Prediction of Fuel Efficiency for Vehicles

***Abstract*—**In this paper, we evaluate the behavior and performance of several classification methods for classifying the city-cycle fuel consumption in miles per gallon based on other specification of a car. We implement the codes in R, and we use cross validation as evaluation method, and the misclassification rate as evaluation metric.

1 INTRODUCTION

The idea is to implement a classification learner to classify the automobiles into two groups: high miles per gallon or low miles per gallon(MPG). This learner can be part of an online Vehicle Fitness Certificate system. The system holds profile for the vehicles on the road in a specific city. First, feature extraction is performed to identify the key information and generate the vector representation of vehicles specification. These vectors are fed to a classification learner to estimate if the miles per gallon of the vehicle are high or low. Upon issuing a call for re-certification, the exact MPG can be measured in a mechanic shop.

We evaluate linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), naive bayes regression, logistic regression, and k-nearest neighbor.

Problem Statement: The objective of our analysis is to classify the automobiles into high or low MPG based on other specification of the vehicles. We define high MPG as if the greater than or equal to the median MPG of all cars in database, and low if otherwise.

Data Set: The dataset provided includes 398 records and 8 variables. For each record, there are 3 multivalued discrete and 5 continuous attributes including: number of cylinders, displacement, horsepower, acceleration, overall weight of the vehicle, model year, origin, and car name (Quinlan, 1993). We remove the car name from our analysis to protect from discrimination against brands. We define

$$mpg01 = 1 \;\; if \; MPG \;\geq median(MPG), \qquad mpg01 = 0 \;\; if \; MPG \; < median(MPG)$$

2 EXPLORATORY DATA ANALYSIS

We start our analysis by perfuming initial investigations on data to understand the features and to discover any patterns or anomalies in the data. We use summary statistics and visual representations to investigate the variation of features.

Size of dataset: Training set includes 353 records and test set includes 39 records.

Variable Range and Outliers: The features are in different range. Thus, it is necessary to standardize the variables. we investigate the outliers using boxplot but we do not to remove the outliers. We may have rare values for some make and brands, but it is still expected and indicates of a vehicle specification.

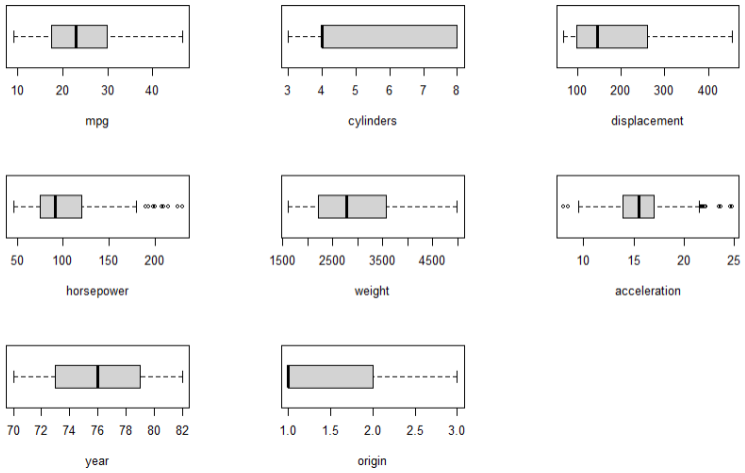

Figure 1: Boxplot; visualization; We can see variables are in different ranges.

Distributions: The histogram is skewed to the right, and even removing one or two observations, doesn't cause the distribution to be normal.

