Machine Learning Applications in the Automotive Industry: Predictive Modeling for Vehicle Prices and Market Segmentation

Bessie Pengjin Wang

Barnard College, Columbia University

Introduction

The primary focus of this report is to predict the resale price of cars in the United States by constructing and evaluating machine learning models. Additionally, the project aims to identify influential factors affecting the resale prices, segment customers based on clusters of similar cars, and recommend vehicles to users due to their preferences.

The analysis includes seven parts:

1. Data cleaning, exploration, and partitioning into training and testing subsets.
2. Feature selection to identify a suitable set of predictors for the models.
3. Fitting various machine learning models to the two sets of predictors, respectively.
4. Model evaluation.
5. Identification of influential factors affecting resale prices.
6. Identification of vehicle clusters for customer segmentation.
7. Car recommendation and prediction of car prices based on user preferences.

Part I. Data Collection and Exploration

The dataset, titled “Car Features and MSRP,” was retrieved from Kaggle. Named "vehicle," the dataset encompasses various details about used cars, including make, model, year, engine horsepower (Engine.HP), engine cylinders (Engine.Cylinders), number of doors (Number.of.Doors), highway miles per gallon (highway.MPG), city miles per gallon (city.mpg), popularity, and manufacturer's suggested retail price (MSRP).

Initial inspection of the dataset was conducted using functions such as head(), tail(), summary(), and str(). The dataset comprises 11914 rows, each representing a distinct used car and its features, and 16 columns, indicating the presence of 15 potential predictors.

In the data cleaning process, missing values within the dataset were detected using the is.na() function. A total of 105 missing values were identified: 69 in the Engine.HP column,

30 in the Engine.Cylinders column, and 6 in the Number.of.Doors column. These missing values were addressed using an imputation method. As the data for numeric predictors, Engine.HP and Engine.Cylinders, were right-skewed, the median value was employed for imputation to prevent bias. For the categorical predictor Number.of.Doors, the mode was used for imputation.

Further exploration of the data involved constructing a box plot for each numeric predictor to visualize their distribution. For example, the "Year" variable showed that cars were primarily from 2007 to 2016, with outliers dating back to the 1990s. Similarly, "Engine.HP" and "Engine.Cylinders" displayed comparable distributions, with horsepower generally ranging from 200 to 300. The number of engine cylinders mostly varied between 4 to 6. Both highway and city miles per gallon exhibited similar distributions, with medians around 26 and 18, respectively, and a few significant outliers (see box plots in Appendix A).

Potential correlations between predictors, such as between Engine.HP and Engine.Cylinders, and between Highway.MPG and City.MPG, were suspected to impact subsequent predictor selection. Thus, a correlation plot was constructed to examine the relationships among various variables. Relatively high correlations between Engine.HP and Engine.Cylinders (around 0.77), as well as Highway.MPG and City.MPG (around 0.89) are identified, which should be taken into the subsequent predictor selection process (see correlation plot in Appendix A).

Part II. Predictor Selection

To visualize the relationship between each numeric predictor and the outcome variable (MSRP), seven scatter plots were drawn (see scatterplots in Appendix B). As we can observe from the scatterplots, Year, Engine Horsepower and Engine Cylinders exhibit a

polynomial growth trend. Therefore, polynomial regression is taken into consideration in the subsequent process of building machine learning models.

Given the dataset contains 15 potential predictors with a mix of numeric and categorical variables, two approaches were used to select a subset of predictors.

The first subset of predictors comprises all numeric variables in the dataset: Year, Engine.HP, Engine.Cylinders, Highway.MPG, City.MPG, and Popularity. High correlations among these variables are not expected to significantly impact predictions in models such as linear regression, where multicollinearity is less problematic, and in regularized models, where correlations are managed with regularization penalties.

The dataset containing 7 numeric predictors is split into “train_set” and “test_set”, with a proportion of 80:20.

The second subset of predictors were selected by conducting a Random Forest analysis on feature importance of the whole dataset, ranked based on the increase in Mean Squared Error (MSE) when the values of a variable are permuted while all others are left unchanged, in decreasing order. Top ten features included: Engine.HP, Engine.Cylinders, Market.Category, Popularity, Vehicle.Size, Make, Model, Engine.Fuel.Type, highway.MPG, and Year.

Aiming to select a subset of five predictors, including both numerical and categorical, Engine.Cylinders was dropped due to its high correlation with the top-ranked feature, Engine.HP, possibly not adding much predicting power to our model. Market.Category and Model were also removed due to their high level of unique categorical values (72 and 915 respectively), which could lead to overfitting. Moreover, the subsequent Random Forest regression model cannot handle categorical variables with more than 53 levels. Therefore, the selected subset includes: Engine.HP, Popularity, Vehicle.Size, Make, and Engine.Fuel.Type.

Three categorical predictors, Make, Vehicle.Size, and Engine.Fuel.Type are transformed with the as.factor() function to aid future usage in models. The dataset containing top 5 predictors is split into “train_set_selected” and “test_set_selected”, with a proportion of 80:20.

