Question 1: Clean all numerical features so that they can be used in training algorithms. For instance, back_legroom feature is in object format containing both numerical values and text. Extract numerical values (equivalently eliminate the text) so that the numerical values can be used as a regular feature.

Answer: I first identify the columns with mixed numeric and text values, and then create a function called extract_numeric_value. This function checks if a value is a text (string), and if so, it uses a regular expression to find and extract numeric patterns. The extracted numeric values are converted to floating-point numbers. If a valid numeric value is not found, the function returns None. I then apply this extract_numeric_value function to clean the specified columns in both the traindf and testdf datasets. The result is that these columns will contain only the extracted numeric values, making them ready for analysis or modeling.

Question 2: Create at least 5 new features from the existing numerical variables which contain multiple items of information, for example you could extract maximum torque and torque rpm from the torque variable

Answer: i created five features The maximum torque which indicates the maximum torque of cars, legroom ratio to show the ratio of rear legroom to front legroom, fuel efficiency of the average of city and highway fuel economy to reflects the average of city and highway fuel economy, engine power ratio as horsepower/engine displacement to denotes the ratio of horsepower to engine displacement, car sizes as product of length, width and height

Question 3: Impute missing values for all features in both the training and test datasets

Answer: Missing values in numerical features are imputed with the median value, ensuring central tendencies. For categorical features, imputation with the mode (most frequent value) ensures representative values. This data preprocessing enhances dataset completeness for analysis and modeling

Question 4: Encode all categorical variables appropriately as discussed in class.

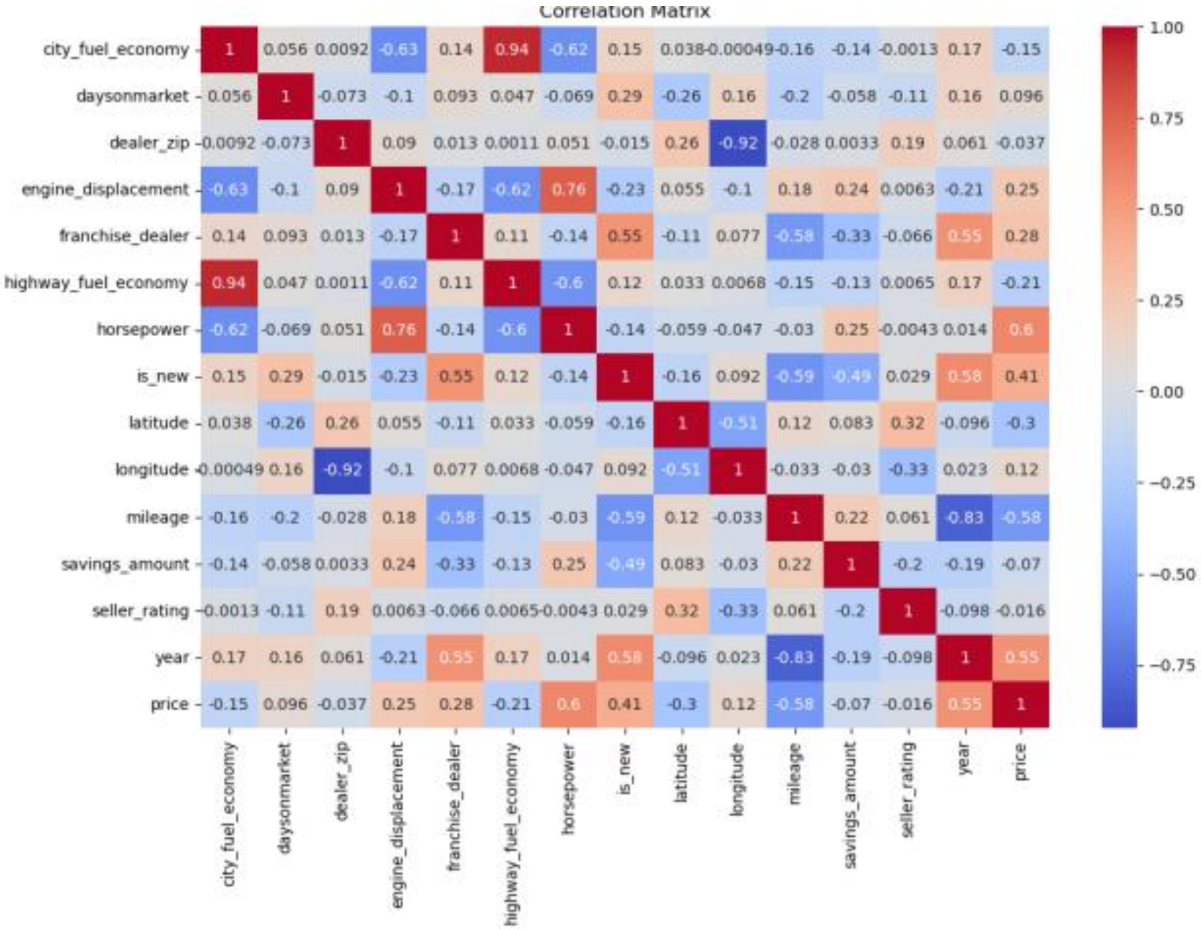
- Where multiple values are given for an observation encode the observation as 'other'.
- Where a categorical feature contains more than 5 unique values, map the features into 5 most frequent values + 'other' and then encode appropriately. For instance, map colours into 5 basic colours + 'other': [red, yellow, green, blue, purple, other] and then encode

