Stats 503 Homework 3

Sam Edds

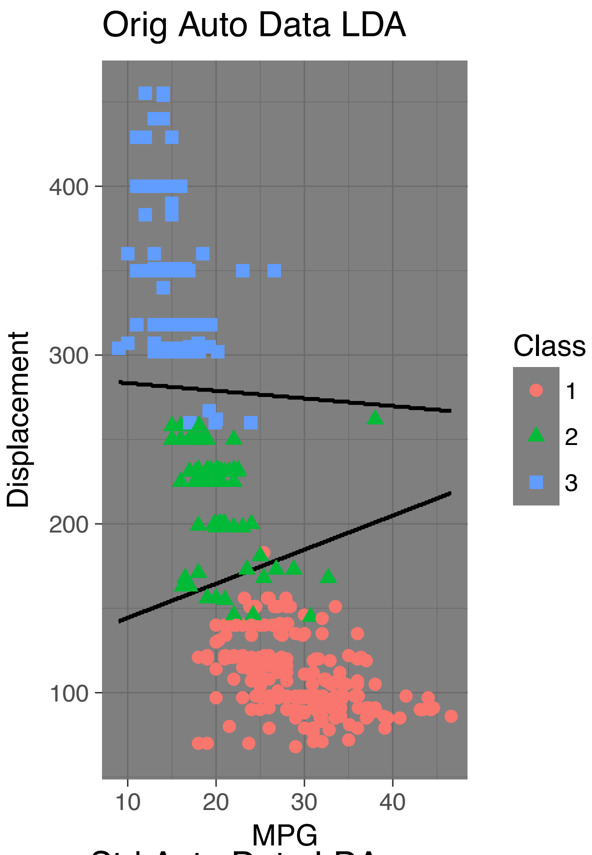
2/17/2018

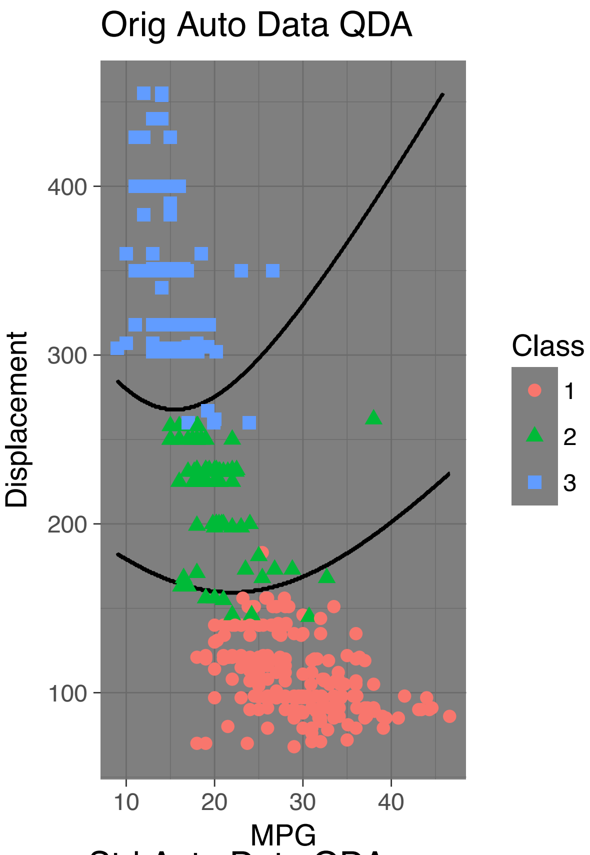
For this analysis we examine different classification methods for our automotive data, breaking it into classes by cylinders in the engine (Y=1 if 5 cylinders or less, Y=2 if 6 cylinders, and Y=3 otherwise). We created three datasets, an original one, as is, a standardized dataset (centered and scaled), and a PCA projected dataset on the first two components. We break our data into train and test sets (75% / 25%) and do all of our modeling on the training dataset before passing the functional form to the train. We graph our train and test samples to check they are visually compatible, which they are.