Part III: Fitting Machine Learning Models

Three categories of models, each comprising two models (totaling six), were fitted to the data:

1. Basic linear and nonlinear models: linear, polynomial
2. Ensemble models: Random Forest, Gradient Boosting
3. Regularized models: Ridge and LASSO

1. Linear & Nonlinear Regression

For the linear model, the lm() function is utilized to fit all independent variables in the training set as predictors for the dependent variable MSRP. For the polynomial model, a 2nd-degree polynomial transformation is applied to the numeric predictors: Engine.HP, Year, and Popularity. Categorical predictors, Make and Vehicle.Size, are treated as factors.

2. Random Forest Regression & Gradient Boosting

Random Forest and Gradient Boosting are ensemble models in machine learning, which combine the predictions of multiple simpler models. Random Forest regression uses the package “randomForest" with the default tuning parameter of 500 trees. Gradient Boosting uses the package “gbm” with tuning parameters were set to Gaussian distribution, 500 trees, an interaction depth of 3, and a shrinkage rate of 0.1.

3. Ridge Regression & LASSO Regression

Both Ridge and LASSO regression were implemented using the “glmnet” package, which stands for "generalized linear model via penalized maximum likelihood". These

models add a penalty to the size of coefficients, to prevent overfitting and improve the model's performance on new data.

The distinction between Ridge and LASSO regression is determined by the alpha parameter: an alpha of 1 an alpha of 0 corresponds to Ridge, and an alpha of 1 corresponds to LASSO. Cross-validation is used to find the optimal tuning parameter lambda of both models, which controls the strength of the penalty.

Part IV. Model Comparison & Validation

To evaluate the models, evaluation metrics (Mean Squared Error - MSE, Root Mean Squared Error - RMSE, Mean Absolute Error - MAE, and R-squared) are calculated for each model using the test set. MSE and R-Squared are visualized through the bar plot below.

Linear and polynomial models generally exhibit a higher MSE value than the ensemble or regularized models, indicating a greater error in predictions. Notably, there was a considerable performance difference between using the seven numeric predictors and the top five predictors, with the latter providing more precise predictions.

Ensemble methods like Random Forest and Gradient Boosting show improved predictions with the seven numeric predictors compared to the linear and non-linear models, however, the performance difference between using the seven numeric predictors and the top five was less significant.

Regularized models, Ridge and LASSO, outperformed all other models when using the seven numeric predictors, exhibiting very low MSEs. Among the two, LASSO has slightly better performance than Ridge.

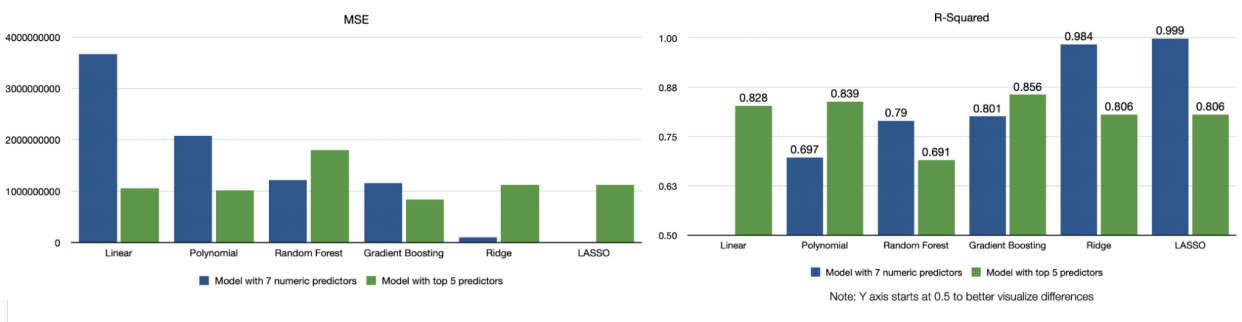
In terms of R-squared values, the linear model using seven numeric predictors had a notably low value, meaning the proportion of variance in the dependent variable MSRP that

can be explained by the independent variables is low. The R-squared of linear and polynomial by using top 5 predictors have a decent R-squared, both over 80%.

Gradient Boosting generally had a higher R-squared compared to Random Forest.

Ridge and LASSO, yielded exceptionally high R-squared values when using the seven numeric predictors, but their performance fell when using the top five predictors, outperformed by either linear, polynomial, or Gradient Boosting.

Overall, the LASSO model with seven numeric predictors performed the best in terms of both MSE and R-squared values. Therefore, LASSO will be used in predicting car MSRP onwards to produce more precise predictions. An alternative could be the Gradient Boosting model with top 5 predictors, since it has a balance of low MSE and high R-squared.



Part V. Influential Factor Identification

To gain insights into the factors influencing car prices, the Gradient Boosting model is used due to its easy-to-interpret relative influence scores for each feature. These relative influence scores represent the extent to which each variable contributes to the model's prediction, with higher scores meaning greater importance.

