ISYE 7406 Homework 3

August 26, 2021

Introduction

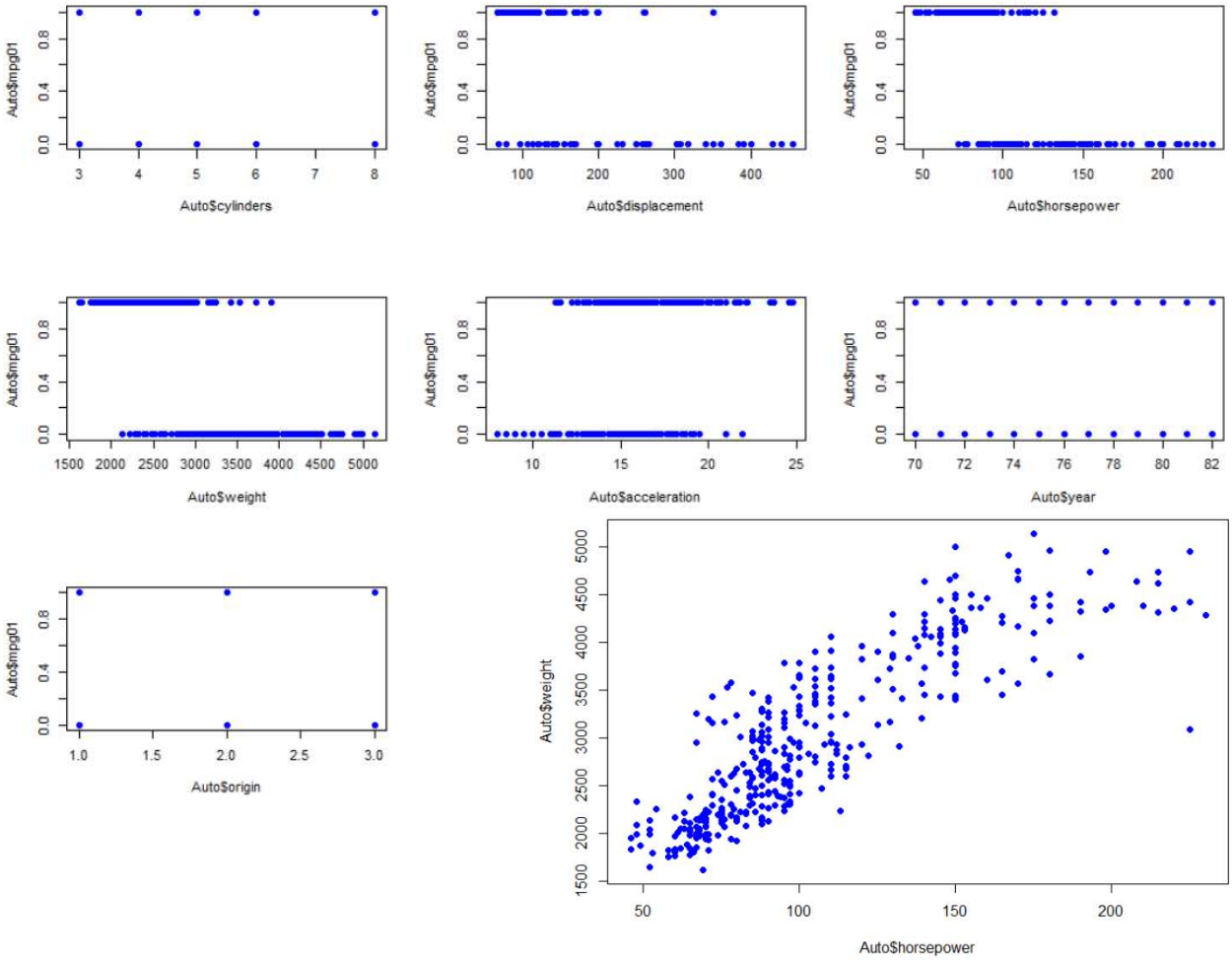
This report will review the Auto+Mpg dataset which includes the mpg, cylinders, displacement, horsepower, weight, acceleration, year and origin of 352 cars. The purpose of analyzing the dataset is to generate models (Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Naïve Bayes, Logistic Regression and KNN with several values of K) to identify the Training & Test errors. Then the KNN models are evaluated by their performance on the test error. The purpose of this report is to also identify other findings and visually explore the dataset for important/unusual patterns.

