

Chapter 4: Classification

The linear model in Ch. 3 assumes the response variable Y is quantitative. But in many situations, the response is categorical.

eg. eye color
cancer diagnosis
which movie I will watch next

In this chapter we will look at approaches for predicting categorical responses, a process known as classification.

Classification problems occur often, perhaps even more so than regression problems. Some examples include

1. A person arrives at emergency room with a set of symptoms that could be attributed to one of three medical conditions. Which of the three conditions does the person have?
2. An online banking service must be able to determine whether a transaction is fraudulent on the basis of user's IP address, past transaction history, etc.
3. Something is in the street in front of the self-driving car you are riding in. Is it a human or another car?

As with regression, in the classification setting we have a set of training observations $(x_1, y_1), \dots, (x_n, y_n)$ that we can use to "build a classifier". We want our classifier to perform well on the training data and also on data not used to fit the model (test data).

most importantly.

We will use the `Default` data set in the `ISLR` package for illustrative purposes. We are interested in predicting whether a person will default on their credit card payment on the basis of annual income and credit card balance.



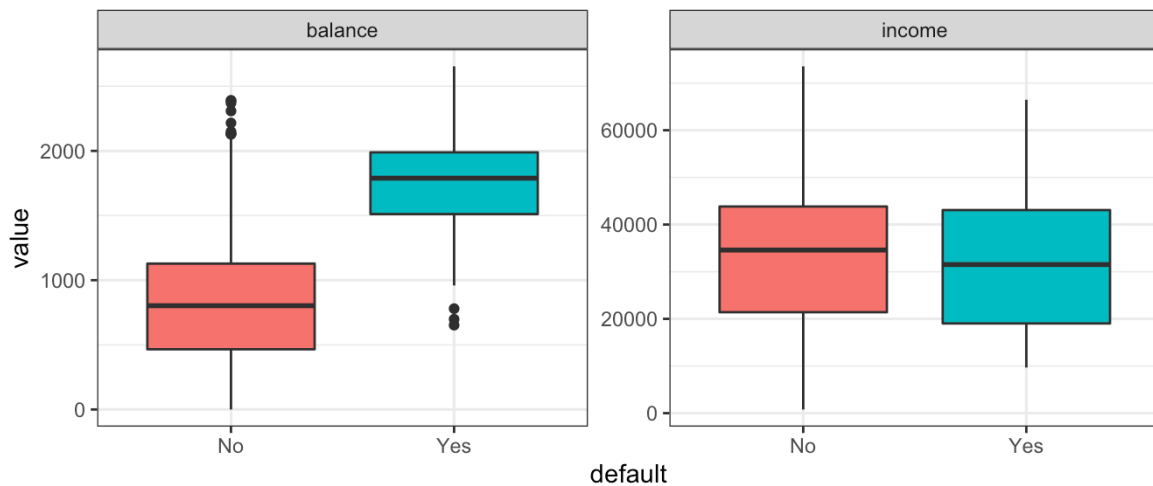
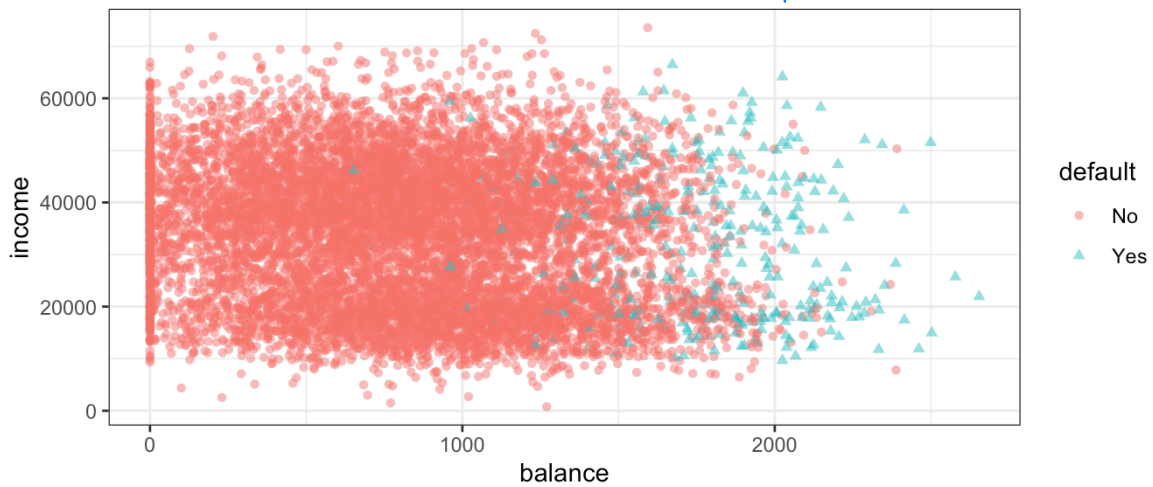
yes or no \Rightarrow categorical.