Answer: I applied custom encoding to the 'listing_color' variable, specifically mapping colors like gray, black, white, silver, and red to themselves, while categorizing all other colors as 'OTHERS.' Similarly, for the 'maximum_seating' variable, I encoded values ranging from 1 to 5 as they are, and categorized any values representing more seats as 'OTHERS.'

Question 5: Perform any other actions you think need to be done on the data before constructing predictive models, and clearly explain what you have done.

Answer: i created a new column 'car age' and added it to the traindf and testdf I believe that the "car age" column is essential and plays a crucial role as it directly reflects the car's durability and longevity.

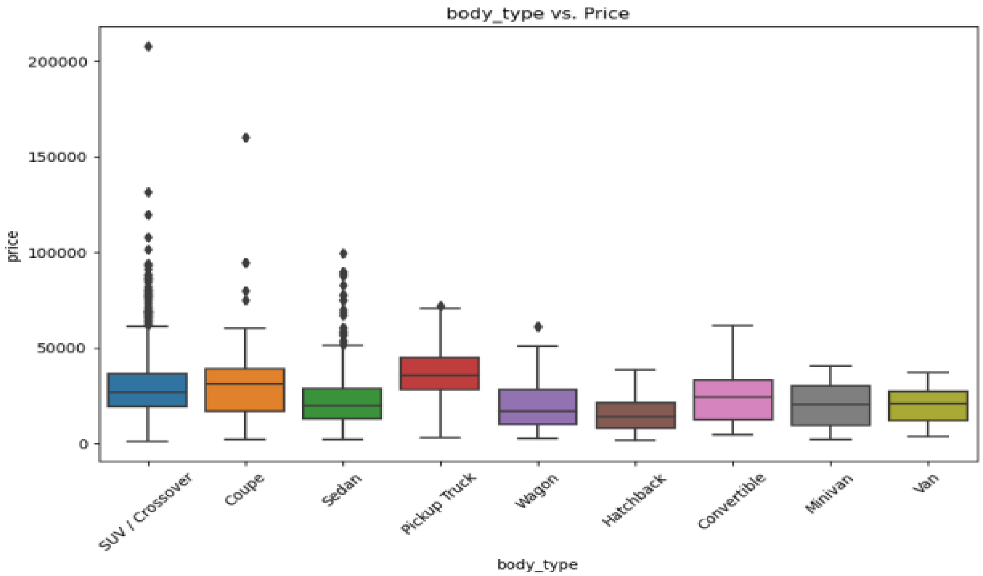
Question 6: Perform exploratory data analysis to measure the relationship between the features and the target and carefully write up your findings



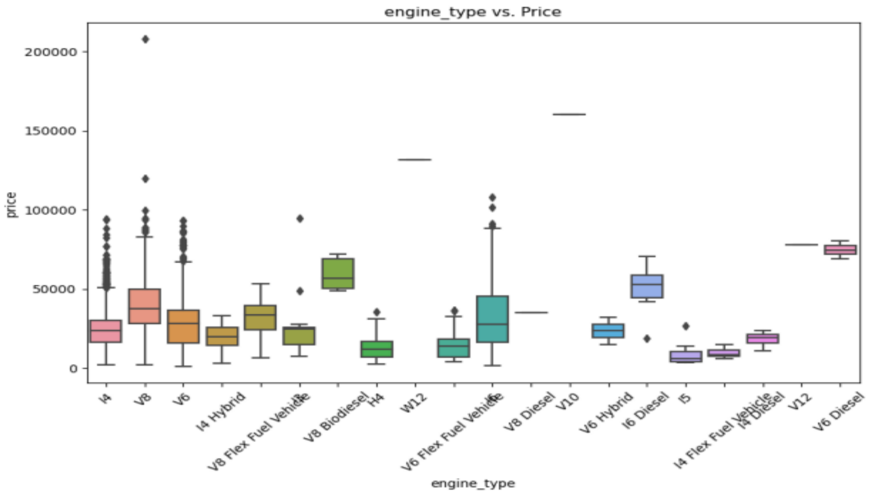
The heat-map displays the relationships, between variables using colors. The warmer colors like reds indicate correlations among all variables. In this case the variable back_legroom stands out as having the correlation with the target variable 'price

