Unleashing Horsepower

Exploring the Dynamic Interplay Between Car Specifications Across Distinct Automotive Clusters

Zaid Mahmood

December 2024

The relationship between a car's horsepower and its specifications is key to understanding automotive performance and design. Beyond power output disparity, a non-optimal horsepower-chassis combination significantly impacts general handling and mileage. This analysis aims to predict horsepower based on various car features, while also categorizing cars into distinct clusters based on shared characteristics.

The primary objective is to identify how various car specifications influence horsepower and to identify unique characteristics of cars within each cluster. By using these clusters, the report breaks down automotive categories to provide targeted insights that link performance metrics with design attributes. This dual-layer approach facilitates a better understanding of the dynamic interplay between a vehicle's technical features and its overall power output.

Dataset

The dataset used for this analysis contains information on 200 cars, which covers diverse automotive specifications and pricing details. Each car is described by 26 features, which include numerical and categorical variables related to its physical dimensions, performance metrics, and other characteristics. Among these features are attributes such as length, width, curb weight, engine size, and fuel system, alongside pricing and other car specifications.

The primary target variable in this analysis is horsepower, which serves as a key indicator of a car's performance capabilities. The aim is to explore the relationship between horsepower and various car attributes to understand the factors that contribute the most significantly to its variance.

Dataset Attributes and Preprocessing

The dataset features both numerical and categorical variables, with numerical features including details like wheelbase, engine size, curb weight, and compression ratio, while categorical features encompass make, fuel type, drive wheels, and body style. The "symboling" and "normalized losses" columns were removed, as these features were not directly correlated with horsepower and deemed not useful for prediction purposes.

To prepare the data for feature selection, missing values and incorrect datatype values were addressed by replacing them with either the median or the mode, depending on whether the variable was numerical or categorical. Outliers with very high horsepower were also identified and removed to improve the generalizability of the analysis (Figure 1).

After preprocessing, the relationships between horsepower and other car specifications were explored. Scatterplots were used to visualize correlations

with numerical predictors (Figure 2), while boxplots depicted relationships for categorical variables (Figure 3-5). The analysis indicated strong positive relationships between horsepower and predictors like engine size, curb weight, and price, while negative correlations were observed with city and highway mileage.

Categorical variables like aspiration showed noticeable differences in horsepower between categories. A correlation matrix (Figure 6) also revealed high multicollinearity between the mileage features. To address this, these mileage values were averaged into a single "mileage" predictor.

Methodology

A combination of exploratory data analysis, clustering techniques, and regression modeling was employed to achieve these objectives. Feature importance from Random Forests was used to streamline the dataset by focusing on impactful features. Hierarchical clustering was then employed to group vehicles into clusters, which were subsequently included as a categorical variable in the final model.

Feature selection

Feature selection was conducted based on feature importance derived from Random Forest models (Figure 7). The most impactful features for predicting horsepower included engine size, curb weight, and compression ratio. Features with lower predictive power, such as the number of doors, body style, height, and drive wheels, were excluded to reduce model complexity and improve accuracy. Price feature was also removed to restrict the focus of the analysis solely on car's specifications.

Clustering

To enhance interpretability, Hierarchical Clustering was applied to segment the dataset into distinct clusters. Clustering considered both numerical features (e.g., engine size, curb weight) and categorical features (e.g., fuel type, make, aspiration). The Gower distance metric was utilized to handle mixed data types effectively. The dataset was segmented into five clusters, representing categories such as luxury performance vehicles and economy compact cars (Table 1-2). A brief overview of each cluster is as follow:

Cluster 1: Small, lightweight compact cars with moderate mileage (e.g., economy cars).
Cluster 2: Large luxury cars with high horsepower but poor mileage (e.g., performance-oriented vehicles).
Cluster 3: Balanced mid-range cars with average dimensions and horsepower (e.g., family cars).

Cluster 4: Lightest cars with the smallest engine sizes and best mileage (e.g., fuel-efficient economy cars).
Cluster 5: Diesel-powered performance cars with moderate horsepower and decent mileage.

These clusters were added as a categorical variable, which enriched the dataset's predictive power.

Model Development

A Random Forest Regressor was employed as the final predictive model for horsepower. This model was selected due to its ability to handle non-linear relationships and complex interactions among features and its robustness in handling different data types. The feature set was iteratively refined based on importance rankings, enhancing interpretability while reducing complexity.