"fit a model"

head (Default)

		y	features		
##	default	student	balance	income	
## 1	No	No	729.5265	44361.625	
## 2	No	Yes	817.1804	12106.135	
## 3	No	No	1073.5492	31767.139	
## 4	No	No	529.2506	35704.494	
## 5	No	No	785.6559	38463.496	
## 6	No	Yes	919.5885	7491.559	

decent separation based on balance



pronounced relationship between balance and default

in a lot of cases, relationship is not so pronounced \Rightarrow classification will be harder.

1 Why not Linear Regression?

I have said that linear regression is not appropriate in the case of a categorical response. Why not?

Let's try it anyways. We could consider encoding the values of `default` in a quantitative response variable Y

$$Y = \begin{cases} 1 & \text{if default} \\ 0 & \text{otherwise} \end{cases}$$

Using this coding, we could then fit a linear regression model to predict Y on the basis of `income` and `balance`. This implies an ^①ordering on the outcome, not defaulting comes first before defaulting and ^②insists the difference between these two outcomes is 1 unit. In practice, there is no reason for this to be true.

we could let $Y = \begin{cases} 0 & \text{if default} \\ 1 & \text{otherwise} \end{cases}$

OR $Y = \begin{cases} 1 & \text{if default} \\ 10 & \text{otherwise} \end{cases}$

there is no natural reason why 0/1 encoding, but it has an advantage:

Using the dummy encoding, we can get a ^{0/1} rough estimate of $P(\text{default}|X)$, but it is not guaranteed to be scaled correctly.

↓
doesn't have to be between 0 and 1
but will provide an ordering.

Real problem: this cannot be easily extended to more than 2 classes.

We can instead use methods specifically formulated for categorical responses.

2 Logistic Regression

Let's consider again the `default` variable which takes values **Yes** or **No**. Rather than modeling the response directly, logistic regression models the probability that Y belongs to a particular category.

e.g. $P(\text{default} = \text{Yes} \mid \text{balance})$

we can abbreviate this as $p(\text{balance}) \in [0, 1]$.

For any given value of `balance`, a prediction can be made for `default`.

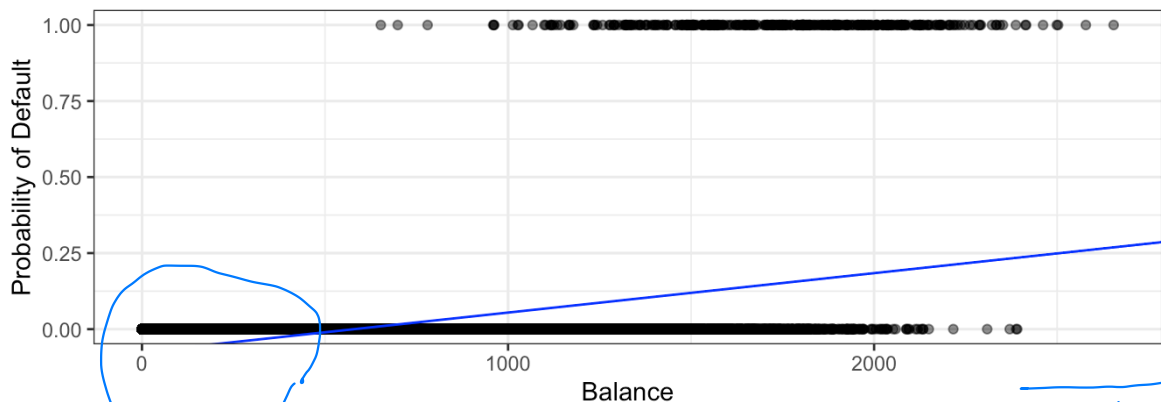
e.g. predict `default = Yes` if $p(\text{balance}) > 0.5$