Exploratory Data Analysis

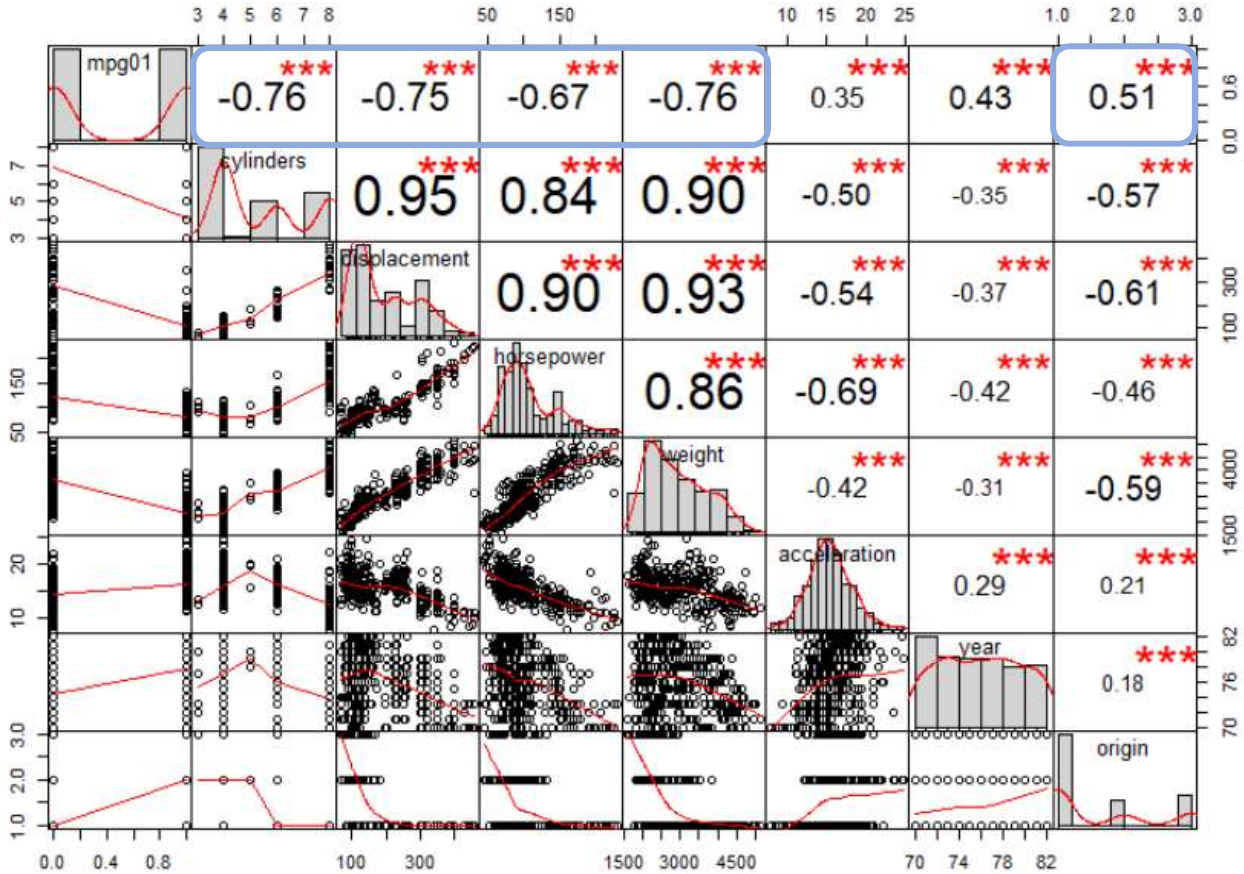
The exploratory data analysis will check for important/unusual patterns in the data through plots and summary statistics. According to the summary statistics shown below, it was interesting to see how cylinders 1st quartile and Median equals 4 suggesting that most of the cars have 4 cylinders & mpg01 (which is also the response binary variable where values greater than or equal to the median are classified as 1, otherwise 0) have an equal proportion of 1's and 0's.

```
> summary(Auto)
   mpg01          cylinders      displacement     horsepower        weight      acceleration
 Mode :logical   Min.   :3.000   Min.   : 68.0   Min.   : 46.0   Min.   :1613   Min.   : 8.00
 FALSE:196       1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0   1st Qu.:2225   1st Qu.:13.78
 TRUE :196       Median :4.000   Median :151.0   Median : 93.5   Median :2804   Median :15.50
                 Mean   :5.472   Mean   :194.4   Mean   :104.5   Mean   :2978   Mean   :15.54
                 3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0   3rd Qu.:3615   3rd Qu.:17.02
                 Max.   :8.000   Max.   :455.0   Max.   :230.0   Max.   :5140   Max.   :24.80
      year           origin
 Min.   :70.00   Min.   :1.000
 1st Qu.:73.00   1st Qu.:1.000
 Median :76.00   Median :1.000
 Mean   :75.98   Mean   :1.577
 3rd Qu.:79.00   3rd Qu.:2.000
 Max.   :82.00   Max.   :3.000
```

Since there are only 7 predictor variables, I decided that I could scatterplot all of them to see which predictors had any relationship with mpg01. Based on the Scatterplot results (shown below), we see that weight has a consistent negative influence on miles per gallon (negative correlation). This makes sense since more gas is needed to run heavier vehicles. Now what's interesting is that not only does horsepower also have a negative correlation with miles per gallon but also it has a positive correlation with weight. Meaning when car shoppers are trying to pick a car that has more horsepower, that will probably end up with a heavier car that has less mpg amongst other cars. It's worth noting that we can see that the displacement increases as mpg decreases and the acceleration decreases when mpg increases.



