How to Quantify Automotive Luxury?

Boyang Wan

MGSC 661: Multivariate Statistics

Instructor: Juan Camilo Serpa

Dec 3, 2024

Introduction

The concept of luxury in the automotive industry is often subjective, driven by perception rather than quantitative metrics. **The objective of this study is to quantify the "luxury perception" associated with car brands**, transforming it into a measurable score. Luxury, as an abstract concept, is difficult to define and standardize, especially for car brands. To address this, a **luxury score** was formulated by engineering features that could objectively capture the luxury aspects of a car.

This analysis also aims to understand **market segmentation** by clustering different car models with their luxury scores, thereby offering business insights that can help brands position their models strategically compared to competitors. These insights are geared towards helping brands adjust specific features to either enhance their luxury perception or position themselves as economical alternatives to target different consumer segments.

Data Exploration and Feature Engineering

Dataset Overview

The dataset used in this analysis includes various attributes of cars such as make, fuel type, engine size, curb weight, etc., totaling 193 observations across 39 variables. For a part of detailed descriptions of features, refer to Table 1.

Data Preprocessing

To prepare the dataset for analysis, several preprocessing steps were undertaken. These included handling missing values, standardizing numerical features, encoding categorical variables, and normalizing the data to ensure compatibility between different features. Additionally, feature scaling was performed to ensure that attributes with larger numerical ranges did not disproportionately affect the model training.

Feature Engineering

To quantify the luxury of car models, multiple new features were engineered:

1. **Power-to-Weight Ratio**

$$\text{Power-to-Weight Ratio} = \frac{\text{Horsepower}}{\text{Curb Weight}}$$

   Higher power-to-weight ratios are typically associated with performance and luxury.

2. **Size Index**

$$\text{Size Index} = \text{Wheel Base} \times \text{Width} \times \text{Height}$$

   The overall size of a car may indicate its luxury status (e.g., larger cars are often considered more premium).

3. **Luxury Brand Indicator**

$$\text{Luxury Brand Indicator} = \begin{cases} 1 & \text{if Make is Jaguar, Mercedes-Benz, Porsche, BMW, or Volvo} \\ 0 & \text{otherwise} \end{cases}$$

These brands were selected based on their high positive price coefficients from the regression analysis, indicating strong associations with higher prices and perceived luxury. Their established reputation in the premium market segment supports their inclusion, ensuring the indicator reflects true luxury brands. For a visualization of the price coefficients by car brands, refer to the figure in the appendix (Figure 1).

4. **Fuel Economy Difference**

$$\text{Fuel Economy Difference} = \text{Highway MPG} - \text{City MPG}$$

The difference between highway and city mileage can indicate performance tuning for efficiency.

5. **Performance Index**

$$\text{Performance Index} = \frac{\text{Engine Size} \times \text{Compression Ratio} \times \text{Peak RPM}}{1000}$$

The Performance Index is a composite metric combining these three factors to provide a holistic view of a car's engine performance:

- **Engine Size**: Contributes to the raw power potential.
- **Compression Ratio**: Indicates efficiency and power optimization.
- **Peak RPM**: Reflects the engine's ability to deliver power quickly.

By combining these, the Performance Index captures a balance between raw power, efficiency, and speed capability, which are key dimensions of a car's performance. The division by 1000 is applied to scale down the values for better interpretability and to prevent extremely large numbers from dominating the analysis.

6. **Weight-to-Size Ratio**

$$\text{Weight-to-Size Ratio} = \frac{\text{Curb Weight}}{\text{Wheel Base} \times \text{Width} \times \text{Height}}$$

This captures the weight distribution relative to the car's size, providing insights into how efficiently the car's weight is managed for its dimensions, which can affect performance and handling.

7. **Compression Efficiency**

$$\text{Compression Efficiency} = \frac{\text{Compression Ratio} \times \text{Horsepower}}{\text{Engine Size}}$$

This metric reflects how efficiently the engine converts fuel to power by balancing the compression ratio and horsepower with engine size, providing insights into the engine's performance optimization.

8. **Doors-to-Weight Ratio**

$$\text{Doors-to-Weight Ratio} = \frac{\text{Number of Doors}}{\text{Curb Weight}}$$

This metric reflects accessibility and weight distribution, indicating how the number of doors scales with the vehicle's overall weight for practical and design considerations.

Multicollinearity and Feature Reduction

During the initial feature selection process, multicollinearity among the variables was addressed by calculating the Variance Inflation Factor (VIF) for each feature. Features with high VIF scores were grouped and dropped based on redundancy and high correlation.

The features dropped are:

- **Highly Correlated with Size Index**:
  - *Wheel Base, Length, Width, Height*: Dropped due to high correlation with *Size Index*, which effectively captured the overall vehicle dimensions.
- **Redundant Weight Metrics**:
  - *Curb Weight, Engine Size*: Removed because they were highly correlated with *Power-to-Weight Ratio*, *Size Index*, and *Performance Index.* These metrics already represented the necessary relationships between weight, size, and engine characteristics.
- **Redundant Performance Indicators**:
  - *Horsepower, Compression Ratio*: Dropped due to their contribution already being encapsulated within *Power-to-Weight Ratio* and *Performance Index.*
- **Fuel Efficiency Overlap**:
  - *Highway MPG, City MPG*: Dropped as their difference (*Fuel Economy Difference*) provided a more meaningful metric for efficiency.
- **Low Predictive Categorical Variables**:
  - *Make*: Dropped in favor of *Luxury Brand Indicator*, which better represented the luxury status of the car brands.
  - *Number of Doors, Number of Cylinders*: Removed due to limited contribution to luxury perception and redundancy with other metrics.

Model Selection and Methodology

Feature Selection

The primary objective of feature selection was to determine the key factors that could influence car prices, which served as a proxy for the concept of "luxury." To achieve this, both linear regression and random forest models were employed to rank and identify the most important features related to price. These selected features would later form the basis of the luxury score.

- **Linear Regression**: Linear regression was initially used to assess the relationship between each feature and the target variable, *price*. The regression coefficients were ranked to determine the most influential features, providing a straightforward approach for feature interpretability.
- **Random Forest**: A Random Forest model was also applied to capture complex non-linear interactions between features. The model provided an importance ranking for each feature, identifying which attributes contributed most significantly to predicting the target variable.

- **Final Feature Selection**: The results from both models were combined and normalized. The top 10 features with the highest average importance scores across both models were selected for further analysis. This approach ensured that only the most impactful features, as confirmed by multiple methods, were used in constructing the luxury score. For a visualization of the top 10 selected features by their normalized importance scores, see Figure 2 in the appendix.

Luxury Score Construction

The **Luxury Score** for each car is calculated as:

$$Luxury\_Score_i = \sum_{j=1}^{N} (w_j \cdot x_{ij})$$

Where:

- i: Represents the car (row in the dataset).
- j: Represents the feature (10 most important features selected using linear regression and random forest).
- N: Total number of selected features.
- w_j: The weight of the j-th feature, derived from the **Average Normalized Importance** across both linear regression and random forest.
- x_{ij}: The normalized value of the j-th feature for the i-th car.
- $Luxury\_Score_i$: The calculated luxury score for the i-th car.

Each car's luxury score is determined by summing the product of its normalized feature values and their corresponding feature importance weights.