or the company could be more conservative predict `default = Yes` if $p(\text{balance}) > \underbrace{0.1}_{\text{threshold}}$.

2.1 The Model

How should we model the relationship between $p(X) = P(Y = 1 \mid X)$ and X ? We could use a linear regression model to represent those probabilities

$$p(X) = \beta_0 + \beta_1 X$$



problem:
for balances close to
zero we predict negative
probability of default

if we had a large
balance would have
predicted prob. > 1
of defaulting

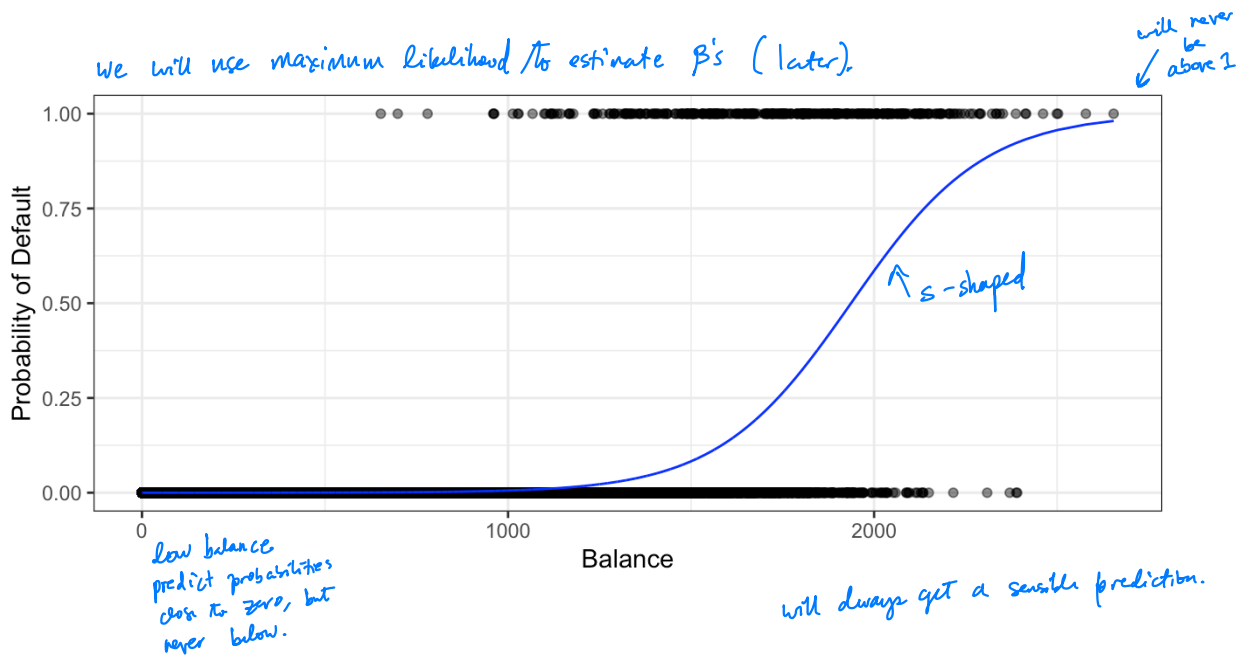
neither of these
make any sense.

standard logistic function

$$f(x) = \frac{e^x}{1 + e^x}$$

To avoid this, we must model $p(X)$ using a function that gives outputs between 0 and 1 for all values of X . Many functions meet this description, but in logistic regression, we use the logistic function,

$$p(x) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}$$



After a bit of manipulation,

By taking the logarithm of both sides we see,

Recall from Ch. 3 that β_1 gives the “average change in Y associated with a one unit increase in X .” In contrast, in a logistic model,

However, because the relationship between $p(X)$ and X is not linear, β_1 does **not** correspond to the change in $p(X)$ associated with a one unit increase in X . The amount that $p(X)$ changes due to a 1 unit increase in X depends on the current value of X .

