Logo

Description automatically generated

**College of Professional Studies**

**Northeastern University San Jose**

**MPS Analytics**

**Course: ALY6020 – Predictive Analytics**

**Module 2 Project:** Building the Car of the Future

**Submitted on:**

Oct 7th , 2023

**Submitted to:**  **Submitted by:**

Professor: BEHZAD AHMADI BHAGYASHRI KADAM

**INTRODUCTION**

**Given Problem Statement:**

A car manufacturer known for making large automobiles is struggling with sales and has asked for your help in designing an energy efficient car. Using data gathered, determine which attributes may contribute to higher gas mileage so that they can design a more fuel efficient automobile.

**Understanding the Dataset**

The given dataset comprises a total of **398 records and 8 columns/attributes** each representing an individual car and detailing their respective attributes such as MPG, Cylinders, Displacement, and so on. The target variable for this dataset is miles per gallon (MPG) based on the attributes if the vehicle.

Below are the datatypes of the attributes:

A screenshot of a computer

Description automatically generated

**PART 1 : DATA CLEANING**

* **Checking the number of missing values for each Attribute in the dataset**

A screenshot of a computer

Description automatically generated

From the above output, we can observe that there are no missing values for any of the attributes.

Further, lets inspect the “horsepower” attribute for its unique values and we observe that there is a “?” present as well as one of the values

A screenshot of a computer code

Description automatically generated

After imputing the “?” with NaN values , 6 missing values were identified which were then imputed with the mean of the “horsepower” attribute leading to removal of all missing values.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer code

Description automatically generated

**DESCRIPTIVE CHARACTERISTICS OF THE DATASET**

**A screenshot of a computer

Description automatically generated**

The above displays the statistical information for several numerical columns related to car attributes.

The key statistical information for the numerical columns in the dataframe is as follows:

* MPG (Miles Per Gallon): This attribute indicates the fuel efficiency of the cars. With a mean value of approximately 23.51, it suggests that on average, cars in this dataset can travel about 23.51 miles per gallon of fuel. The range goes from a low of 9 MPG to a high of 46.6 MPG, indicating a wide variety of fuel efficiencies among the cars.
* Cylinders: The average number of cylinders in the cars is about 5.45, but the data ranges from cars with 3 cylinders to those with 8, suggesting a mix of engine sizes and power.
* Horsepower: With a mean value of approximately 104.47, this attribute gives an idea of the average power of the cars in the dataset. The range from 46 to 230 indicates the presence of both low-powered and high-powered cars.
* Weight: The average weight of the cars is around 2970.42 units (likely pounds or kilograms). This attribute has a wide range, indicating the dataset includes both lightweight and heavyweight vehicles.
* US Made: This is a binary attribute, with a mean of 0.63. This suggests that around 63% of the cars in the dataset are made in the US.
* Acceleration, Model Year: These attributes provide insights into the average acceleration capability and the model year of the cars, respectively. The data suggests a diverse range of car models from different years with varied acceleration capabilities.

**DATA VISUALIZATIONS**

In the EDA, we will start by understanding the distribution of numerical attributes using histogram as below :

A screenshot of a graph

Description automatically generated

A collage of different graphs

Description automatically generated with medium confidence

A screenshot of a graph

Description automatically generated

Below are some key insights from the from the above pairplot –

**MPG (Miles Per Gallon):** The distribution appears to be slightly right skewed, with a peak around 20-25 MPG. There seems to be a negative correlation, indicating that cars with more cylinders tend to have lower MPG (less fuel-efficient).

-Also, MPG vs Displacement & Horsepower show a negative trend, suggesting that cars with higher displacement & horsepower generally have lower fuel efficiency.

-For MPG vs Weight shows a clear negative correlation indicating that heavier cars tend to be less fuel-efficient.

-For MPG vs Acceleration: The relationship is less clear, but there might be a slight positive correlation suggesting that cars with better acceleration might be slightly more fuel-efficient.

**Cylinder vs horsepower, Displacement & weight:** There’s a positive correlation, indicating that cars with more cylinders tend to have higher displacement and horsepower. Also, cars with more cylinders seem to be heavier. Also, cars with more cylinders seem to be heavier.

**Displacement vs horsepower & weight:** These 3 variables also show positive correlations with each other. Cars with higher displacement tend to have more horsepower and are heavier.