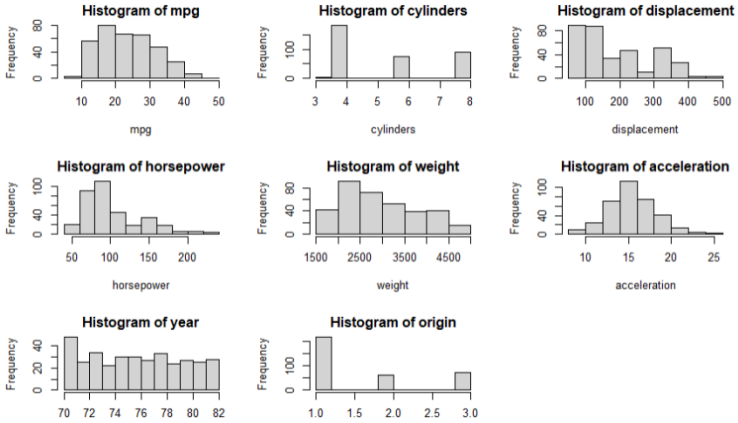

Figure 2: Distribution of MPG is not normal.

Even though, we investigate MPG variation, we use only mpg01 in our learners.

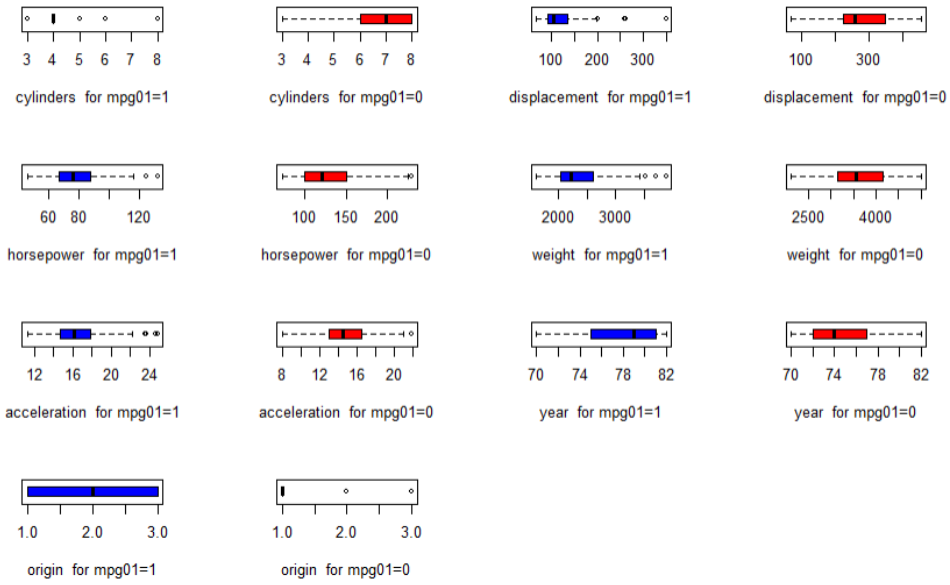

Figure 3: Boxplot; visualization for each class; verifying if there is Equal Variance.

Figure 2 illustrates the box plot for each class of MPG. We can see that for each feature the boxplot for class 1 and 2 are clearly different. This is an **indication** of non-equal variances. This can impact the performance of LDA classifier.

Correlation: Next, we investigate the correlation between each pair of features. But we do not remove the highly correlated features. Note that MPG is highly correlated to cylinders, displacement, horsepower, acceleration, and weight. Moreover, we can see that cylinders, displacement, horsepower, and weight are highly correlated.

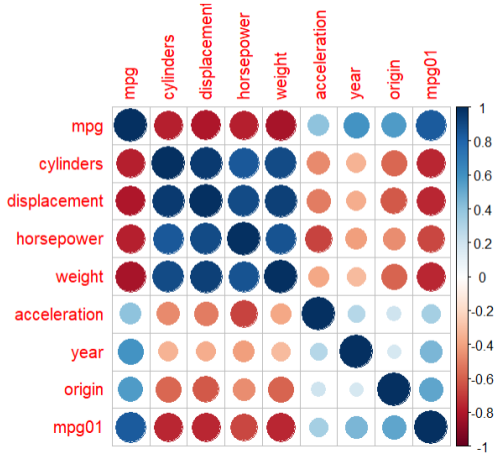

Figure 4: Correlation Heat map

Variation: Next, we explore the variation of each predictor for different response. The goal is to develop an **intuitive** understanding about the data.