2.2 Estimating the Coefficients

The coefficients β_0 and β_1 are unknown and must be estimated based on the available training data. To find estimates, we will use the method of *maximum likelihood*.

The basic intuition is that we seek estimates for β_0 and β_1 such that the predicted probability $\hat{p}(x_i)$ of default for each individual corresponds as closely as possible to the individual's observed default status.

```
logistic_spec <- logistic_reg()

logistic_fit <- logistic_spec |>
  fit(default ~ balance, family = "binomial", data = Default)

logistic_fit |>
  pluck("fit") |>
  summary()

##
## Call:
## stats::glm(formula = default ~ balance, family = stats::binomial,
##   data = data)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.2697  -0.1465  -0.0589  -0.0221   3.7589
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.065e+01  3.612e-01  -29.49  <2e-16 ***
## balance      5.499e-03  2.204e-04   24.95  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 2920.6  on 9999  degrees of freedom
## Residual deviance: 1596.5  on 9998  degrees of freedom
## AIC: 1600.5
##
## Number of Fisher Scoring iterations: 8
```

2.3 Predictions

Once the coefficients have been estimated, it is a simple matter to compute the probability of `default` for any given credit card balance. For example, we predict that the default probability for an individual with `balance` of \$1,000 is

In contrast, the predicted probability of default for an individual with a balance of \$2,000 is

2.4 Multiple Logistic Regression

We now consider the problem of predicting a binary response using multiple predictors. By analogy with the extension from simple to multiple linear regression,

Just as before, we can use maximum likelihood to estimate $\beta_0, \beta_1, \dots, \beta_p$.

```
logistic_fit2 <- logistic_spec |>
  fit(default ~ ., family = "binomial", data = Default)

logistic_fit2 |>
  pluck("fit") |>
  summary()
```

```
##
## Call:
## stats::glm(formula = default ~ ., family = stats::binomial, data =
## data)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.4691  -0.1418  -0.0557  -0.0203   3.7383
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.087e+01  4.923e-01 -22.080  < 2e-16 ***
## studentYes   -6.468e-01  2.363e-01  -2.738  0.00619 **
## balance       5.737e-03  2.319e-04  24.738  < 2e-16 ***
## income       3.033e-06  8.203e-06   0.370  0.71152
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 2920.6  on 9999  degrees of freedom
## Residual deviance: 1571.5  on 9996  degrees of freedom
## AIC: 1579.5
##
## Number of Fisher Scoring iterations: 8
```

By substituting estimates for the regression coefficients from the model summary, we can make predictions. For example, a student with a credit card balance of \$1,500 and an income of \$40,000 has an estimated probability of default of

A non-student with the same balance and income has an estimated probability of default of

2.5 Logistic Regression for > 2 Classes

We sometimes wish to classify a response variable that has more than two classes. There are multi-class extensions to logistic regression (“multinomial regression”), but there are far more popular methods of performing this.

3 LDA

Logistic regression involves direction modeling $P(Y = k|X = x)$ using the logistic function for the case of two response classes. We now consider a less direct approach.

Idea:

Why do we need another method when we have logistic regression?

1.

2.

3.

3.1 Bayes' Theorem for Classification

Suppose we wish to classify an observation into one of K classes, where $K \geq 2$.

$$\pi_k$$

$$f_k(x)$$

$$P(Y = k|X = x)$$

In general, estimating π_k is easy if we have a random sample of Y 's from the population.

Estimating $f_k(x)$ is more difficult unless we assume some particular forms.