**Acceleration:** The relationships between acceleration & other variables are less distinct. However, there might be a slight negative correlation between acceleration and horsepower, suggesting that cars with higher horsepower might have slightly lower acceleration.

Further, the distribution of cylinders is with peaks at 4, 6, and 8 cylinders. The majority of cars seem to have 4 cylinders, followed by 8 and then 6.

For displacement, distribution is somewhat bimodal, with peaks in the lower and upper ranges. This suggests the presence of both small and large engine cars.

For horsepower, the distribution is right-skewed, with a peak in the lower horsepower range. There are also a few cars with very high horsepower, but they are less common.

For weight, the distribution appears to be slightly right skewed, with a peak around the mid-weight range. This suggests that most cars are of average weight, with fewer very light or very heavy cars.

For acceleration, the distribution seems to be approximately normal, centered around 15-16 units (likely seconds). This indicates that most cars have average acceleration capabilities.

For Model year , the distribution is uniform across the years, suggesting that the dataset contains a balanced number of cars from each model year.

**Correlation Map**

A screenshot of a graph

Description automatically generated

The above correlation map helps us to understand how different car attributes relate to each other.

Below are some key insights for correlation of MPG with the rest of the attributes:

MPG vs. Cylinders: There's a strong negative correlation of approximately -0.78, indicating that cars with more cylinders tend to have lower MPG (less fuel-efficient).

MPG vs. Displacement: A strong negative correlation of about -0.81 suggests that cars with higher displacement generally have lower fuel efficiency.

MPG vs. Horsepower: A negative correlation of around -0.78 indicates that cars with higher horsepower tend to have lower MPG.

MPG vs. Weight: A strong negative correlation of approximately -0.83 suggests that heavier cars tend to be less fuel-efficient.

MPG vs. Acceleration: A slight negative correlation of about -0.42.

MPG vs. Model Year: A positive correlation of around 0.58 indicates that newer models tend to be more fuel-efficient.

**Part 2 : Model Building & Analysis**

Before building the model, I have divided the data in to target variable (MPG) and the rest if the features and then split the data in to training set (80%) & test set (20%).

Thereafter, I have added a constant to the feature matrix . The sm.add\_constant() function adds a column of ones to the X\_train dataframe. This column represents the constant term or intercept (�0β0​) in the linear regression equation. By adding this constant, we ensure that the regression model has an intercept.

Further, I have initialized the Ordinary Least squares (OLS) regression model using sm.OLS() function which initializes the OLS regression model with the dependent variable y\_train and the independent variables X\_train\_ols (which includes the added constant).

After initializing the model, I have fit the model using the fit() which estimates the coefficients of the OLS regression model using the training data.

And Finally, once the model is fitted , I have evaluated the model’s performance using summary() method which provides a comprehensive summary of the regression results, including coefficients, R-squared values, p-values, and other statistics.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a data sheet

Description automatically generated

A number on a white background

Description automatically generated

*Model results summary*

Below are some key insights from the above results summary:

1. The R-squared value is 0.819, which means that approximately 81.9% of the variability in the MPG can be explained by the independent variables in the model. This is a relatively high R-squared value, indicating a good fit of the model to the data.
2. The adjusted R-squared value is 0.815, which adjusts the R-squared value based on the number of predictors in the model. It's very close to the R-squared value, suggesting that most of the variables contribute to the explanation of the variance in MPG.
3. **Coefficients**:

Cylinders: The coefficient is -0.1744, but it's not statistically significant (p-value > 0.05). This suggests that the number of cylinders might not be a strong predictor for MPG in this model.

Displacement: Positive coefficient of 0.0195 and is statistically significant. This indicates that as displacement increases, MPG tends to increase slightly.

Horsepower: The coefficient is -0.0139, but it's not statistically significant.

Weight: Negative coefficient of -0.0070 and is statistically significant. This suggests that as the weight of the car increases, the MPG decreases.

Acceleration: The coefficient is 0.0733, but it's not statistically significant.

Model Year: Positive coefficient of 0.8230 and is statistically significant. This indicates that newer car models tend to have higher MPG.

US Made: Negative coefficient of -2.7904 and is statistically significant. This suggests that US-made cars tend to have lower MPG compared to non-US-made cars.

