Classification

Lu Haibo

Mondy, Mar 11, 2019

The linear regression model discussed in the last chapter assumes that the response variable Y is quantitative. But in many situations, the response variable is instead *qualitative* (or *categorical*).

1. A person arrives at the emergency room with a set of symptoms that could possibly be attributed to one of three medical conditions. Which of the three conditions does the individual have?

2. An online banking service must be able to determine whether or not a transaction being perfomed on the site is fraudulent, on the basis of the users's IP address, past transaction history, and so forth.

3. On the basis of DNA sequence data for a number of patients with and without a given disease, a biologist would like to figure out which DNA mutations are delerious and which are not.

Predicting a qualitative response for an observation can be referred to as *classifying* that observation, since it involves assinging the observation to a category, or class.

- *logistic regression*
- *Naive Bayesian classification*
- *linear discriminant analysis*
- *K-nearest neighbors*

Logistic Regression

Consider the `Default` data set, where the response `default` falls into one of two categories, `Yes` or `No`. Rather than modeling this response Y directly, logistic regression models the *probability* that Y belongs to a particular category.

default	student	balance	income
No	No	729.53	44361.63
No	Yes	817.18	12106.13
No	No	1073.55	31767.14
No	No	529.25	35704.49
No	No	785.66	38463.50
No	Yes	919.59	7491.56

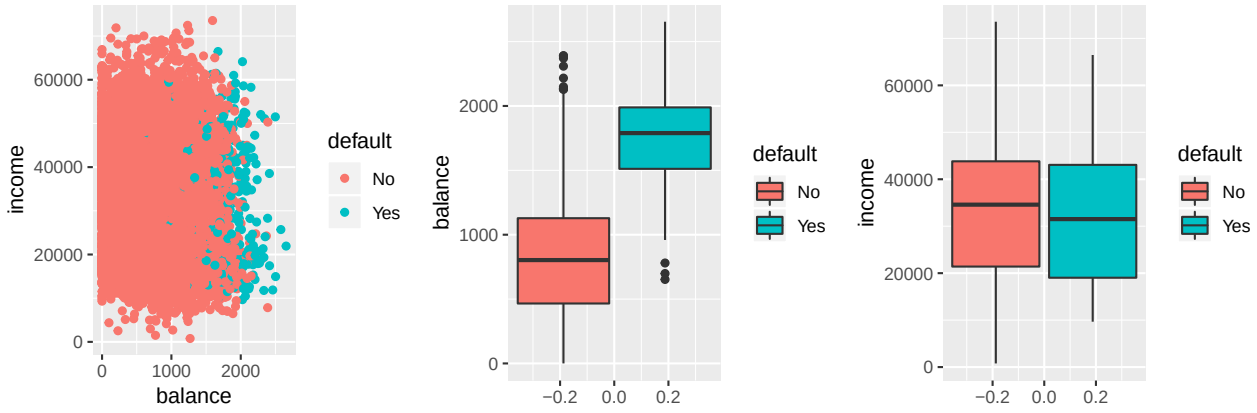

Figure 1: The Default data set. Left: The annual incomes and monthly credit card balances of a number of individuals. The individuals who defaulted on their credit card payments are shown in orange, and those who did not are shown in blue. Center: Boxplots of balance as a function of default status. Right: Boxplots of income as a function of default status

