MONASH University

ETC3250/5250 Introduction to Machine Learning

Week 7: Explainable artificial intelligence (XAI)

Professor Di Cook

etc3250.clayton-x@monash.edu

Department of Econometrics and Business Statistics

Overview

We will cover:

- Global explainability
- Local explainability
    - LIME
    - Counterfactuals
    - Anchors
    - Shapley values

Global explainability

Variable importance (1/4)

Remember:

- Model coefficients on standardised data
- Effect of collinearity
- Importance from permutation

Variable importance (2/4)

Model coefficients on standardised data

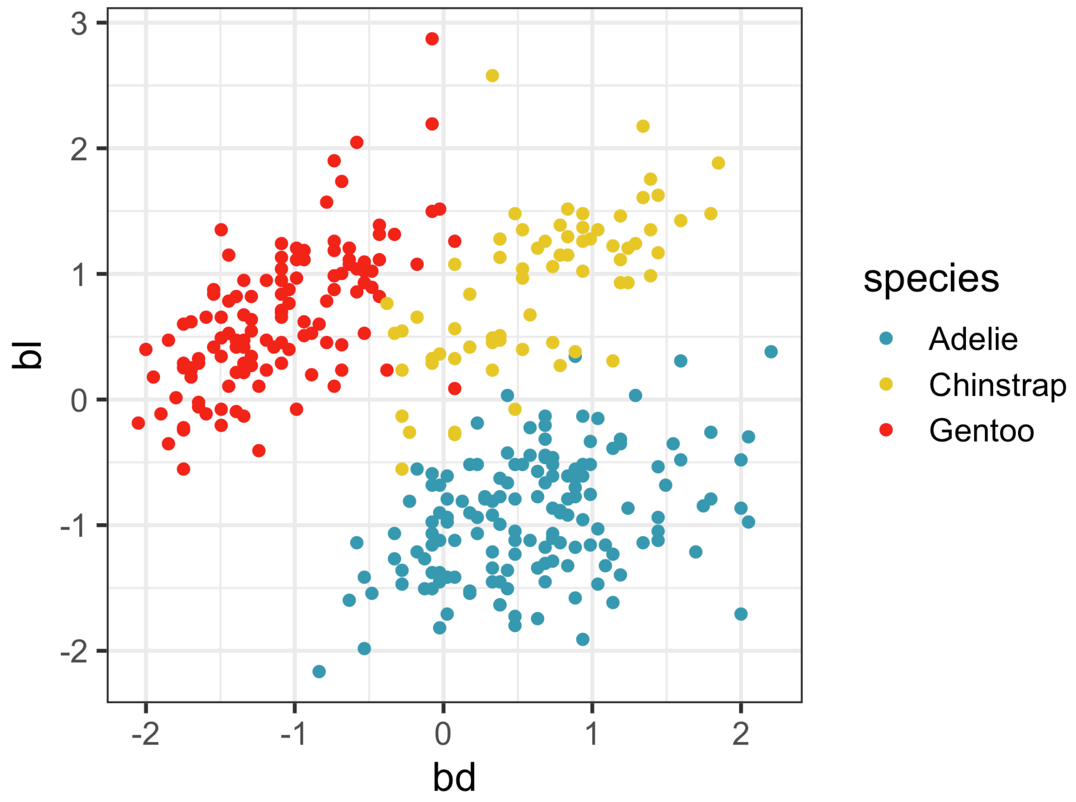
```
parsnip model object

Call:
lda(species ~ ., data = data, prior = ~c(1/3, 1/3, 1/3))

Prior probabilities of groups:
   Adelie Chinstrap    Gentoo 
     0.33      0.33      0.33 

Group means:
             bl    bd    fl    bm
Adelie    -0.94  0.61 -0.78 -0.62
Chinstrap  0.90  0.64 -0.36 -0.58
Gentoo     0.66 -1.10  1.16  1.09
```

