Classification with Linear Discriminant Analysis

Classification with LDA

LDA

We have existing observations

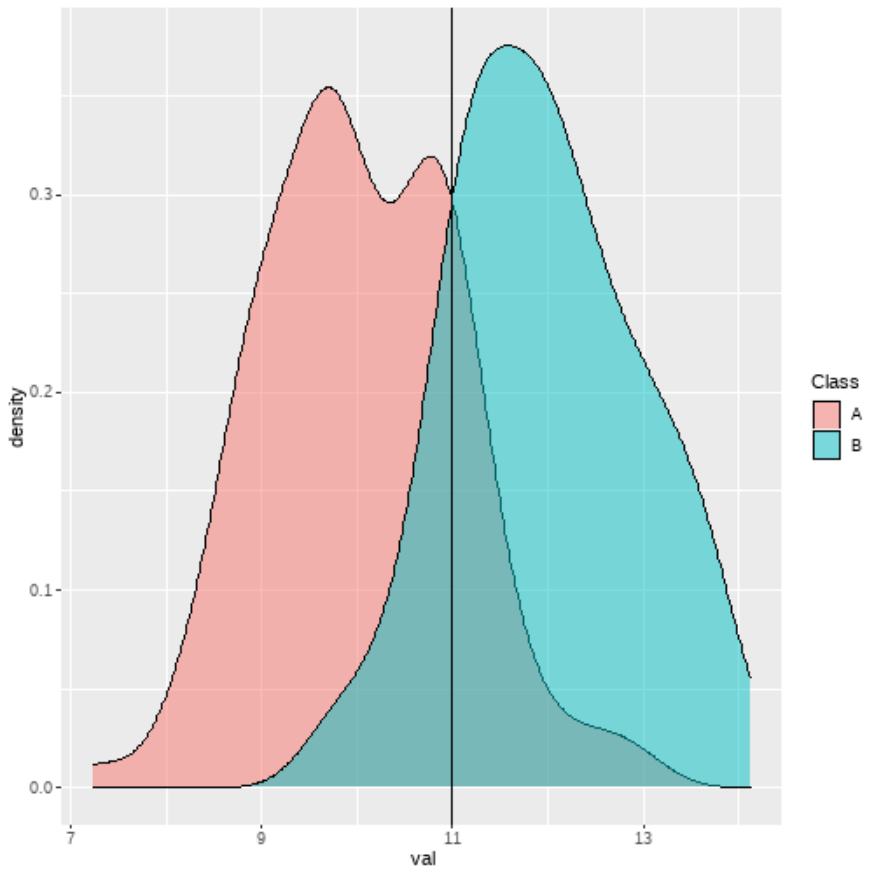
$$(x_1, C_1), \ldots (x_n, C_n)$$

where the C_i are categories.

Given a new observation x_{new}, how do we predict C_{new}?

Come up with a "cutoff": if $x_{new} > $ cutoff, predict class A, if not, predict class B.