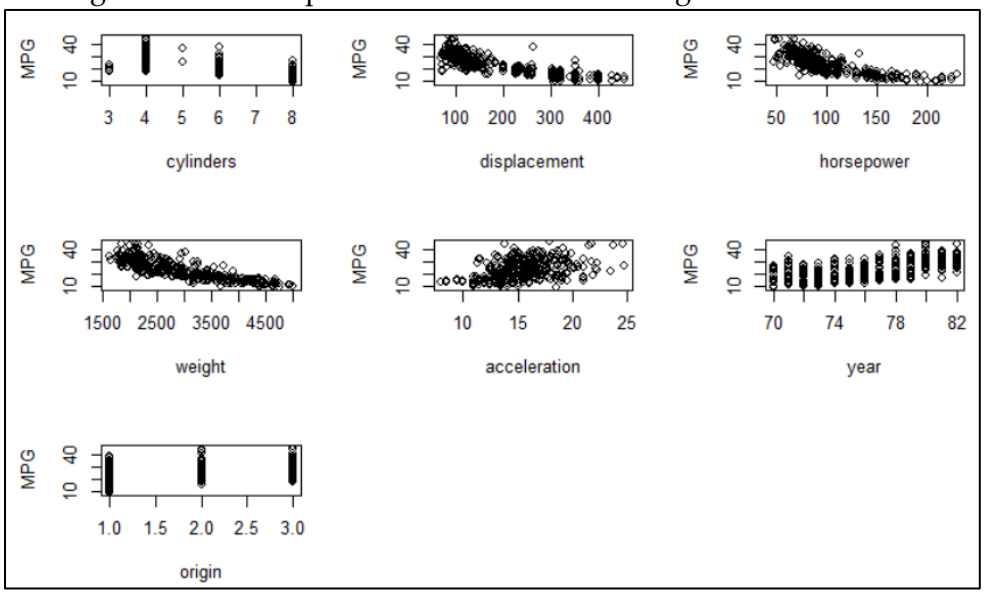

Figure 5: variation of MPG vs. each feature.

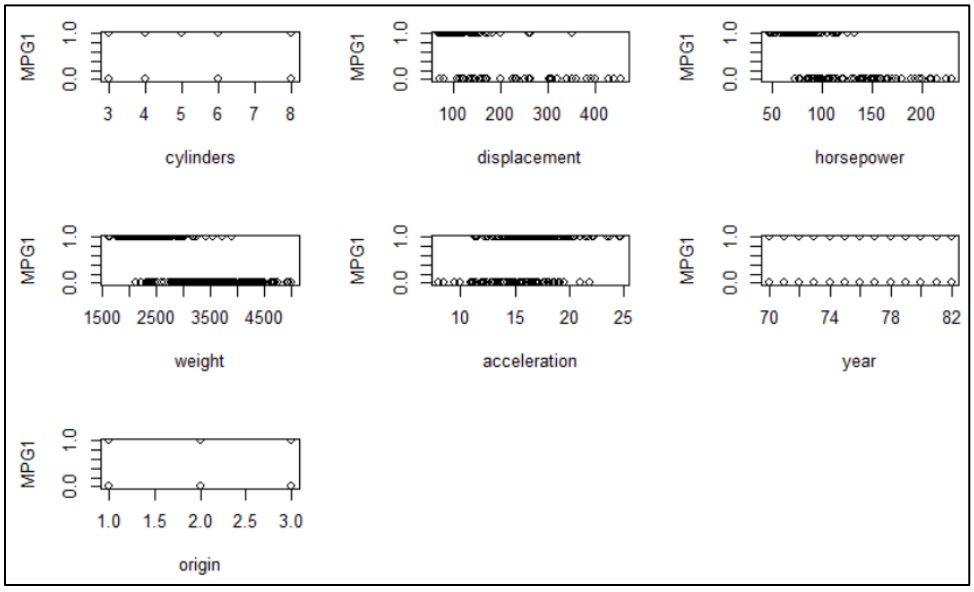

Figure 6: Variation of response/mpg01 vs. each feature.