Instead of hyperparameter tuning, an iterative feature removal approach was adopted to simplify the model while maintaining performance. Features were removed in the following order:

- Fuel Type: Showed minimal impact on horsepower predictions.
- Make: Its removal slightly improved model generalizability and reduced potential biases.
- Length: Had a low direct correlation with horsepower, and its removal streamlined the model.

Adding the cluster variable significantly enhanced interpretability, allowing the model to contextualize predictions within distinct automotive categories and offering insights into how horsepower varies across different types of cars (e.g., luxury vs. economy vehicles).

Results

The Random Forest model was trained using the most significant features as identified by feature selection. Key metrics from the model include:

- Number of Trees: 500
- Variables Tried at Each Split: 4
- Mean of Squared Residuals (MSR): 66.009
- Explained Variance: 94.16

These metrics demonstrate the model's high accuracy in predicting the optimal horsepower of vehicles.

Feature Importances

The feature importances from the Random Forest model provide insight into the predictive power of each variable, expressed as the percentage increase in mean squared error (%IncMSE) if the predictor is removed. The most significant predictors of horsepower are

- Mileage: 23.17
- Engine Size: 19.55
- Fuel System: 18.72
- Compression Ratio: 18.98
- Cluster: 16.03
- Curb Weight: 15.96

Mileage emerged as the top predictor, followed closely by engine size and fuel system. Clusters also showed significance in ensuring that predictions align with the characteristics of each car category. A detailed view of %IncMSE and IncNodePurity can be found in Figure 8.

Residual Analysis

Residuals, calculated as the difference between actual and predicted horsepower values, were distributed close to zero, ranging from -35.82 to 32.31. The small mean residual reflects the high accuracy of the model in predicting ideal horsepower (Figure 9).

Classification/Predictions and Conclusions

The analysis revealed that mileage, engine size, fuel system, and curb weight are the most critical predictors of horsepower. The Random Forest Regressor, combined with clustering, provided a nuanced approach to predicting horsepower across diverse vehicle categories. The model achieved 94.16% explained variance, highlighting its robustness and reliability.

Clustering played a crucial role in segmenting vehicles into distinct categories, which enhanced interpretability. Insights gained from predictions include:

- Clusters 1 and 4 (economy and highly fuel-efficient cars) showed the best alignment between predictions and actual horsepower, reflecting the model's strength in handling simpler configurations.

- Clusters 2 and 5 (luxury and diesel-powered performance cars) presented larger residual variations, driven by the complexities of high-performance features and unique diesel characteristics.
- Cluster 3 (mid-range cars) demonstrated consistent predictions with moderate residuals, balancing diversity in specifications.

Business Value and Applications

The insights derived from this model can help manufacturers design vehicles that are better tailored specific market needs. By identifying key factors like engine size, fuel system, and mileage as key predictors of horsepower, manufacturers can make data-driven decisions to optimize vehicle performance. For high-performance vehicle segments, this model emphasizes the importance of focusing on engine advancements and fuel system efficiency to meet consumer expectations in premium markets.

The clustering analysis adds an additional layer of interpretability, enabling manufacturers to segment their product lines effectively. For example, Cluster 4 represents vehicles designed for cost-conscious buyers who prioritize fuel efficiency, offering opportunities for manufacturers to refine their economy car offerings. On the other hand, Cluster 2 highlights the demand for high-horsepower, luxury vehicles, encouraging manufacturers to invest in premium features and branding strategies tailored to this segment.

Moreover, the model's ability to predict horsepower with high accuracy can streamline research and development efforts, reducing costs associated with trial-and-error design processes. By leveraging predictive insights, manufacturers can focus on developing innovative technologies, such as advanced fuel systems and lightweight materials, to stay ahead of competitors in both economy and high-performance segments.

The segmentation provided by the clusters also holds potential for targeted marketing and sales strategies. Automotive companies can use the cluster insights to align their advertising campaigns with specific customer profiles, ensuring that each product is effectively positioned in the market. Additionally, the integration of clustering into the model enhances its practical usability, enabling analysts to draw actionable insights that bridge the gap between technical predictions and strategic business decisions.