1. **Significance**: The p-values associated with each coefficient determine the significance of that variable in the model and with p-values less than 0.05 are considered statistically significant. **In this model, Displacement, Weight, Model Year, and US Made are statistically significant predictors.**
2. **Building the Proper Car:**

Based on the significant attributes from the regression model, here's how they can help in building a proper car with good MPG:

Displacement: While the model suggests a positive relationship between displacement and MPG, it's essential to consider other factors like horsepower, torque, and the type of fuel used. A balanced approach to engine size and performance can lead to better fuel efficiency.

Weight: Reducing the car's weight can lead to better MPG. This can be achieved by using lightweight materials, optimizing the car's design, and removing unnecessary components.

Model Year: Incorporating the latest technology and design improvements can lead to better fuel efficiency. This includes advancements in engine technology, aerodynamics, and fuel management systems.

US Made vs. Non-US Made: While the model suggests that non-US-made cars tend to be more fuel-efficient, it's essential to consider the specific design and manufacturing practices that lead to this outcome. Adopting best practices from global car manufacturers can help improve MPG.

In conclusion, to build a car with optimal MPG, focus on optimizing the engine's displacement, reducing the car's weight, incorporating the latest technological advancements, and adopting best practices from both US and global car manufacturers.

**Part 3 : Model Optimization**

I have further refined and build the model using only the statistically significant car attributes. The refined model, which only includes the statistically significant predictors **(Displacement, Weight, Model Year, and US Made)**, explains approximately **81.7%** of the variability in MPG which is the R2 value.

Weight and US Made attributes have a negative impact on MPG. Reducing the car's weight and considering manufacturing practices from global car manufacturers can lead to better MPG.

Model Year has a positive impact on MPG, indicating that newer car models are more fuel-efficient.

Displacement has a borderline significant positive impact on MPG. However, this relationship should be interpreted with caution, as larger engines (higher displacement) are often less fuel-efficient.

A screenshot of a computer program

Description automatically generated

A screenshot of a cell

Description automatically generated

In conclusion we can say the below observing the refined model results—

To achieve higher MPG, focus on reducing the car's weight, incorporating the latest technological advancements, and considering best practices from both US and global car manufacturers. The Model Year attribute plays a crucial role, suggesting that advancements over the years have led to more fuel-efficient cars.

**Checking for Multi-Collinearity for the refined model:**

**A screenshot of a computer

Description automatically generated**

From the above VIF for each feature from the refined model, we can observe that Displacement and Weight have VIF values close to or above the threshold of 5, suggesting potential multicollinearity issues. This means that these predictors might be correlated with each other or with other predictors in the model. This multicollinearity can inflate the variance of the regression coefficients, making them unstable and harder to interpret.

Model Year and US Made have VIF values well below the threshold, indicating that they are not highly correlated with the other predictors in the model.

Further, in order to decide whether if we need to drop these attributes depends on their correlation with the target variable (MPG).

A screenshot of a computer code

Description automatically generated

From above, we can understand that both of these attributes (Displacement and Weight) have strong negative correlations with MPG, suggesting that they are significant predictors in determining a car's fuel efficiency. When designing a car for higher MPG, it would be beneficial to consider reducing both the engine displacement and the overall weight of the vehicle.

Lets evaluate the above model on the test set :

A screenshot of a computer program

Description automatically generated

The refined OLS regression model performs well on the test set, with an R-squared score of **84.06%.** This suggests that the model can explain a significant portion of the variability in the MPG based on the selected car attributes (Displacement, Weight, Model Year, and US Made).

The high R-squared score on the test set confirms the model's ability to generalize well to new, unseen data.

In conclusion, the refined OLS regression model provides a reliable tool for predicting a car's MPG based on its attributes. The selected attributes play a significant role in determining a car's fuel efficiency, and the model can be used to guide decisions in car design and manufacturing to achieve higher MPG.

The scatter plot visualizes the relationship between the predicted MPG values and the actual MPG values for the test set using the refined OLS regression model.

**Scatter plot of Predicted vs Actual MPG :**

The below scatter plot visualizes the relationship between the predicted MPG values and the actual MPG values for the test set using the refined OLS regression model.

A graph with blue dots and a red line

Description automatically generated

Line of Best Fit: The red line represents the line of best fit, where the predicted MPG values would be equal to the actual MPG values. Ideally, all points would lie on this line, indicating perfect predictions.

The scatter points represent individual observations in the test set. The x-axis shows the actual MPG values, while the y-axis shows the predicted MPG values.

Distribution: Most of the scatter points are clustered around the line of best fit, indicating that the model's predictions are generally close to the actual values. This suggests that the refined OLS regression model is performing well on the test set.