Detailed Explanation

The luxury score calculation is designed to quantify the perceived luxury level of each car by integrating the contributions of the selected features, weighted by their relative importance. Below, we break down the components of the formula in more detail:

1. **Selected Features (x_{ij})**: The features used in this formula are the top 20 features identified during the feature selection process. Each feature represents an important attribute that significantly affects the price, which acts as a proxy for luxury perception.
2. **Normalization of Features**: Each feature value (x_{ij}) is normalized to bring all values to a common scale, typically between 0 and 1. This prevents any feature with larger raw values from disproportionately influencing the luxury score.
3. **Weights (w_j)**: The weights (w_j) are derived from the average importance of each feature, as determined by both the linear regression and random forest models. The importance scores from each model were normalized and averaged to ensure a balanced representation of feature significance, accounting for both linear and non-linear effects.

4. **Weighted Sum for Luxury Score ($Luxury\_Score_i$)**: The final luxury score for each car is calculated by taking the weighted sum of the normalized feature values. This ensures that each feature contributes to the luxury score proportionally to its importance, providing a composite measure that reflects the combined effect of all selected features on the perception of luxury.

The resulting **Luxury Score** is a numerical value that allows for comparison across different car models, indicating their relative luxury level. Cars with higher luxury scores are those that perform well across the most significant features, thus embodying characteristics that contribute strongly to the perception of luxury. Refer to Appendix Figure 3 and Figure 4 for the distribution of luxury scores and the relationship between luxury scores and car prices, respectively.

Clustering Analysis

A clustering methodology was employed to segment car models into distinct groups for market positioning insights. The goal was to understand how different car features relate to perceived luxury and identify natural groupings in the data that could guide business decisions.

Dataset Preparation

A subset of 10 key features was selected for clustering, including engineered features such as:

- `power_to_weight_ratio`
- `size_index`
- `luxury_brand_indicator.1` (representing luxury branding)
- `fuel_economy_difference`
- `performance_index`
- `weight_to_size_ratio`
- `compression_efficiency`
- `doors_to_weight_ratio`
- `Luxury_Score`
- `price`

The inclusion of `Luxury_Score` and `price` was intended to ensure that the clustering captured both the engineered aspects of luxury and the market valuation.

Methodology

K-means clustering was used to segment the car models into clusters based on the selected features. The clustering process was guided by silhouette analysis to determine the optimal number of clusters. As shown in Figure 5, the silhouette scores for cluster numbers ranging from 2 to 10 were evaluated, and the optimal number of clusters (k) was determined to be 3. The resulting clusters and their spatial distribution are visualized in Figure 6.

Results

Feature Importance for Luxury Score

To determine which features most significantly contribute to the luxury score, both linear regression and random forest models were used. Features were ranked based on importance from each model, normalized, and then aggregated to select those with the highest significance.

These top features effectively quantify the luxury aspects of car brands, aligning with the objective to make the concept of luxury measurable. The combined rankings from linear and non-linear models capture a holistic view of luxury.

The top features selected for the luxury score calculation include:

- `luxury_brand_indicator.0`
- `weight_to_size_ratio`
- `size_index`
- `power_to_weight_ratio`
- `luxury_brand_indicator.1`
- `compression_efficiency`
- `performance_index`
- `fuel.type.diesel`
- `bore`
- `fuel.system.mpfi`

This set of features effectively captures various elements of performance, branding, size, and engine characteristics that contribute to the overall luxury perception of a vehicle.

Clustering Results

Cluster 1: Mid-Range Practical Cars

The luxury score for this cluster is moderate, with a mean of 0.43, indicating a blend of basic and mid-range features. The price range is moderate to high, with a mean of 1.06, reflecting mid-range pricing.

- **Performance and Power**: The Power-to-Weight Ratio is high with a mean of 1.08, offering decent acceleration and handling, while the Performance Index is moderate at a mean of 0.41, indicating a reasonable focus on performance.
- **Design and Practicality**: The Size Index is moderate with a mean of 0.37, indicating balanced car sizes suitable for daily use. The Weight-to-Size Ratio is high at 1.12, reflecting heavier builds relative to their size, and the Doors-to-Weight Ratio is low at -0.66, suggesting a preference for practicality over design complexity.
- **Luxury Indicator**: The Luxury Brand Indicator.1 has a high mean of 0.97, reflecting a strong presence of mid-range vehicles from luxury brands.

- **Fuel Economy and Efficiency**: The Fuel Economy Difference is slightly below average with a mean of -0.11, indicating adequate fuel efficiency, and Compression Efficiency is near average at -0.03, showcasing typical engine tuning.

Cluster 1 represents practical, mid-range vehicles that focus on offering a balanced driving experience. These cars are designed for buyers seeking reliable, moderately priced vehicles with good performance and practicality. While some luxury brands are represented, the overall focus is not on luxury but rather on value and usability.

Cluster 2: High-Performance Large Cars

The luxury score for this cluster is high, with a mean of 0.84, indicating strong luxury features. The price range is moderate to high, with a mean of 0.63, reflecting higher-end pricing due to size and performance.

- **Performance-Focused**: The Power-to-Weight Ratio is low with a mean of -1.09, emphasizing fuel efficiency over raw power. Compression Efficiency is very high at 2.73, showcasing advanced engine tuning, and the Performance Index is also very high at 2.44, indicating a strong focus on high performance.
- **Design and Size**: The Size Index is large with a mean of 1.35, reflecting significant car sizes, while the Doors-to-Weight Ratio is balanced at 0.04, suggesting practical design.
- **Fuel Economy**: The Fuel Economy Difference is very low with a mean of -1.27, indicating poor fuel efficiency.
- **Luxury Indicator**: The Luxury Brand Indicator.1 has a high mean of 0.41, reflecting a strong presence of luxury-branded cars.

Cluster 2 vehicles are designed for performance-oriented buyers who value size, cutting-edge technology, and luxury. These vehicles likely represent premium SUVs or large sports sedans.

Cluster 3: Economical Compact Cars

The luxury score for this cluster is low, with a mean of -0.27, indicating limited luxury features. The price range is low, with a mean of -0.52, making these the most economical cars.

- **Compact and Lightweight**: The Size Index is small with a mean of -0.32, reflecting compact car sizes, and the Weight-to-Size Ratio is low at -0.51, suggesting lightweight designs. The Doors-to-Weight Ratio is high with a mean of 0.27, indicating practical designs.
- **Fuel Economy and Efficiency**: The Fuel Economy Difference is above average with a mean of 0.20, reflecting better fuel efficiency, while Compression Efficiency is below average at -0.32, indicating less advanced engine tuning.
- **Performance**: The Power-to-Weight Ratio is below average with a mean of -0.33, reflecting moderate performance, while the Performance Index is low at -0.47, showing limited focus on performance.
- **Luxury Indicator**: The Luxury Brand Indicator.1 has a low mean of -0.46, reflecting minimal luxury branding.

Cluster 3 cars appeal to cost-conscious buyers seeking compact, practical, and fuel-efficient vehicles with limited luxury features. These cars represent affordable, small-sized offerings with moderate performance.

Summary of Clusters

- **Cluster 1**: Mid-range practical cars offering reliability and balanced features at moderate pricing.
- **Cluster 2**: High-performance, large luxury cars with advanced engine technology and moderate fuel efficiency.
- **Cluster 3**: Economical compact cars with practical designs, better fuel efficiency, and limited luxury branding.

The distribution of luxury scores across the clusters reveals significant differentiation in perceived luxury, as illustrated in Figure 7. The radar chart in Figure 8 visually compares the clusters across key features, highlighting their unique characteristics and relative strengths. Figure 9 compares the mean values of selected features across the three clusters, providing insight into the unique characteristics of each group.