I decided to run a correlation test as well to display the correlation matrix and provide an easier birds-eye view visual analysis. This was my first time using the "Performance Analytics" package in R to run the correlation and I'm glad I did it because it made looking at the data very easy. You can see how mpg01 has a strong correlation with weight, horsepower, displacement, cylinders, and origin but also a very strong positive correlation between displacement and cylinders which I didn't know until running this plot. Based off research, the correlation values >= +/- 0.50 are considered the significant values associated with mpg01 (blue outlined in the diagram). You can also see the histograms and how the distribution is shaped which I thought was also a great nice-to-have.



Methodology – Modeling for Training and Test errors

I performed the following (Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Naïve Bayes, Logistic Regression and KNN with several values of K) to retrieve training and test errors when predicting mpg01. The testing data is used to evaluate the model's true performance since the training data may have excluded/included certain data that may have a significant positive/negative affect on the training error. For this exercise, I performed a 100-run cross validation for each model to obtain the test errors and observed which one had the best test error. Running a Monte Carlo cross-validation helps reduce the variance compared to just running one test run and depicts which model is truly the most accurate on a large scale.

Results & Findings

Linear Discriminant Analysis (LDA) – When running the (LDA) model, the testing error resulted at 0.1538462 which was larger compared to the training error of 0.09348442. I also created a confusion table to see how the errors occur. The results showed that the Specificity was 77% and Sensitivity was 90%. A high Sensitivity means that there are very few FALSE negative results meaning fewer cases of mpg01 being misclassified as 1. Specificity being lower shows that the model tends to more likely classify mpg01 as 1 even though the true value is 0.

pred1test	FALSE	TRUE
FALSE	14	2
TRUE	4	19

When running the cross-validation, the testing error increased to 0.09974359 which came 2nd in the ranking of best test errors. LDA is preferred when you have more than two classes because Logistic Regression is limited to two-class. LDA also helps decrease the number of features to provide a controllable number for classification.

Quadratic Discriminant Analysis (QDA) – When running the (QDA) model, the testing error resulted at 0.1282051 which was fairly larger compared to the training error of 0.10481586. It's interesting to see that the QDA testing error was smaller than the LDA and may show that the classes may exhibit distinct covariances. The QDA tends to fit data better than LDA and allows for non-linear separation of data but has more parameters to estimate. It's said that QDA works better when response classes are separable and normality assumption holds. Running the cross-validation led to a test error of 0.10230769 which was ranked 3rd of the best test errors between all the other models (tied with Logistic Regression). As you can see below, the confusion table depicts a stronger accuracy compared to LDA's confusion table with False Positives and False Negatives being equal.

pred2test	FALSE	TRUE
FALSE	16	3
TRUE	2	18

Naïve Bayes – When running the Naïve Bayes model, the testing error resulted at 0.1282051 which was larger compared to the training error of 0.09631728. Naïve Bayes is primarily useful to predict the class probability given diverse dimensions. The model generally performs well and even better than other models like logistic regression when the predictors are independent which is not the case for this dataset. However, Naïve Bayes performs poorly when predictors aren't independent which is rare since sets of predictors are usually never fully independent. The cross validation performed with a testing error of 0.09461538 which ranked 1st of the best test errors but was not too far off from the LDA model. The confusion table shows that this model led to a couple of False negatives and a few False positives.

	FALSE	TRUE
FALSE	15	2
TRUE	3	19

Logistic Regression – For the Logistic Regression model, I decided to use the intuitive approach where (cutoff value) c*=0.5 because the proportion of 0's to 1's is pretty even in the training data so it wouldn't make much difference compared to c*= proportion of Y=0. I then ran the Logistic Regression model with all predictors and got a test error of 0.1538462. This was higher than the other models so I decided to also run the model with all the predictors but surprisingly that led to a lower test error of 0.1282051. The cross-validation test error resulted at 0.09051282 for the model with all predictors vs. the significant predictors at 0.10230769. This made me think whether or not I had chosen the correct predictor variables but I decided to leave it as is since I considered the correlation matrix with a value of >= +/- 0.50 a significant factor which is common based on peer reviewed articles. Still, the significant predictor model still could not outperform the Naïve Bayes model.

pred.glm	FALSE	TRUE
0	14	2
1	4	19

KNN model with several K values using only variables that are most associated with mpg01 – For the KNN model, the instructions stated to only include variables that were significantly associated with mpg01 which based on the correlation diagram is cylinders, displacement, horsepower and weight. Therefore, I filtered the training and test data to only include mpg01 (since knn is a supervised model) and these selected columns when performing the k-nearest neighbor model and the choice of k's are an odd range from 1-15. The optimal k value resulted at KNN5 with a test error value of 0.05128205 which was smaller than all the other test errors but the cross-validation results showed that KNN13 had the smallest test error at 0.11538462. KNN5 in the cross-validation actually resulted in a higher test error of .11692308 which is a good example of why cross-validation is needed to choose the optimal model & its parameters.