Correlation Strength: When a variable has a correlation it means that as that variables value increases the cars 'price' tends to increase as well. On the hand if there is a negative correlation, an increase in that variables value is associated with a decrease, in the 'price.' Positive and Negative Correlations: By looking at the heatmap we can determine which variables have correlations (represented by colors) and which ones have negative correlations (represented by cool colors) with 'price.'

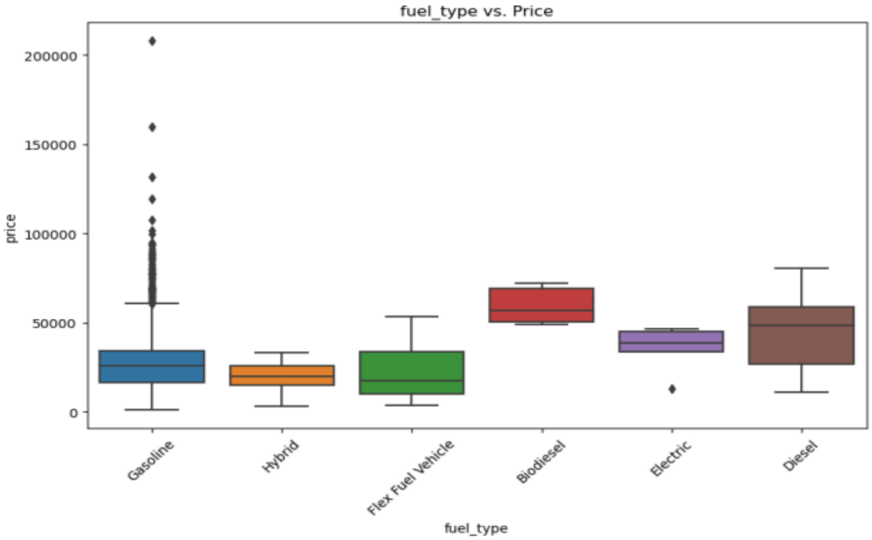
Comparing Categorical Features to Price: When examining the boxplots for features we gain extra insights, into how these features are connected to the target variable (price):



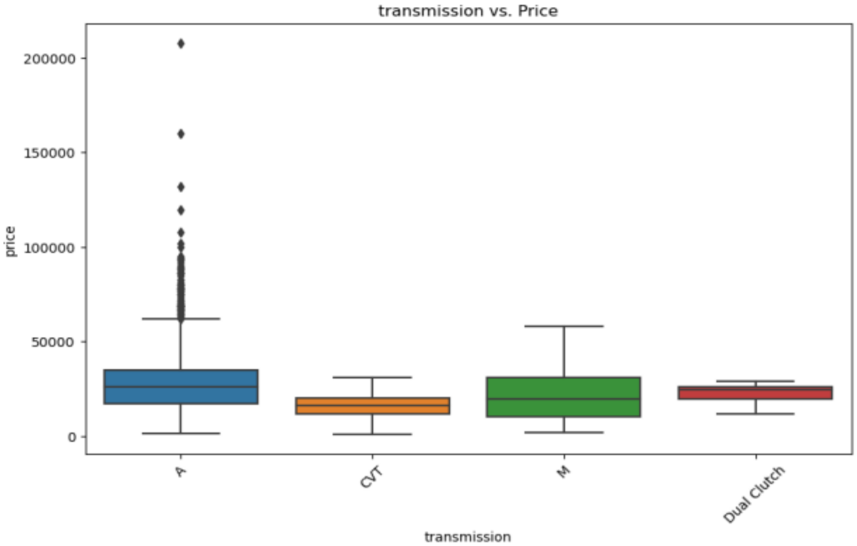
Body Type and Price: The boxplot for body_type vs price indicates that pickup trucks generally have the highest average price, followed by coupes. However it's important to note that SUV/Crossover and Sedan categories have outliers suggesting that certain cars, in these categories have prices exceeding those of pickup trucks.



