STATISTICS 452/652: Statistical Learning and Prediction

November 2, 2020

Lecture 18: Linear Methods for Classification

(Reading: Section 4.2, 4.3, 4.4)

1 Goals of lecture

2 Linear Methods for Classification

- As we have seen, the simplest form of decision boundary between classes is a linear boundary
- There are numerous ways to make linear decision boundaries between classes
  - When true structure of class probabilities forms boundaries between true classes are (approximately) linear, expect these methods to do well
    - Low bias
    - (potentially) Low variance
  - When boundaries are highly nonlinear, then these methods will have high bias and potentially poor misclassification rates.

2.1 Linear regression (not used!)

A bad way to create linear boundaries is to fit a linear regression to indicator functions for each response

1. Convert classification variable into numerical by creating indicators for group membership
   (a) $Y_1 = I(Y = 1), \ldots, Y_K = I(Y = K)$

Figure 1: Comparison of linear and logistic regression classifiers. From ISLR.

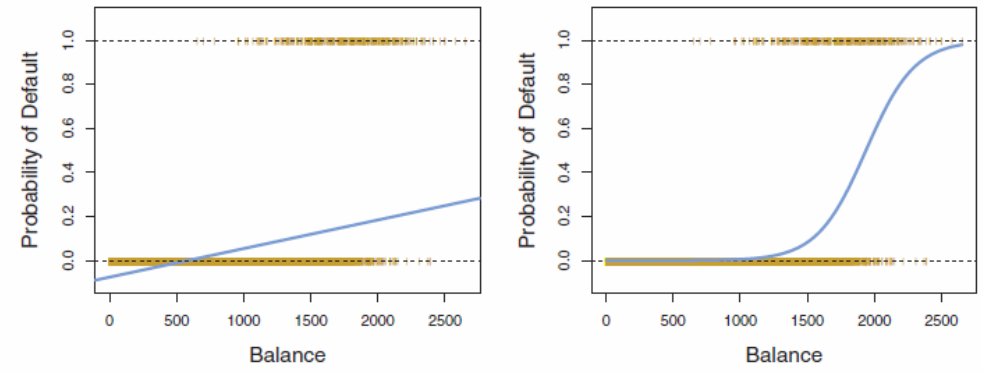


FIGURE 4.2. *Classification using the* `Default` *data.* Left: *Estimated probability of* `default` *using linear regression. Some estimated probabilities are negative! The orange ticks indicate the 0/1 values coded for* `default` *(*`No` *or* `Yes`*).* Right: *Predicted probabilities of* `default` *using logistic regression. All probabilities lie between* 0 *and* 1.

(b) These add to 1 for every observation

Y	Y_1	Y_2	Y_3	Y_4
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

2. It turns out mathematically, that the mean value of an indicator function equals the probability of a 1.

   (a) Sample proportion is average of sample of 1's and 0's

3. Could try using a linear model separately fitting each Y_k vs X and estimate parameters by least squares

4. Then $\hat{f}(X) = \hat{p}_k(X) = \widehat{\Pr}(Y = k|X)$

5. Decision boundaries where pairs of $\hat{p}_k(X)$'s are equal creates linear boundaries

   (a) At $X = x_0$ classify according to which k has the largest $\hat{p}_k(x_0)$

Problems

1. Linear regression does not constrain $\hat{p}_k(\mathbf{x})$ to lie between 0 and 1!

   (a) Simple example with $p = 1$ and $K = 2$ demonstrates this. See Figure 1.

   (b) Modeling $E(Y)$, not thinking of it as a probability

   (c) Not too much of a problem if goal is merely to create linear boundaries and estimate $G(\mathbf{x})$.

2. Linear regression has even more problems with $K > 2$
   (a) Can miss classes completely depending on how they are arranged in X

3 Logistic Regression

Logistic regression is form of regression developed specifically for binary responses ($K = 2$):

- With $K = 2$, note that $p_2(x) = 1 - p_1(x)$
  - So only need to model one of these
- Rewrite Y as a single indicator function, Y =1 ("Success") or 0 ("Failure")
- Probability of success = $p(X)$
- Assume that Y has a Bernoulli/Binomial distribution
- Assume a linear model for "logit" (log-odds of success) instead of mean