3.2 $p = 1$

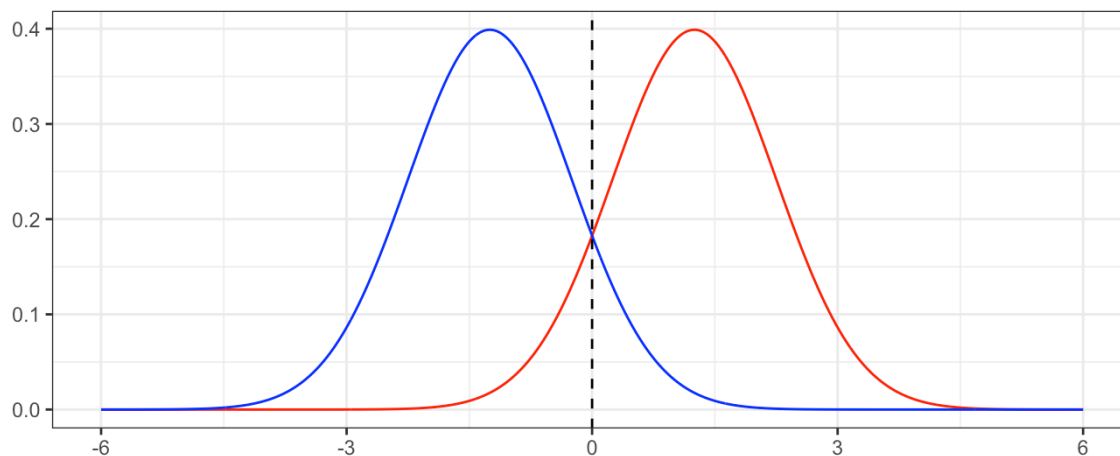
Let's (for now) assume we only have 1 predictor. We would like to obtain an estimate for $f_k(x)$ that we can plug into our formula to estimate $p_k(x)$. We will then classify an observation to the class for which $\hat{p}_k(x)$ is greatest.

Suppose we assume that $f_k(x)$ is normal. In the one-dimensional setting, the normal density takes the form

Plugging this into our formula to estimate $p_k(x)$,

We then assign an observation $X = x$ to the class which makes $p_k(x)$ the largest. This is equivalent to

Example 3.1 Let $K = 2$ and $\pi_1 = \pi_2$. When does the Bayes classifier assign an observation to class 1?



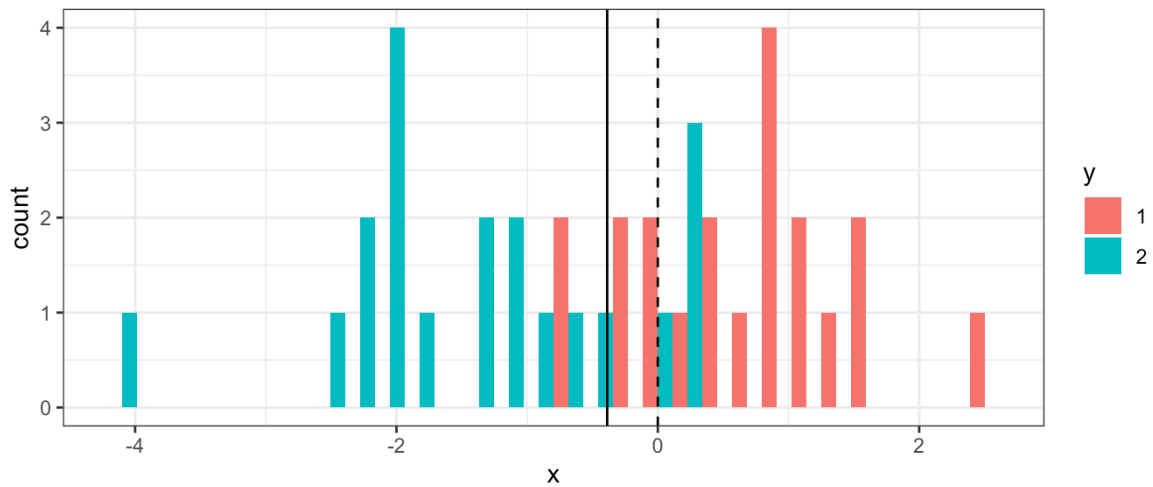
In practice, even if we are certain of our assumption that X is drawn from a Gaussian distribution within each class, we still have to estimate the parameters

$$\mu_1, \dots, \mu_K, \pi_1, \dots, \pi_K, \sigma^2.$$

The *linear discriminant analysis* (LDA) method approximated the Bayes classifier by plugging estimates in for π_k, μ_k, σ^2 .