Single-run Results

k_value	value
1	0.10256410
3	0.07692308
5	0.05128205
7	0.10256410
9	0.12820513
11	0.10256410
13	0.12820513
15	0.12820513

Cross Validation Results

```
> apply(TEALL,2,mean)
        LDA         QDA Naive Bayes    Log Reg.        KNN1        KNN3        KNN5        KNN7        KNN9
 0.09974359  0.10230769  0.09461538  0.10230769  0.13153846  0.11589744  0.11692308  0.11769231  0.11846154
      KNN11       KNN13       KNN15
 0.11666667  0.11538462  0.11897436

> apply(TEALL,2,var)
        LDA         QDA Naive Bayes    Log Reg.        KNN1        KNN3        KNN5        KNN7        KNN9
0.001844812 0.002158269 0.001921848 0.002277808 0.002811215 0.002676070 0.002727073 0.002923316 0.002852722
      KNN11       KNN13       KNN15
0.002502009 0.002304438 0.002384396
```

Conclusion

When analyzing all the cross-validated models, it is clear that the Naïve Bayes model performed the best in terms of average test error performance but LDA performed better on average for the variance. We can see that the KNN model did not perform as well as the other models regardless of using several k values. I also noticed that the results were extremely close amongst most of the models so I decided to run different seeds to get a better understanding of the average performance. However, I doubled checked with 3 different seeds (shown below) and it was interesting to see how Naïve Bayes outperformed all models in each instance and most likely more if additional seeds were run. This leads me to believe that the Naïve Bayes model is the clear winner and would provide consistent accurate results for other datasets similar to this one. I was curious and ended up coding all predictors and saw that the cross-validation test results were better and favored a QDA model. In the future, I plan on re-attempting this type of problem set for my own work and hopefully will be able to better understand each classification model's strengths in a personal work-based application.

Set.seed(666)

```
> apply(TEALL,2,mean)
       LDA         QDA Naive Bayes    Log Reg.        KNN1        KNN3        KNN5        KNN7        KNN9
 0.1094872   0.1110256   0.1053846   0.1102564   0.1317949   0.1202564   0.1230769   0.1261538   0.1266667
     KNN11       KNN13       KNN15
 0.1253846   0.1243590   0.1292308

> apply(TEALL,2,var)
       LDA         QDA Naive Bayes    Log Reg.        KNN1        KNN3        KNN5        KNN7        KNN9
0.002335717 0.002484477 0.002389377 0.002516951 0.003002012 0.002851593 0.002749387 0.002532624 0.002481355
     KNN11       KNN13       KNN15
0.002455787 0.002555137 0.002615504
```

Set.seed(999)

```
> apply(TEALL,2,mean)
       LDA         QDA Naive Bayes    Log Reg.        KNN1        KNN3        KNN5        KNN7        KNN9
 0.1043590   0.1082051   0.1010256   0.1023077   0.1400000   0.1169231   0.1169231   0.1215385   0.1269231
     KNN11       KNN13       KNN15
 0.1256410   0.1258974   0.1292308

> apply(TEALL,2,var)
       LDA         QDA Naive Bayes    Log Reg.        KNN1        KNN3        KNN5        KNN7        KNN9
0.001823030 0.002584424 0.002109457 0.001706679 0.003073735 0.002687227 0.002620817 0.002903725 0.002887189
     KNN11       KNN13       KNN15
0.002570079 0.002817059 0.002735043
```

Set.seed(444)

```
> apply(TEALL,2,mean)
       LDA         QDA Naive Bayes    Log Reg.        KNN1        KNN3        KNN5        KNN7        KNN9
 0.1110256   0.1125641   0.1074359   0.1125641   0.1389744   0.1220513   0.1212821   0.1276923   0.1317949
     KNN11       KNN13       KNN15
 0.1297436   0.1292308   0.1310256

> apply(TEALL,2,var)
       LDA         QDA Naive Bayes    Log Reg.        KNN1        KNN3        KNN5        KNN7        KNN9
0.002298528 0.002216710 0.002214054 0.002216710 0.002486137 0.002551750 0.002256025 0.002244403 0.002537140
     KNN11       KNN13       KNN15
0.002587612 0.002655350 0.002641736
```