`bl` and `bd` are the most important variables

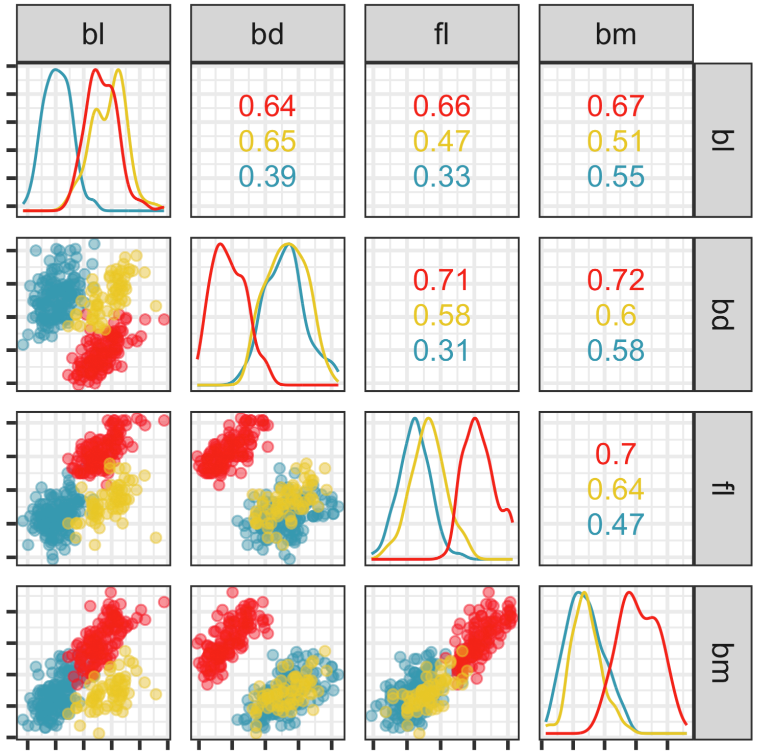


Variable importance (3/4)

When predictors are strongly linearly associated, interpreting coefficients purely on magnitude can be incorrect.

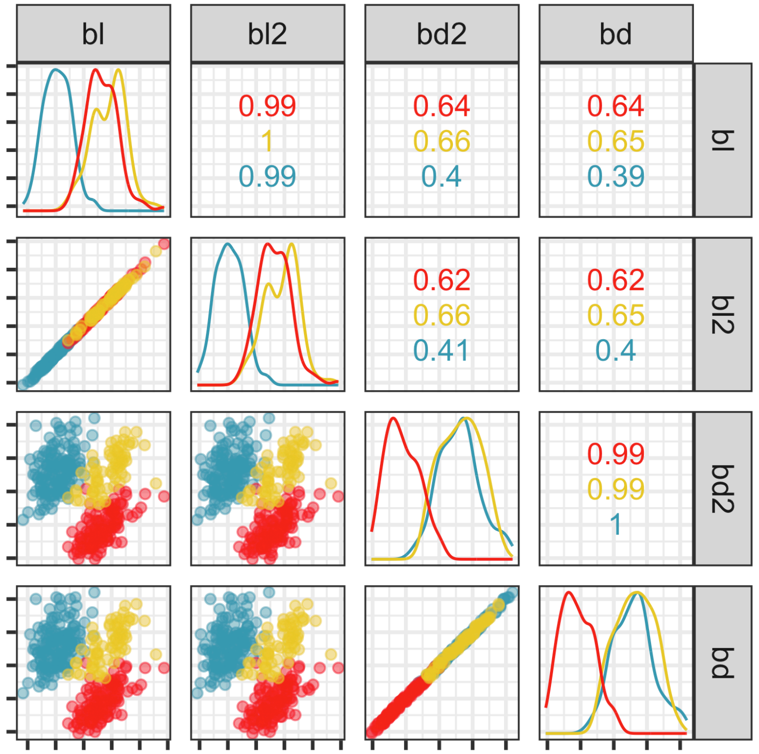
Original

```
      LD1    LD2
bl -0.24 -2.319
bd  2.04  0.172
fl -1.22  0.062
bm -1.18  1.257
```