Sometimes we have knowledge of class membership probabilities π_1, \dots, π_K that can be used directly. If we do not, LDA estimates π_k using the proportion of training observations that belong to the k th class.

The LDA classifier assigns an observation $X = x$ to the class with the highest value of



```
##      pred
## y      1      2
## 1 18966 1034
## 2  3855 16145
```

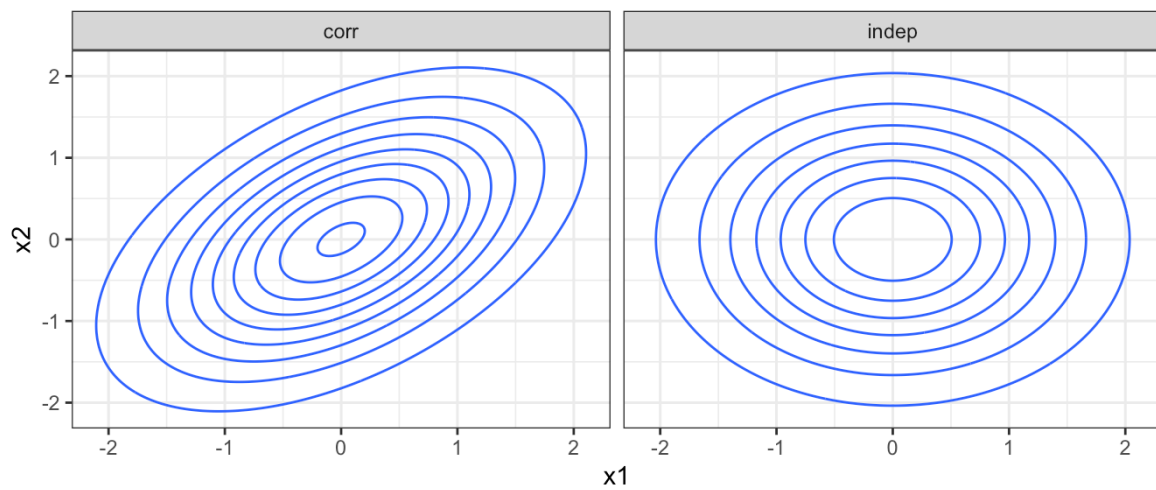
The LDA test error rate is approximately 12.22% while the Bayes classifier error rate is approximately 10.52%.

The LDA classifier results from assuming that the observations within each class come from a normal distribution with a class-specific mean vector and a common variance σ^2 and plugging estimates for these parameters into the Bayes classifier.

3.3 $p > 1$

We now extend the LDA classifier to the case of multiple predictors. We will assume

Formally the multivariate Gaussian density is defined as



In the case of $p > 1$ predictors, the LDA classifier assumes the observations in the k th class are drawn from a multivariate Gaussian distribution $N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma})$.

Plugging in the density function for the k th class, results in a Bayes classifier

Once again, we need to estimate the unknown parameters $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K, \pi_1, \dots, \pi_K, \boldsymbol{\Sigma}$.

To classify a new value $X = x$, LDA plugs in estimates into $\delta_k(x)$ and chooses the class which maximized this value.