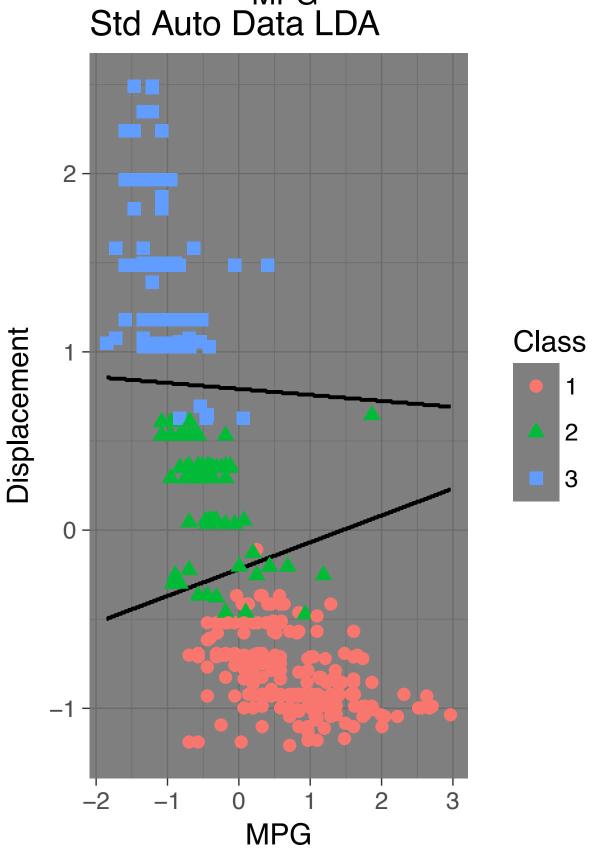
Our misclassification rates show that our original and standardized data have the lowest out-of-sample error rate for both LDA and QDA. Overall QDA, which does not assume equal variance, shows lower rates of misclassification than LDA, which assumes equal variance. When we plot our results, we do so using MPG and Displacement for our original and standardized data because they have the most visual separation. Our plots draw two lines through which they classify our data into the three different cylinder categories. These add a visual aspect to the error rates, we can see the boundaries splitting the different classes and see that the original dataset has the least number of observations misclassified, and overall that QDA has less misclassification than LDA. The parabolic curve better hugs and separate the different groups. Our standardized automative data performs the same as original, because we are just centering and scaling our data, not inherently changing it. PCA rotates our data so our lines are now vertical (these are our two principal components that most explain our data and are a mix of all of our continuous predictors). As a result the data are much more closely aligned, all up and down the y-axis, instead of being separate much more along both axes.

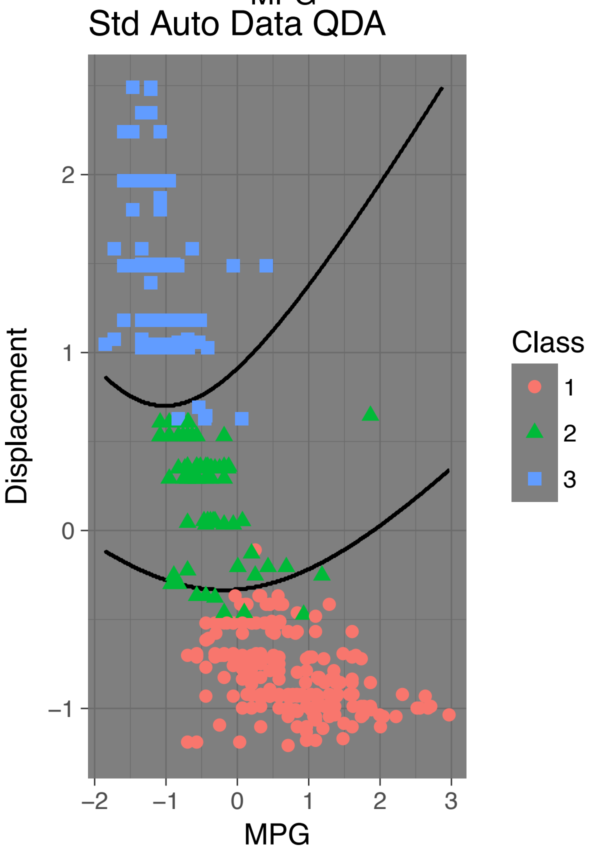
```
##   Error Rate       Type
## 1   Orig LDA 0.04040404
## 2    Std LDA 0.04040404
## 3    PCA LDA 0.06060606
```