Correlated variables

```
      LD1    LD2
bl    1.80 -1.921
bl2  -1.57 -0.407
bd2   0.49  1.548
bd   -2.54 -1.375
fl    1.21  0.052
bm    1.21  1.270
```



Permutation variable importance (1/2)

For trained model $\hat{f}$, which depends on data $\mathbf{X}$ to predict response y, with loss function $L(y, \hat{f})$ (e.g. misclassification rate, error),

1. Estimate $L(y, \hat{f})$ on the data, L^{orig}.
2. For each variable $j \in 1, \ldots, p$,
   - Generate data matrix $\mathbf{X}^{\text{perm}}$ by permuting variable j. This breaks the association between variable j and observed y.
   - Compute the $L(y, \hat{f})$ on the permuted data, L^{perm}.
   - Compare L^{orig} and L^{perm}, e.g. $|L^{\text{orig}} - L^{\text{perm}}|$
3. Most important variables have larger values.

Permutation variable importance (2/2)

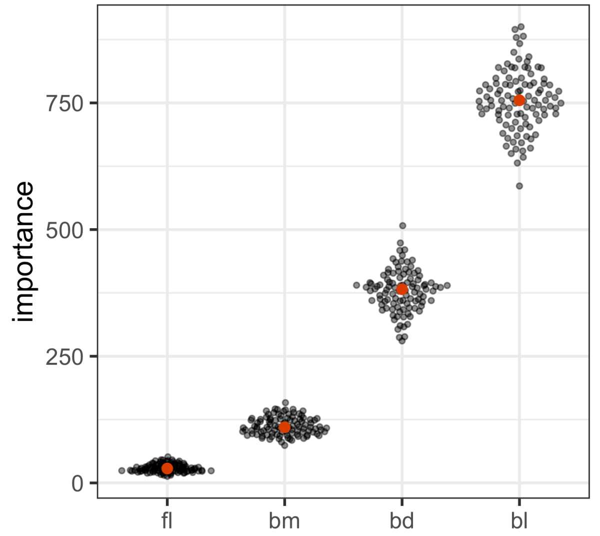
Random forests have this baked into the model fitting (using the out-of-bag cases).

Generally, should be conducted on the test set.

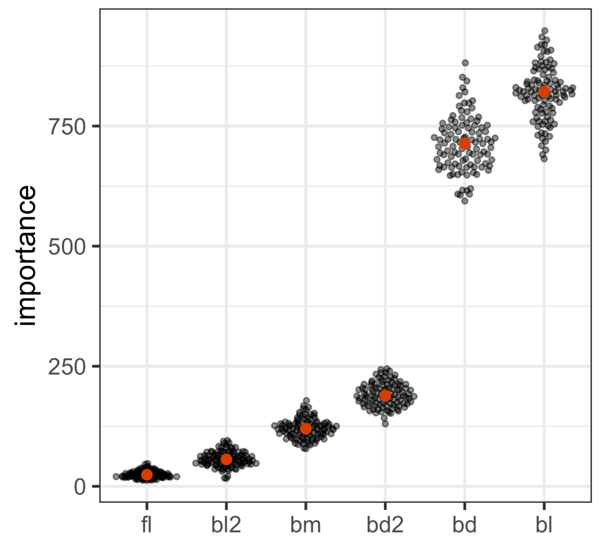
```r
# Using DALEX with tidymodels
# https://www.tmwr.org/explain
# https://ema.drwhy.ai/featureImportance.html
vip_features <- colnames(p_std)[2:5]

vip_train <-
  p_std |>
  select(all_of(vip_features))

explainer_lda <-
  explain_tidymodels(
    lda_fit,
    data = vip_train,
    y = p_std$species,
    verbose = FALSE
  )
vip_lda <- model_parts(explainer_lda,
                        B=100)
```