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$$

- Estimate parameters $(\beta_0, \beta_1, \ldots, \beta_p)$ using "maximum likelihood" (ML)
  - $\hat{p}(\mathbf{x}) = \frac{\exp(\hat{\beta}_0+\hat{\beta}_1 X_1+\ldots+\hat{\beta}_p X_p)}{1+\exp(\hat{\beta}_0+\hat{\beta}_1 X_1+\ldots+\hat{\beta}_p X_p)} = \frac{1}{1+\exp[-(\hat{\beta}_0+\hat{\beta}_1 X_1+\ldots+\hat{\beta}_p X_p)]}$
- When $p = 1$, probability curve looks like right side of Figure 1.
  - Can make decision boundary at all x where $\hat{p}(X) = 0.5$
  - Choose class 1 if $\hat{p}(X) > 0.5$
- When $p > 1$, creates linear (or hyperplane) boundaries between classes

3.1 Extension to $K > 2$ classes

Not quite as well known but still pretty classic stuff

1. Use multinomial distribution, where class probabilities depend on X, $p_k(X)$, $k = 1, \ldots, K$
2. Use "multi-response logit" link, comparing each class to some baseline category (doesn't matter which one; predictions are the same)
   (a) For example, if the last class is chosen as the baseline, then make $K - 1$ logit ratios

$$\log\left(\frac{p_k(X)}{p_K(X)}\right) = \beta_{k0} + \beta_{k1} X_1 + \ldots + \beta_{kp} X_p,\ k = 1, \ldots, K-1$$

(b) Results in probabilities

$$\Pr(G = k|X) = \frac{\exp[\beta_{k0} + \beta_{k1}X_1 + \ldots + \beta_{kp}X_p]}{1 + \sum_{l=1}^{K-1} \exp[\beta_{l0} + \beta_{l1}X_1 + \ldots + \beta_{lp}X_p]}, \; k = 1, \ldots, K-1$$

and

$$\Pr(G = K|X) = \frac{1}{1 + \sum_{l=1}^{K-1} \exp[\beta_{l0} + \beta_{l1}X_1 + \ldots + \beta_{lp}X_p]}$$

i. Parameters again estimated by ML

3. Also produces linear boundaries (can be proved mathematically)
4. Then the classifier $\hat{f}(x)$ returns class k if $\hat{p}_k(x)$ is the largest class probability.
5. Results in $(p+1)(K-1)$ parameters.

(a) In vehicle image data, this is $19 \times 3 = 57$!

3.2 Notes

1. One advantage of logistic regression is that wealth of inference procedures that follow it, thanks to ML estimation (assuming no model selection)

(a) Confidence intervals for parameters and for probability estimates

(b) Tests for individual parameters or groups of them in ANOVA-style tests

(c) Variable-selection methods

i. All subsets, stepwise based on information criteria, which were developed explicitly for ML estimates

ii. LASSO has a version that works with ML to shrink estimates and select variables

EXAMPLE: Logistic Regression on Wheat Data (L18 Logistic Regression Wheat.R)

The package `nnet` that does neural nets has a function that fits multinomial models, `multinom()`. Evidently, it just skips the hidden layer and combines the inputs directly into a linear combination, transformed by a sigmoidal output function. It can be slow and fail to converge with too few iterations. Also, it requires that explanatory variables be scaled to lie approximately between 0 and 1, or else it may give very strange results.

```
> summary(mod.fit)
Call:
multinom(formula = type ~ ., data = set1.rescale, trace = TRUE)

Coefficients:
       (Intercept)   density  hardness      size      weight
```

```
Scab     16.096363 -18.81505 -4.915380 6.736446 -18.2989406
Sprout    8.761246 -12.50927 -4.052665 1.803399  -0.9999576
         moisture    classnum
Scab    0.1109796 -0.77462360
Sprout -1.6636584 -0.04881767

Std. Errors:
       (Intercept)  density hardness     size   weight
Scab      2.518195 3.176054 2.086737 3.423757 3.861902
Sprout    1.981611 2.593197 1.574296 2.264907 1.797390
       moisture  classnum
Scab   1.599686 0.8898911
Sprout 1.076341 0.6012836

Residual Deviance: 250.127
AIC: 278.127
>
> # Adding a function to perform LR Tests.
> #  Legitimate here since I am fitting one model
> library(car)
> Anova(mod.fit)
Analysis of Deviance Table (Type II tests)

Response: type
          LR Chisq Df Pr(>Chisq)
density     68.440  2  1.375e-15 ***
hardness     8.402  2    0.01498 *
size         3.874  2    0.14413
weight      38.271  2  4.893e-09 ***
moisture     3.195  2    0.20244
classnum     0.873  2    0.64620
```

The first part of the output gives the logic regression coefficients for the two logit models. At the end is a test for the significance of each variable, testing the null hypotheses that a given variable could be dropped completely (from both logits) avainst the alternative that it cannot. We see that `density` and `weight` seem to be dominant variables here, while `hardness` may also be important. Other variables seem less important, given the rest of the model.