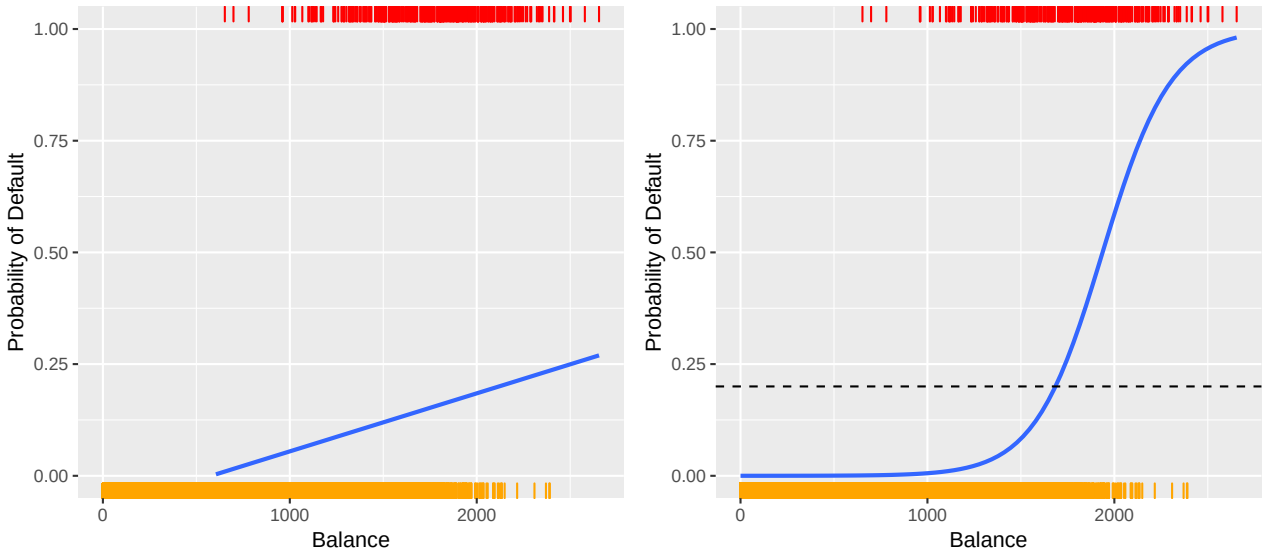

Figure 2: Classification using the Default data. Left: Estimated probability of default using linear regression. Some estimated probabilities are negative! The orange ticks indicate the 0/1 values coded for default(No or Yes). Right: Predicted probabilities of default using logistic regression. All probabilities lie between 0 and 1.

For the `Default` data, logistic regression models the probability of default. For example, the probability of default given `balance` can be written as

$$Pr(default = Yes \mid balance)$$

The value of $Pr(default = Yes \mid balance)$, which we abbreviate $p(balance)$, will range between 0 and 1. Then for any given value of `balance`, a prediction can be made for `default`. For example, one might predict $default = Yes$ for any individual for whom $p(balance) > 0.2$. Alternatively, if a company wishes to be conservative in predicting individuals who are at risk for default, then they may choose to use a lower *threshold*, such as $p(balance) > 0.1$.

The Logistic Model

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X + \epsilon$$

The left-hand side is called the *log-odds* or *logit*.

Generalized linear model

$$E(Y \mid X) = \mu(x) = g^{-1}(X\beta)$$

`https://en.wikipedia.org/wiki/Generalized_linear_model`

Estimating the Regression Coefficients

In general, we use *maximum likelihood* to estimate the unknown linear regression coefficients.

Table 1: For the Default data, estimated coefficients of the logistic regression model that predicts the probability of default.

	Dependent variable:		
	default		
	(1)	(2)	(3)
balance	0.005***		0.006***
	(0.0002)		(0.0002)
income			0.00000
			(0.00001)
studentYes		0.405***	−0.647***
		(0.115)	(0.236)
Constant	−10.651***	−3.504***	−10.869***
	(0.361)	(0.071)	(0.492)
Observations	10,000	10,000	10,000
Log Likelihood	−798.226	−1,454.342	−785.772
Akaike Inf. Crit.	1,600.452	2,912.683	1,579.545
Note:		*p<0.1; **p<0.05; ***p<0.01	

Making Predictions

Using the coefficient estimates given in Table 1, we predict that the defalut probability for an individual with a `balance` of $1,000 is

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}}$$

Multiple Logistic Regression

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

Table 1, model (3) shows the coefficient estimates for a logistic regression model that uses `balance`, `income` (in thousands of dollars), and `student` status to predict probability of `default` . There is a surprising result here. The p-values associated with `balance` and the dummy variable for `student` status are very small, indicating that each of these variables is associated with the probability of `default`. However, the coefficient for the dummy variable is negative, indicating that students are less likely to default than nonstudents. In contrast, the coefficient for the dummy variable is positive in model (2). How is it possible for student status to be associated with an increase in probability of default in model (2) and a decrease in probability of default in model (3)?

