Homework #4

Justin Robinette

February 5, 2019

No collaborators for any problem

Question 4.7.3, pg 168: This problem relates to the QDA model, in which the observations within each class are drawn from a normal distribution with a class-specific mean vector and a class-specific covariance matrix. We conside the simple case where p = 1; i.e., there is only one feature.

Suppose that we have K classes, and that if an observation belongs to the kth class then X comes from a one-dimensional normal distribution. X ~ N(μ_k, $\sigma_k{}^2$).

Recall that the density function for one-dimensional normal distribution is given in (4.11). Prove that in this case, the Bayes' classifier is not linear. Argue that it is in fact quadratic.

*Hint: For this problem, you should follow the arguments laid out in Section 4.4.2, but without making the assumption that $\sigma_1^2 = \ldots = \sigma_k{}^2$.

Results:

Step 1 - starting from equation 4.12 on page 139 of the *ISLR:

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma_k} \exp(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2)}{\sum \pi_l \frac{1}{\sqrt{2\pi}\sigma_k} \exp(-\frac{1}{2\sigma_k^2}(x-\mu_l)^2)}$$

The costant term that isn't variable by k:

$$Constant(c) = \frac{\frac{1}{\sqrt{2\pi}}}{\sum \pi_l \frac{1}{\sqrt{2\pi}\sigma_k} \exp(-\frac{1}{2\sigma_k^2}(x-\mu_l)^2)}$$

Then the equation from 4.12 becomes:

$$p_k(x) = c\frac{\pi_k}{\sigma_k} \exp(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2)$$

Log of both sides:

$$log(p_k(x)) = log(c) + log(\pi_k) - log(\sigma_k) + (-\frac{1}{2\sigma_k^2}(x-\mu_k)^2)$$

Simplify:

$$log(p_k(x)) = (-\frac{1}{2\sigma_k^2}(x^2 + \mu_k^2 - 2x\mu_k)) + log(\pi_k) - log(\sigma_k) + log(C')$$

The x^2 shows the quadratic

Question 4.7.5, pg 169: We now examine the differences between LDA and QDA.

Part A: If the Bayes decision boundary is linear, do we expect LDA or QDA to perform better on the training set? On the test set?

Results: If the Bayes decision boundary is linear, then the **LDA** would be expected to perform better on the test set and the **QDA** could overfit the training set. Because of this overfitting, the **QDA** would be expected to perform better on the training set.

Part B: If the Bayes decision boundary is non-linear, do we expect LDA or QDA to perform better on the training set? On the test set?

Results: With the Bayes decision boundary being non-linear, I would expect the **QDA** to perform better on both the training and test data sets.

Part C: In general, as the sample size (n) increases, do we expect the test prediction accuracy of QDA relative to LDA to improve, decline, or be unchanged? Why?

Results: This is discussed on page 150 of the *ISLR*, where it states that "**LDA** tends to be a better than **QDA** if there are relatively few training observations…" This is because you want to reduce the variance more if there are relatively few training observations. With a larger sample size, reducing the variance of the classifier is not as big of a concern. Because of this, I would expect the **QDA** to increase in prediction accuracy, relative to the **LDA**, as the sample size is increased.

Part D: True or False: Even if the Bayes decision boundary for a given problem is linear, we will probably achieve a superior test error rate using QDA rather than LDA because QDA is flexible enough to model a linear decision boundary. Justify your answer.

Results: FALSE If the Bayes decision boundary is linear, than the **QDA** decision boundary is inferior, because there is a higher variance without corresponding decrease in bias. Although the QDA may perform better on the training set, it is likely to overfit and perform worse on the test data set. This is because the **QDA** is a more flexible method.

Question 4.7.10, pg 171: This question should be answered using the *Weekly* data set, which is part of the *ISLR* package. This data set is similar in nature to the *Smarket* data from this chapter's lab, except that it contains 1,089 weekly returns for 21 years, from the beginning of 1980 to the end of 2020.

Part E: Repeat (d) using LDA.

Results: First, I reloaded the data from the ISLR package. I subset into training and test just as I did in Homework 3.