In our analysis, the Make of the car was found to have the highest relative influence score (66.66), suggesting that the brand of the car significantly impacts its MSRP. The next influential factor was Engine Horsepower' (29.08), implying that a car's power also substantially influences its price. Other variables, including Vehicle Size (1.591281), Popularity (1.43), and Engine Fuel Type (1.24), had relatively lower influence scores,

suggesting their impacts are less pronounced compared to 'Make' and 'Engine Horsepower' (visualized by relative influence plot in Appendix C).

It's worth noting that the results are generalized influential factors, as they consider all types of buyers and sellers as one group. In the following parts of the paper, clustering methods will be used to distinguish between different types of sellers and buyers to offer a more detailed understanding of influential factors across market segments.

Part VI. Clustering Using K-Means Clustering

This part of the report focuses on unsupervised learning, specifically clustering the vehicle data into distinct groups using K-means and visualizing the results with Principal Component Analysis (PCA).

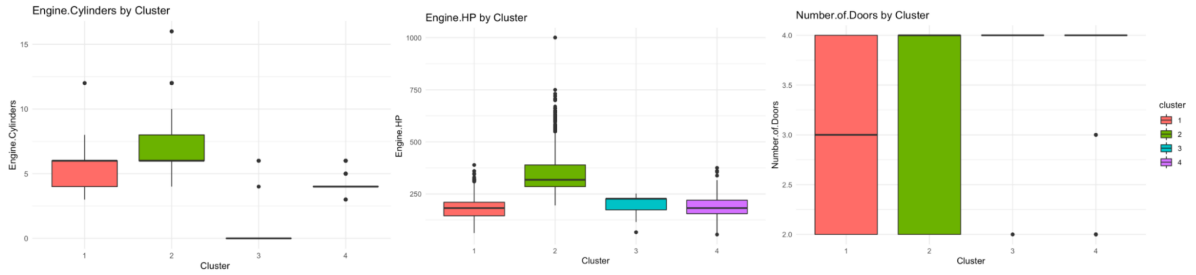
K-means clustering was conducted on the dataset following steps: data scaling, determining the optimal number of clusters (k) through the Elbow Method, and applying k-means clustering with k=4.

The Elbow Method calculates the within-cluster sum of squares (WSS) for different values of k and plotting these against an increasing value of k. As to our dataset, the plot did not clearly show an 'elbow', signifying a distinct optimal value for k. For the ease of interpretability, k was set to 4.

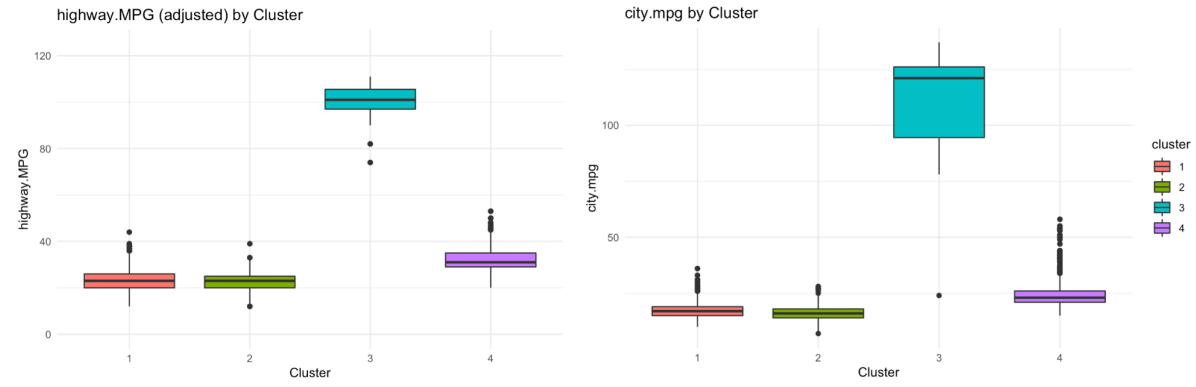
PCA was used to visualize the clustered, multi-dimensional data. The scatter plot of the first two principal components, color-coded by cluster assignment is drawn (for elbow plot and PCA visualization, see Appendix D).

To gain a deeper understanding of the characteristic differences between clusters, box plots were constructed for each numerical feature, grouped by cluster assignment. The numeric features were synthesized into four categories: functionality, fuel efficiency, year & popularity, and price.

Functionality: Refers to the number of engine cylinders, engine horsepower, and the number of doors. Group 2 vehicles generally have the highest engine horsepower and the most engine cylinders, showing a focus on power and performance. Vehicles in groups 1 and 2 exhibit a variety in the number of doors, suggesting a mix of different vehicle types, while groups 3 and 4 are characterized primarily by vehicles with four doors, indicating a potential focus on family or utility vehicles.

