ISYE 7406

Data Mining & Statistical Learning

Hassan Moughanieh

HW3

1. Introduction

1.1 Problem Description

The Auto MPG dataset is a classic dataset in machine learning and statistics that contains information about various car models, including their attributes and fuel efficiency. The objective of this project is to find the best classification model that can predict if the car has an above or below average milage (mpg) based on 7 car features.

1.2 Dataset

In the original dataset, there are 398 rows (i.e., 398 different kinds of cars), and 9 columns (the car attributes and name). However, some datapoints had missing data, and thus, were removed. The dataset in this project includes 392 datapoints with 8 different features (excluding the car name).

Here's a description of the key features in the Auto MPG dataset:

1. **mpg** (Miles per Gallon): This is the target variable and represents the fuel efficiency of the car in miles per gallon. It measures how many miles a car can travel per gallon of fuel.
2. **cylinders**: The number of cylinders in the engine. It is often used as a measure of engine size and power.
3. **displacement**: The engine displacement, which represents the total volume of all cylinders in cubic inches.
4. **horsepower**: The horsepower rating of the engine, indicating the power output.
5. **weight**: The weight of the car in pounds. Heavier cars tend to have lower fuel efficiency.
6. **acceleration**: The acceleration performance of the car, measured in seconds from 0 to 60 miles per hour.
7. **model year**: The year in which the car model was manufactured.
8. **origin**: The origin of the car, typically represented as a categorical variable (e.g., 1 for American, 2 for European, 3 for Japanese).

2. Exploratory Data Analysis

Initially our target variable, mpg, was a continuous variable representing the fuel efficiency of the car in miles per gallon as the name suggests. However, in this project, our aim is to classify cars as either below or above average. For that, we calculated the median of mpg and labeled all cars above that as 1 and below it as 0. We then looked at boxplots of all the features by class. Some parameters showed significant variation by class like displacement, horsepower and weight, while acceleration and year showed smaller impact on mpg (fig. 2.1 & 2.2).

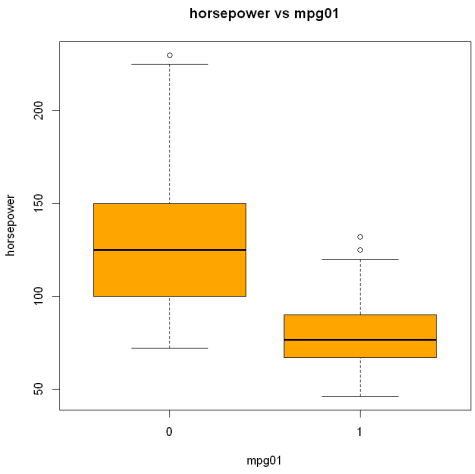

Fig 2.1

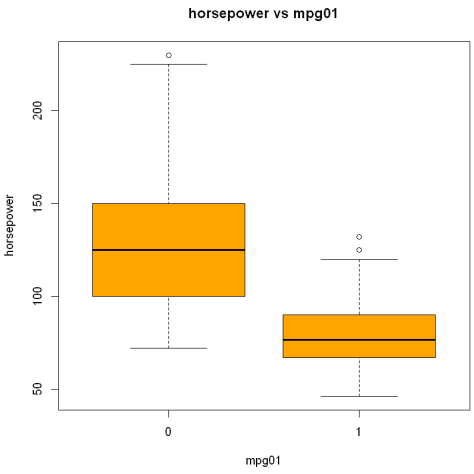

Fig 2.1

The correlation matrix illustrated a similar pattern with acceleration, year, and origin having the lowest correlations with the predictor, mpg01 (fig 2.3).