Then, I used Linear Discriminant Analysis and only the **Lag2** variable as the predictor of **Direction**, just as we had done for part (d) of Homework 3. I used the training set to build the model. I then used this *LDA* model to predict the direction in the test data set and printed the confusion matrix and overall fraction of correct predictions, just as we had been instructed in part (d). I also included the percentage of accuracy for easier analysis.

The model predicted the test set correct 62.5% of the time. The sensitivity was (56/61 = 91.8%). The specificity was (9/43 = 20.9%).

```
## Call:
## lda(Direction ~ Lag2, data = Weekly.training)
##
## Prior probabilities of groups:
##      Down        Up
## 0.4477157 0.5522843
##
## Group means:
##             Lag2
## Down -0.03568254
## Up    0.26036581
##
## Coefficients of linear discriminants:
##            LD1
## Lag2 0.4414162

##         Predicted
## Observed Down Up
##     Down    9 34
##     Up      5 56

## [1] "The percentage of accurate predictions in LDA test set is: 62.5"

## [1] "The overall fraction of correct predictions in the LDA test set is: 65 / 104"
```

Part F: Repeat (d) using QDA

Results: Here, I used Quadratic Discriminant Analysis using only the **Lag2** variable as the predictor of **Direction**, just as we had done for part (d) of Homework 3. I used the training set to build the model. I then used this *QDA* model to predict the direction in the test data set and printed the confusion matrix and overall fraction of correct predictions, just as we had been instructed in part (d). I also included the percentage of accuracy for easier analysis.

The model predicted the test set correct 62.5% of the time. The sensitivity was (61/61 = 100%). The specificity was (0/43 = 0%).

```
## Call:
## qda(Direction ~ Lag2, data = Weekly.training)
##
## Prior probabilities of groups:
##      Down        Up
## 0.4477157 0.5522843
##
## Group means:
##             Lag2
## Down -0.03568254
## Up    0.26036581

##         Predicted
## Observed Down Up
##     Down    0 43
##     Up      0 61

## [1] "The percentage of accurate predictions in QDA test set is: 58.654 (rounded to 3 de
cimals"

## [1] "The overall fraction of correct predictions in the QDA test set is: 61 / 104"
```

Part G: Repeat (d) using KNN with K = 1

Results: I used the K-Nearest Neighbor Classification method here. I set a seed to create reproducibility. To fit the *knn()* function, I created matrices from training and test **Lag2** variables. I also created a 'direction' variable from the training Direction variable. Per homework instructions, I printed the confusion matrix, the fraction of correct predictions and I also included the percentage correct for easier readability. This model only had a prediction accuracy of 50%. The sensitivity was (31/61 = 50.8%). The specificity was (21/43 = 48.8%).

```
##          Predicted
## Observed Down Up
##     Down   21 22
##     Up     30 31
```

```
## [1] "The percentage of accurate predictions in KNN test set is: 50"

## [1] "The overall fraction of correct predictions in the KNN test set is: 52 / 104"
```

Part H: Which of these methods appears to provide the best results on this data?

Results: Using the training and test sets from the previous exercises and the Lag2 variable as the predictor (as instructed on the homework), we see that the GLM and LDA models perform the best at 62.5% accuracy on the test set. The QDA model is slightly lower at 58.65% and the KNN model with K=1 was only accurate 50% of the time.

```
## Call:
## glm(formula = Direction ~ Lag2, family = binomial, data = Weekly.training)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.536  -1.264   1.021   1.091   1.368
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.20326    0.06428   3.162  0.00157 **
## Lag2         0.05810    0.02870   2.024  0.04298 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1354.7  on 984  degrees of freedom
## Residual deviance: 1350.5  on 983  degrees of freedom
## AIC: 1354.5
##
## Number of Fisher Scoring iterations: 4
```

```
##          Predicted
## Observed Down Up
##     Down    9 34
##     Up      5 56
```

Accuracy on Test Data Set by Method

GLM Accuracy (%)	LDA Accuracy (%)	QDA Accuracy (%)	KNN Accuracy (%)
62.5	62.5	58.65385	50

Part I: Experiment with different combinations of predictors including possible transformation and interactions, for each of the methods. Report the variables, method, and associated confusion matrix that appears to provide the best results on the held out data. Note that you should also experiment with values for K in the KNN classifier.