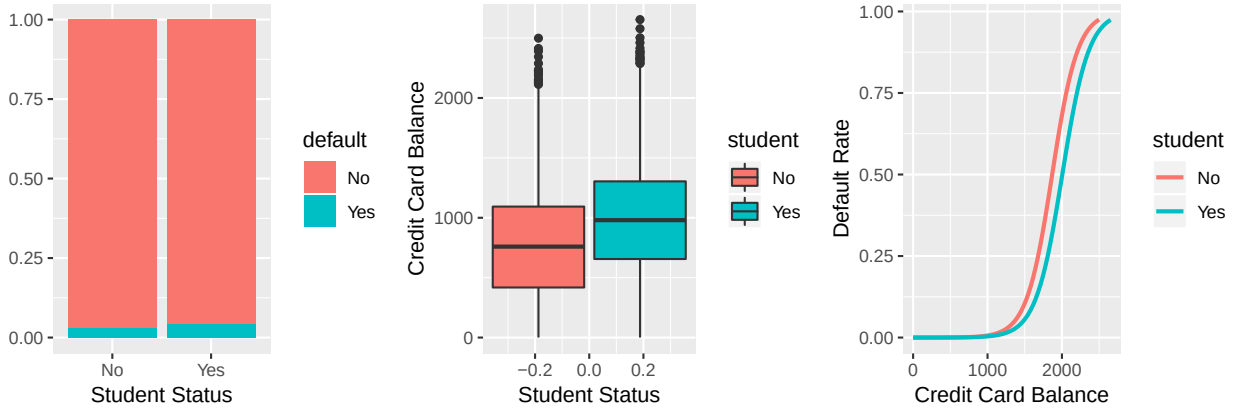

Figure 3: Confounding in the Default data.

Confusion Matrix

A **confusion matrix** is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. The confusion matrix itself is relatively simple to understand, but the related terminology can be confusing.

```
# For the Default data, the predicted probility of the logistic regression
# model using thresholds 0.05
library(rsample)
```

```
data <- initial_split(Default, prop = 0.7)
train <- training(data)
test <- testing(data)

logist_fit <- glm(default ~., data = train, family="binomial")

logist_fit %>%
  augment(newdata = test) %>%
  mutate(
    y_fitted = exp(.fitted)/(1+exp(.fitted)),
    pred = if_else(y_fitted >= 0.05, "Yes", "No")
  ) %>%
  select(default, pred) %>%
  table()
```

```
##         pred
## default   No  Yes
##     No  2604  295
##     Yes   20   81
```

- **true positives** (TP): These are cases in which we predicted yes (they have the disease), and they do have the disease.
- **true negatives** (TN): We predicted no, and they don't have the disease.
- **false positives** (FP): We predicted yes, but they don't actually have the disease. (Also known as a "Type I error.")
- **false negatives** (FN): We predicted no, but they actually do have the disease. (Also known as a "Type II error.")

This is a list of rates that are often computed from a confusion matrix for a binary classifier:

- **Accuracy**: Overall, how often is the classifier correct? $(TP + TN)/total$
- **Misclassification Rate**: Overall, how often is it wrong? $(FP + FN)/total$ equivalent to 1 minus Accuracy, also known as "Error Rate".
- **True Positive Rate**: When it's actually yes, how often does it predict yes? $TP/actualyes$ also known as "Sensitivity" or "Recall"
- **False Positive Rate**: When it's actually no, how often does it predict yes? $FP/actualno$
- **True Negative Rate**: When it's actually no, how often does it predict no? $TN/actualno$ equivalent to 1 minus False Positive Rate, also known as "Specificity".
- **Precision**: When it predicts yes, how often is it correct? $TP/predictedyes$

- **Prevalence**: How often does the yes condition actually occur in our sample? *actualyes/total*

A couple other terms are also worth mentioning:

- **Null Error Rate**: This is how often you would be wrong if you always predicted the majority class. (In our example, the null error rate would be $105/3000 = 0.035$ because if you always predicted "No", you would only be wrong for the 333 "Yes" cases.) This can be a useful baseline metric to compare your classifier against. However, the best classifier for a particular application will sometimes have a higher error rate than the null error rate, as demonstrated by the *Accuracy Paradox*.
- **Cohen's Kappa**: This is essentially a measure of how well the classifier performed as compared to how well it would have performed simply by chance. In other words, a model will have a high Kappa score if there is a big difference between the accuracy and the null error rate. (*More details about Cohen's Kappa*)
- **F Score**: This is a weighted average of the true positive rate (recall) and precision. (*More details about the F Score*)
- **ROC Curve**: This is a commonly used graph that summarizes the performance of a classifier over all possible thresholds. It is generated by plotting the True Positive Rate (y-axis) against the False Positive Rate (x-axis) as you vary the threshold for assigning observations to a given class. (*More details about ROC Curves*)