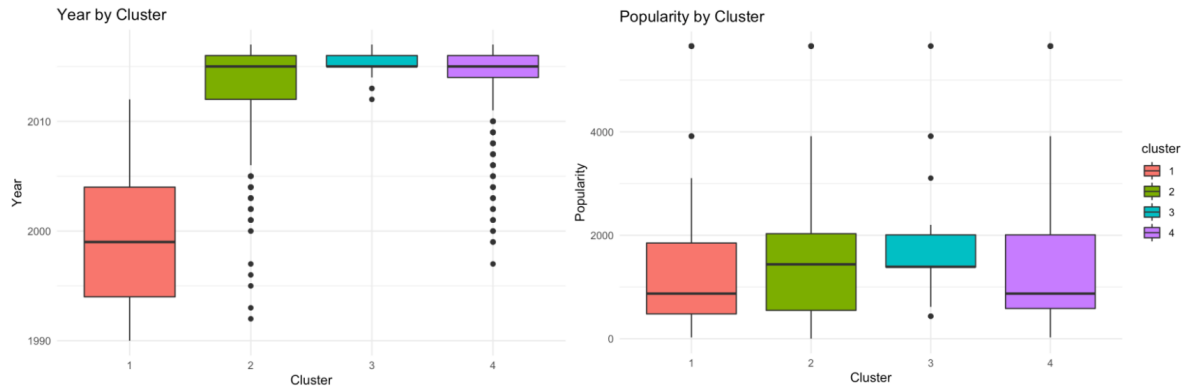


Fuel efficiency: The highway and city miles per gallon (MPG) are distributed similarly across the clusters, with Group 3 vehicles demonstrating significantly higher MPG. This suggests that Group 3 may consist of more fuel-efficient or potentially smaller vehicles. Highway MPG graph is adjusted to showcase most of the values since it has a few extreme outliers.

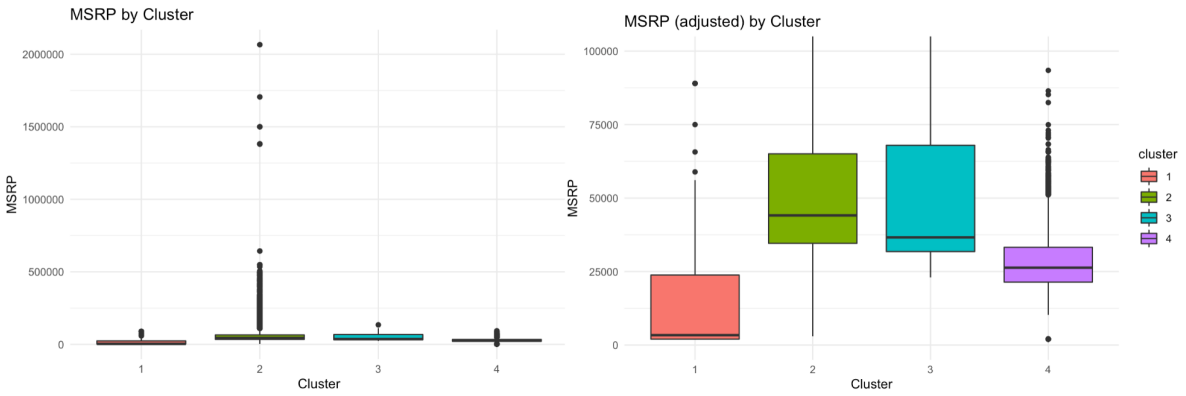


Year and Popularity: Group 1 predominantly contains older vehicles, with a median year around 1998. In contrast, Group 3 appears to contain the newest vehicles, with both its

median and lower IQR year close to 2015. In terms of popularity, Group 3 again stands out with generally higher popularity compared to the other groups.



Price: The Manufacturer's Suggested Retail Price (MSRP) shows variability across groups. Group 1 generally has the lowest prices, suggesting it may consist of older or budget models. Group 2, having some exceptionally high-priced outliers, exhibits a median MSRP similar to Group 3, but is slightly higher.



These findings may have further implications in marketing strategies, as potential buyer segments can be proposed into four groups:

Group 1: “Economy buyers”, who are more interested in older models with affordable prices; Group 2: “Performance Enthusiasts”, who favor cars with higher engine horsepower and cylinders.; Group 3: “Modern Efficiency Advocates”, who prefer newer, more popular, and fuel-efficient cars; Group 4: “All Rounders”, who don’t show a distinct focus on a certain aspect of cars.

Part VII. Car Recommendation and Car Price Prediction

The final part of this report focuses on practical applications of our data and machine learning models: personalized car recommendations and car price predictions.

First, a car recommendation algorithm is designed to match customer preferences with existing vehicle data. It utilizes Euclidean distance to measure the similarity between a customer's preferences and the attributes of each vehicle in our dataset. Customer preferences are specified for several features, including horsepower, cylinders, popularity, year, number of doors, highway miles per gallon, city miles per gallon, and MSRP. After computing the Euclidean distances, we sort the vehicles based on these distances. The top three unique brands are recommended to the user.

Secondly, a tool for predicting the Manufacturer's Suggested Retail Price (MSRP) of second-hand cars is introduced, using two of our most effective models, LASSO and Gradient Boosting. Users can input their desired car features, and the tool will output the predicted MSRP in a range.

Selected R code for the two widgets are attached in Appendix E.