Below I show a sample of the estimated class probabilities from the test set, along with the test set confusion matrix. While the confusion matrix is better looking than the ones from KNN, there are still relatively large numbers of misclassifications across the entire table.

```
> pred.probs.2 <- predict(mod.fit, newdata=set2.rescale,
+                         type="probs")
```

```
> round(head(pred.probs.2),3)
   Healthy  Scab Sprout
3    0.623 0.022  0.355
11   0.396 0.403  0.201
16   0.686 0.017  0.297
17   0.147 0.759  0.094
18   0.760 0.021  0.218
19   0.207 0.650  0.143
>
> # Test set confusion matrix
> table(set2$type, pred.class.2, dnn=c("Obs","Pred"))
         Pred
Obs       Healthy Scab Sprout
  Healthy      16    4      5
  Scab          4   14      4
  Sprout        7    6     15
```

The `glmnet` package that fits the LASSO fits a different version of the problem. It does a logistic regression on each binary indicator $Y_1, Y_2, \ldots Y_k$. So it provides a set of coefficients for all K classes instead of for $K-1$ comparisons with the baseline class. This is not the same analysis. It might be fine in its own right, but it is not the usual multinomial regression.

Finally, I show the list of all test error rates achieved so far:

Method	Training	Test	Test "SE"
KNN-1	0	0.47	0.06
KNN-Opt(12)	0.28	0.49	
Logistic	0.28	0.40	
Logist-glmnet	0.28	0.40	
Logist-LASSO-min	0.28	0.43	

Using the binomial distribution to approximate the number of misclassified observations in the test set, we can calculate that the standard error for an estimated error rate between 0.4 and 0.5 is about 0.06. So we see that the error rate from Logistic more than a standard error lower than from KNN.

4 Linear Discriminant Analysis

- Logistic regression is useful, but it has one really painful quirk: **Parameter estimates and boundaries become unstable when classes have little overlap.**
- When classes in sample are completely separated, parameter estimation in logistic regression actually fails
  - Slope wants to be infinite
  - Infinitely many intercepts provide identical perfect fit

- This is frustrating, because (near-) separable classes is the case where classification methods ought to be easy to fit!
- Fortunately, there are other, very different ways to make linear boundaries

4.1 LDA Concepts

- Linear discriminant analysis (LDA) is a very popular tool for classification
- LDA operates in the reverse direction from regression-type methods like logistic
  - Regression methods model the distribution of the response, given the value of the explanatory variable
    - Logistic regression models $P(Y|X)$ using bi-/multinomial
  - LDA instead models distribution of the explanatory variables, given the response
    - Consider each class separately—what does distribution of X look like?
    - Then use a conditional probability formula called Bayes' rule to estimate the probability of each class, given X
- For example, suppose we wanted to understand the effects of, say, blood pressure on death due upon contracting Covid-19.
  - We look at distribution of blood pressure for people who die from Covid-19
  - We look at distribution of blood pressure for people who survive Covid-19
  - For any particular BP, x, the ratio of these two distributions at x gives a measure of how likely x is to have come from the death group than vs. the survive group. See Figure 2
  - If death and survival were equally likely overall (*yikes!*) then Bayes' rule would classify according to which group is more likely to have a blood pressure measurement at x
  - More generally, we also factor in the total fractions of the population in each group and formalize all of this into probability calculation
  - End result is exactly $p_k(x) = p(Y = k|X = x)$
- In Discriminant Analysis, we *assume* that X has a multivariate normal distribution (MVN) in each class
  - Each variable X_j has a normal distribution within each class
    - For class k, mean is μ_{jk} and variance is σ^2_{jk}
    - The means and variance may differ across classes in general (more on this in a minute)
  - Each pair of variables $X_j, X_{j'}$ has correlation within each class $\rho_{jj',k}$
  - See Figure

Figure 2: Example of conditional distributions of X for $K = 2$ classes. When each class is equally likely, Bayes rule says to classify according to highest curve. (From ISLR.)