LDA

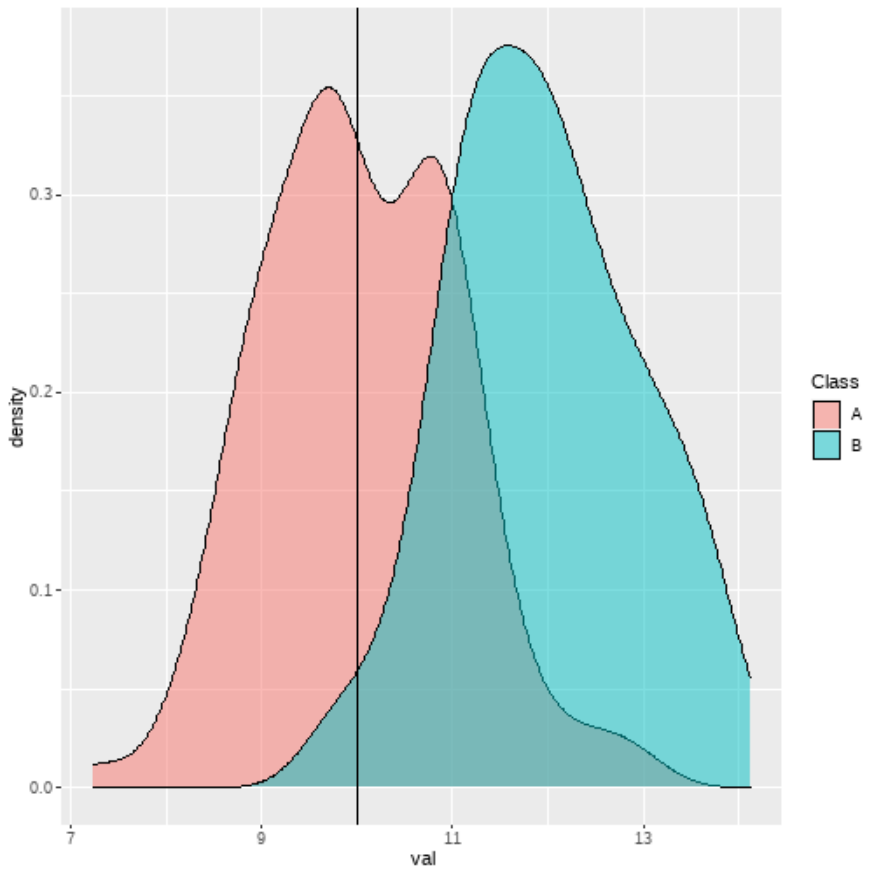


LDA

Cutoff of 11:

```
## # A tibble: 4 x 3
##   Class pred_class     n
##   <chr> <chr>      <int>
## 1 A     A             84
## 2 A     B             16
## 3 B     A             16
## 4 B     B             84
```

LDA



LDA

Cutoff of 10:

```
dat %>%
  mutate(
    pred_class = case_when(
      val > 10 ~ "B",
      TRUE ~ "A"
    )
  ) %>%
  count(Class, pred_class)
```

```
## # A tibble: 4 x 3
##   Class pred_class     n
##   <chr> <chr>      <int>
## 1 A     A             52
## 2 A     B             48
## 3 B     A              3
## 4 B     B             97
```

LDA

To perform classification with Linear Discriminant Analysis, we choose the *best dividing line* between the two classes.

The Big Questions:

- What is our definition of best?
- What if we allow the line to "wiggle"?

Example

Example

Let's keep hanging out with the insurance dataset.

Suppose we want to use information about insurance charges to predict whether someone is a smoker or not.

```
ins <- read_csv("https://www.dropbox.com/s/bocjjyo1ehr5auz/insurance.csv?dl=1")

ins <- ins %>%
  mutate(
    smoker = factor(smoker)
  ) %>%
  drop_na()
```

Quick Quiz

What do we have to change?

The model?

The recipe?

The workflow?

The fit?

Example

Just the model needs to change, of course!

```
lda_mod <- discrim_linear() %>%
  set_engine("MASS") %>%
  set_mode("classification")
```

Example

Fit our model:

```
lda_fit_1 <- lda_mod %>%
  fit(smoker ~ charges, data = ins)

lda_fit_1$fit %>% summary()
```

```
##         Length Class  Mode
## prior   2      -none- numeric
## counts  2      -none- numeric
## means   2      -none- numeric
## scaling 1      -none- numeric
## lev     2      -none- character
## svd     1      -none- numeric
## N       1      -none- numeric
## call    3      -none- call
## terms   3      terms  call
## xlevels 0      -none- list
```

Example

```
lda_fit_1
```

```
## parsnip model object
##
## Fit time:  0ms
## Call:
## lda(smoker ~ charges, data = data)
##
## Prior probabilities of groups:
##     no    yes
## 0.7981 0.2019
##
## Group means:
##     charges
## no     7528
## yes   31152
##
## Coefficients of linear discriminants:
##             LD1
## charges 0.00014
```

Example

```
preds <- lda_fit_1 %>% predict(ins)

ins <- ins %>%
  mutate(
    pred_smoker = preds$.pred_class
  )

ins %>%
  accuracy(truth = smoker,
           estimate = pred_smoker)
```

```
## # A tibble: 1 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.921
```

What if we want to use more than one predictor?