Summary

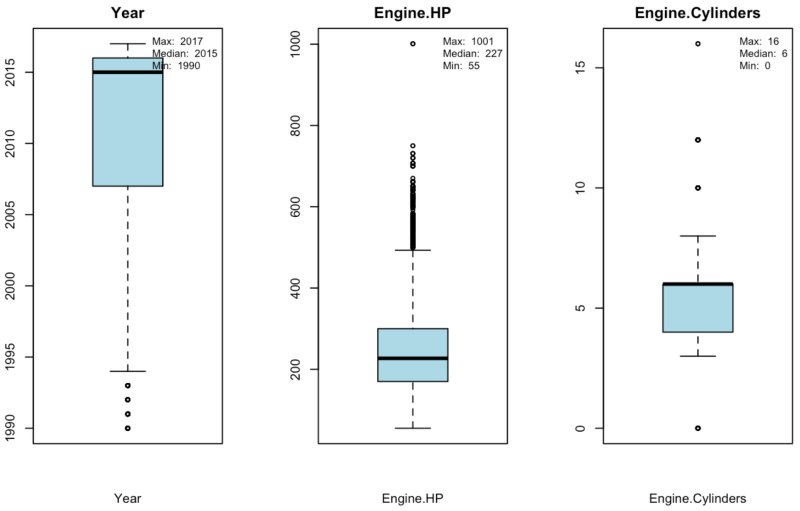
In this analysis, we applied machine learning models to predict vehicle prices in the second-hand car market. Our findings shows that the car brand and engine horsepower significantly influence the Manufacturer's Suggested Retail Price (MSRP) of cars. Using k-means clustering, we identified distinctive buyer groups, each with unique preferences for specific car features. To provide practical applications for our research, we used LASSO and Gradient Boosting models to develop tools for predicting car prices. These tools demonstrate the real-world usability and impact of our models.

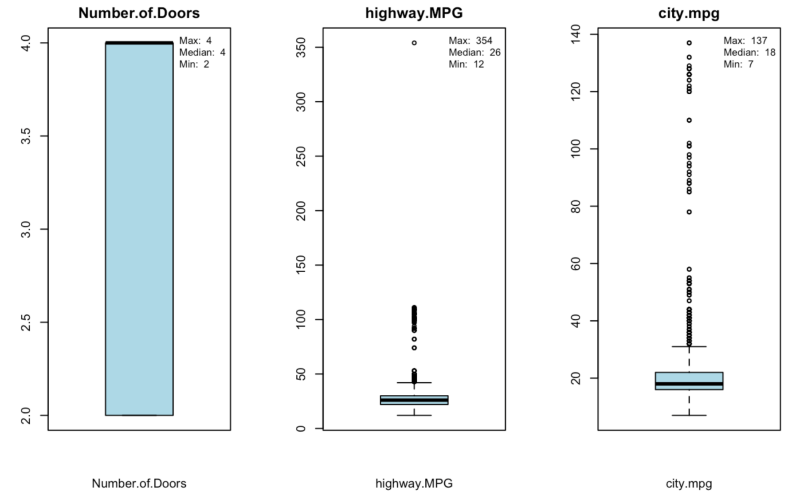
References

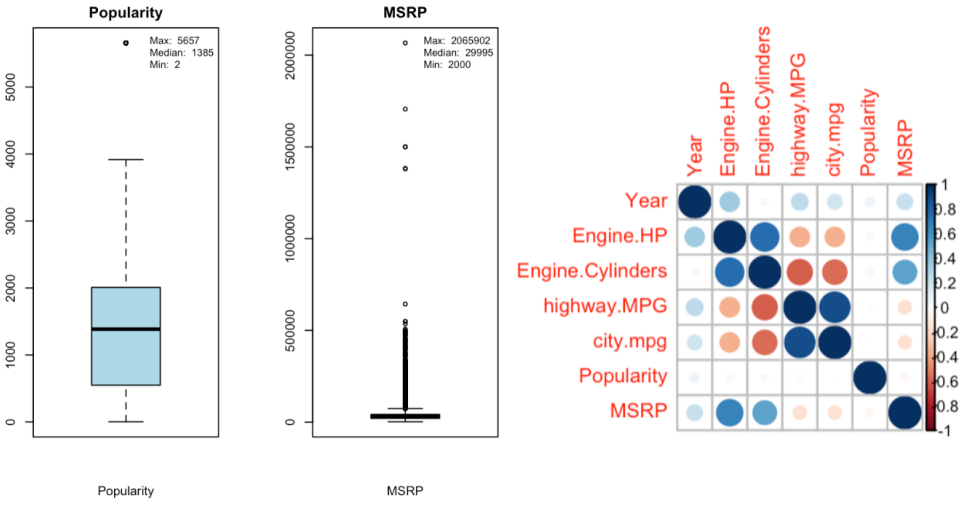
Cooper Union. (2017). Car specifications from 1992 to 2017 [Data set]. Kaggle. https://www.kaggle.com/datasets/CooperUnion/cardataset