Business Insights

The luxury score and selected features provide car brands with insights to refine their products. By understanding which features impact luxury perception the most, brands can strategically enhance those areas. This can be particularly useful for product development or repositioning a model to better meet market demands.

Clustering analysis helps car brands identify distinct market segments, enabling tailored marketing and product design strategies. For instance, brands can focus on fuel efficiency for economical clusters or enhance luxury features for high-end segments.

For Toyota, these insights are directly applicable. By benchmarking their high-end models, such as the Lexus LS, against luxury competitors like BMW and Mercedes, Toyota can identify gaps in features like size index and weight-to-size ratio. Improving these aspects can boost the luxury perception of Lexus vehicles. Additionally, clustering insights help Toyota understand its market position and identify opportunities to align Lexus more closely with premium segments.

Conclusion

This project combines the development of a **Luxury Score** and **clustering analysis for market segmentation** to provide a data-driven framework for understanding and quantifying automotive luxury. The Luxury Score translates subjective perceptions of luxury into a measurable index by leveraging features most strongly associated with car pricing. It offers actionable insights for product enhancement and market positioning. Clustering analysis further segments car models into three distinct groups—mid-range practical cars, high-performance luxury vehicles, and economical compact cars—highlighting key market dynamics and customer preferences. Together, these methodologies empower automotive brands to refine strategies, optimize offerings, and effectively target diverse consumer segments.

Appendix

Table 1: Data Dictionary for Selected Features

Feature Name	**Description**
`power_to_weight_ratio`	Ratio of a car's power output to its weight, indicating acceleration capability.
`size_index`	A measure representing the overall size of the car.
`luxury_brand_indicator.1`	Binary indicator (1 if luxury brand, 0 otherwise).
`fuel_economy_difference`	Difference in fuel economy compared to a baseline value.
`performance_index`	An index indicating the performance capability of the car.
`weight_to_size_ratio`	Ratio of a car's weight to its size, indicating build compactness.
`compression_efficiency`	Engine's compression efficiency, reflecting power and fuel efficiency balance.
`doors_to_weight_ratio`	Ratio of the number of doors to the car's weight, indicating design compactness.
`Luxury_Score`	Composite score representing the luxury characteristics of the car.
`price`	Price of the car in normalized units.
`luxury_brand_indicator.0`	Binary indicator (0 if non-luxury brand, 1 otherwise).
`fuel.type.diesel`	Binary indicator (1 if diesel fuel type, 0 otherwise).
`bore`	Diameter of the cylinder bore in the engine, affecting engine displacement.
`fuel.system.mpfi`	Indicator for multi-point fuel injection system.