Let's perform LDA on the `Default` data set to predict if an individual will default on their CC payment based on balance and student status.

```
lda_spec <- discrim_linear(engine = "MASS")

lda_fit <- lda_spec |>
  fit(default ~ student + balance, data = Default)

lda_fit |>
  pluck("fit")

## Call:
## lda(default ~ student + balance, data = data)
##
## Prior probabilities of groups:
##      No      Yes
## 0.9667 0.0333
##
## Group means:
##      studentYes  balance
## No    0.2914037  803.9438
## Yes   0.3813814 1747.8217
##
## Coefficients of linear discriminants:
##                      LD1
## studentYes -0.249059498
## balance    0.002244397
```

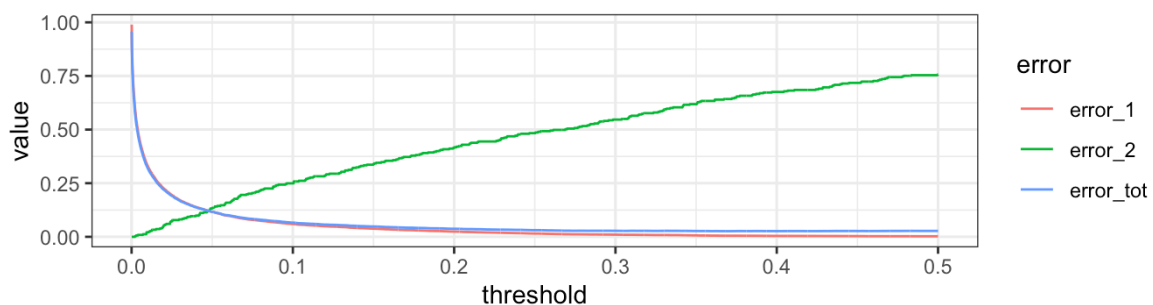
```
# training data confusion matrix
lda_fit |>
  augment(new_data = Default) |>
  conf_mat(truth = default, estimate = .pred_class)
```

```
##           Truth
## Prediction   No  Yes
##           No 9644 252
##           Yes  23   81
```

Why does the LDA classifier do such a poor job of classifying the customers who default?

```
lda_fit |>
  augment(new_data = Default) |>
  mutate(pred_lower_cutoff = factor(ifelse(.pred_Yes > 0.2, "Yes",
                                           "No"))) |>
  conf_mat(truth = default, estimate = pred_lower_cutoff)
```

```
##           Truth
## Prediction   No  Yes
##           No 9432 138
##           Yes 235 195
```



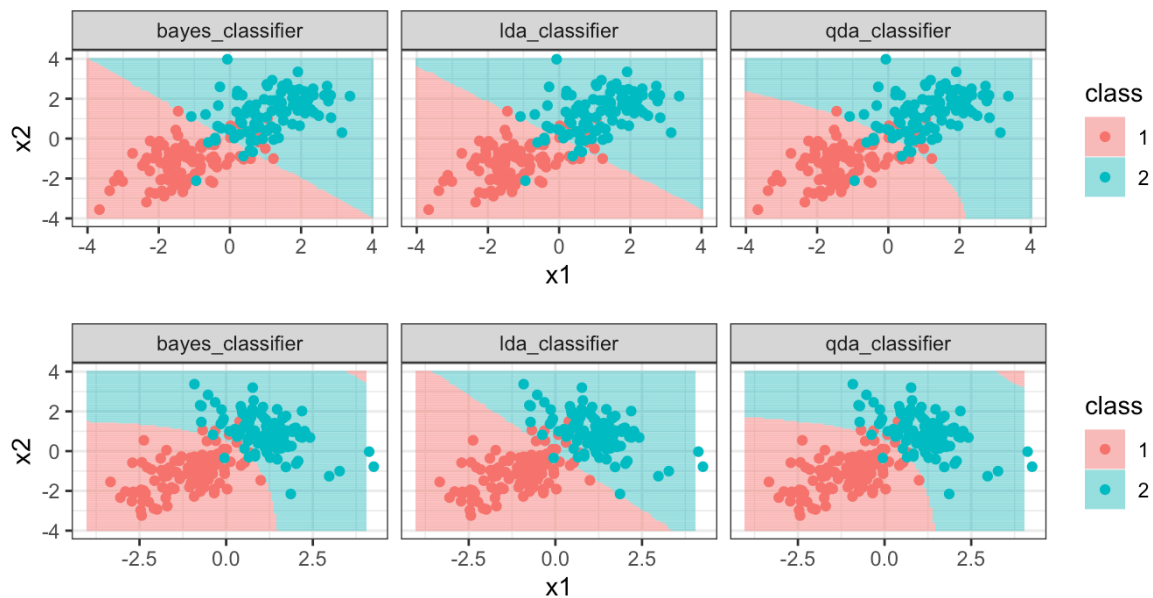
3.4 QDA

LDA assumes that the observations within each class are drawn from a multivariate Gaussian distribution with a class-specific mean vector and a common covariance matrix across all K classes.