Conclusion

The predictive capabilities of this model, combined with the clustering insights, can be extended to the development of electric vehicles (EVs) to address CO2 emissions challenges. By identifying key predictors of performance, such

as energy efficiency (analogous to mileage in internal combustion vehicles), battery capacity (analogous to engine size), and drivetrain configurations, manufacturers can design EVs optimized for specific market segments. For instance, clusters could represent energy-efficient urban EVs or high-performance electric sports cars, guiding design and production strategies. Furthermore, by aligning vehicle features with CO2 emission reduction goals, this model can support manufacturers in transitioning to sustainable transportation solutions, thereby contributing to global decarbonization efforts.

Appendix

Cluster	Wheelbase (in)	Length (in)	Width (in)	Curb Weight (lbs)	Engine Size (L)	Horsepower (HP)	Mileage (MPG)
1	95	169	64.9	2355	101.6	96	26.3
2	104	187	68.1	3280	200	160	20
3	101	179	66.7	2691	127.9	115	24.7
4	95	164	64.3	2114	99.6	74	33.3
5	104	182	67.4	2899	135.7	84	32.5

Table 1: Car Specifications by Clusters 1

Cluster	Common Make	Fuel Type	Aspiration	Cylinders
1	Subaru	Gasoline	Standard	Four
2	BMW	Gasoline	Standard	Six
3	Toyota	Gasoline	Standard	Four
4	Honda	Gasoline	Standard	Four
5	Peugeot	Diesel	Turbo	Four

Table 2: Car Specifications by Clusters 2

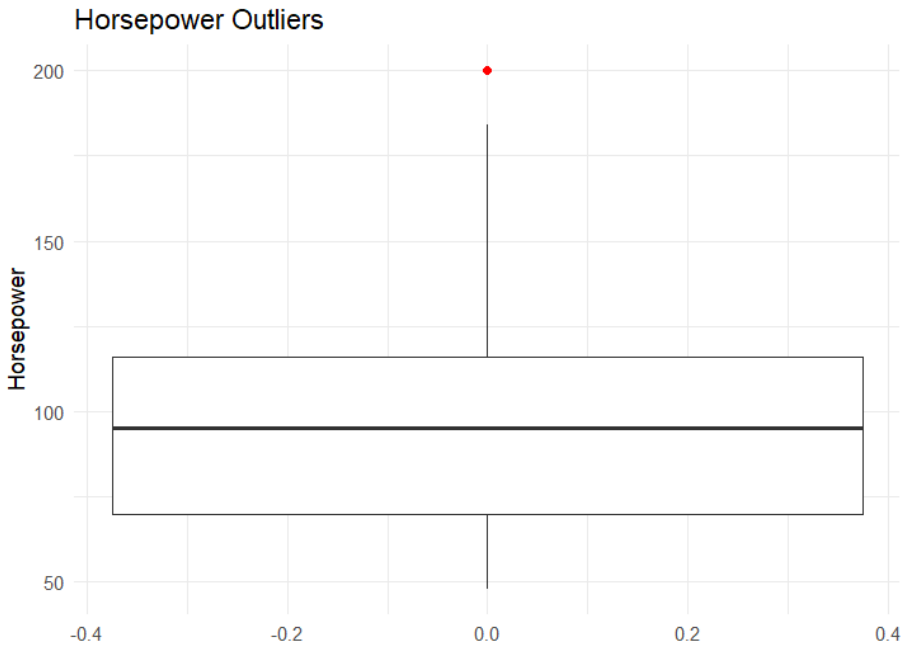

Figure 1: Outliers in Target Variable (Horsepower)