The ROC curve [1]

[1] `https://www.dataschool.io/roc-curves-and-auc-explained/`

An ROC curve is the most commonly used way to visualize the performance of a binary classifier, and AUC is (arguably) the best way to summarize its performance in a single number. [2]

[2] Understanding ROC curves, `http://www.navan.name/roc/`

- a popular graphic for simultaneously displaying the two types of errors for all possible threshold
- The overall performance of a classifier, summarized over all possible thresholds, is given by the *area under the (ROC) curve* (AUC). An ideal ROC curve will hug the top left corner, so the larger area under the (ROC) curve the AUC the better the classifier
- a classifier that performs no better than chance to have an AUC of 0.5

Naive Bayesian classification [3]

[3] Naive Bayes Classifier: theory and R example

Logistic regression involves directly modeling $p(Y = k \mid X = x)$ using the logistic function for the case of two response classes. We now

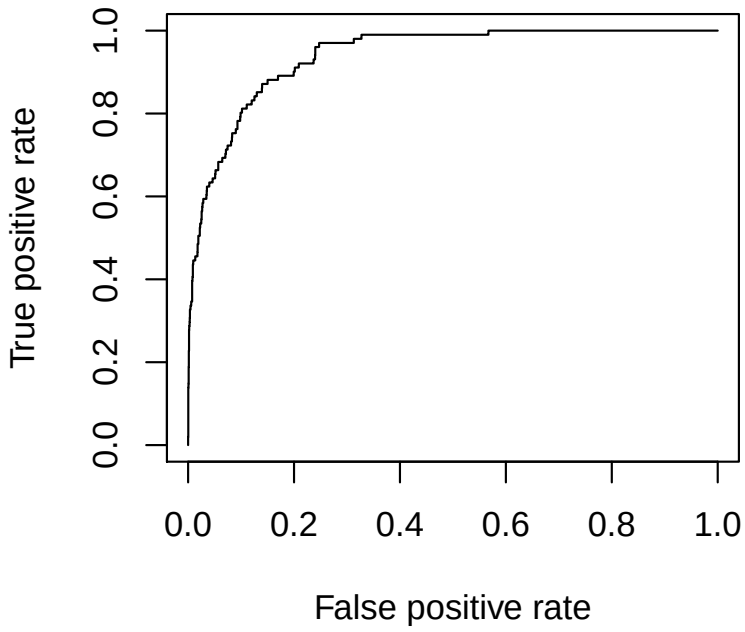

Figure 4: A ROC curve for the logistic regression classifier on the Default data. It traces out two types of error as we vary the threshold value for the posterior probability of default. The actual thresholds are not shown. The true positive rate is the sensitivity: the fraction of defaulters that are correctly identified, using a given threshold value. The false positive rate is 1-specificity: the fraction of non-defaulters that we classify incorrectly as defaulters, using that same threshold value. The ideal ROC curve hugs the top left corner, indicating a high true positive rate and a low false positive rate. The dotted line represents the "no information" (random guess) classifier; this is what we would expect if student status and credit card balance are not associated with probability of default.

consider an alternative and less direct approach to estimating these probabilities using Bayes' theorem:

$$p(Y = k \mid X = x) = \frac{p(Y = k) \times p(X = x \mid Y = k)}{p(X = x)},$$

- $\pi_k := p(Y = k)$ represent the overall or *prior* probability that a randomly chosen observation comes from the kth class
- $f_k(x) := p(X = x \mid Y = k)$ denote the *density function* of X for an observation that comes from the k th class.
- $p_k(x) := p(Y = k \mid X = x)$ as the *posterior* probability that an observation $X = x$ belongs to the kth class. That is, it is the probability that the observation belongs to the k th class, *given* the predictor value for that observation.

The above equation can be written as

$$p_k(x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)} \tag{1}$$

Example

American Cancer Society estimates that about 1.7% 0f women have breast cancer.

Susan G.Komen For the Cure Foundation states that mammography correctly identified about 78% of women who truly have breast cancer.

An article published in 2003 suggests that up tp 10% of all mammograms are false positive.

- Prior to any testing and any information exchange between the patient and the doctor, what probability should a doctor assign to a female patient having breast cancer?
- When a patient goes through breast cancer screening there are two competing claims: patient has cancer and patient doesn't have cancer. If a mammogram yields a positive result, what is the probability that patient has cancer?
- Since a positive mammogram doesn't necessarily mean that the patient actually has breast cancer, the doctor might decide to retest the patient. What is the probability of having breast cancer if this second mammogram also yields a positive result?