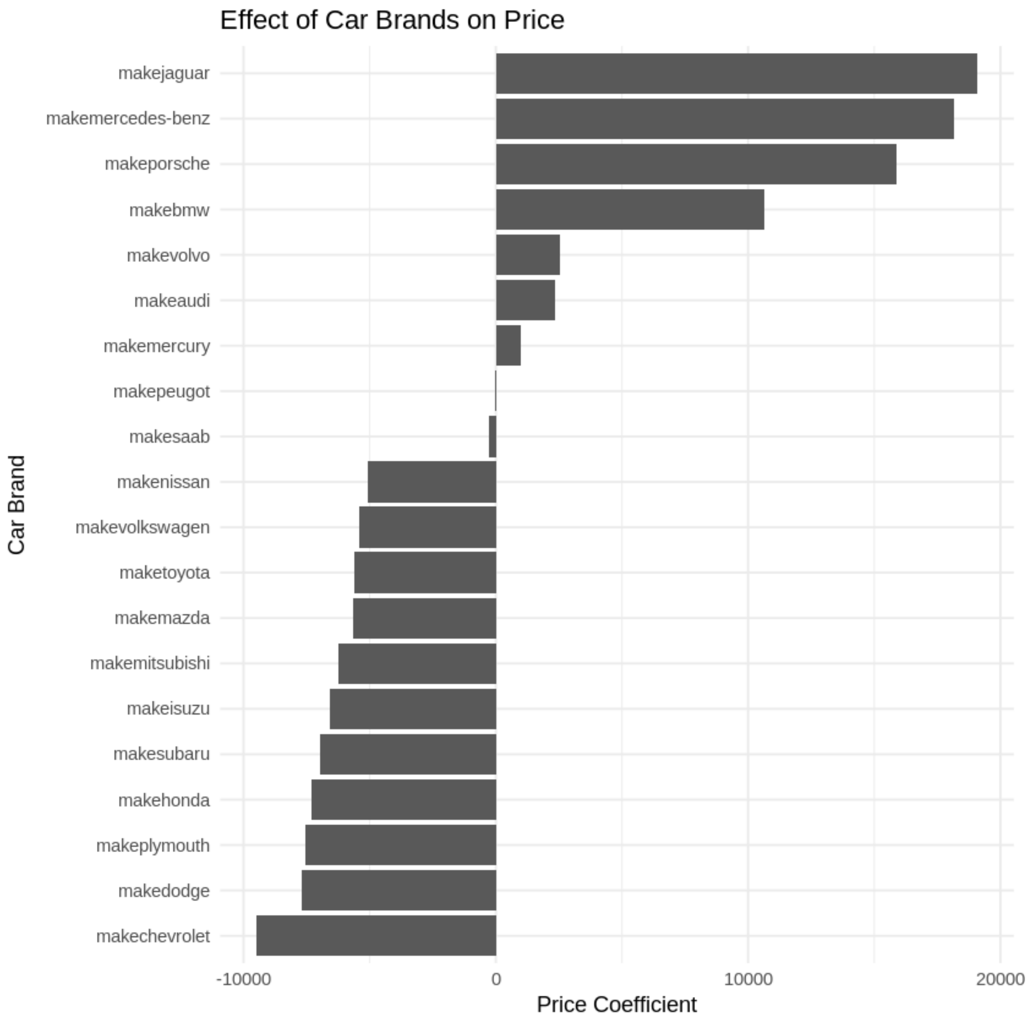

Figure 1: Effect of Car Brands on Price

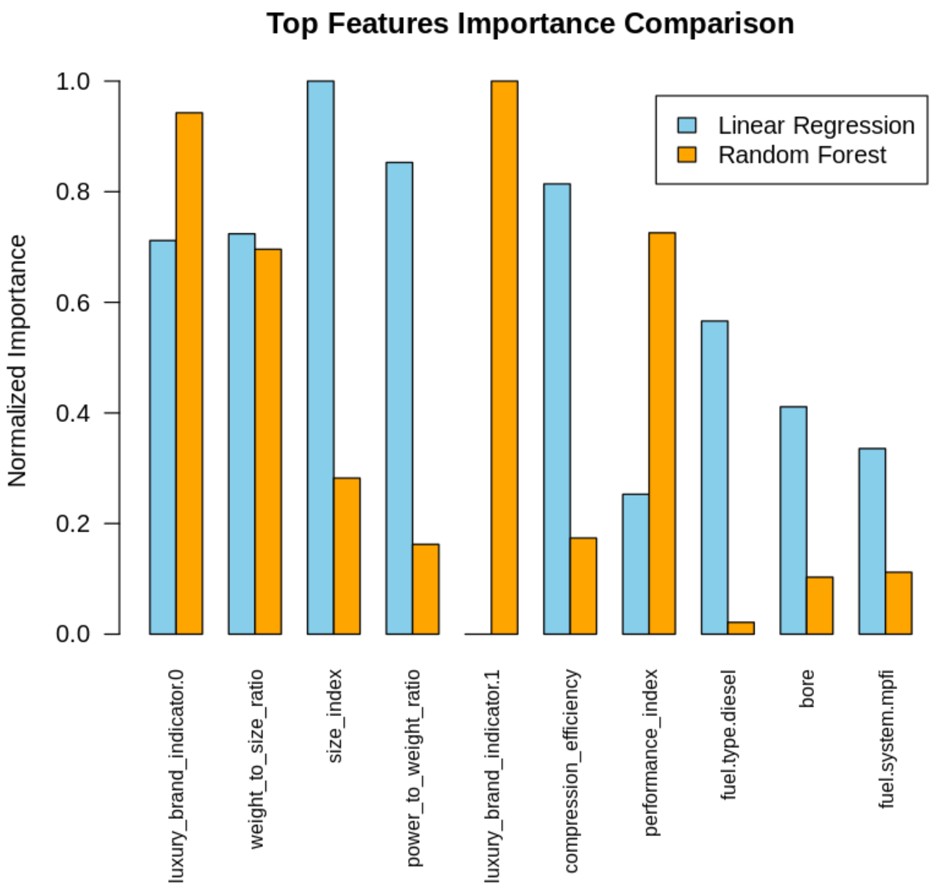

Figure 2: Top 10 Features Selected by Average Normalized Importance Scores

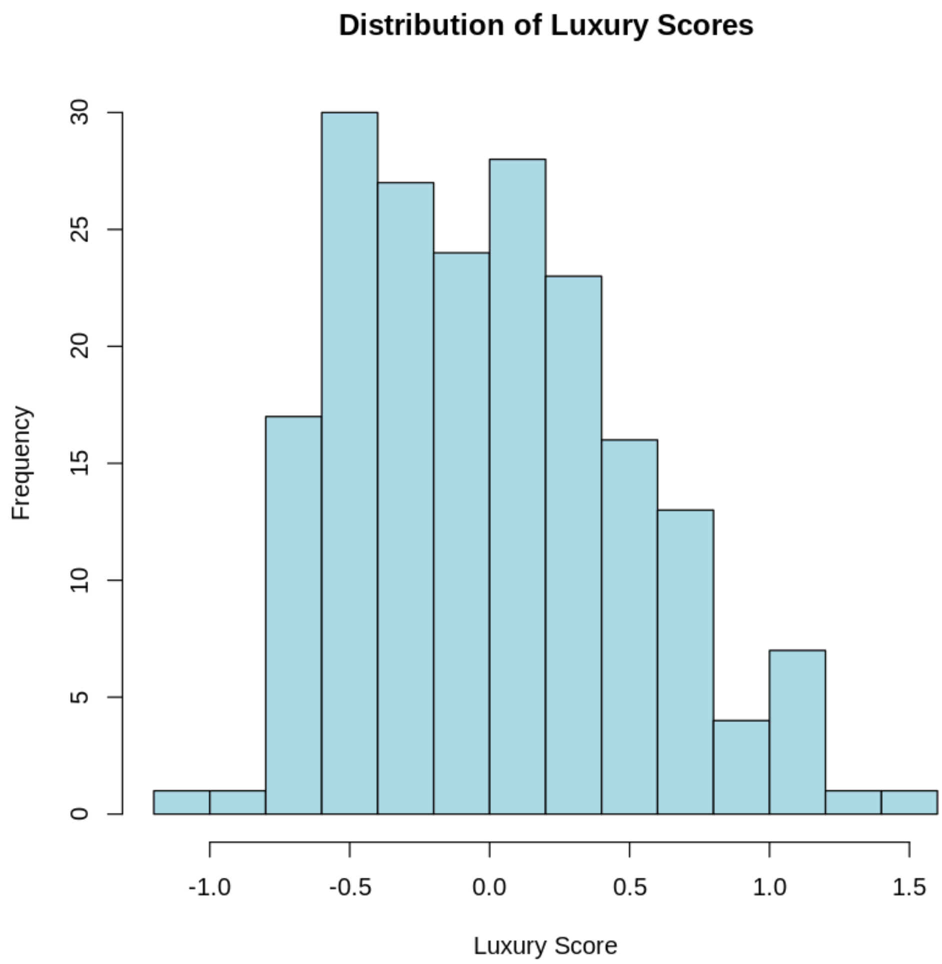

Figure 3: Distribution of Luxury Scores across all car models. This histogram visualizes how the luxury scores are distributed within the dataset, showing the clustering of scores.

Figure 4: Relationship between Luxury Score and Price. This scatter plot illustrates the correlation between a car's luxury score and its price, highlighting how luxury metrics relate to pricing.