Example 2

```
lda_fit_2 <- lda_mod %>%
  fit(smoker ~ charges + age, data = ins)

lda_fit_2
```

```
## parsnip model object
##
## Fit time:  0ms
## Call:
## lda(smoker ~ charges + age, data = data)
##
## Prior probabilities of groups:
##     no    yes
## 0.7981 0.2019
##
## Group means:
##     charges   age
## no     7528 38.30
## yes   31152 36.62
##
## Coefficients of linear discriminants:
##                LD1
## charges  0.0001718
## age     -0.0449953
```

Example

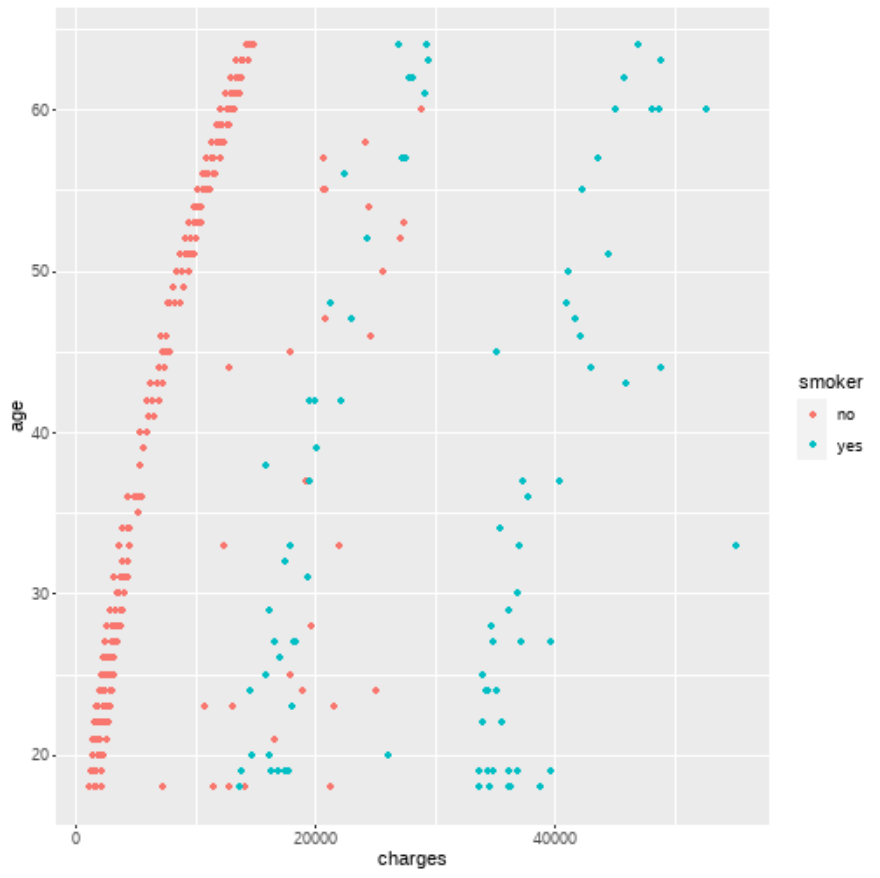
```
lda_fit_2$fit$scaling
```

```
##                LD1
## charges  0.0001718
## age     -0.0449953
```