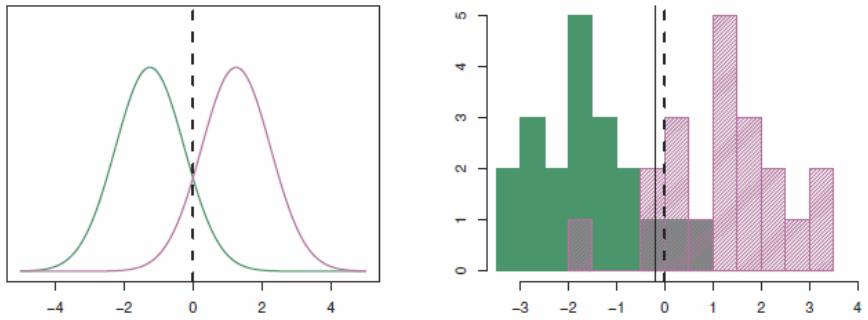


FIGURE 4.4. Left: *Two one-dimensional normal density functions are shown. The dashed vertical line represents the Bayes decision boundary.* Right: *20 observations were drawn from each of the two classes, and are shown as histograms. The Bayes decision boundary is again shown as a dashed vertical line. The solid vertical line represents the LDA decision boundary estimated from the training data.*

Figure 3: 3D pictures of normal distributions for $p = 2$ with different correlations. (From ISLR.)

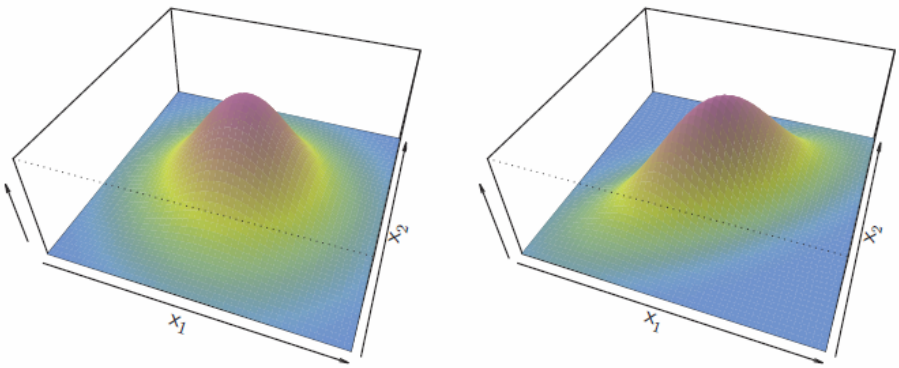


FIGURE 4.5. *Two multivariate Gaussian density functions are shown, with* $p = 2$. Left: *The two predictors are uncorrelated.* Right: *The two variables have a correlation of* 0.7.

Figure 4: 3D pictures of normal distributions for $p = 2$ with different correlations. (From ISLR.)

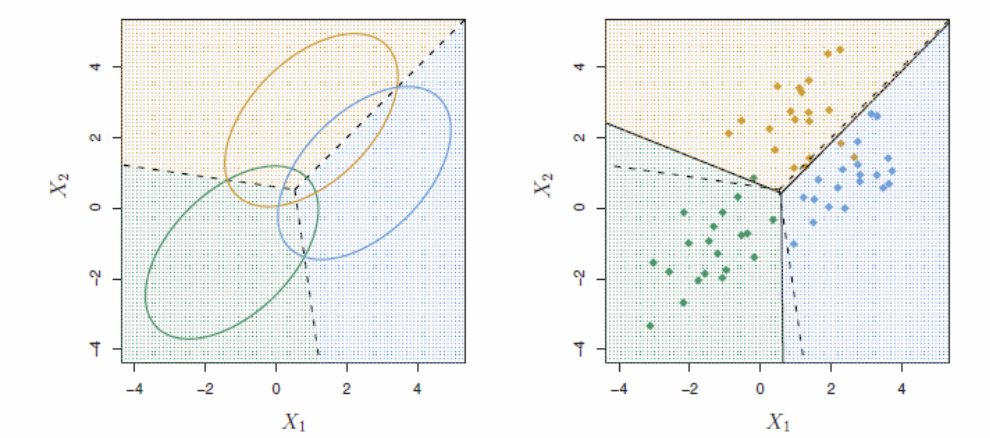


FIGURE 4.6. *An example with three classes. The observations from each class are drawn from a multivariate Gaussian distribution with $p = 2$, with a class-specific mean vector and a common covariance matrix.* Left: *Ellipses that contain 95 % of the probability for each of the three classes are shown. The dashed lines are the Bayes decision boundaries.* Right: 20 *observations were generated from each class, and the corresponding LDA decision boundaries are indicated using solid black lines. The Bayes decision boundaries are once again shown as dashed lines.*