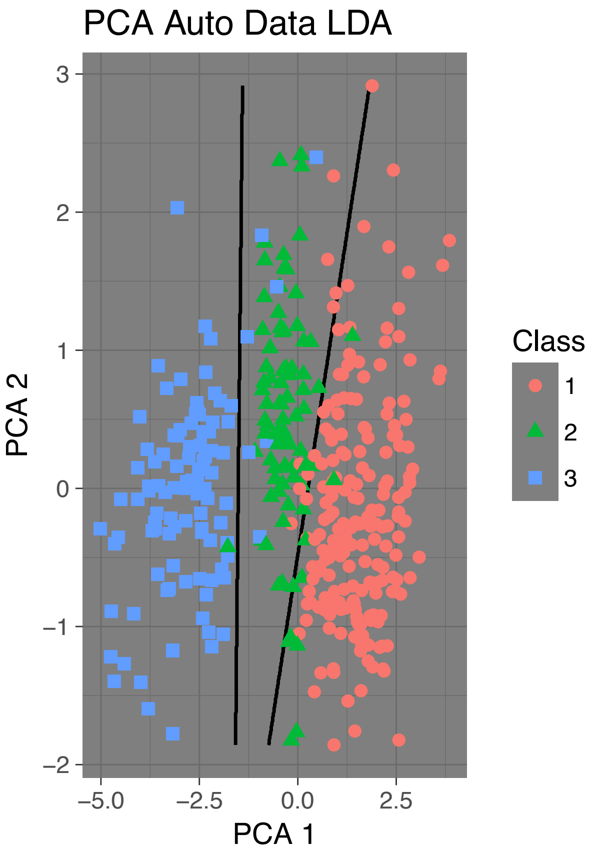
```
##   Error Rate       Type
## 1   Orig qda 0.02020202
## 2    Std qda 0.02020202
## 3    PCA qda 0.05050505
```