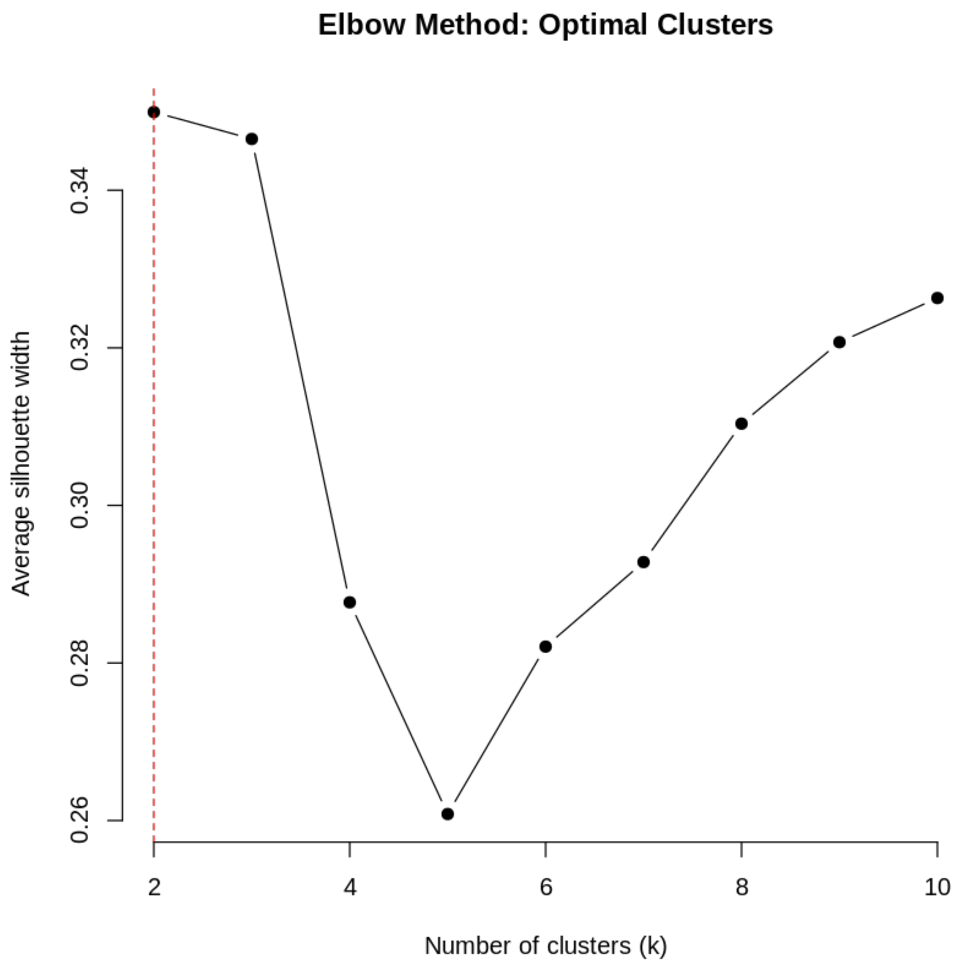

Figure 5: Silhouette Analysis for Optimal Number of Clusters

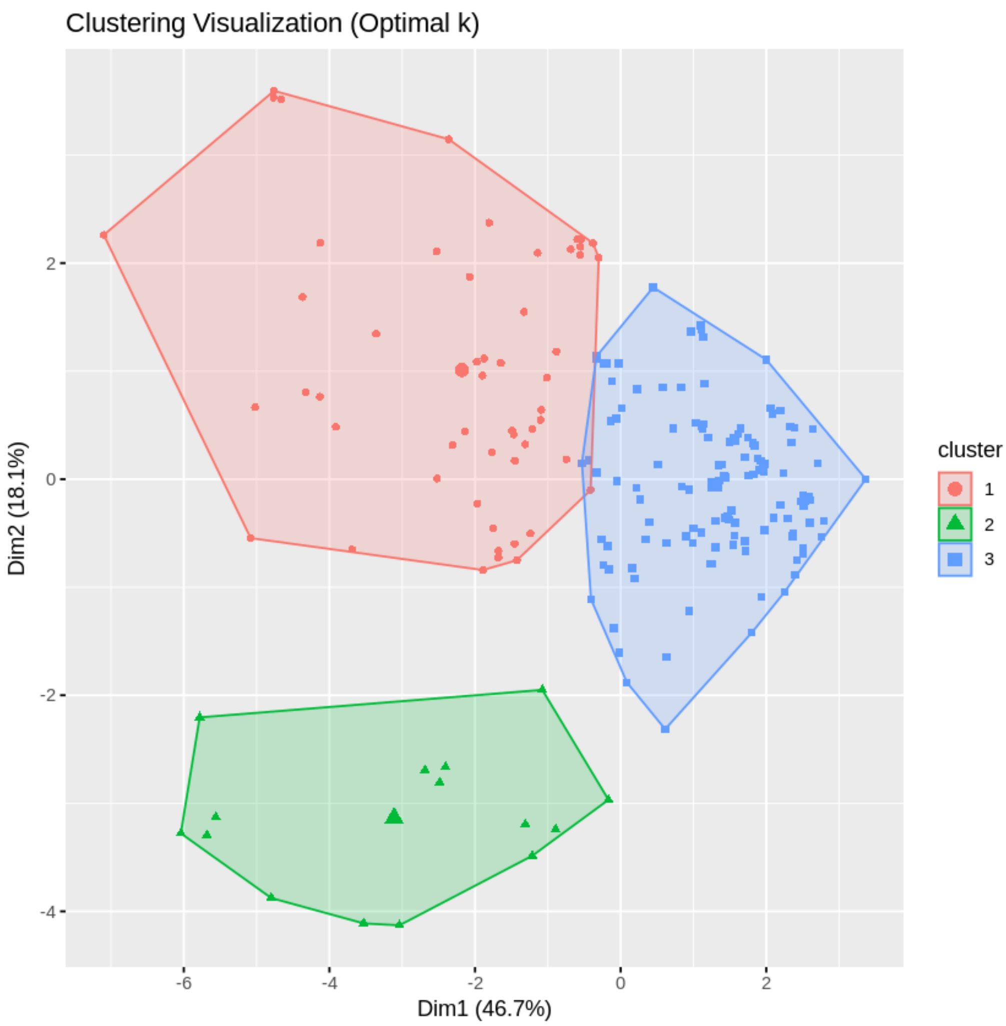

Figure 6: Visualization of Clustering Results with Optimal k

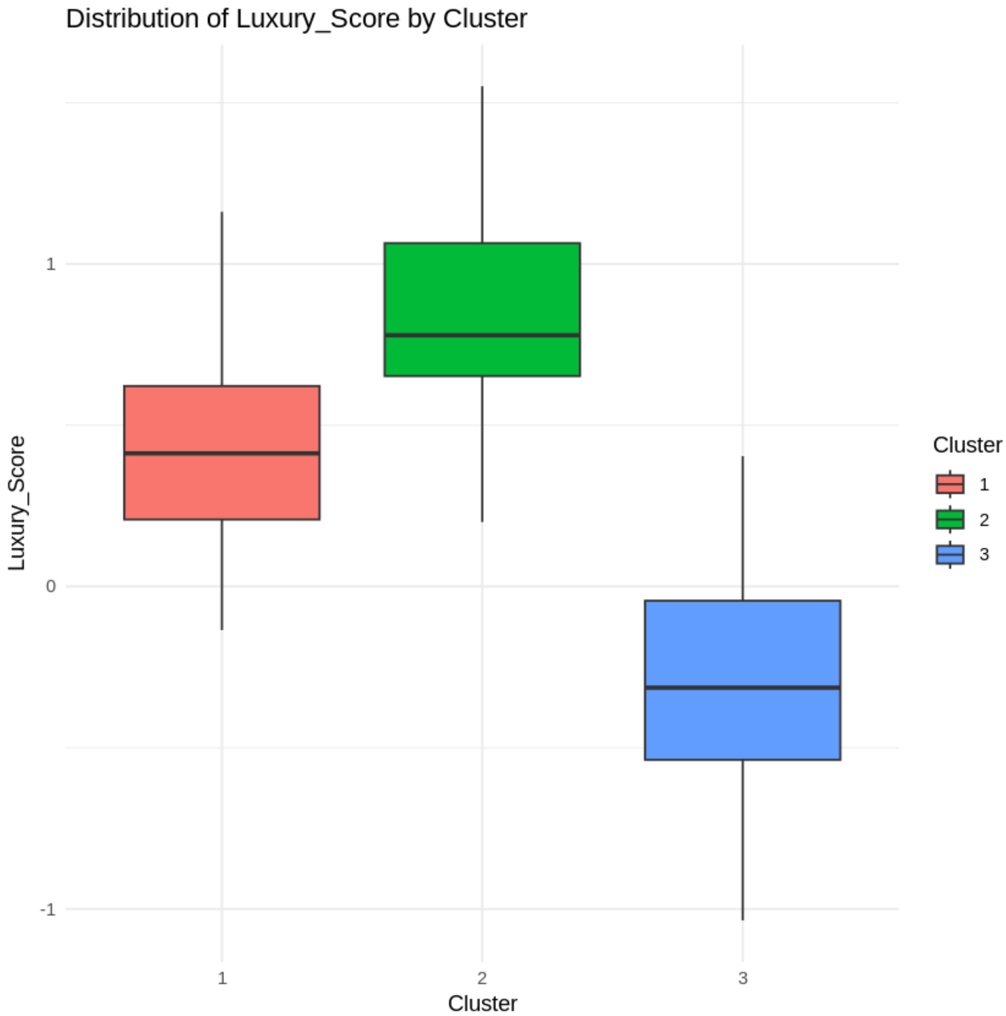

Figure 7: Distribution of Luxury Scores by Cluster. This boxplot highlights the variation in luxury scores across the three clusters, reflecting distinct levels of perceived luxury.