  - FYI: Normal is also called "Gaussian" (ISLR does calls it Gaussian)

- **Linear** Discriminant Analysis further assumes that the variances and correlations are not changing across k, but that the means may.
  - Identical variances and correlations forces the distributions have the same "shape" (See Figure 4)
  - Decision boundaries turn out to be linear
- Practically speaking, LDA produces $K-1$ "linear discriminants", say $Z_1, \ldots, Z_{K-1}$
  - Linear combinations of explanatory variables, like Z in PCA, PPR, PLS
    - First one "discriminates" the K classes (normal distributions) as much as possible
    - Second discriminates what first did not, etc through $K-1$
  - Plotting combinations of them can be interesting.
- If we knew the population means, variances, and correlations, then LDA would return Bayes classifier
  - In practice, have to estimate parameters, so merely approximate Bayes classifier
    - K means

    - p variances
    - $p(p-1)/2$ correlations
  - As n grows relative to $K + p(p+1)/2$, LDA test error rate approaches error rate of Bayes Classifier

4.2 Quadratic Discriminant Analysis

- LDA is great if the the variance and correlation structure for X is the same in all classes
  - If it's not, then LDA is biased
    - True decision boundaries are not linear
    - LDA can't adapt
- Alternative version is *not* to assume that variance/correlation structures are identical across classes
  - Let them be different in each class
  - Results in decision boundaries that are quadratic surfaces
  - Quadratic Discriminant Analysis (QDA)
- Which is better, LDA or QDA?
  - Answer is usual bias/variance tradeoff
  - Quadratic can better match certain surfaces than linear, and can choose to be flat if covariance matrices between two classes are about the same
    - Hence reduced bias
  - But extra parameters to be estimated, not all of which may be necessary
    - Has $(K-1)p(p+1)/2$ more parameters than LDA
    - Hence extra variance
- Both of these tools are useful and are in frequent use in practice.
  - However, both rely on assumption of normality for X in each class
  - Still useful even when normal assumptions are obviously false, as with binary variables
    - But no longer capable of mimicking Bayes classifier
    - Take on bias that more flexible methods might be able to eliminate

EXAMPLE: Discriminant Analysis on Wheat Data (L18 Discrim Wheat.R)

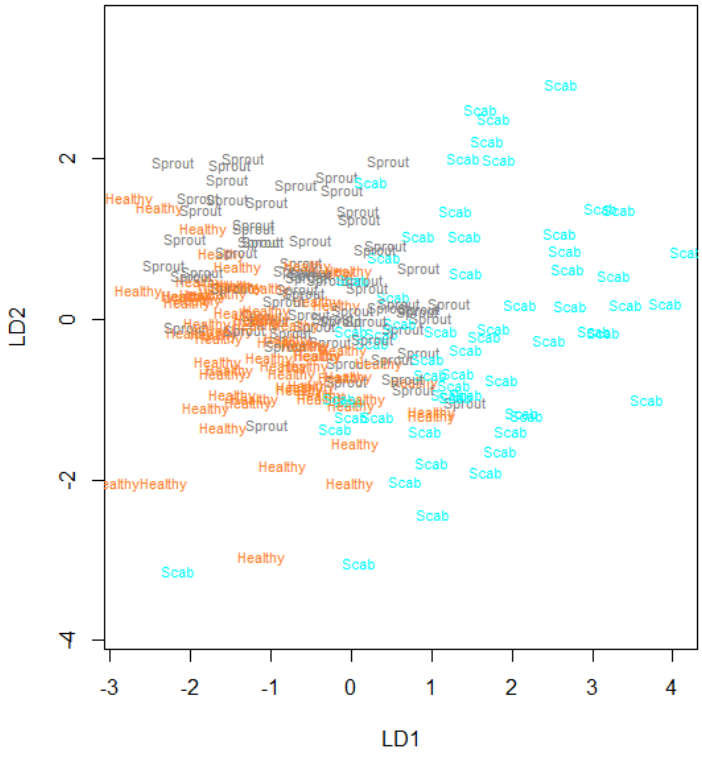
The package `MASS` has functions `lda()` and `qda()` that do linear and quadratic discriminant analysis, respectively. As usual, there are a variety of helper functions for examining the results. If we first standardize the explanatory variables to have mean 0 and variance 1 in the combined data, then when the separate class means are estimated, we can see which ones are most different. See below.

```
> set1s <- apply(set1[,-6], 2, scale)
> set1s <- data.frame(set1s,type=set1$type)
> lda.fit.s <- lda(data=set1s, type~.)
> lda.fit.s
Call:
lda(type ~ ., data = set1s)

Prior probabilities of groups:
Healthy    Scab  Sprout
  0.355   0.305   0.340

Group means:
            density   hardness       size     weight
Healthy  0.70102694  0.1265443  0.2897454  0.3743367
Scab    -0.82241704  0.1705631 -0.7327091 -0.9532671
Sprout   0.00580186 -0.2851323  0.3547549  0.4642851
           moisture      classnum
Healthy  0.15005446  0.0721051156
Scab    -0.07828193 -0.0845800172
Sprout  -0.08645102  0.0005870271

Coefficients of linear discriminants:
                  LD1         LD2
density  -0.84687425 -0.73968176
hardness -0.14168390 -0.44763549
size      0.07351008  0.21444413
weight   -1.01267254  0.53878781
moisture  0.01751022 -0.35676216
classnum -0.02125836  0.07436082

Proportion of trace:
   LD1    LD2
0.8999 0.1001
```