Appendix A

Data exploration graphs: box plots and numeric correlation plot

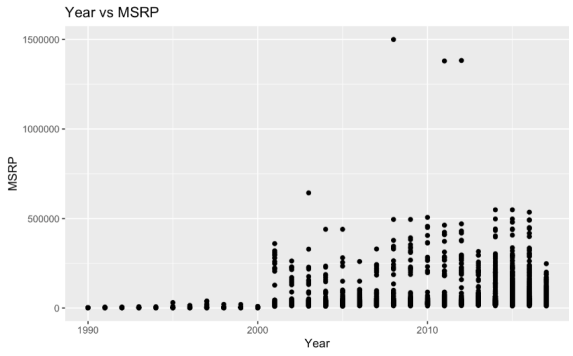


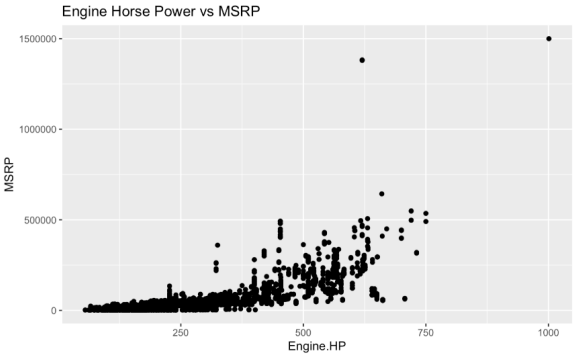


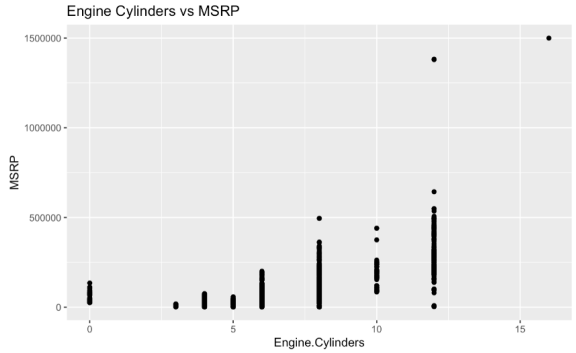


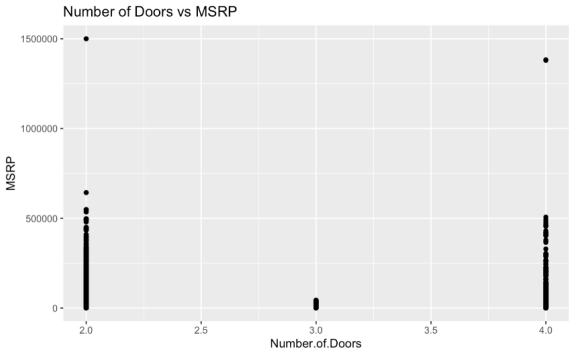
Appendix B

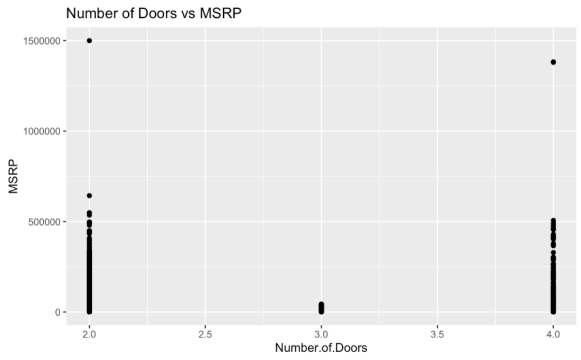
Data exploration graphs: numeric variable scatterplots