Results: Here I wanted to test a few things with the various models. First, I wanted to compare the performance of the original models from Part H with a model that uses a 2nd order polynomial (of Lag2) as the predictor. I compare each of the 3 models (GLM, LDA, and QDA) using this strategy. Only the QDA model was improved by introducing a 2nd order Polynomial of Lag2 - which is to be expected I supposed.

I also wanted to compare the effectiveness of the KNN method on the data using various K values in my models. I tested KNNs equal to 1, 5, 10, 20, and 100. As we see below, the KNN of 20 performed the best, and all increases in K improved the model accuracy versus the K=1.

Lastly, I printed a table showing the accuracy, by model, in descending order. Both GLM models, the LDA model with the simple **Lag2** predictor, and the QDA model with the 2nd order polynomial were the best performers. The KNN models performed the worst, with the K=1 KNN model being the worst predictor of Direction.

Accuracy (%) for GLM by Predictor

Lag2	Lag2 2nd Order Polynomial
62.5	62.5

Accuracy (%) for LDA by Predictor

Lag2	Lag2 2nd Order Polynomial
62.5	61.53846

Accuracy (%) for QDA by Predictor

Lag2	Lag2 2nd Order Polynomial
58.65385	62.5

Accuracy (%) of KNN Models with Different K Values

K=1	K=5	K=10	K=20	K=100
50	52.88462	56.73077	58.65385	55.76923

```
##                                               Model AccuracyPercentage
## 1                         GLM w/ Lag2 as Predictor           62.50000
## 2  GLM w/ 2nd Order Polynomial (Lag2) as Predictor           62.50000
## 3                         LDA w/ Lag2 as Predictor           62.50000
## 4  QDA w/ 2nd Order Polynomial (Lag2) as Predictor           62.50000
## 5  LDA w/ 2nd Order Polynomial (Lag2) as Predictor           61.53846
## 6                         QDA w/ Lag2 as Predictor           58.65385
## 7                  KNN w/ Lag2 as Predictor (k=20)           58.65385
## 8                  KNN w/ Lag2 as Predictor (k=10)           56.73077
## 9                 KNN w/ Lag2 as Predictor (k=100)           55.76923
## 10                  KNN w/ Lag2 as Predictor (k=5)           52.88462
## 11                  KNN w/ Lag2 as Predictor (k=1)           50.00000
```

Question 4.7.11, pg 172: In this problem, you will develop a model to predict whether a given car gets high or low gas mileage based on the *Auto* data set.

Part D: Perform LDA on the training data in order to predict **mpg01** using the variables that seemed most associated with **mpg01** in (b). What is the test error of the model obtained?

Results: Based on last week's assignment, and the performance of my 2 GLMs, I included the following predictors: cylinders, weight, displacement, horsepower, and year.

Here I performed LDA with the training set and predicted the test set **mpg01** values. I printed a confusion matrix showing that the model is better at predicting when the mpg is greater than the median (47/48) than it is at predicting that an mpg is less than the median (42/50). In my opinion, both are very good rates of success though. The overall accuracy is nearly 91% and the fraction of accuracy is 89/98. The test error rate is 9.1837%.

```
##   mpg01 mpg cylinders displacement horsepower weight acceleration year
## 1     0  18         8          307        130   3504         12.0   70
## 2     0  15         8          350        165   3693         11.5   70
## 3     0  18         8          318        150   3436         11.0   70
##   origin                      name
## 1      1 chevrolet chevelle malibu
## 2      1         buick skylark 320
## 3      1        plymouth satellite
```

```
##         Predicted
## Observed  0  1
##        0 42  8
##        1  1 47

## [1] "The percentage of accurate predictions in LDA test set is: 90.8163"

## [1] "The overall fraction of correct predictions in the LDA test set is: 89 / 98"

## [1] "The test error in the LDA test set is: 9.1837%"
```

Part E: Perform QDA on the training data in order to predict **mpg01** using the variables that seemed most associated with **mpg01** in (b). What is the test error of the model obtained?

Results: Based on last week's assignment, and the performance of my 2 GLMs, I included the following predictors: cylinders, weight, displacement, horsepower, and year.