Agreeing with the logistic regression analysis, the class mean for `density` and for `weight` seem most different from one another. We also see the coefficients that create the linear

discriminants, Z_1 and Z_2. Figure 5 plots the three classes against these two linear discriminants. We can see that the first one (`LD1`) seems to mostly separate Scab from the other groups, while the second tries—with limited success—to separate healthy and sprout.

Figure 5: Plot of three classes on the two linear discriminants for the Wheat Data



The test error for LDA is in the table below. I also run QDA in the program, and its test error is also below. LDA is mostly indistinguishable from logistic regression, and QDA does not seem to help.

Method	Training	Test	Test "SE"
KNN-1	0	0.47	0.06
KNN-Opt(12)	0.28	0.49	
Logistic	0.28	0.40	
Logist-glmnet	0.28	0.40	
Logist-LASSO-min	0.28	0.43	
LDA	0.28	0.39	
QDA	0.26	0.43	

5 What to take away from this

1. Linear methods for classification are well-studied and easy to use (sort of).
2. There are many more bells and whistles than what I've shown, so follow-up diagnostics are possible
    (a) In particular, especially when $K > 2$ it can sometimes be instructive to look at the misclassified cases
        i. Do they tend to be all from certain groups or are they spread out more equally?
3. Extensions have been proposed to make them more robust or better classifiers.
    (a) These extensions represent additional tools to consider.

6 Exercises

Application

Return to the Vehicle data used in the previous lecture. Use the same split as before.

```
set.seed(46685326, kind="Mersenne-Twister")
perm <- sample(x=nrow(vehdata))
set1 <- vehdata[which(perm <= 3*nrow(vehdata)/4), ]
set2 <- vehdata[which(perm > 3*nrow(vehdata)/4), ]
```

1. Run logistic regression using `multinom()`.

   (a) Scale the training data to lie between 0 and 1, and use the same min and max values to scale the test data. Run `summary()` in each scaled data set to confirm that you are doing this correctly. **Report the summary of the first 3 variables in each set.**

   (b) Run the logistic regression model using all explanatory variables.

      i. Run `Anova()` on the object. **Report the table of test results and comment on which variables seem to be important or unimportant.**

      ii. Compute and **report training and test error**. **Does test error seem better or worse than optimal KNN?** (Use the standard error computed before to help you make a sensible comment here.)

      iii. **Report the confusion matrix and comment sensibly on what it tells you.**

2. Run a LASSO version of logistic regression, using CV to estimate optimal shrinkage in the logistic regression parameters using minimum CV-error.

   (a) **Report a list of the variables included/excluded in each logit. Does the pattern seem somewhat consistent with the ANOVA results from earlier? Explain in a sentence.**

   (b) Compute and **report training and test error**. **How does test error compare to other methods?**

3. Run LDA on these data.

   (a) Make a colour plot of the classes against pairs of linear discriminants. The `plot()` function will do this for you automatically. Use these colours:
   `class.col <- ifelse(set1$class==1,y=53,n= ifelse(set1$class==2,y=68,n= ifelse(set1$class==3,y=203,n=464)))`
   Present the plot and write a sentence for each linear discriminant, explaining how it seems to separate classes.

(b) **Report training and test error. How does test error compare to other methods?**

4. Run QDA on these data. **Report training and test error. How does test error compare to other methods?**