Variability: There is some variability in the predictions, especially for cars with higher MPG values. This is evident from the scatter points that deviate from the line of best fit. However, the majority of the predictions are reasonably accurate.

Consistency: The model seems to be consistent across different ranges of MPG values, as the scatter points are distributed around the line of best fit throughout the plot.

In conclusion, the scatter plot provides a visual confirmation of the model's performance on the test set. The close alignment of most scatter points with the line of best fit indicates that the refined OLS regression model is capable of making accurate predictions for MPG based on the selected car attributes.

**Conclusion**

Throughout our exploration and analysis of the car dataset, we aimed to build a linear regression model to accurately predict miles per gallon (MPG) based on various attributes of a vehicle.

1. Data Exploration and Visualization: Initial visualizations, such as pair plots and correlation heatmaps, provided insights into the relationships between different attributes and their individual distributions. We identified strong correlations between certain attributes like `Displacement`, `Weight`, and `MPG`.

2. Model Building and Refinement: The initial OLS regression model incorporated all available attributes. However, to enhance the model's performance and interpretability, we refined it by selecting only statistically significant attributes. This refined model included `Displacement`, `Weight`, `Model Year`, and `US Made`.

3. Multicollinearity: We addressed concerns of multicollinearity by analyzing Variance Inflation Factor (VIF) values. While `Displacement` and `Weight` exhibited high VIF values, suggesting potential multicollinearity, their strong individual correlations with `MPG` made them crucial for the model. Strategies like combining attributes or using regularization techniques were discussed as potential solutions.

4. Model Evaluation: The refined model's performance on the test set was commendable, with an R-squared score of 84.06%, indicating that it could explain approximately 84% of the variability in `MPG`. Scatter plots further visualized the accuracy of the model's predictions, showing a close alignment between predicted and actual MPG values.

5. Significant Attributes: Among the attributes, `Displacement` and `Weight` stood out as significant predictors negatively influencing MPG. In contrast, `Model Year` had a positive impact, suggesting that newer models tend to be more fuel-efficient. The `US Made` attribute also played a role, indicating potential differences in manufacturing practices or consumer preferences between domestic and imported vehicles.

In summary, our refined linear regression model offers a robust tool for predicting a car's MPG based on key attributes. For stakeholders, such as car manufacturers or consumers, understanding these relationships can guide decisions in car design, manufacturing, and purchasing. By focusing on reducing engine displacement, weight, and incorporating advancements from newer model years, one can aim to achieve higher fuel efficiency and more environmentally friendly vehicles.

**References**

1. <https://www.analyticsvidhya.com/blog/2020/03/what-is-multicollinearity/>
2. <https://georgepaskalev.medium.com/first-steps-to-understand-and-improve-your-ols-regression-part-1-dc7a8e911684>
3. <https://northeastern.instructure.com/courses/160443/pages/lesson-2-4-interpreting-linear-regression-results?module_item_id=9500042>

