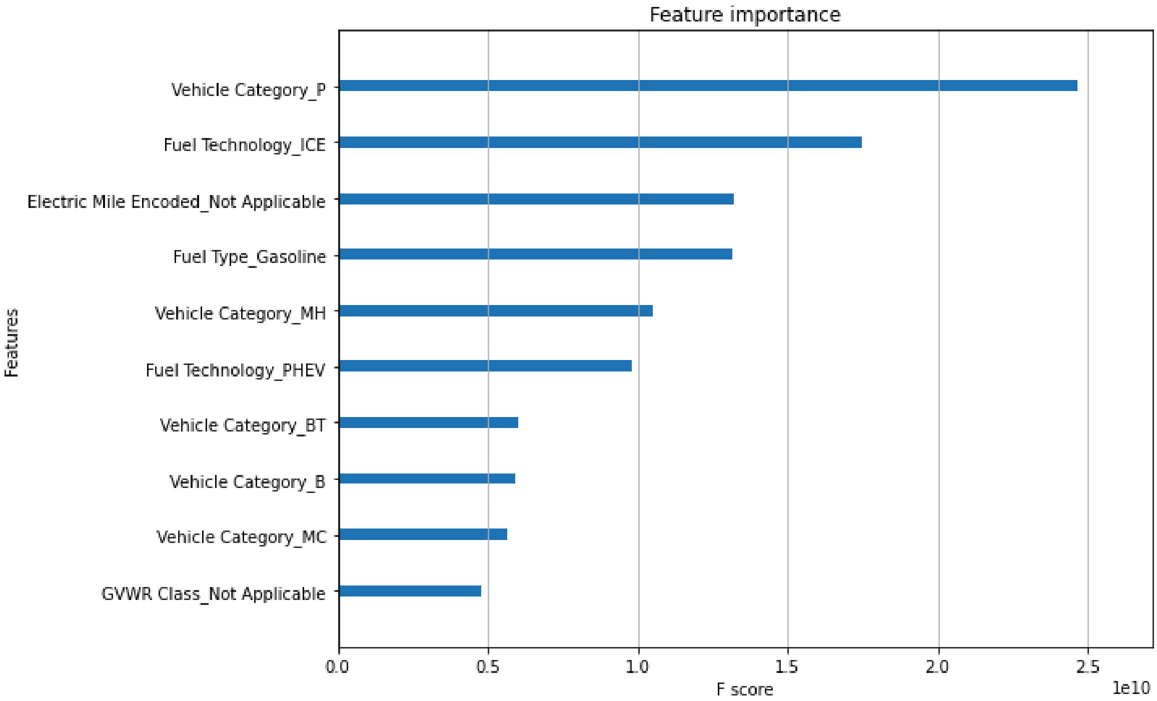
As for our model performance on test data, we used our XGBoost model to make predictions on scoring.xlsx and obtained an RMSE of approximately 6000. This means that, on average, the model's predictions deviate by 6000 from the actual values, with the original variable ranging from 1 to 300,000. This deviation accounts for approximately 2% of the total range.