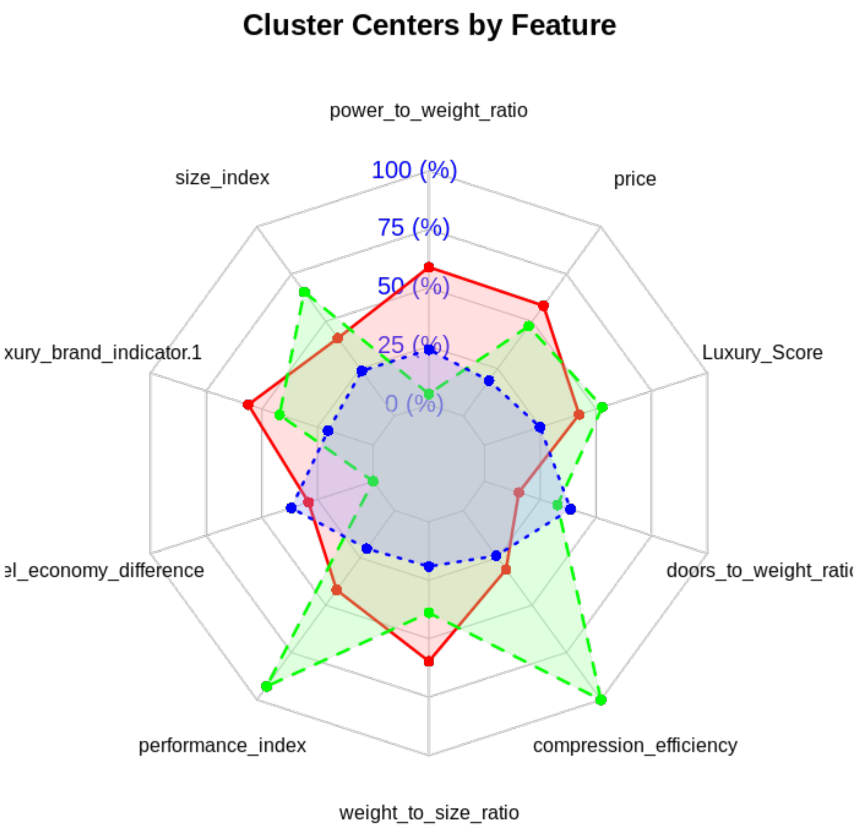

Figure 8: Radar Chart of Clusters Across Key Features. This chart illustrates the relative feature values for each cluster, providing a clear comparison of their profiles.

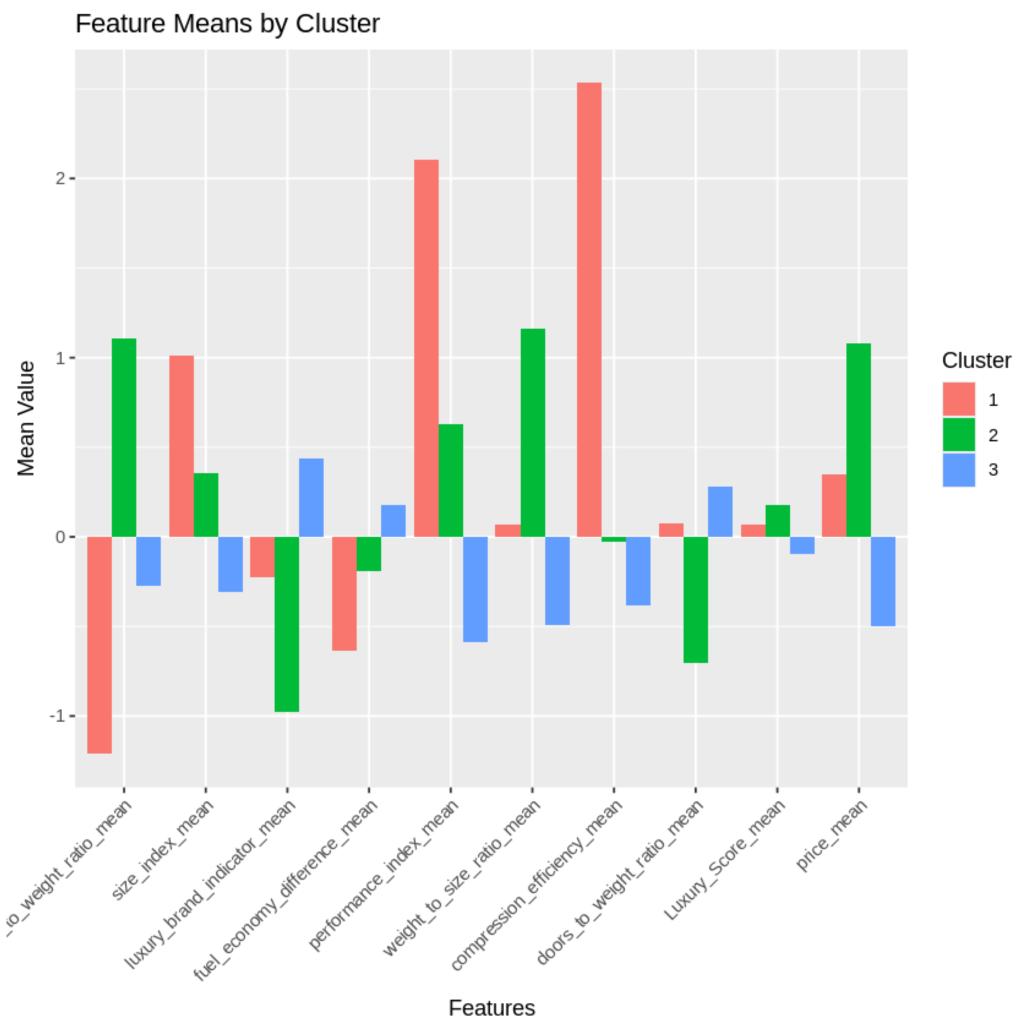

Figure 9: Feature Means by Cluster. This bar chart compares the average values of key features across clusters, highlighting their distinguishing characteristics.

Appendix: R Code

To see more structured code with output, check my Jupyter Notebook at this link: https://colab.research.google.com/drive/1brAXXqhMLEVESfciPNT3NgpVmBgJheZw?usp=sharing.

```
# -*- coding: utf-8 -*-
#"""Automobile.ipynb

# -------------------------------------------------------------
# Load Dataset and Packages
# -------------------------------------------------------------
install.packages("dplyr")
install.packages("car")
install.packages("e1071")
install.packages("caret")
install.packages("cluster")
install.packages("factoextra")
install.packages("fmsb")

df = read.csv('Dataset5_Automobile_data.csv')

library(fmsb)
library(dplyr)
library(car)
library(e1071)
library(caret)
library(cluster)
library(factoextra)

attach(df)

# -------------------------------------------------------------
# Data Preprocessing
# -------------------------------------------------------------

## Missing Values Handling
missing_counts <- sapply(df, function(col) sum(col == "?" | is.na(col)
    | col == "", na.rm = TRUE))
missing_df <- data.frame(Column = names(missing_counts), Missing_Count
    = missing_counts)
missing_df

# Replace "?" with NA for easier handling
df[df == "?"] <- NA

# Remove rows with any missing values
df <- na.omit(df)

# -------------------------------------------------------------
# Drop Irrelevant Features
# -------------------------------------------------------------
df <- df[, !(names(df) %in% c("normalized.losses", "symboling"))]