Score = 0.001718 *charges* + *-0.0444* age

Predict "smoker" if Score > 0

Example



Example

Predict "smoker" if Score > 0

$$0 = 0.001718 \; charges + -0.0444 \; \text{age}$$

$$\text{age} = (0.00178/0.0444)*\text{charges}$$

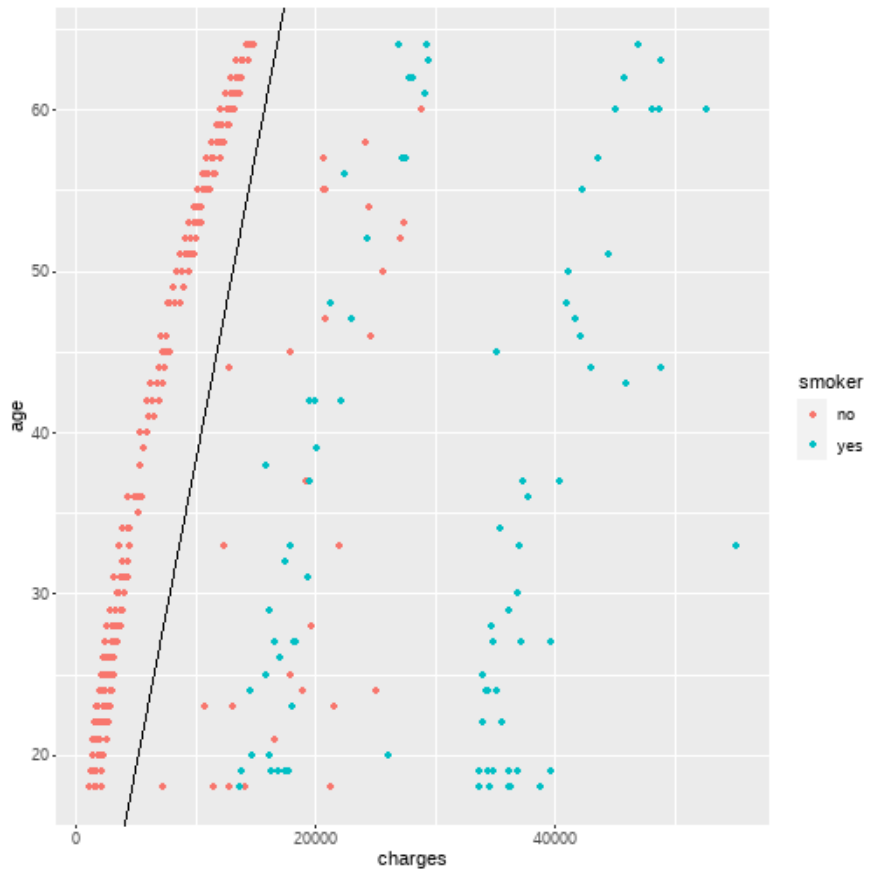
Example

```
lda_fit_2
```

```
## parsnip model object
##
## Fit time:  0ms
## Call:
## lda(smoker ~ charges + age, data = data)
##
## Prior probabilities of groups:
##     no    yes
## 0.7981 0.2019
##
## Group means:
##     charges   age
## no     7528 38.30
## yes   31152 36.62
##
## Coefficients of linear discriminants:
##                LD1
## charges  0.0001718
## age     -0.0449953
```

```
my_slope = lda_fit_2$fit$scaling[1]/(-1*lda_fit_2$fit$scaling[2])
```

Example



Try it!

(you know the drill...)

Open Activity-Classification-2.Rmd

Select the best LDA model for predicting smoker status

Compare the accuracy to your KNN and Logistic Regression models.