Here I performed QDA with the training set and predicted the test set **mpg01** values. I printed a confusion matrix showing that the model is better at predicting when the mpg is greater than the median (46/48) than it is at predicting that an mpg is less than the median (44/50). In my opinion, both are very good rates of success though. The overall accuracy is over 91% and the fraction of accuracy is 90/98 (one better than with LDA). The test error rate is 8.1633%.

```
##         Predicted
## Observed  0  1
##        0 44  6
##        1  2 46

## [1] "The percentage of accurate predictions in QDA test set is: 91.8367 (rounded to 4 d
ecimals"

## [1] "The overall fraction of correct predictions in the QDA test set is: 90 / 98"

## [1] "The test error in the QDA test set is: 8.1633%"
```

Part G: Perform KNN on the training data, with several values of K, in order to predict **mpg01**. Use only the variables that seemed most associated with **mpg01** in (b). What test errors do you obtain? Which value of K seems to perform the best on this data set?

Results: Based on last week's assignment, and the performance of my 2 GLMs, I included the following predictors: cylinders, weight, displacement, horsepower, and year.

Here I performed KNN with the training set and predicted the test set **mpg01** values. To do so, I created matrices of our predictor variables from the training and test sets. I then created a factor variable for mpg01 from training.

From here, I used KNN to predict **mpg01** - whether or not the miles per gallon of the automobile was greater than or less than the median.

Per the homework instructions, I utilized multiple k values (1, 5, 10, 20, 40, 60) to try and determine the best value for prediction rate.

As we see in the table below of error rates on the test data set, the best predicting models were the ones that utilized K values of 5, 40 and 60. Here, or test error rate is 10.2%. The model accuracy improves going from k=1 to k=5 and then performs worse going from k=5 to k=10. Then we receive our worst error rate going to k=20 (12.24%). The error rate from the k values tested then reverts back to our superior error rate also seen in k=5.

Test Error (%) of KNN Models with Different K Values

K=1	K=5	K=10	K=20	K=40	K=60
11.22449	10.20408	11.22449	12.2449	10.20408	10.20408

Question 5: Read the paper "Statistical Classification Methods in Consumer Credit Scoring: A Review" posted on D2L. Write a one page (no more, no less) summary.

Results: Credit, as defined by this paper, refers to "an amount of money that is loaned to a consumer by a financial institution which must be repaid." The goal of this paper is to look at credit scoring methods, limitations that exist among the methods as well as in the process as a whole, and discuss some areas that cause additional difficulty in the scoring process.

The paper mentions the voluminous data that exists in credit scoring databases. Well over 100,000 applicants with more than 100 predictor variables is quite common. Furthermore, when a database is used for behavioral scoring, which focuses on past repayment behavior, the paper mentions that these databases can be much larger. Some of the more typical predictor variables include: time at current residence, home status (owner, rent, etc), postcode, telephone, annual income, age, court judgments against the applicant, marital status, and others.

When working with so many independent variables, and such large amounts of data, missing values are common. This makes the variable selection process even more challenging. Theoretically, three approaches are discussed for variable selection: the use of expert knowledge and experience, stepwise approaches, and information value of the variable. In practice, however, all three approaches are typically used, as the paper explains.

Some statistically scoring methods used in credit scoring mentioned are: discriminant analysis, linear regression, logistic regression, mathematical programming methods, recursive partitioning, expert systems, neural networks, smoothing nonparametric methods, and time varying models. Despite the many tools available for statistically scoring credit applicants, there is no overall "best" method, according to the paper.

The data structure, predictors used, and our ability to separate applicants into classes using those predictors helps to determine the best method. Additionally, figuring out what classification rate is to be used to determine best model can change our determination of the best model. Overall misclassification rate might be good, but if your model is very good at predicting if someone with not default, but less successful at figuring out those who will default, this could be very costly to the organization.

Another consideration is the overall speed of classification for purposes of issuing a decision. Being able to offer an instant decision is much more appealing to a borrow - especially, I would suspect, those borrowers that can get quick decisions elsewhere. Neural networks are good for credit situations where there is not as much understanding of the data structure. Classification methods are easier to understand and much more appealing to the organization and to the borrowers. They make it easier for the organization to explain their decision to the borrower.