```

```r
# ---------------------------------------------------------------
# Data Type Alignment
# ---------------------------------------------------------------
str(df)

## Mapping for num.of.doors and num.of.cylinders
door_mapping <- c("two" = 2, "four" = 4)
df$num.of.doors <- door_mapping[df$num.of.doors]

cylinder_mapping <- c("two" = 2, "three" = 3, "four" = 4, "five" = 5, "six" = 6, "eight" = 8, "twelve" = 12)
df$num.of.cylinders <- cylinder_mapping[df$num.of.cylinders]

## Convert Columns from Character to Numeric
cols_to_convert <- c("bore", "stroke", "horsepower", "peak.rpm", "price")
df[cols_to_convert] <- lapply(df[cols_to_convert], function(x) as.numeric(as.character(x)))

# ---------------------------------------------------------------
# Feature Engineering
# ---------------------------------------------------------------

## Power-to-Weight Ratio
df <- df %>%
  mutate(power_to_weight_ratio = horsepower / `curb.weight`)

## Size Index
df <- df %>%
  mutate(size_index = `wheel.base` * width * height)

## Luxury Brand Indicator
df$make <- as.factor(df$make)
model <- lm(price ~ make, data = df)
summary(model)

luxury_brands <- c("jaguar", "mercedes-benz", "porsche", "bmw", "volvo")
df <- df %>%
  mutate(luxury_brand_indicator = ifelse(tolower(make) %in% luxury_brands, 1, 0)) %>%
  mutate(luxury_brand_indicator = as.factor(luxury_brand_indicator))

## Fuel Economy Difference
df <- df %>%
  mutate(fuel_economy_difference = highway.mpg - city.mpg)

## Performance Index
df <- df %>%
  mutate(performance_index = (engine.size * `compression.ratio` * peak.rpm) / 1000)

## Weight-to-Size Ratio
```

```r
df <- df %>%
  mutate(weight_to_size_ratio = `curb.weight` / size_index)

## Compression Efficiency
df <- df %>%
  mutate(compression_efficiency = (`compression.ratio` * horsepower) / engine.size)

## Doors-to-Weight Ratio
df <- df %>%
  mutate(doors_to_weight_ratio = `num.of.doors` / `curb.weight`)

# ---------------------------------------------------------------
# Multicollinearity & Feature Selection
# ---------------------------------------------------------------

## Drop Redundant Features
df <- df %>%
  select(-wheel.base, -length, -width, -curb.weight, -engine.size, -highway.mpg, -city.mpg, -horsepower, -compression.ratio, -make)

## Calculate VIF Scores
numeric_vars <- sapply(df, is.numeric)
numeric_df <- df[, numeric_vars]
vif_scores <- vif(lm(price ~ ., data = numeric_df))
vif_df <- data.frame(Variable = names(vif_scores), VIF = vif_scores)
vif_df <- vif_df[order(-vif_df$VIF), ]
vif_df

# ---------------------------------------------------------------
# Outlier Analysis
# ---------------------------------------------------------------

## Detect Outliers Using IQR Method
detect_outliers <- function(x) {
  q1 <- quantile(x, 0.25)
  q3 <- quantile(x, 0.75)
  iqr <- q3 - q1
  lower_bound <- q1 - 1.5 * iqr
  upper_bound <- q3 + 1.5 * iqr
  outliers <- x[x < lower_bound | x > upper_bound]
  return(length(outliers))
}

outlier_counts <- sapply(df[, sapply(df, is.numeric)], detect_outliers)
outlier_df <- data.frame(Column = names(outlier_counts), Outlier_Count = outlier_counts)
outlier_df

# ---------------------------------------------------------------
# Boxplots for Selected Features
# ---------------------------------------------------------------
```

```r
par(mfrow = c(1, 2))
boxplot(df$fuel_economy_difference, main = "Fuel Economy Difference",
    ylab = "Fuel Economy Difference")
boxplot(df$performance_index, main = "Performance Index", ylab = "
    Performance Index")

# ---------------------------------------------------------------
# Dummification of Character/Factor Columns
# ---------------------------------------------------------------

## Dummify All Character/Factor Variables
char_factor_cols <- names(df)[sapply(df, function(x) is.character(x) |
     is.factor(x))]
for (col in char_factor_cols) {
  if (is.character(df[[col]])) {
    df[[col]] <- as.factor(df[[col]])
  }
  dummy_vars <- dummyVars(paste("~", col), data = df)
  dummy_df <- data.frame(predict(dummy_vars, newdata = df))
  df <- cbind(df, dummy_df)
  df <- df[, -which(names(df) == col)]
}

# ---------------------------------------------------------------
# Standardization
# ---------------------------------------------------------------
numeric_cols <- sapply(df, is.numeric)
df_numeric <- df[, numeric_cols]

# Standardize the numeric columns
df_scaled <- scale(df_numeric)

# Convert scaled data back to data frame
df_scaled <- as.data.frame(df_scaled)

str(df_scaled)

# ---------------------------------------------------------------
# Feature Selection
# ---------------------------------------------------------------

## Linear Regression Model
lm_model <- lm(price ~ ., data = df_scaled)

# Extract coefficients as feature importance
lm_importance <- summary(lm_model)$coefficients[, "Estimate"]
lm_importance <- data.frame(Feature = names(lm_importance), Importance
     = lm_importance)
print(lm_importance)

## Random Forest Model

# Load Random Forest Library
```

```r
library(randomForest)

# Fit Random Forest Model
rf_model <- randomForest(price ~ ., data = df_scaled, importance = TRUE)

# Extract Feature Importance
rf_importance <- data.frame(Feature = rownames(rf_model$importance),
                            Importance = rf_model$importance[, "IncNodePurity"])
print(rf_importance)

# Plot Feature Importance
barplot(rf_importance$Importance, names.arg = rf_importance$Feature, las = 2, main = "Random Forest Feature Importance")

# ---------------------------------------------------------------
# Final Feature Selection and Normalization
# ---------------------------------------------------------------

# Normalize a Vector to Range 0-1
normalize <- function(x) {
  return((x - min(x)) / (max(x) - min(x)))
}

# Normalize Feature Importance from Linear Regression and Random Forest
lm_importance$Normalized_Importance <- normalize(abs(lm_importance$Importance))
rf_importance$Normalized_Importance <- normalize(rf_importance$Importance)

# Rename Columns for Clarity Before Merging
colnames(lm_importance) <- c("Feature", "Linear_Importance", "Linear_Normalized")
colnames(rf_importance) <- c("Feature", "RandomForest_Importance", "RandomForest_Normalized")

# Merge Importance from Both Models by Feature
combined_importance <- merge(lm_importance, rf_importance, by = "Feature", all = TRUE)

# Fill NA Values with 0 (In Case a Feature is Missing from One of the Models)
combined_importance[is.na(combined_importance)] <- 0

# Calculate Average Normalized Importance Across Methods
combined_importance$Average_Normalized_Importance <- rowMeans(
  combined_importance[, c("Linear_Normalized", "RandomForest_Normalized")])

# Sort by Average Normalized Importance
combined_importance <- combined_importance[order(-combined_importance$
```

```
  Average_Normalized_Importance), ]

# Select Top N Features
top_features <- head(combined_importance, 10)

# Print Combined Importance Table and Top Features
print(combined_importance)
print(top_features)

# Adjust Plot Size and Margins for Visualization
par(mar = c(12, 5, 4, 2))

# Plot Top Features
barplot(
  height = t(as.matrix(top_features[, c("Linear_Normalized", "RandomForest_Normalized")])),
  beside = TRUE,
  names.arg = top_features$Feature,