Quadratic Discriminant Analysis

One more time: wiggly style

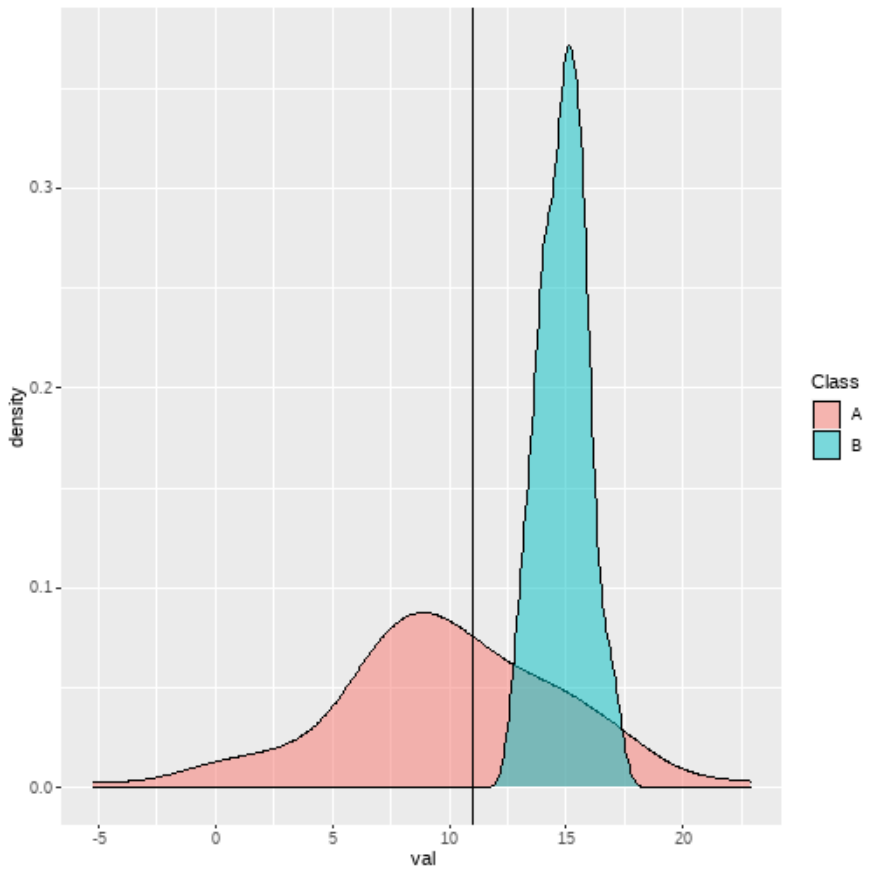
QDA

What if we allow the separating line to be non-linear?

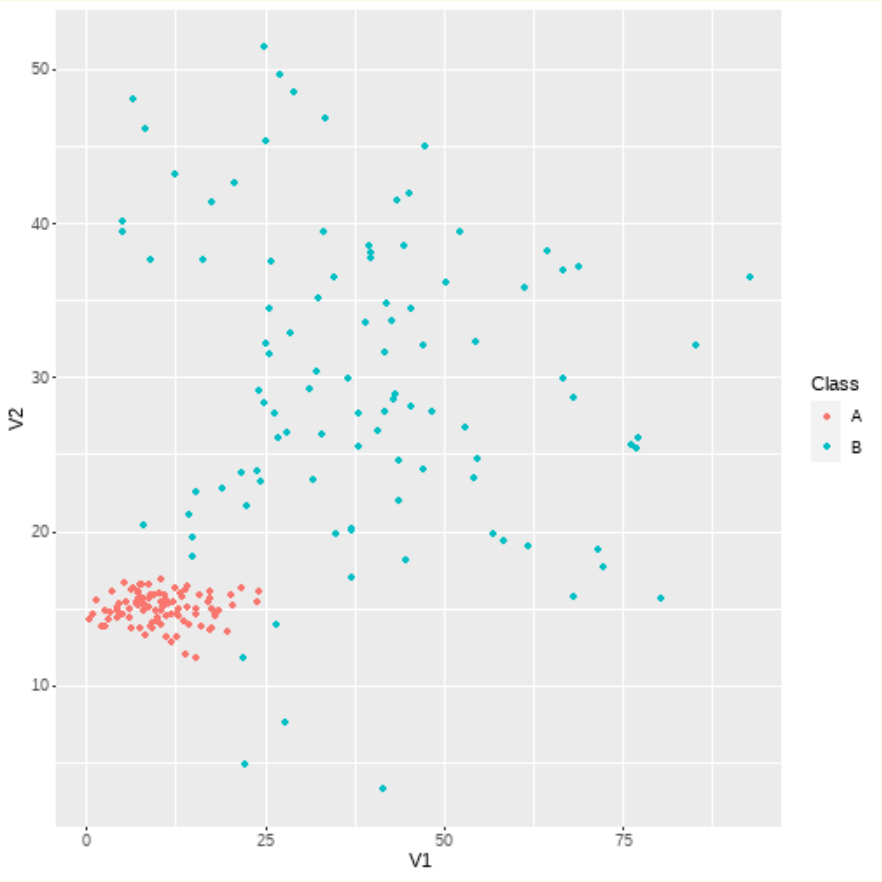
```
qda_mod <- discrim_regularized(frac_common_cov = 0) %>%
             set_engine('klaR') %>%
             set_mode('classification')
```