Now the "naive" conditional independence assumptions come into play: assume that each feature X_i is *conditionally independent* of every other feature X_j for $j \neq i$, given the category k. This means that

$$p(x_1, x_2, \ldots, x_p \mid k) = p(x_1 \mid k) \times p(x_2 \mid k) \times \cdots \times p(x_p \mid k)$$

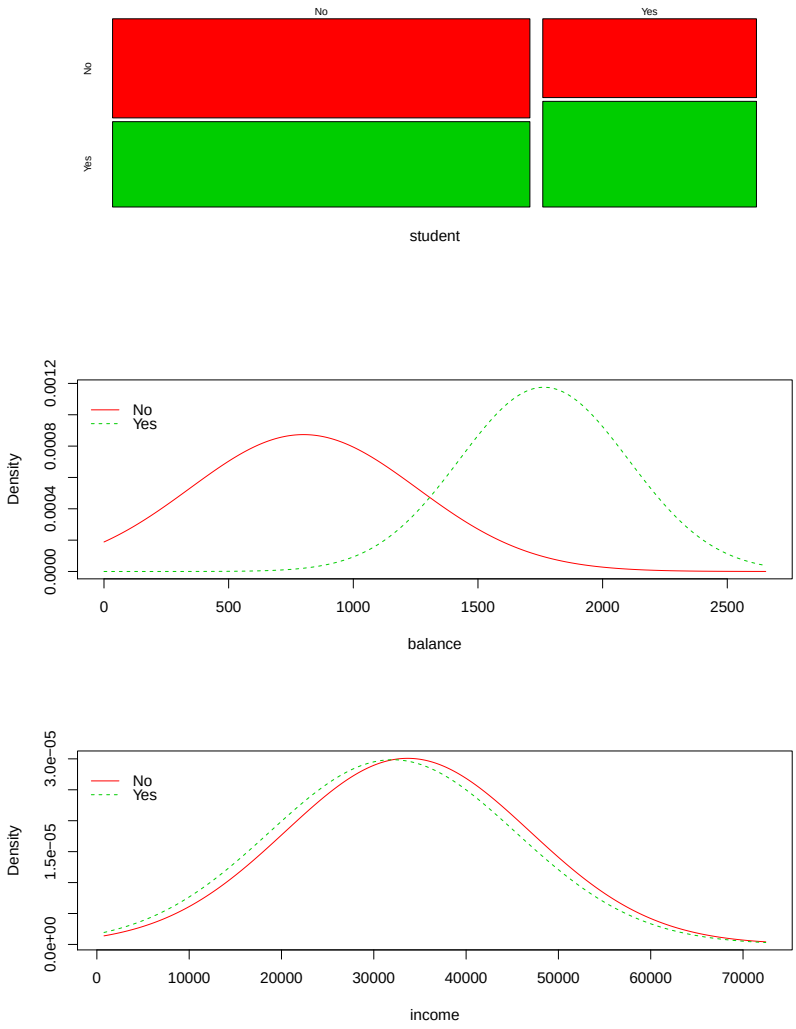

Figure 5: The Naive Bayes classifier for the Default data, normal distribution density for the continuous predictors.