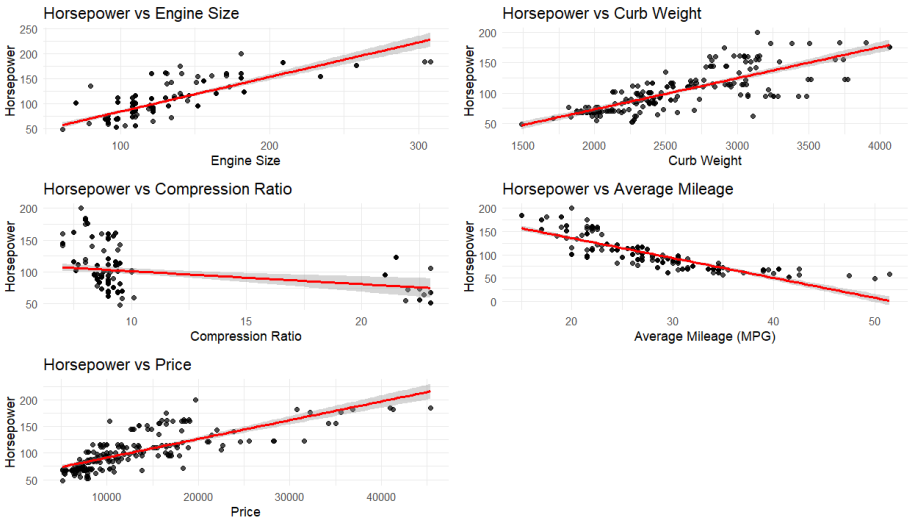

Figure 2: Numerical Predictors Correlation with Horsepower

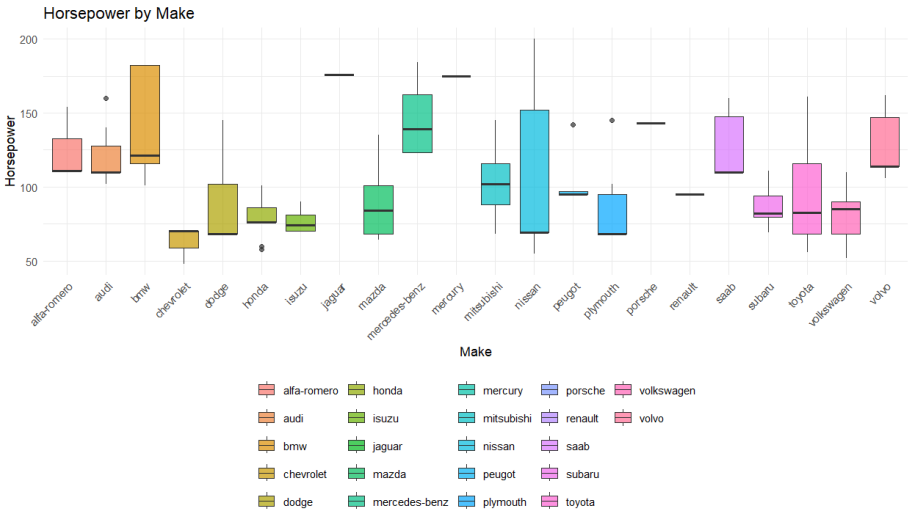

Figure 3: Car Make Correlation with Horsepower

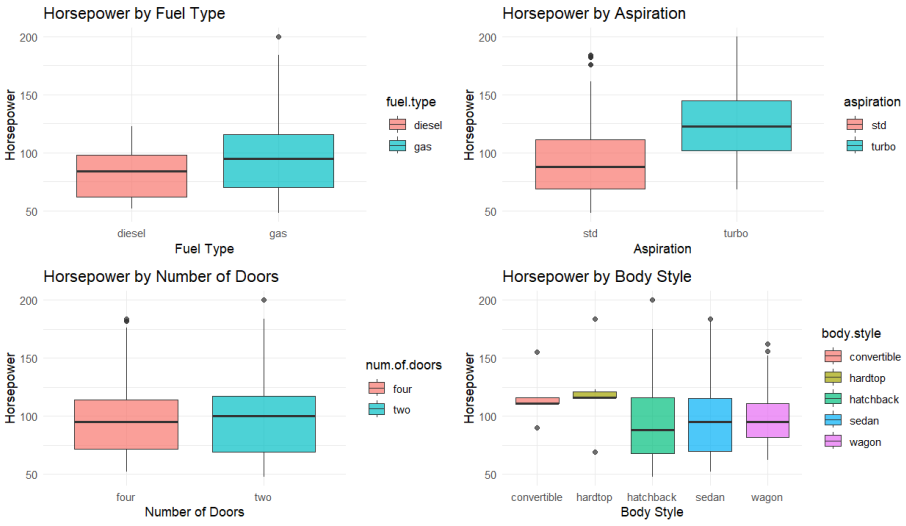

Figure 4: Categorical Predictors Correlation with Horsepower 1

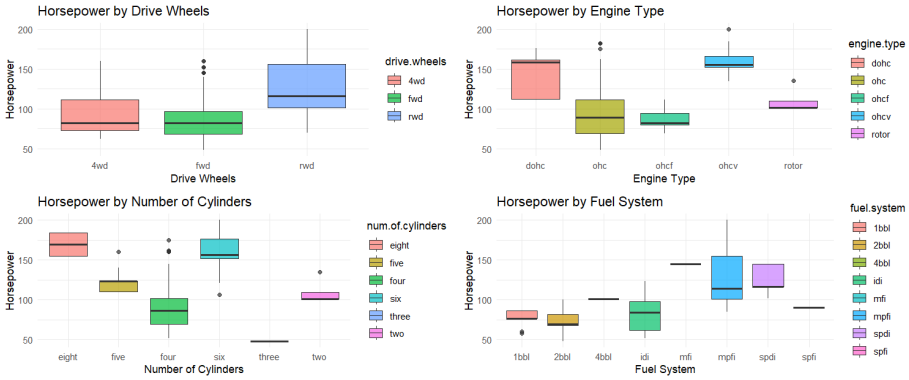

Figure 5: Categorical Predictors Correlation with Horsepower 2

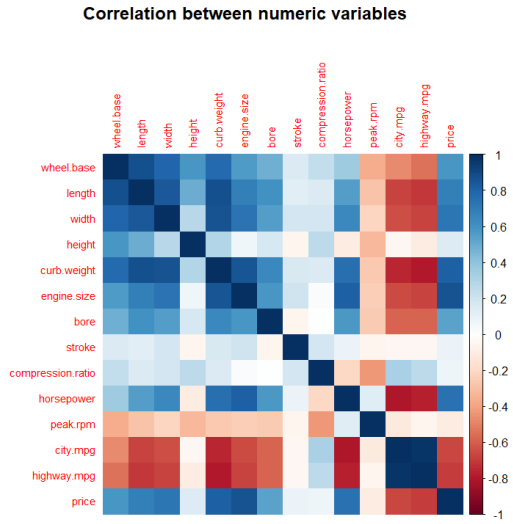

Figure 6: Numerical Predictors Correlation Matrix

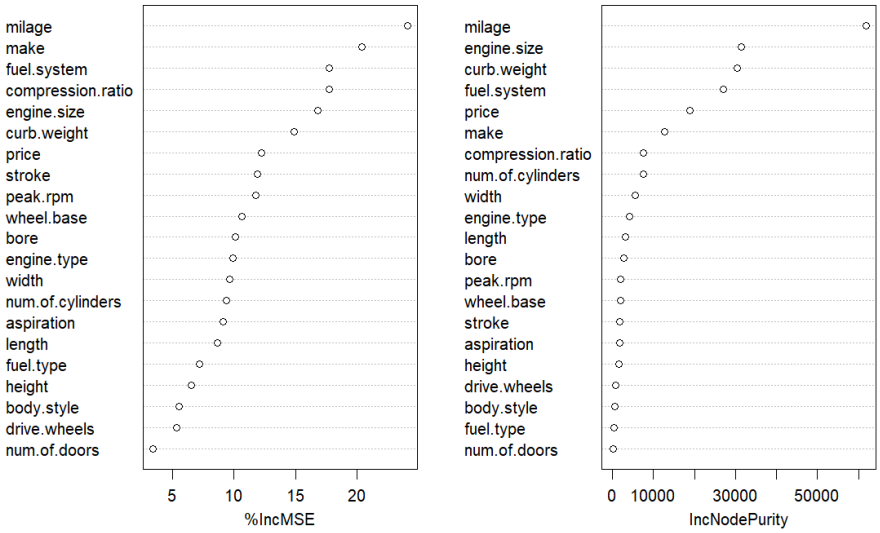

Figure 7: Feature Importances for feature selection

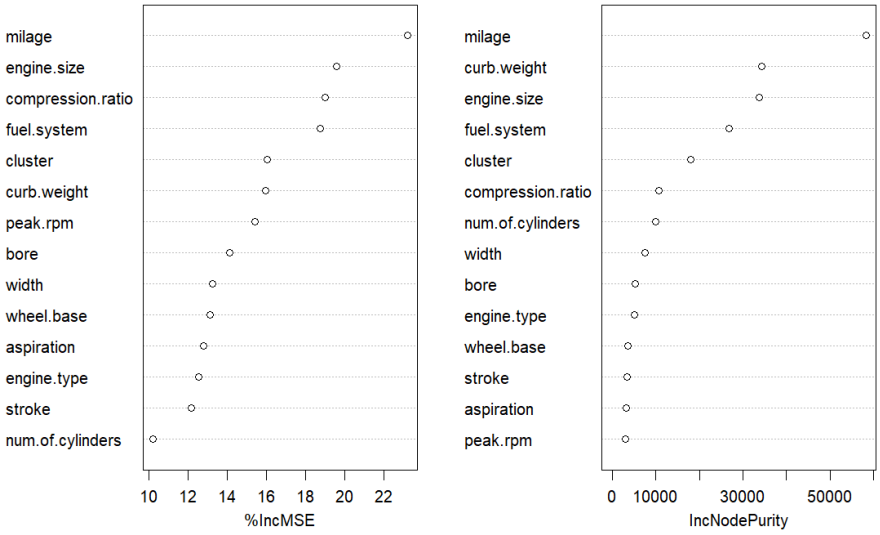

Figure 8: Final Model Feature Importances

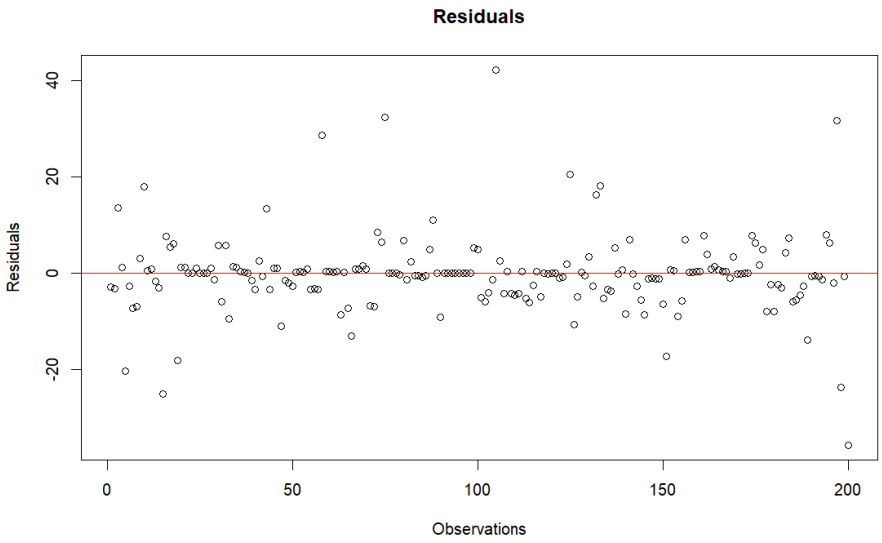

Figure 9: Residuals Plot for model predictions

R Code

```
library(dplyr)
library(ggplot2)
library(corrplot)
library(cluster)
library(factoextra)
library(randomForest)
library(Metrics)
library(gridExtra)
library(car)
library(caret)
library(clusterR)
library(tidyr)

set.seed(71)

# Importing dataset
carsData = read.csv("Automobile_data.csv")

# Removing the irrelevant columns
data = carsData %>%
  select(-symboling, -normalized.losses)

# Replace "?" with NA for all columns
data[data == "?"] = NA

# Check for missing values or '?' in the dataset
data %>%
  summarise(across(everything(), ~ sum(is.na(.))))

# Convert columns with missing values to appropriate data types
data$horsepower = as.numeric(data$horsepower)
data$peak.rpm = as.numeric(data$peak.rpm)
data$num.of.doors = as.character(data$num.of.doors)
data$price = as.numeric(data$price)
data$bore = as.numeric(data$bore)
data$stroke = as.numeric(data$stroke)

# Handle missing values for numerical columns by replacing with the
    median
data$horsepower[is.na(data$horsepower)] = median(data$horsepower, na.rm
    = TRUE)
data$peak.rpm[is.na(data$peak.rpm)] = median(data$peak.rpm, na.rm = TRUE)
data$price[is.na(data$price)] = median(data$price, na.rm = TRUE)
data$bore[is.na(data$bore)] = median(data$bore, na.rm = TRUE)
data$stroke[is.na(data$stroke)] = median(data$stroke, na.rm = TRUE)