(i.e., we allow the data in the different categories to have different variances)

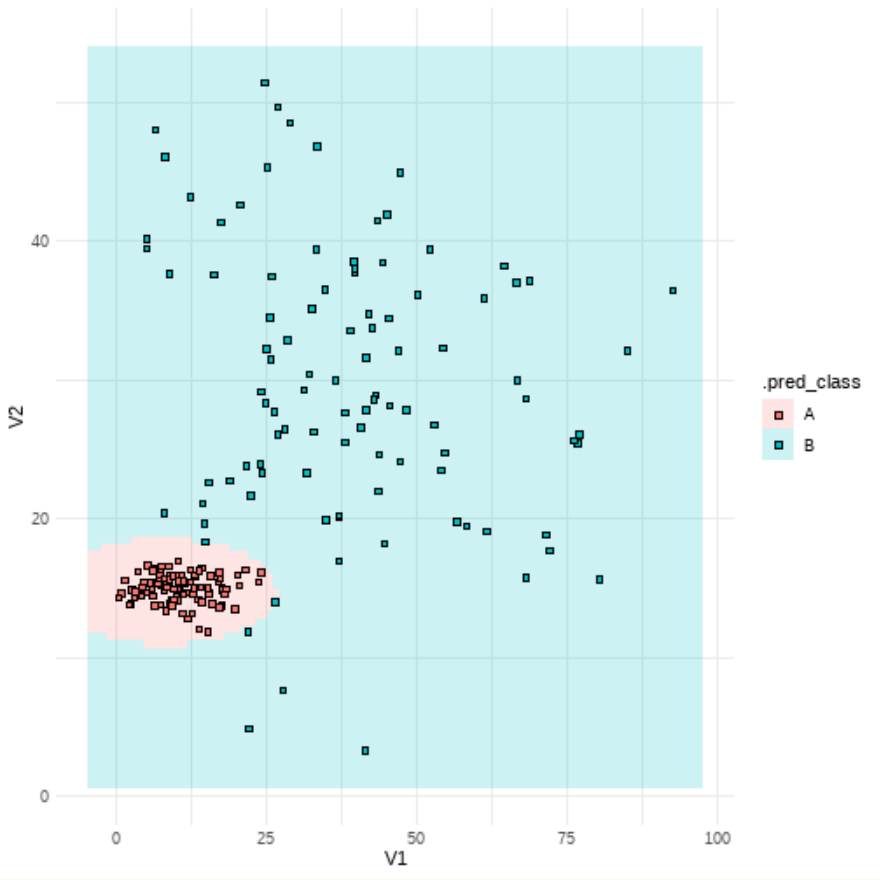
QDA



QDA



QDA



Questions to ponder

- What if we have a categorical variable where 99% of our values are Category A?
- What if we have a categorical variable with more than 2 categories?
- Are there other ways to do classification besides these logistic regression and KNN and Discriminant Analysis?

Try it!

Open Activity-Classification-2.Rmd again

Select the best QDA model

Compare to prior models