There are also other issues that must be considered in the process. One is the legality that plays a role in credit decisions. Legislation prevents the use of certain characteristics playing a role in the process - whether or not those characteristics would make the decisions more accurate. Additionally, how credit limits play a role in the modeling of statistical methods can complicate the process. Figuring out what interest rate to charge borrowers in order for the company lending the credit to be profitable is another factor. When do we act on delinquent loans? Is it even worthwhile to do so? These are other factors.

The paper asks some very good questions and presents some very useful explanations of how statistics plays a role in the credit decision making process. The methods described are important, but as the paper mentions, improvements in this process will likely be a product of gaining a better understanding of the other issues discussed in the previous paragraph.

Question 6: Explore this website (https://archive.ics.uci.edu/ml/datasets.html) that contains open data sets that are used in machine learning. Find one data set with a classification problem and write a description of the dataset and problem. I don't expect you to do the analysis for this homework, but feel free to if you want!

Results: I chose the credit-screening data set from the website above. (https://archive.ics.uci.edu/ml/datasets/Credit+Approval). This data set concerns credit card applications, as the website says. Looking at the summary below, we can see that there are some missing values that may require imputation. These are denoted by **?** in the dat aset. A1, for example, has 12 missing, or **?** values.

The dependent variable is the factor **A16**. A '+' indicates approval where as a '-' indicates a decline on the credit application.

What I found most intriguing about this data set was that the variable names have been removed to protect the confidentiality of the applicants. I had not though about the reality that this is probably done frequently in some industries - healthcare and finance being two that I am very interested in. I think this, on the surface, makes the data seem more "daunting". In reality though, I think it could be useful in removing biases that we have. One binary factor independent variable, for example, is **A9**. It lists a 't' and an 'f'. Maybe that variable actually represents "College Student", "Previous Bankruptcy", "Income > 150k", etc. This would inherently bias my initial exploration.

My first step in working with this data set would be to attempt to impute values based on the other values in the variable. In doing so, I would change all '?' values to 'NA'. Next I would begin the imputation process and determine if any observations should be removed (too many NAs, for example). Then I could begin the process of looking for correlations and building models.

```
##   A1    A2   A3 A4 A5 A6 A7   A8 A9 A10 A11 A12 A13   A14 A15 A16
## 1  b 30.83 0.00  u  g  w  v 1.25  t   t   1   f   g 00202   0   +
## 2  a 58.67 4.46  u  g  q  h 3.04  t   t   6   f   g 00043 560   +
## 3  a 24.50 0.50  u  g  q  h 1.50  t   f   0   f   g 00280 824   +

##  A1            A2            A3         A4       A5            A6
##  ?: 12   ?      : 12   Min.   : 0.000   ?:  6   ? :  6   c      :137
##  a:210   22.67  :  9   1st Qu.: 1.000   l:  2   g :519   q      : 78
##  b:468   20.42  :  7   Median : 2.750   u:519   gg:  2   w      : 64
##          18.83  :  6   Mean   : 4.759   y:163   p :163   i      : 59
##          19.17  :  6   3rd Qu.: 7.207                    aa     : 54
##          20.67  :  6   Max.   :28.000                    ff     : 53
##          (Other):644                                     (Other):245
##        A7            A8         A9      A10         A11       A12
##  v      :399   Min.   : 0.000   f:329   f:395   Min.   : 0.0   f:374
##  h      :138   1st Qu.: 0.165   t:361   t:295   1st Qu.: 0.0   t:316
##  bb     : 59   Median : 1.000                   Median : 0.0
##  ff     : 57   Mean   : 2.223                   Mean   : 2.4
##  ?      :  9   3rd Qu.: 2.625                   3rd Qu.: 3.0
##  j      :  8   Max.   :28.500                   Max.   :67.0
##  (Other): 20
##  A13          A14           A15           A16
##  g:625   00000  :132   Min.   :     0.0   -:383
##  p:  8   00120  : 35   1st Qu.:     0.0   +:307
##  s: 57   00200  : 35   Median :     5.0
##          00160  : 34   Mean   :  1017.4
##          00080  : 30   3rd Qu.:   395.5
##          00100  : 30   Max.   :100000.0
##          (Other):394
```