```

```
# Handle missing values for categorical columns by replacing with the
    highest occuring value
data$num.of.doors[is.na(data$num.of.doors)] = "four"

# Replacing a typo with the most occuring value
data$engine.type[data$engine.type == "l"] = "ohc"
# Check again for missing values
data %>%
  summarise(across(everything(), ~ sum(is.na(.))))

# Converting categorical columns into factors
categoricalColumns = names(data)[!sapply(data, is.numeric)]

for (col in categoricalColumns) {
  data[[col]] = as.factor(data[[col]])
}

attach(data)

# Exploratory Data analysis

# Summary Statistics
summary(data)

# Correlation Analysis (for numeric variables)
numeric_vars = data %>%
  select_if(is.numeric)

# Checking and removing highly correlated variables
cor_matrix = cor(numeric_vars, use = "complete.obs")
corrplot(cor_matrix, method = "color", tl.cex = 0.75, title =
    "Correlation between numeric variables", mar = c(0, 0, 1.5, 0))

# Extracting highly correlated variables
highlyCorrelated = findCorrelation(cor_matrix, cutoff = 0.9)
highlyCorrelated
colnames(cor_matrix)[highlyCorrelated]

#Since highway and cily mileage is exactly correlated to each other, we
    will take average and then
#remove both columns
data = data %>%
  mutate(mileage = rowMeans(select(., highway.mpg, city.mpg), na.rm =
      TRUE))
data = data %>%
  select(-highway.mpg, -city.mpg)

# For outliers detection of Target Variable (Horsepower)
ggplot(data, aes(y = horsepower)) +
  geom_boxplot(outlier.color = "red", outlier.shape = 16) +
```

```
  theme_minimal() +
  labs( title = "Horsepower Outliers", y = "Horsepower")

# Removing outliers
data$zScoreHP = scale(data$horsepower)

# Remove outliers based on Z-score (greater or less than 2.5)
data = data %>%
  filter(abs(zScoreHP) <= 2.5)

data = data %>%
  select(-zScoreHP)

attach(data)
# Relationships Between Horsepower and Other Variables
# a. Scatter plots for numeric variables
ggplot(data, aes(x = engine.size, y = horsepower)) +
  geom_point(alpha = 0.7) +
  geom_smooth(method = "lm", color = "red") +