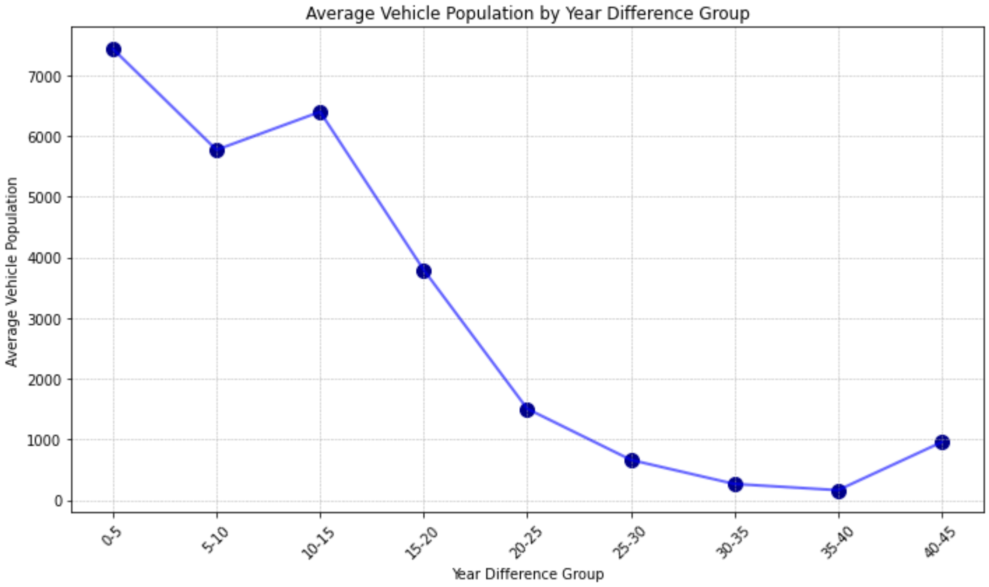
6. Model Evaluation

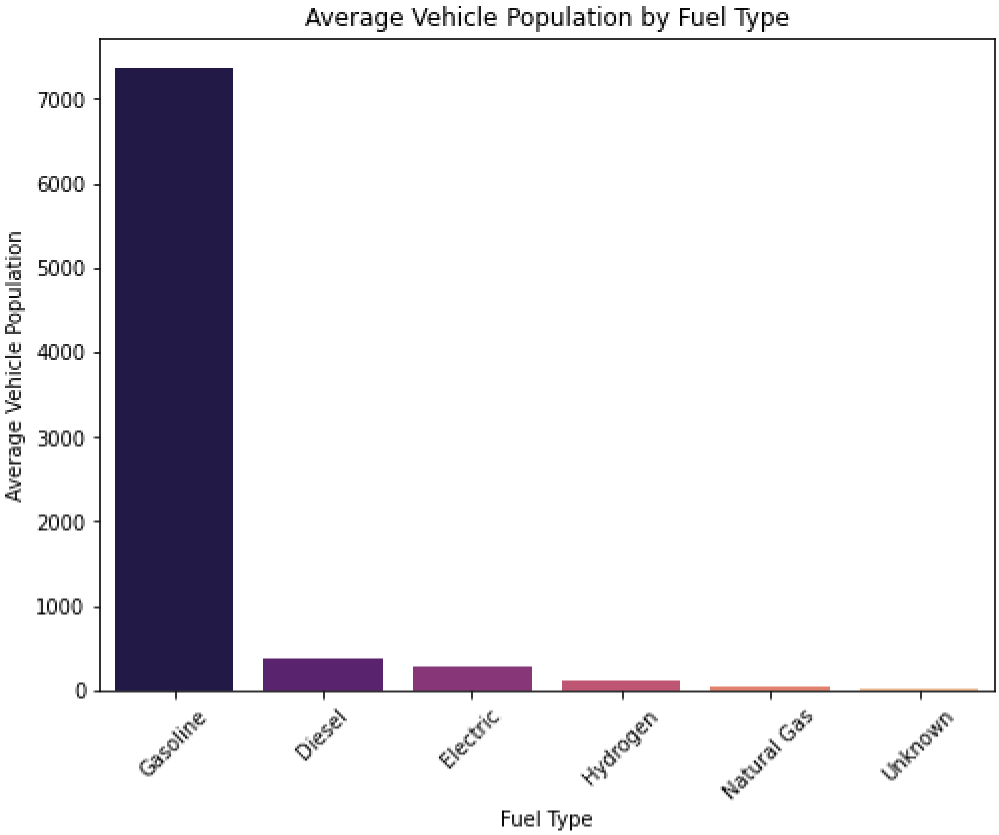
Aside from fine-tuning the model, we made improvements through feature engineering. These included creating a new feature, **"Vehicle Age"** (calculated as Date - Model Year), performing variable categorization, and filling missing values. These enhancements collectively contributed to improved predictive performance on the test data.

7. Model Interpretation

To analyze the impact of variables on predictions, I generated a plot highlighting the top 10 most influential features. The most impactful variable is Vehicle Category, indicating that the classification of a vehicle plays the largest role in the model's predictions. Other key variables include Fuel Technology, Electric Mile Range (if applicable), and GVWR Class (if applicable).







8. Conclusion and Insights

On average, the model's predictions deviate by 6000 from the actual values, 2% deviation within the original variable's range of 1 to 300,000.

New Variable (Vehicle Age) helps enhance model performance, while Vehicle Category plays the most significant role in prediction.