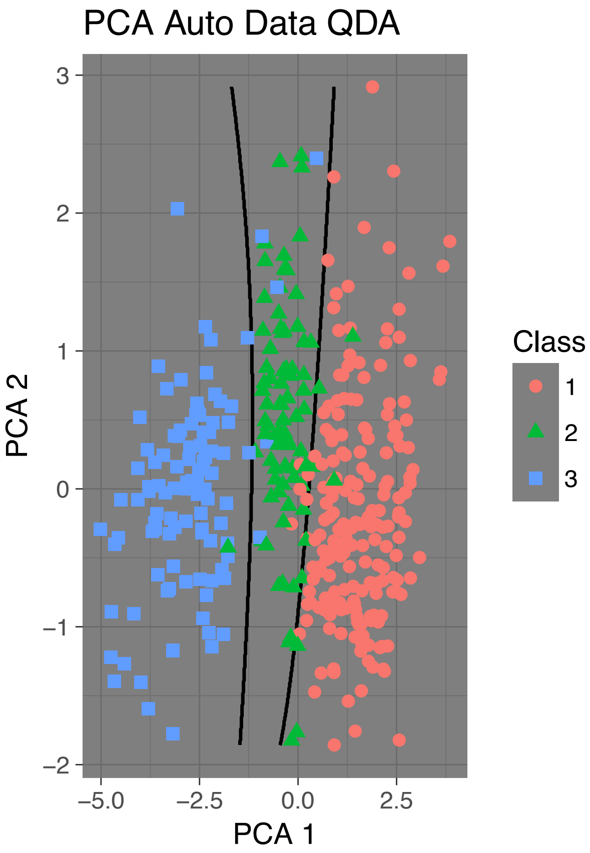

Orig Auto Data LDA


Orig Auto Data QDA


Std Auto Data LDA


Std Auto Data QDA



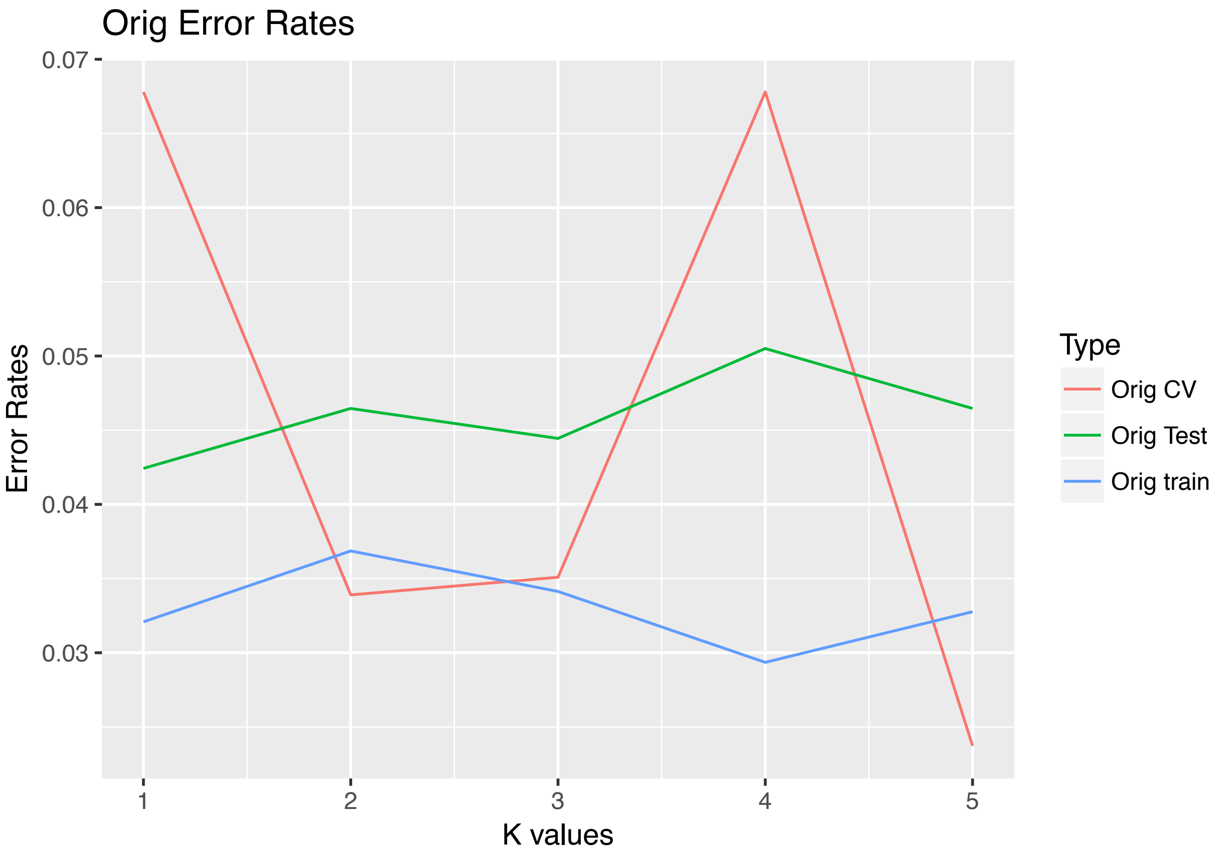


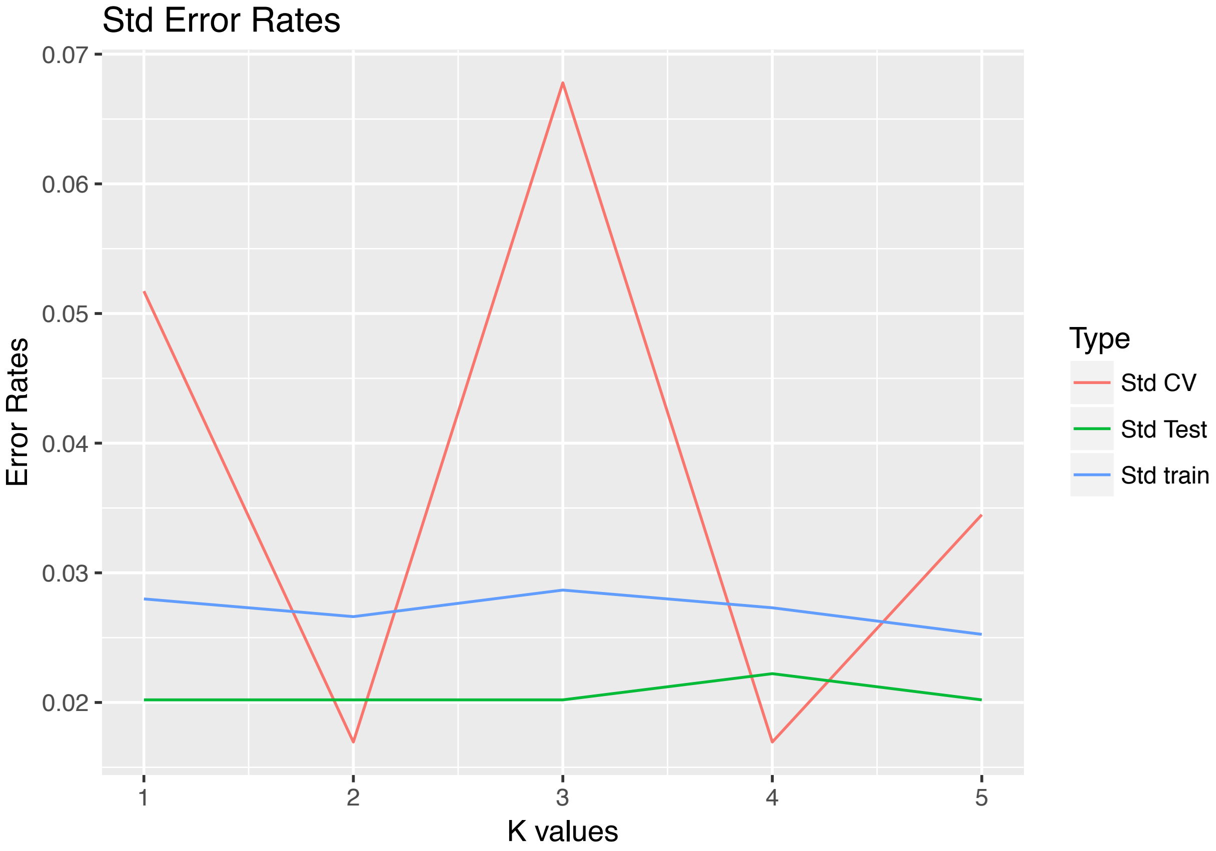
Our logistic regression error rates unsurprisingly are lowest for for our training datasets over our tests because we are fitting on train, so always have a lower error rate than for test, because we are better fitting the idiosyncracies from the train data. We notice that for our test results the lowest mean error rate is lowest for our original dataset, then standardized, and highest for our PCA data.