Quadratic Discriminant Analysis (QDA) also assumes the observations within each class are drawn from a multivariate Gaussian distribution with a class-specific mean vector but now each class has its own covariance matrix.

Under this assumption, the Bayes classifier assigns observation $X = x$ to class k for whichever k maximizes

When would we prefer QDA over LDA?

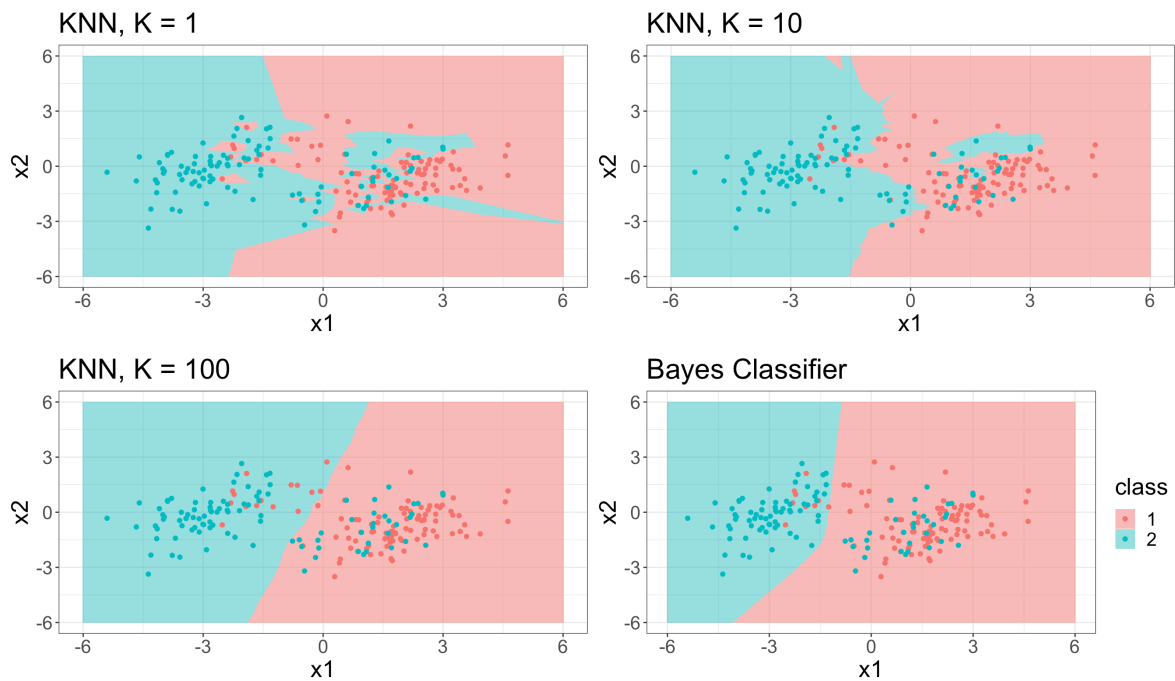


4 KNN

Another method we can use to estimate $P(Y = k|X = x)$ (and thus estimate the Bayes classifier) is through the use of K -nearest neighbors.

The KNN classifier first identifies the K points in the training data that are closest to the test data point $X = x$, called $\mathcal{N}(x)$.

Just as with regression tasks, the choice of K (neighborhood size) has a drastic effect on the KNN classifier obtained.



5 Comparison

LDA vs. Logistic Regression

(LDA & Logistic Regression) vs. KNN

QDA