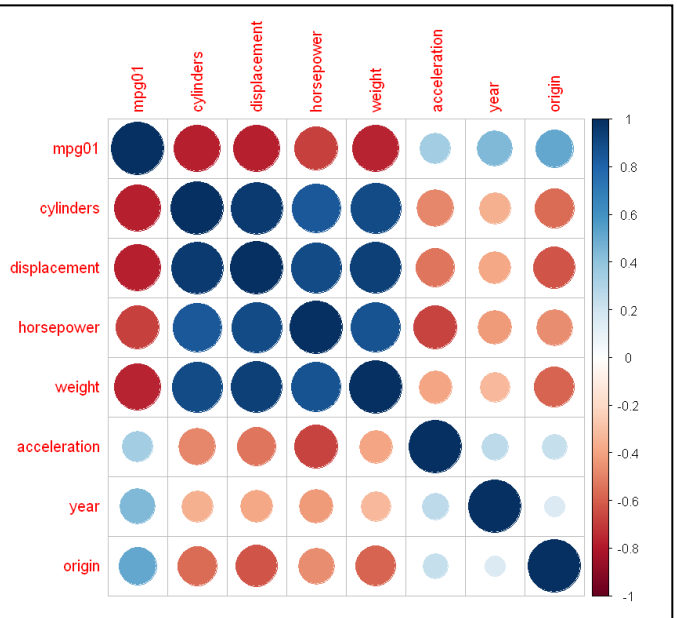

Fig 2.3

We then proceeded to split the data into 80/20 training and testing, respectively. We did that by randomly sampling 80% of the dataset for training and leaving the rest for testing. The resultant datasets were, trainAuto with 313 datapoints, and testAuto with 79 datapoints.

By looking at the distribution of values, we can see that more than 150 cars in the dataset had 4 cylinders with a few having 3 or 5. 6 & 8 cylinders were present in similar amounts with around 60 to 70 cars. Displacement, horsepower, and weight all had heavy right tails, while acceleration had an almost perfect normal distribution. We can also note that a significantly larger number of cars in the study were made in 70, 71 and almost one third of the cars came from origin 1 (American).

In this report, we are only interested in parameters that are most associated with the predictor, mpg01. Thus, we removed acceleration, year, and origin as they all have correlations around 0.5 and below.

3. Methods

The following 6 models were tested to find the best performing model with the lowest classification error:

1. Linear Discriminant Analysis (LDA).
2. Quadrant Discriminant Analysis (QDA).
3. Naïve Bayes.
4. Logistic Regression.
5. K-Nearest Neighbors.
6. Support Vector Machine.

Each one of these algorithms is based on different assumptions and expected to behave differently. Our objective is to find the best performing model producing the lowest training & testing errors. Due to the small size of the dataset (392 datapoints) we also performed Mont Carlo cross validation with 100 iterations to ensure the robustness of the model.

Our first model (mod1) is Linear Discriminant Analysis (LDA). LDA performs classification and dimensionality reduction by projecting the data onto a lower-dimensional space that maximizes the separation between the classes. It does this by finding a set of linear discriminants that maximize the ratio of between-class variance to within-class variance. It assumed that the data is linearly separable, which may not be the case.

The second model is Quadrant Discriminant Analysis (QDA). Unlike LDA, QDA does not typically perform dimensionality reduction as a part of its process. It focuses on modeling the data distribution more flexibly allowing for quadratic decision boundaries.

Next, we used Gaussian Naive Bayes classifier. This algorithm assumes no pair of features are dependent which is not generally correct. Especially in our case, here we can see significant correlations between the different parameters. Despite that, it often works well in practice.

Our fourth model is Logistic regression. “cylinders” & “displacement” had p-values of 0.94836 & 0.08973, respectively, making them statistically insignificant at 95% confidence interval. We even performed Wald test and they both had zeros in their range implying the same conclusion. We calculated the overall significance of the model using the Chi-squared test and it had a p-value of 0 indicating it is significant and can make good predictions. We then tested the goodness of fit using

Deviance & Pearson residual tests. They returned contradictory conclusion indicating that the model might be a good fit. By inspecting the QQplot and histogram of residuals, we can see that the normality assumption does not hold (fig 3.1 & 3.2). We conclude that the model is not a good fit and can't be used for inference, However, it is good for prediction.