**Appendix**

|  |
| --- |
| import numpy as np  import pandas as pd  import matplotlib.pyplot as plt  import seaborn as sns  #Loading data#  df = pd.read\_csv("/Users/bhagyashrikadam/Documents/NEU\_ASSIGNMENTS/ALY6020/Module2/car.csv")  df.shape  print(df.dtypes)  df.head(10)  ## PART1 ##  ## Data Cleaning ##  # Checking the number of missing values ##  missing\_values = df.isnull().sum()  print(missing\_values)  # Inspect the unique values in the 'Horsepower' column  unique\_horsepower = df['Horsepower'].unique()  unique\_horsepower  # Replace '?' with NaN in the 'Horsepower' column  df['Horsepower'].replace('?', value=np.nan, inplace=True)  # Convert 'Horsepower' to numeric data type  df['Horsepower'] = pd.to\_numeric(df['Horsepower'])  # Check for missing values in the dataset  missing\_values = df.isnull().sum()  missing\_values  # Impute missing values in the 'Horsepower' column with the mean value of the column  df['Horsepower'].fillna(df['Horsepower'].mean(), inplace=True)  # Verify if there are any missing values left in the dataset  missing\_values\_after\_imputation = df.isnull().sum()  missing\_values\_after\_imputation  # Descriptive Statistics #  df.describe()  ######## EDA ########  import matplotlib.pyplot as plt  import seaborn as sns  # Set the style of the visualization  sns.set(style='whitegrid')  # Plot pairplot to visualize the relationships between variables  sns.pairplot(df)  plt.show()  # 3. Correlation Heatmap for Numerical Features  # Compute the correlation matrix  corr\_matrix = df.corr()  # Plot heatmap to visualize the correlation between variables  plt.figure(figsize=(10, 8))  sns.heatmap(corr\_matrix, annot=True, cmap='coolwarm', fmt='.2f')  plt.title('Correlation Matrix')  plt.show()  ## mpg --> acceleration , model year |
| ## PART 2 : MODEL BUILDING & ANALYSIS ###  ## Linear Regression Model ##  from sklearn.model\_selection import train\_test\_split  from sklearn.linear\_model import LinearRegression  from sklearn.metrics import mean\_squared\_error, r2\_score  # Selecting the features (independent variables) and the target (dependent variable)  X = df.drop('MPG', axis=1) # Features  y = df['MPG'] # Target  # Splitting the data into training and testing sets  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)  import statsmodels.api as sm  #Adding a constant to the feature matrix for OLS with all features  X\_train\_ols = sm.add\_constant(X\_train)  # Initializing the OLS model with all features  ols\_model = sm.OLS(y\_train, X\_train\_ols)  # Fitting the OLS model with all features  ols\_results = ols\_model.fit()  # Displaying the OLS model summary with all features  ols\_results.summary()  # Selecting only the statistically significant features identified by the OLS model  significant\_features = ['Displacement', 'Weight', 'Model Year', 'US Made']  # Creating a new feature matrix (X) with only the significant features  X\_refined = df[significant\_features]  y\_refined = df['MPG'] # Target remains the same  # Splitting the refined data into training and testing sets  X\_train\_refined, X\_test\_refined, y\_train\_refined, y\_test\_refined = train\_test\_split(X\_refined, y\_refined, test\_size=0.2, random\_state=42)  # Adding a constant to the refined feature matrix for OLS with significant features  X\_train\_ols\_refined = sm.add\_constant(X\_train\_refined)  # Initializing the OLS model with significant features  ols\_model\_refined = sm.OLS(y\_train\_refined, X\_train\_ols\_refined)  # Fitting the OLS model with significant features  ols\_results\_refined = ols\_model\_refined.fit()  # Displaying the OLS model summary with significant features  ols\_results\_refined.summary()  ## Calculating Multicollinearity##  from statsmodels.stats.outliers\_influence import variance\_inflation\_factor  # Calculating the Variance Inflation Factor (VIF) for each predictor variable in the refined model  vif\_data = pd.DataFrame()  vif\_data['Feature'] = X\_train\_ols\_refined.columns  vif\_data['VIF'] = [variance\_inflation\_factor(X\_train\_ols\_refined.values, i) for i in range(X\_train\_ols\_refined.shape[1])]  # Displaying the VIF for each predictor variable  vif\_data  # Calculating the correlation of 'Displacement' and 'Weight' with 'MPG'  displacement\_mpg\_corr = X\_train\_refined['Displacement'].corr(y\_train\_refined)  weight\_mpg\_corr = X\_train\_refined['Weight'].corr(y\_train\_refined)  # Displaying the correlation of 'Displacement' and 'Weight' with 'MPG'  {'Displacement-MPG Correlation': displacement\_mpg\_corr, 'Weight-MPG Correlation': weight\_mpg\_corr}  ## Evaluating model on the test set##  import statsmodels.api as sm  from sklearn.metrics import mean\_squared\_error, r2\_score  # Adding a constant to the refined test feature matrix for OLS  X\_test\_ols\_refined = sm.add\_constant(X\_test\_refined)  # Predicting the MPG on the test set using the OLS model  y\_pred\_ols\_refined = ols\_results\_refined.predict(X\_test\_ols\_refined)  # Calculating the Mean Squared Error (MSE) on the test set for the OLS model  mse\_ols\_refined = mean\_squared\_error(y\_test\_refined, y\_pred\_ols\_refined)  # Calculating the R-squared score on the test set for the OLS model  r2\_ols\_refined = r2\_score(y\_test\_refined, y\_pred\_ols\_refined)  # Displaying the model evaluation metrics on the test set for the OLS model  print('OLS Mean Squared Error:', mse\_ols\_refined)  print('OLS R-squared Score:', r2\_ols\_refined) |