  theme_minimal() +
  labs(title = "Horsepower vs Engine Size", x = "Engine Size", y =
      "Horsepower")

ggplot(data, aes(x = curb.weight, y = horsepower)) +
  geom_point(alpha = 0.7) +
  geom_smooth(method = "lm", color = "red") +
  theme_minimal() +
  labs(title = "Horsepower vs Curb Weight", x = "Curb Weight", y =
      "Horsepower")

ggplot(data, aes(x = compression.ratio, y = horsepower)) +
  geom_point(alpha = 0.7) +
  geom_smooth(method = "lm", color = "red") +
  theme_minimal() +
  labs(title = "Horsepower vs Compression Ratio", x = "Compression
      Ratio", y = "Horsepower")

ggplot(data, aes(x = mileage, y = horsepower)) +
  geom_point(alpha = 0.7) +
  geom_smooth(method = "lm", color = "red") +
  theme_minimal() +
  labs(title = "Horsepower vs Average Mileage", x = "Average Mileage
      (MPG)", y = "Horsepower")

ggplot(data, aes(x = price, y = horsepower)) +
  geom_point(alpha = 0.7) +
  geom_smooth(method = "lm", color = "red") +
  theme_minimal() +
  labs(title = "Horsepower vs Price", x = "Price", y = "Horsepower")
```

```
# b. Box plots for categorical variables
ggplot(data, aes(x = make, y = horsepower, fill = make)) +
  geom_boxplot(alpha = 0.7) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  theme(legend.position = "bottom", legend.title = element_blank()) +
  labs(title = "Horsepower by Make", x = "Make", y = "Horsepower")

ggplot(data, aes(x = fuel.type, y = horsepower, fill = fuel.type)) +
  geom_boxplot(alpha = 0.7) +
  theme_minimal() +
  labs(title = "Horsepower by Fuel Type", x = "Fuel Type", y =
      "Horsepower")

ggplot(data, aes(x = aspiration, y = horsepower, fill = aspiration)) +
  geom_boxplot(alpha = 0.7) +
  theme_minimal() +
  labs(title = "Horsepower by Aspiration", x = "Aspiration", y =
      "Horsepower")

  ggplot(data, aes(x = num.of.doors, y = horsepower, fill =
      num.of.doors)) +
  geom_boxplot(alpha = 0.7) +
  theme_minimal() +
  labs(title = "Horsepower by Number of Doors", x = "Number of Doors", y
      = "Horsepower")