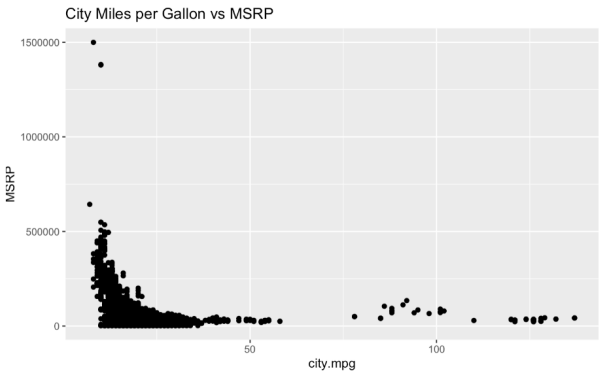


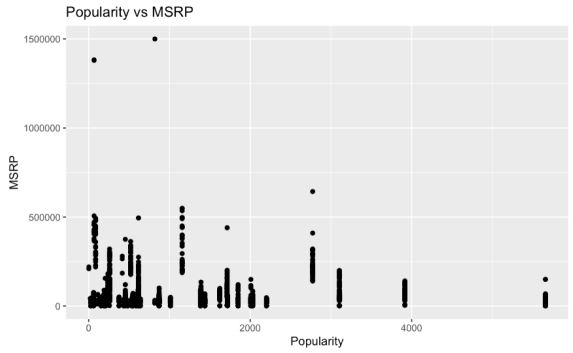






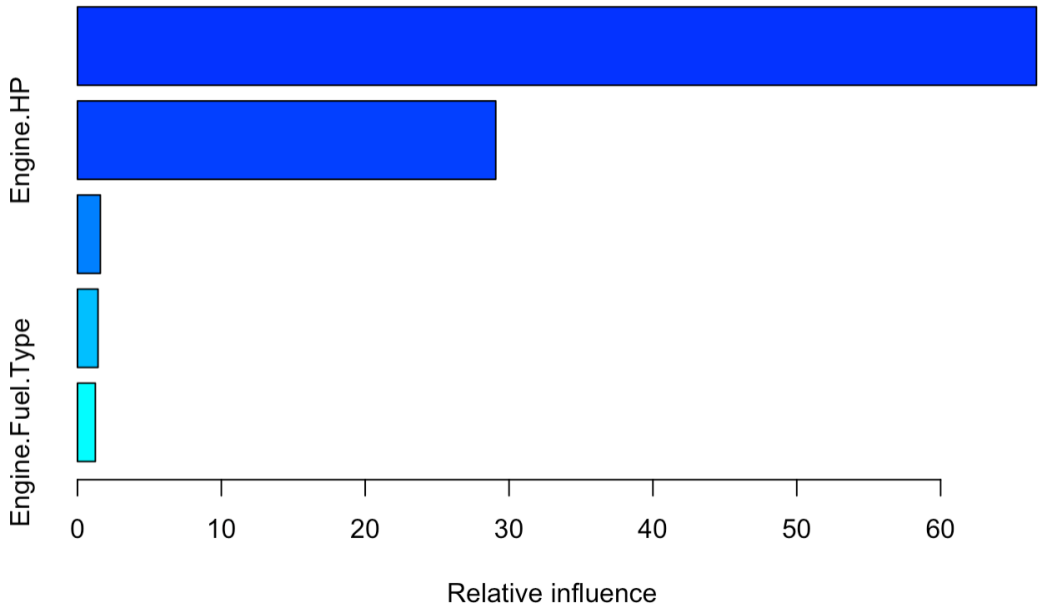






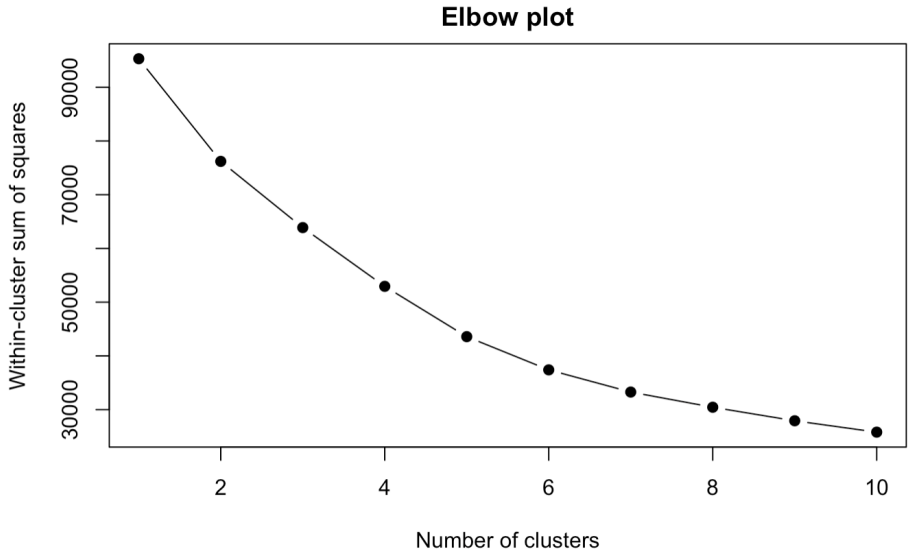
Appendix C

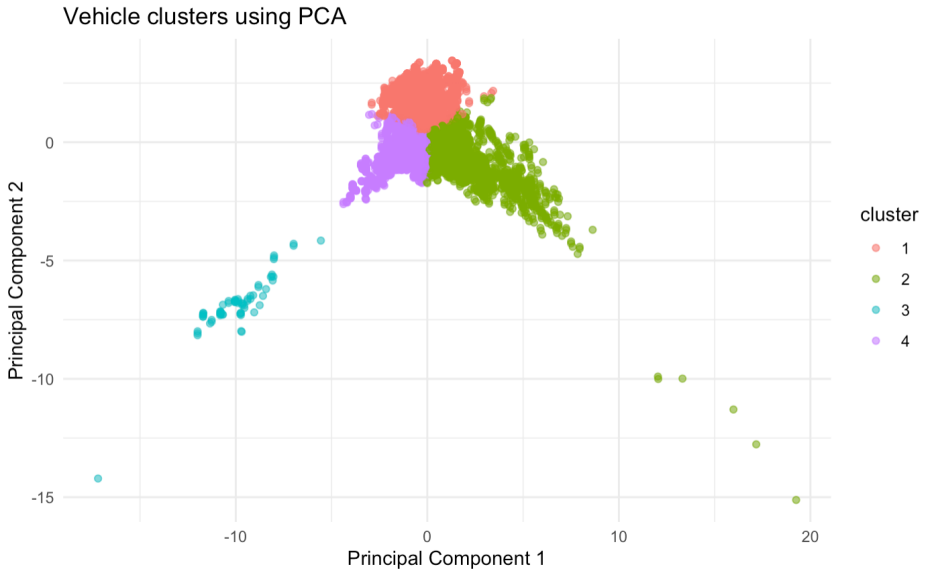
Feature selection graph: Gradient Boosting model relative influence