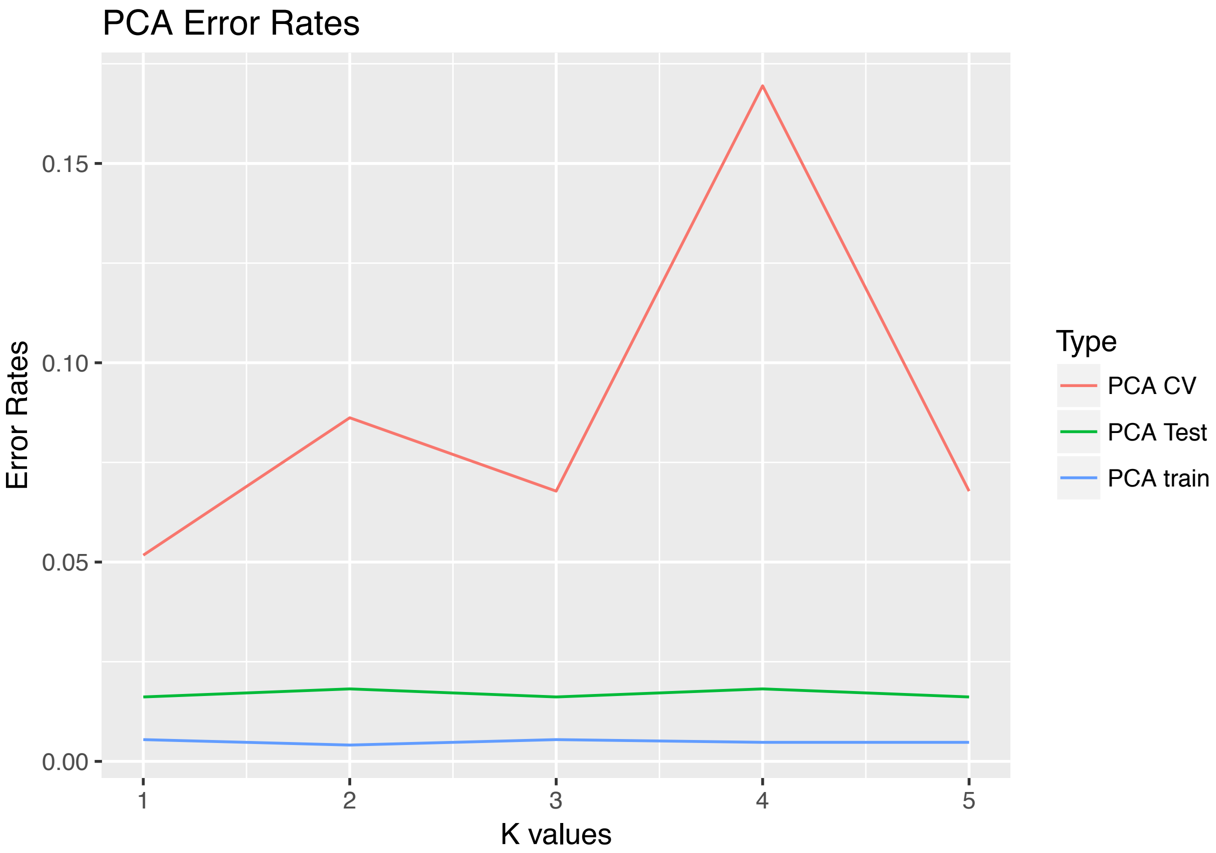
When compared to error rates from LDA and QDA we see logistic regression performs a bit better for all datasets compared to LDA. Perhaps because they are not the same model constraints imposed as with LDA. For standardized and PCA the models perform about the same between QDA and logistic, and logistic performs better than even QDA for the original dataset.

```
##    Error Rate                Type
## 1 0.000000000 Orig Train Logistic
## 2 0.010101010  Orig Test Logistic
## 3 0.003412969  Std Train Logistic
## 4 0.020202020   Std Test Logistic
## 5 0.044368601  PCA Train Logistic
## 6 0.050505051   PCA Test Logistic
```

We finally use KNN on our train, test, and cross-validated models. Our distance metric is chosen as Euclidean distance (as chosen by the R package, which seems reasonable). We use k-fold cross validation (choosing k = 5) to find the optimal number of nearest neighbors for each of our 3 datasets through mean error rate. We find for our original that we should choose our 5 nearest neighbors, for standardized we should choose 4, and for PCA we should choose only our nearest neighbor. We show our train error rates to note again that we fit much better on train, and that our CV results do not necessarily line up with our lowest mean errors for each k of train or test. Again this makes sense because we are training on multiple iterations and not trying to fit the idiosyncracies in our train data (and we do not see the test data beforehand). It is interesting to note our CV error rates vary strongly while neither our train, nor our test do. This would warrant looking into our splits and require more testing to figure out why this might be. Finally we will comment on different datasets.


Orig Error Rates
0.07
0.06
0.05
0.04
0.03
Error Rates
1
2
3
4
5
K values
Type
Orig CV
Orig Test
Orig train


Std Error Rates
0.07
0.06
0.05
0.04
0.03
0.02
Error Rates
1
2
3
4
5
K values
Type
Std CV
Std Test
Std train



Our PCA results have extremely low test errors (remember the optimal KNN is 1), across the board, and are the lowest of the datasets. This is true for train and test. For our standardized results they have higher error rates than PCA, but somehow the test have lower error rates than train which doesn't seem plausible. Finally, our original error rates have the highest test error rates. These results are fairly different from our logistic and LDA/QDA results in which the original data have the lowest error rates, and PCA the highest. Accuracy can be impacted by the different methods, for example, KNN is very localized, while LDA/QDA assume a Gaussian model, and logistic regression is conditional. This leads to different error rates. Similarly, our choice of dataset, if we center and scale our results, or if we use PCA and maximize variance, all give us different optimization answers. Finally, we have a small dataset so because we have low, close error rates, this could be the difference between classifying one more point as an error. Broadly speaking it is difficult to have too many takeaways because we have a small dataset and with different seeds our results shift.