ggplot(data, aes(x = body.style, y = horsepower, fill = body.style)) +
  geom_boxplot(alpha = 0.7) +
  theme_minimal() +
  labs(title = "Horsepower by Body Style", x = "Body Style", y =
      "Horsepower")

ggplot(data, aes(x = drive.wheels, y = horsepower, fill = drive.wheels))
    +
  geom_boxplot(alpha = 0.7) +
  theme_minimal() +
  labs(title = "Horsepower by Drive Wheels", x = "Drive Wheels", y =
      "Horsepower")

ggplot(data, aes(x = engine.location, y = horsepower, fill =
    engine.location)) +
  geom_boxplot(alpha = 0.7) +
  theme_minimal() +
  labs(title = "Horsepower by Engine Location", x = "Engine Location", y
      = "Horsepower")
# After outliers, we can ignore this variable since there is no variation
data = data %>%
  select(-engine.location)
attach(data)
```

```
ggplot(data, aes(x = engine.type, y = horsepower, fill = engine.type)) +
  geom_boxplot(alpha = 0.7) +
  theme_minimal() +
  labs(title = "Horsepower by Engine Type", x = "Engine Type", y =
      "Horsepower")

ggplot(data, aes(x = num.of.cylinders, y = horsepower, fill =
    num.of.cylinders)) +
  geom_boxplot(alpha = 0.7) +
  theme_minimal() +
  labs(title = "Horsepower by Number of Cylinders", x = "Number of
      Cylinders", y = "Horsepower")

ggplot(data, aes(x = fuel.system, y = horsepower, fill = fuel.system)) +
  geom_boxplot(alpha = 0.7) +
  theme_minimal() +
  labs(title = "Horsepower by Fuel System", x = "Fuel System", y =
      "Horsepower")

# Train a Random Forest model for feature selection
rf_model = randomForest(horsepower ~ ., data = data, importance = TRUE)

# Check feature importance
importance(rf_model)

# Plot feature importance
varImpPlot(rf_model)

# Create a subset with only the important features
dataSelected = data %>%
  select(-num.of.doors, -body.style, -height, -drive.wheels)

# Removing price for a more specs focused analysis
dataSelected = dataSelected %>%
  select(-price)

# Apply hierarchical clustering
gowerDist = daisy(dataSelected, metric = "gower")
hc = hclust(gowerDist)
plot(hc)

# Adding clusters
clusters = cutree(hc, k = 5)