Appendix D

Clustering graphs: K-Means elbow plot and PCA cluster visualization





Appendix E

R Widget selected code: car recommendation and MSRP prediction

Car Recommendation

```{r}
# Example usage, user can adjust input values according to their preferences
customer_preferences <- data.frame(Engine.HP = 5, Engine.Cylinders = 4, Popularity = 100, Year = 2010, Number.of.Doors = 4, highway.MPG = 20, city.mpg = 20, MSRP = 2000)

# Scale dataset
customer_preferences_scaled <- scale(customer_preferences, center = attr(vehicle_numeric_scaled, "scaled:center"), scale = attr(vehicle_numeric_scaled, "scaled:scale"))

# Compute the distances
euclidean_distances <- apply(vehicle_numeric_scaled, 1, function(vehicle) sqrt(sum((vehicle - customer_preferences_scaled)^2)))
distances_and_brands <- data.frame(Distance = euclidean_distances, Brand = vehicle$Make)
sorted_distances_and_brands <- distances_and_brands[order(distances_and_brands$Distance),]

# Loop for 3 different brands
top_3_vehicles <- data.frame()
brands <- c()
for (i in 1:nrow(sorted_distances_and_brands)) {
  current_brand <- sorted_distances_and_brands$Brand[i]
  if (!(current_brand %in% brands)) {
    top_3_vehicles <- rbind(top_3_vehicles, vehicle[vehicle$Make == current_brand,][1,])
    brands <- c(brands, current_brand)
  }
  if (length(brands) == 3) {
    break
  }
}
top_3_vehicles
```

Sample output

	Make <chr>	Model <chr>	Year <int>	Engine.Fuel.Type <chr>	Engine.HP <int>	Engine.Cylinders <dbl>
36	Mercedes-Benz	190-Class	1991	regular unleaded	130	4
1288	Oldsmobile	Achieva	1996	regular unleaded	150	4
2342	Buick	Cascada	2016	premium unleaded (recommended)	200	4

Predicting MSRP

```{r}
predict_MSRP_lasso <- function(year, engine_hp, engine_cylinders, number_of_doors, highway_mpg, city_mpg, popularity) {
  user_input <- data.frame(Year = year,
                           Engine.HP = engine_hp,
                           Engine.Cylinders = engine_cylinders,
```

```
                    Number.of.Doors = number_of_doors,
                    highway.MPG = highway_mpg,
                    city.mpg = city_mpg,
                    Popularity = popularity
                    )
  user_mat <- as.matrix(user_input)
  predicted_MSRP <- predict(lasso_model, s = best_lambda_lasso, newx = user_mat)
  return(predicted_MSRP)
}
predict_MSRP_gbm <- function(year, engine_hp, engine_cylinders, number_of_doors, highway_mpg,
city_mpg, popularity, vehicle_size, engine_fuel_type, make) {
  user_input <- data.frame(Year = year,
                    Engine.HP = engine_hp,
                    Engine.Cylinders = engine_cylinders,
                    Number.of.Doors = number_of_doors,
                    highway.MPG = highway_mpg,
                    city.mpg = city_mpg,
                    Popularity = popularity,
                    Vehicle.Size = vehicle_size,
                    Engine.Fuel.Type = engine_fuel_type,
                    Make = make
                    )
  predicted_MSRP <- predict(gbm_model_selected, newdata = user_input, n.trees = 500)
  return(predicted_MSRP)
}

# Example usage, user can adjust input values according to their preferences
year <- 2000
engine_hp <- 300
engine_cylinders <- 4
number_of_doors <- 4
highway_mpg <- 30
city_mpg <- 20
popularity <- 2000
vehicle_size <- "Compact" # Enter "Compact" "Midsize", or "Large"
engine_fuel_type <- "diesel" # For simplicity, enter "diesel", "electric", or "natural gas"
make <- "Volvo" # Enter a car brand, as long as the brand is in the dataset
predicted_MSRP_LASSO <- predict_MSRP_lasso(year, engine_hp, engine_cylinders,
number_of_doors, highway_mpg, city_mpg, popularity)
predicted_MSRP_GB <- predict_MSRP_gbm(year, engine_hp, engine_cylinders, number_of_doors,
highway_mpg, city_mpg, popularity, vehicle_size, engine_fuel_type, make)
cat("Your dream car's price might be around", predicted_MSRP_LASSO, " ~ ",
predicted_MSRP_GB)
```

Sample output

```
Your dream car's price might be around 49052.95  ~  39822.47
```