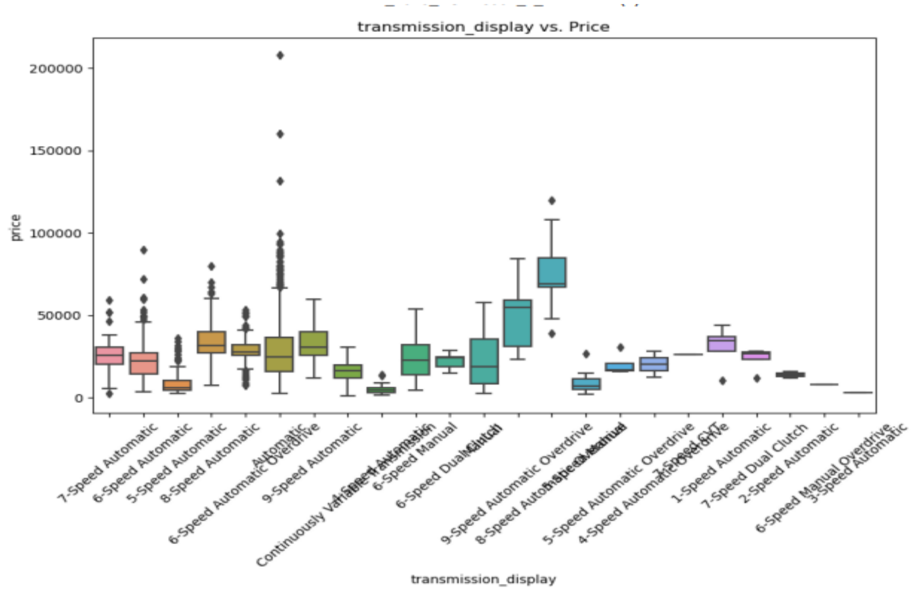
Engine. Price: By looking at the engine_type vs price boxplot we can observe the price ranges associated with engine types. This information helps us understand what we can expect in terms of car prices. Notably some mid range cars equipped with V6 Diesel engines appear to have prices.



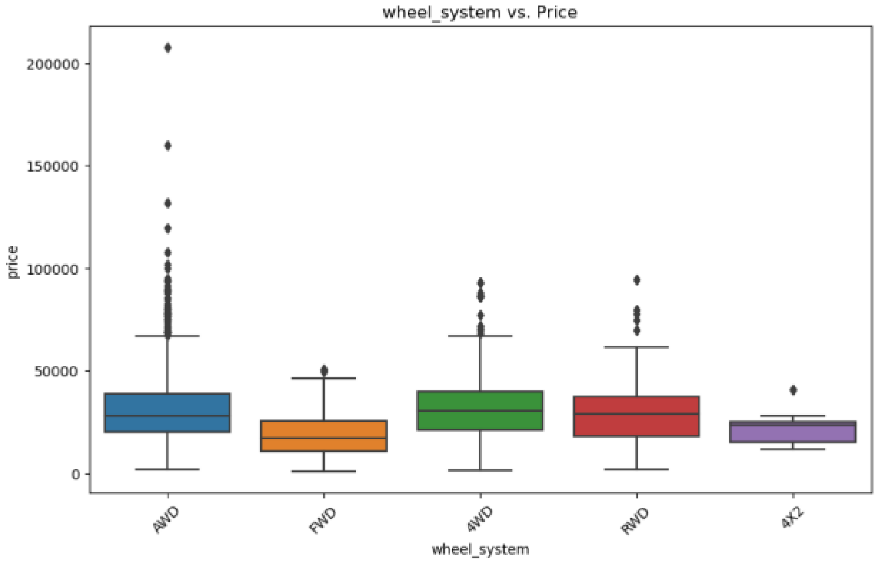
Fuel Type vs. Price: The fuel_type vs price boxplot indicates that the most expensive cars are typically fueled by biodiesel. Gasoline-fueled cars are the most common, and there are outliers in various categories, suggesting that some cars in these categories can have notably high prices.



Transmission vs. Price: The transmission vs price boxplot shows that cars with 'A' (Automatic) transmission are the most common across all price ranges, including their outliers. Other transmission types appear to be more randomly scattered.



Transmission Display vs. Price: The transmission_display vs price boxplot displays different types of transmission displays and their distribution in the dataset. Notably, cars with '9-speed Automatic Overdrive' have the highest price range, while those with '8-speed Automatic' are spread out and contain outliers



Wheel System vs. Price: The 'wheel_system' boxplot illustrates the distribution of various wheel systems in the dataset. 'AWD' (All-Wheel Drive) is the most common wheel system, followed by '4WD' (Four-Wheel Drive), and other categories.