From figure 4-6, we **anticipate** that *displacement*, *horsepower*, *weight*, and *acceleration* would be more useful in predicting mpg01. From figure 5, it seems 'year' won't be useful, and that is counter intuitive.

3 METHODOLOGY

First, we feed the *displacement*, *horsepower*, *weight*, and *acceleration* features to different classification models: LDA, QDA, Naive Bayes, Logistic Regression, and KNN. 'mpg01' is used as response value. Then, we calculate and compare the misclassification rate for both training and testing set. Misclassification occurs when the predicted class doesn't match with the actual class. For KNN, we used K= 1,3,5,7,9,11,13 and 15.

Standardization: We standardize our training set, but not the target value. Later, we perform standardization on test set using the mean and variance from training set. Note that we do not perform standardization on whole set to avoid information leakage from testing set to training set (impact on mean and sigma).

Cross-Validation: To properly compare these methods, we utilize Monte Carlo Cross-Validation algorithm with B=100. In each step, we randomly split the dataset into training and test set with a 9:1 ratio. Then, we standardize the data as discussed, train the model, and calculate the misclassification rate.

Finally, we compute and compare the "average" performances of each model.

4 RESULTS

Results from different models are listed in Table 1. Misclassification or error rate is recorded as percentage, and up to 4 decimal places. We can see that error rate for KNN is lower than error rate for all other models. Best error rate seems to be for KNN with K = 5. After KNN, QDA seems to outperform the rest. Note that Standard deviation doesn't vary significantly across different models.

	Without CV		**With CV**	
Method	**Training Error Rate**	**Testing Error Rate**	**Mean Error Rate**	**SD Error Rate**
LDA	10.2	15.38	11.1538	4.8669
QDA	10.2	10.26	9.8973	4.5519
Naive Bayes	11.05	17.95	11.1021	4.5671
Logistic Regression	9.92	12.82	10.0767	4.9741

KNN (k=1, overfit)	0	10.26	9.692	4.589
KNN (k=3, overfit)	5.95	10.26	8.5636	4.5846
KNN (k=5)	**6.8**	**7.69**	**8.4871**	**4.5331**
KNN (k=7)	7.37	10.26	8.4877	4.4441
KNN (k=9)	8.5	10.26	8.7436	4.39
KNN (k=11)	8.22	10.26	8.7432	4.5826
KNN (k=13)	8.78	12.82	8.8718	4.6165
KNN (k=15)	9.07	12.82	8.9231	4.6744

Table 1: mean squared error for each method without using cross-validation.

5 FINDINGS

It is obvious the nature of the relationship between the variables and response is non-linear. KNN doesn't assume any specific shape of the decision boundary, and that's why KNN has the lowest misclassification. Even though, KNN performs better, it does not specify which features are more important. It is more challenging to present the KNN findings to audience with management and business background. Next, QDA performs better because it assumes a quadratic decision boundary. Logistic regression, LDA and Naive Bayes are linear models and underperform KNN and QDA if decision boundaries are highly non-linear. LDA assumes that Gaussian distribution for each class and that all the classes share the same covariance matrix. We illustrated in Figure 3 that this assumption is not applicable here. Native bay assumes independence among features whereas displacement, horsepower, and weight are highly correlated based on the correlation heat map in Figure 4. Both assumptions are not met here, and that is why both models underperform logistic regression.

6 REFERENCES

1. Quinlan,R. (1993). Combining Instance-Based and Model-Based Learning. In Proceedings on the Tenth International Conference of Machine Learning, 236-243, University of Massachusetts, Amherst. Morgan Kaufmann