  las = 2,
  col = c("skyblue", "orange"),
  legend.text = c("Linear Regression", "Random Forest"),
  main = "Top Features Importance Comparison",
  ylab = "Normalized Importance",
  cex.names = 0.8
)

# ---------------------------------------------------------------
# Luxury Score Construction
# ---------------------------------------------------------------

## Luxury Score Calculation
calculate_luxury_score <- function(df, top_features) {
  # Extract Top Features and Their Normalized Importance
  selected_features <- top_features$Feature
  feature_weights <- top_features$Average_Normalized_Importance

  # Ensure Feature Weights Sum to 1
  feature_weights <- feature_weights / sum(feature_weights)

  # Create Luxury Score Column
  df$Luxury_Score <- rowSums(df[, selected_features] * feature_weights)

  return(df)
}

# Apply Function to Scaled Dataset
df_scaled_with_luxury_score <- calculate_luxury_score(df_scaled, top_features)

# Plot Distribution of Luxury Scores
hist(
  df_scaled_with_luxury_score$Luxury_Score,
```

```r
  main = "Distribution of Luxury Scores",
  xlab = "Luxury Score",
  col = "lightblue",
  border = "black",
  breaks = 15
)

# Scatter Plot: Luxury Score vs Price
plot(
  df_scaled_with_luxury_score$Luxury_Score,
  df_scaled_with_luxury_score$price,
  main = "Luxury Score vs Price",
  xlab = "Luxury Score",
  ylab = "Price",
  col = "darkblue",
  pch = 19
)

# -------------------------------------------------------------
# Clustering Analysis
# -------------------------------------------------------------

## Subset Dataset for Clustering
clustering_features <- df_scaled_with_luxury_score[, c(
  "power_to_weight_ratio",
  "size_index",
  "luxury_brand_indicator.1",
  "fuel_economy_difference",
  "performance_index",
  "weight_to_size_ratio",
  "compression_efficiency",
  "doors_to_weight_ratio",
  "Luxury_Score",
  "price"
)]

# -------------------------------------------------------------
# Silhouette Analysis to Determine Optimal Clusters
# -------------------------------------------------------------

silhouette_analysis <- function(data, max_clusters = 10) {
  sil_width <- numeric(max_clusters - 1)

  for (k in 2:max_clusters) {
    kmeans_model <- kmeans(data, centers = k, nstart = 25)
    sil <- silhouette(kmeans_model$cluster, dist(data))
    sil_width[k - 1] <- mean(sil[, 3])
  }

  return(sil_width)
}

# Perform Silhouette Analysis
```

```r
max_clusters <- 10
sil_width <- silhouette_analysis(clustering_features, max_clusters)

# Plot Silhouette Scores for Each k
plot(2:max_clusters, sil_width, type = "b", pch = 19, frame = FALSE,
     xlab = "Number of clusters (k)", ylab = "Average silhouette width",
     main = "Elbow Method: Optimal Clusters")
abline(v = which.max(sil_width) + 1, col = "red", lty = 2)

# Determine Optimal Number of Clusters
optimal_k <- which.max(sil_width) + 1
cat("Optimal number of clusters based on silhouette score:", optimal_k, "\n")

# -------------------------------------------------------------
# K-Means Clustering with Optimal Number of Clusters
# -------------------------------------------------------------

final_kmeans <- kmeans(clustering_features, centers = 3, nstart = 25)

# Visualize Clustering Results
library(factoextra)
fviz_cluster(final_kmeans, data = clustering_features, geom = "point",
             main = "Clustering Visualization (Optimal k)")

# -------------------------------------------------------------
# Cluster Analysis
# -------------------------------------------------------------

# Add Cluster Labels to Dataset
df_scaled_with_luxury_score$cluster <- final_kmeans$cluster

# Calculate Cluster-Level Statistics
cluster_stats <- df_scaled_with_luxury_score %>%
  group_by(cluster) %>%
  summarize(
    mean_luxury_score = mean(Luxury_Score),
    median_luxury_score = median(Luxury_Score),
    sd_luxury_score = sd(Luxury_Score),
    count = n()
  )

print(cluster_stats)

# -------------------------------------------------------------
# Visualization: Boxplot of Feature Distribution by Cluster
# -------------------------------------------------------------

# Select Feature for Boxplot (e.g., 'Luxury_Score')
feature_to_plot <- "Luxury_Score"

# Create Boxplot for Selected Feature by Cluster
```

```
ggplot(df_scaled_with_luxury_score, aes(x = factor(cluster), y = .data[[feature_to_plot]], fill = factor(cluster))) +
  geom_boxplot() +
  labs(title = paste("Distribution of", feature_to_plot, "by Cluster"),
       x = "Cluster",
       y = feature_to_plot,
       fill = "Cluster") +
  theme_minimal()

# ------------------------------------------------------------
# Radar Chart of Cluster Centers by Feature
# ------------------------------------------------------------

cluster_centers <- as.data.frame(final_kmeans$centers)
cluster_centers <- rbind(rep(max(cluster_centers), ncol(cluster_centers)),
                         rep(min(cluster_centers), ncol(cluster_centers)),
                         cluster_centers)

radarchart(cluster_centers,
           axistype = 1,
           pcol = c("red", "green", "blue"),
           pfcol = adjustcolor(c("#FF9999", "#99FF99", "#9999FF"), alpha.f = 0.3),
           plwd = 2,
           cglcol = "grey", cglty = 1, cglwd = 0.8,
           vlcex = 0.8,
           title = "Cluster Centers by Feature")

# ------------------------------------------------------------
# Cluster-Level Statistics for Specified Variables
# ------------------------------------------------------------

cluster_stats <- df_scaled_with_luxury_score %>%
  group_by(cluster) %>%
  summarize(
    across(
      c(
        "power_to_weight_ratio",
        "size_index",
        "luxury_brand_indicator.1",
        "fuel_economy_difference",
        "performance_index",
        "weight_to_size_ratio",
        "compression_efficiency",
        "doors_to_weight_ratio",
        "Luxury_Score",
        "price"
      ),
      list(mean = mean)
    ),
```

```r
    count = n()
  )

print(cluster_stats)

# -------------------------------------------------------------
# Visualization: Feature Means by Cluster
# -------------------------------------------------------------

cluster_means <- tibble::tibble(
  cluster = c(1, 2, 3),
  power_to_weight_ratio_mean = c(-1.2112051, 1.1070278, -0.2719148),
  size_index_mean = c(1.0100569, 0.3585888, -0.3047082),
  luxury_brand_indicator_mean = c(-0.2277566, -0.9761798, 0.4399394),
  fuel_economy_difference_mean = c(-0.6336319, -0.1890356, 0.1762587),
  performance_index_mean = c(2.1013806, 0.6273455, -0.5847224),
  weight_to_size_ratio_mean = c(0.06967552, 1.16250959, -0.49277906),
  compression_efficiency_mean = c(2.53229070, -0.02752784,
     -0.37975287),
  doors_to_weight_ratio_mean = c(0.07208247, -0.70114069, 0.27958218),
  Luxury_Score_mean = c(0.06761755, 0.18050763, -0.09856919),
  price_mean = c(0.3484391, 1.0797236, -0.5015142)
)

cluster_means_melted <- melt(cluster_means, id.vars = "cluster")

ggplot(cluster_means_melted, aes(x = variable, y = value, fill =
   factor(cluster))) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Feature Means by Cluster",
       x = "Features",
       y = "Mean Value",
       fill = "Cluster") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Listing 1: Quantify Luxury and Clustering for Market Segmentation