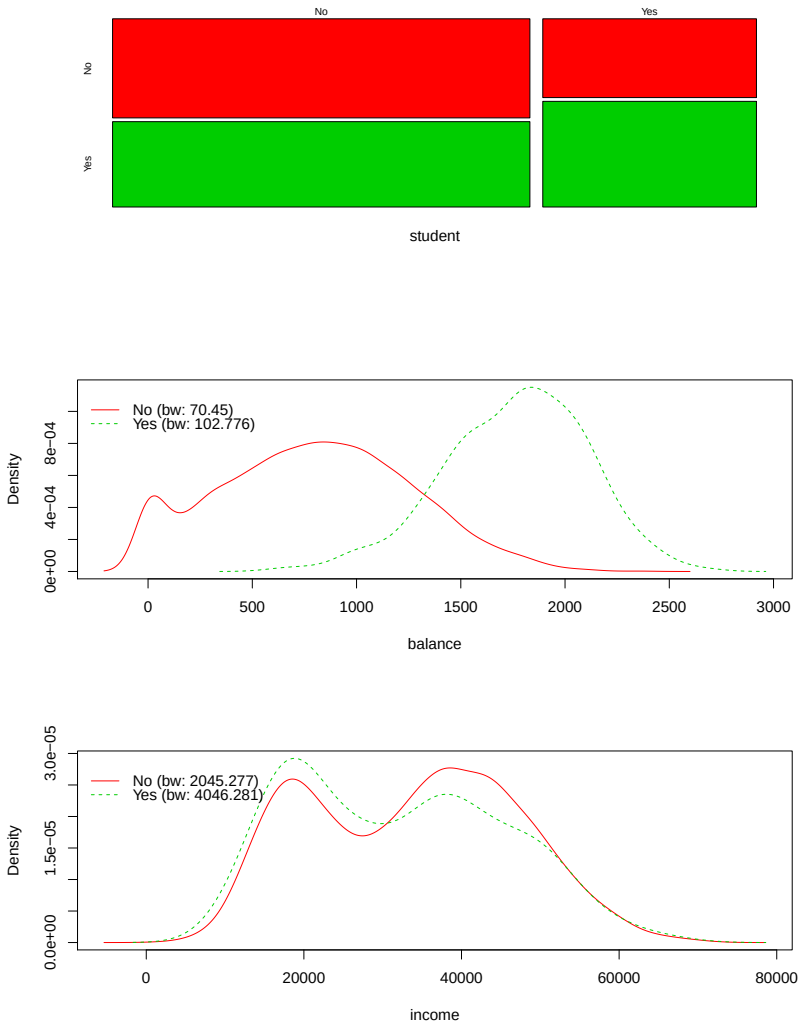

Figure 6: The Naive Bayes classifier for the Default data, kernel based density for the continuous predictors.

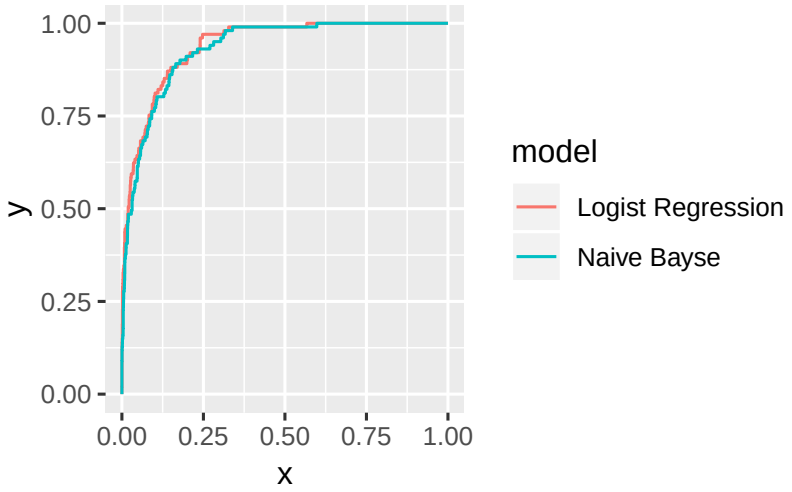

Figure 7: A ROC curve for the Logist classifier and Naive Bayes classifier on the Default data.

Linear Discriminant Analysis (Numerical predictor)

If X is a numerical predictor, in general, estimating $f_k(x)$ tends to be more challenging, unless we assume some simple forms for these densities:

Suppose we assume that $f_k(x)$ is *normal* or *Gaussian*, i.e.

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\Sigma_k|^{1/2}} exp\left(-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)\right), \qquad (2)$$

where μ_k and Σ_k are the mean and covariance matrix for the kth class.

- Linear discriminant analysis (LDA)
  - Assumes that $\Sigma_1 = \ldots = \Sigma_K$, i.e., there is a shared variance term across all K classes.
  - The LDA classifier assigns an observation $X = x$ to the class for which
    $$\delta_k(x) = x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + \log\pi_k$$
    is largest.
- Quadratic discriminant analysis (QDA)
  - Unlike LDA, QDA assumes that each class has its own covariance matrix Σ_k.
  - The QDA classifier assigns an observation $X = x$ to the class for which
    $$\delta_k(x) = -\frac{1}{2}x^T\Sigma_k^{-1}x + x^T\Sigma_k^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma_k^{-1}\mu_k + \log\pi_k$$
    is largest.