# Add the clusters to your data
dataSelected$cluster = as.factor(clusters)
dataSelected$cluster

# Cluster Numerical summary
```

```
clusterNumSummary <- dataSelected %>%
  group_by(cluster) %>%
  summarise(across(where(is.numeric), list(mean = ~ mean(.), sd = ~
      sd(.)), .names = "{.col}_{.fn}"))

categorical_vars = c("make", "fuel.type", "aspiration", "body.style",
    "drive.wheels", "num.of.cylinders")

# Mode function to calculate mods of categorical variables
mode_function <- function(x) {
  uniqx <- unique(x)
  uniqx[which.max(tabulate(match(x, uniqx)))]
}

# Calculating modes and adding it to mode_resukt
mode_results = list()

for (var in categorical_vars) {
  mode_results[[var]] = dataSelected %>%
    group_by(cluster) %>%
    summarise(mode = mode_function(!!sym(var)))
}

finalSummary = clusterNumSummary

# Add categorical mode results
for (var in categorical_vars) {
  finalSummary[[paste(var, "mode", sep = "_")]] =
      mode_results[[var]]$mode
}

# Final Summary
finalSummary

# Writing it to a csv file to explore it
write.csv(finalSummary, "summary.csv", row.names = FALSE)

# Train the Random Forest model
rf_model1 = randomForest(horsepower ~ ., data = dataSelected, importance
    = TRUE)
rf_model1

importance(rf_model1)
varImpPlot(rf_model1)

# Removing the weakest feature
dataSelected = dataSelected %>%
  select(-fuel.type )
```

```
# Train the Random Forest model
rf_model2 = randomForest(horsepower ~ ., data = dataSelected, importance
    = TRUE, ntree = 500, do.trace=50)
rf_model2

importance(rf_model2)
varImpPlot(rf_model2)

dataSelected1 = dataSelected %>%
  select(-make, -length)


# Train the Random Forest model
rf_model3 = randomForest(horsepower ~ ., data = dataSelected1,
    importance = TRUE, ntree = 500, do.trace=50)
rf_model3

importance(rf_model3)
varImpPlot(rf_model3)

prediction = rf_model3$predicted
actual = dataSelected1$horsepower

residuals = actual - prediction
residuals
plot(residuals, main = "Residuals", ylab = "Residuals", xlab =
    "Observations")
abline(h = 0, col = "red")
```