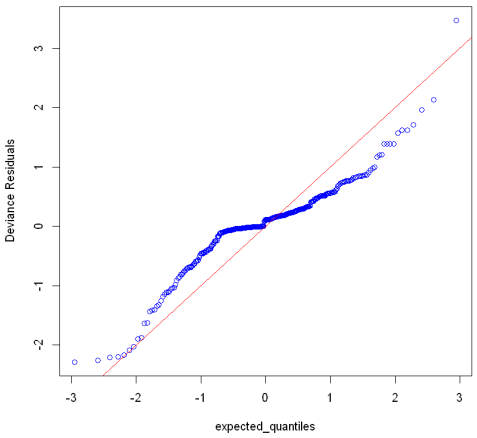

Fig 3.1

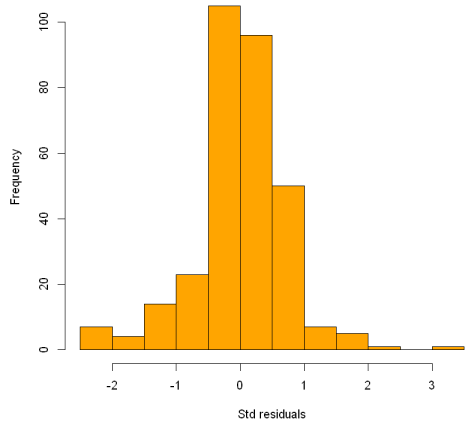

Fig 3.2

Next, we built a KNN model using 3, 5, 7, 9, 11, 13 & 15 as K values. (Fig 3.3) shows KNN5 as the best performing model judging by the training error. However, testing error, which is a better measure for the performance shows a better performance with K = 9 (fig. 3.4).

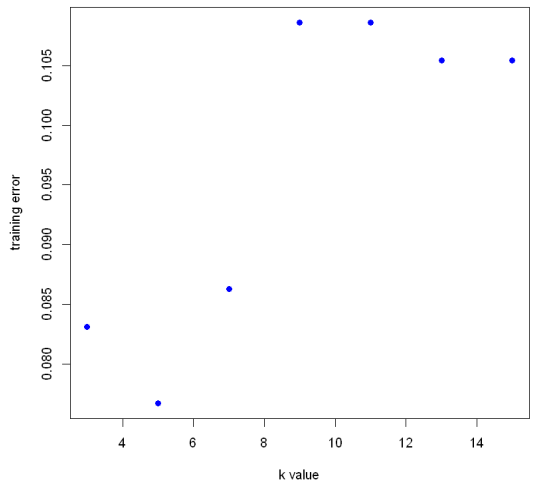

Fig 3.3

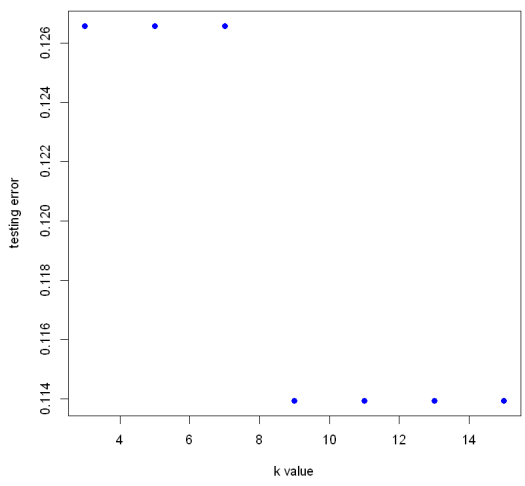

Fig 3.4

Lastly, we tried Support Vector Machine (SVM) with radial kernel. We ran the model c values ranging from 1 to 1000 looking for the optimal value producing the lowest training error. (fig 3.5) shows that the performance of the model improves gradually up until c = 124 and flatters after that. The model is then trained and tested with C=124.

All the models were then cross validated using Monte Carlo method with 100 iterations.

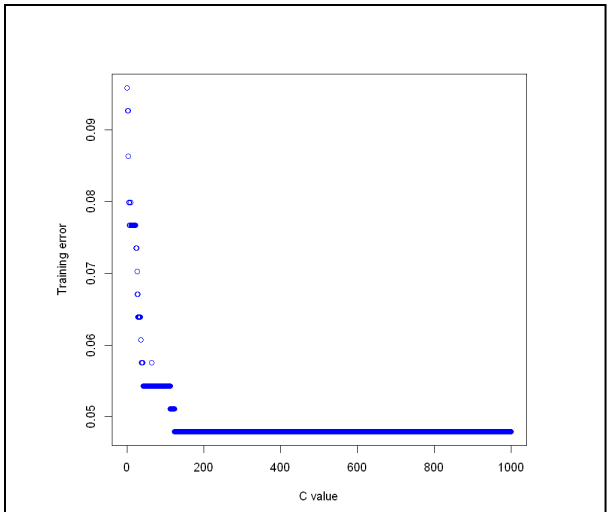

Fig 3.5

4. Results

As previously mentioned, we calculated the training and testing error of all the models, in addition to, the testing errors with CV. The results are as follows:

Model	Training Error	Testing Error
LDA	0.0926518	0.1139241
QDA	0.0926518	0.1012658
Naïve Bayes	0.0990415	0.1139241
Logistic Regression	0.0990415	0.1265823
KNN3	0.0830671	0.1265823
KNN5	0.0766773	0.1265823
KNN7	0.0862620	0.1265823
KNN9	0.1086262	0.1139241
KNN11	0.1086262	0.1139241
KNN13	0.1054313	0.1139241
KNN15	0.1054313	0.1139241
SVM	**0.0479233**	**0.0886076**

Model	CV Mean Testing Error	CV Variance
LDA	0.10227848	0.00108659
QDA	0.10886076	0.00102612
Naïve Bayes	0.10227848	0.00096359
Logistic Regression	0.10886076	0.00104231
KNN3	0.12189873	0.00121275
KNN5	0.11708861	0.00138664
KNN7	0.12367089	0.00141409
KNN9	0.12405063	0.00127537
KNN11	0.12379747	0.00140997
KNN13	0.12632911	0.00142421
KNN15	0.12746835	0.00146071
SVM	**0.08658228**	**0.00086337**

5. Findings

We can observe from the tables above that training errors were slightly smaller than testing errors indicating no signs of overfitting. The results after cross validation are also consistent with this finding where almost all the testing error values remained unchanged. Most models performed comparably similar with cv testing error ranging between 10 & 13%. SVM with radial was the best performing model with an error of 8.6% and a variance of 0.00086337. The best performing KNN model without CV had a K = 9, while with CV had a K = 5. As we have noted before, CV ensures robustness of the model and avoids overfitting. Logistic Regression, on the other hand, was one of the worst performing with testing error of 0.1265823 along with KNN3, KNN%, and KNN7. However, with CV, its error dropped down to 0.10886076, outperforming all KNN models.

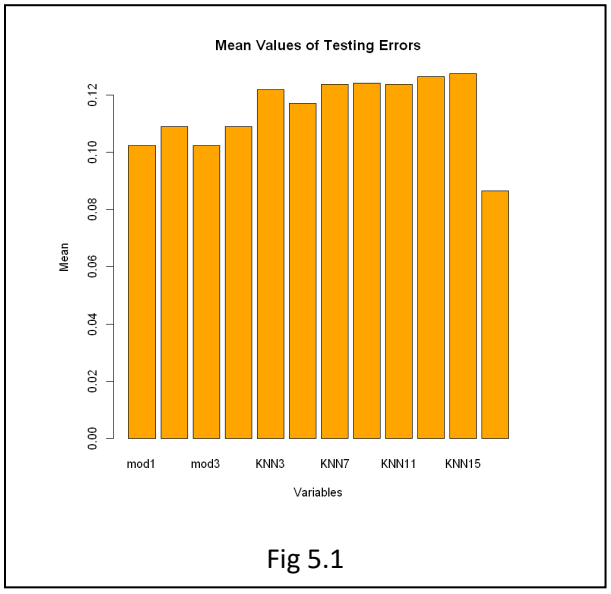

Fig 5.1

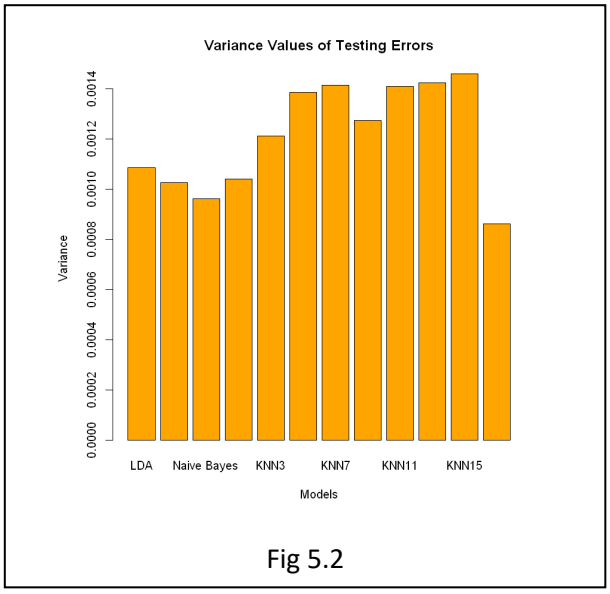

Fig 5.2

6. Appendix

Here you can find the R code associated with this report.