- The Bias-Variance trade-off
  - Roughly speaking, LDA tends to be a better bet than QDA if there are relatively few training observations and so reducing variance is crucial.
  - In contrast, QDA is recommended if the training set is very large, so that the variance of the classifier is not a major concern, or if the assumption of a common covariance matrix for the K classes is clearly untenable.

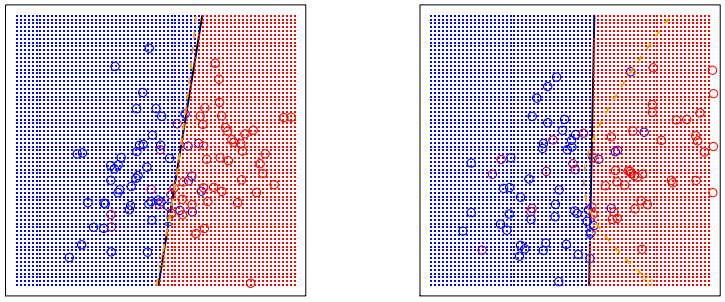

Figure 8: The LDA and QDA decision boundary for a two-class problem with $\Sigma_1 = \Sigma_2$ (Left), and $\Sigma_1 \neq \Sigma_2$ (Right).

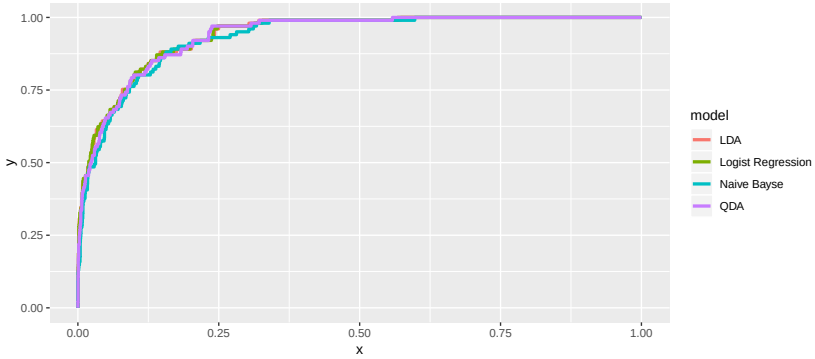

Figure 9: A ROC curve for the Logist, Naive Bayes, LDA and QDA classifier on the Default data.

K-nearst neighbors (KNN)

- non-parametric method
- given a positive integer K and a test observation x_0
- identifies the K points in the training data that are closest to x_0. respresented by N_0

- estimates the conditional probability for class j as the fraction of points in N_0 whose response values equal j:

$$Pr(Y = J \mid X = x_0) = \frac{1}{K} \sum_{i \in N_0} I(y_i = j)$$

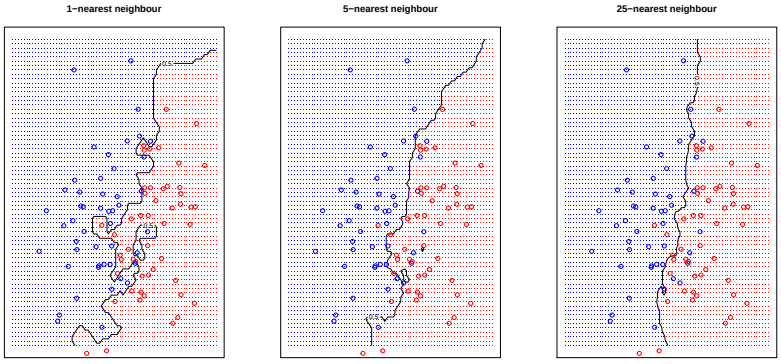

Figure 10: The knn decision boundary for a two-class problem with $\Sigma_1 = \Sigma_2$

A Comparison of Classification Methods

- Both Logistic regression and LDA produce linear decision boundaries, the only difference is:
  - Logistic regression using maximum likelihood;
  - LDA using the estimated mean and variance from a normal distribution;
  - When the Gaussian assumptions are correct, LDA can outperform Logistic regression; Conversely, Logistic regression can outperform LDA if these Gaussian assumptions are not met.
- KNN is a completely non-parametric approach, no assumptions are made about the shape of the decision boundary
  - when the decision boundary is highly non-linear, this approach to dominate LDA and logistic regression
  - But KNN does not tell us which predictors are important
- QDA serves as a compromise between the non-parametric KNN method and the linear LDA and logistic regression.