Data with additional correlated variables



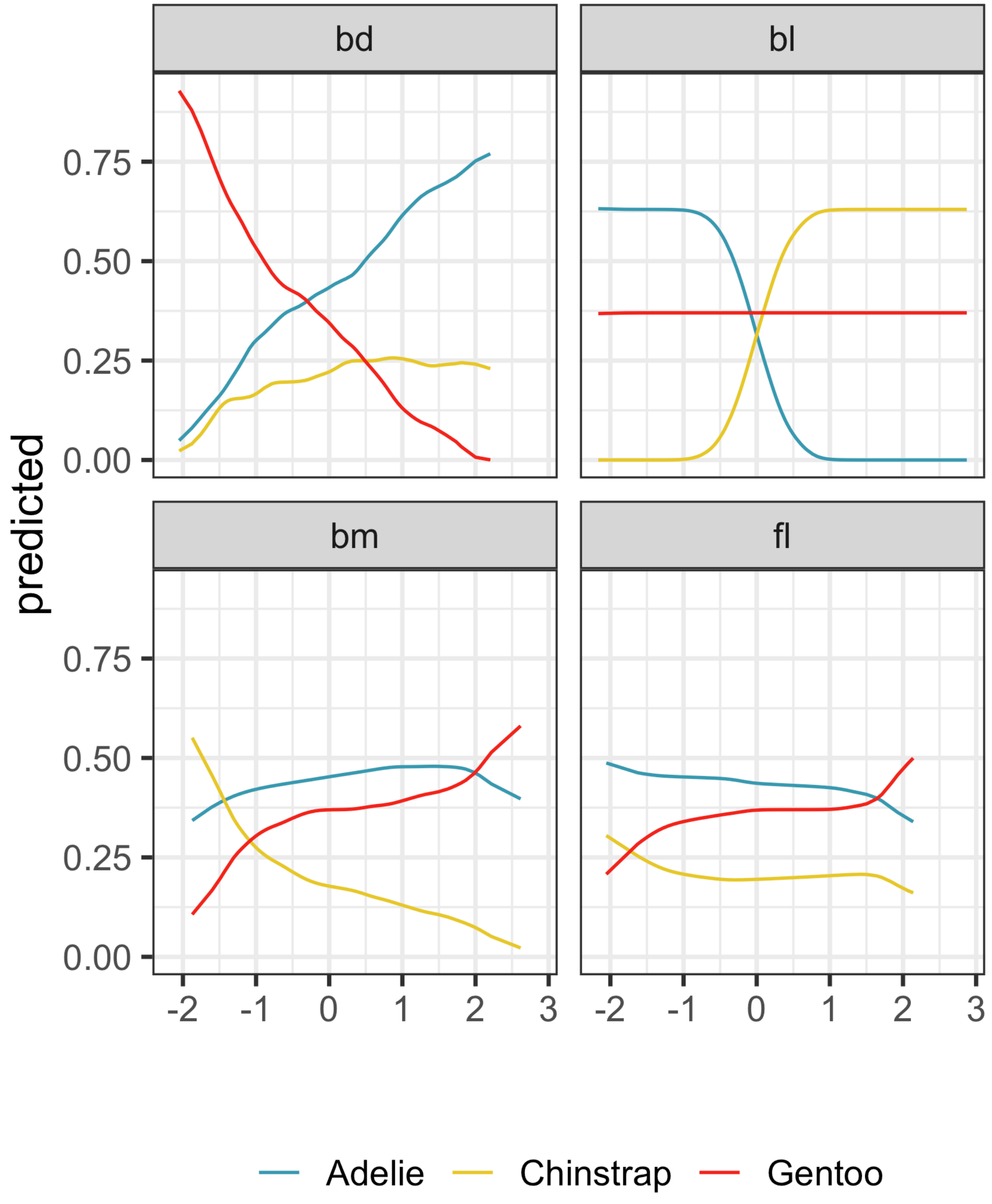
- Variables with correlation still can affect results.
- Variables can mask the importance of others.

Partial dependence profiles (1/2)

Partial dependence profiles show how the model prediction changes across different values of an explanatory variable.

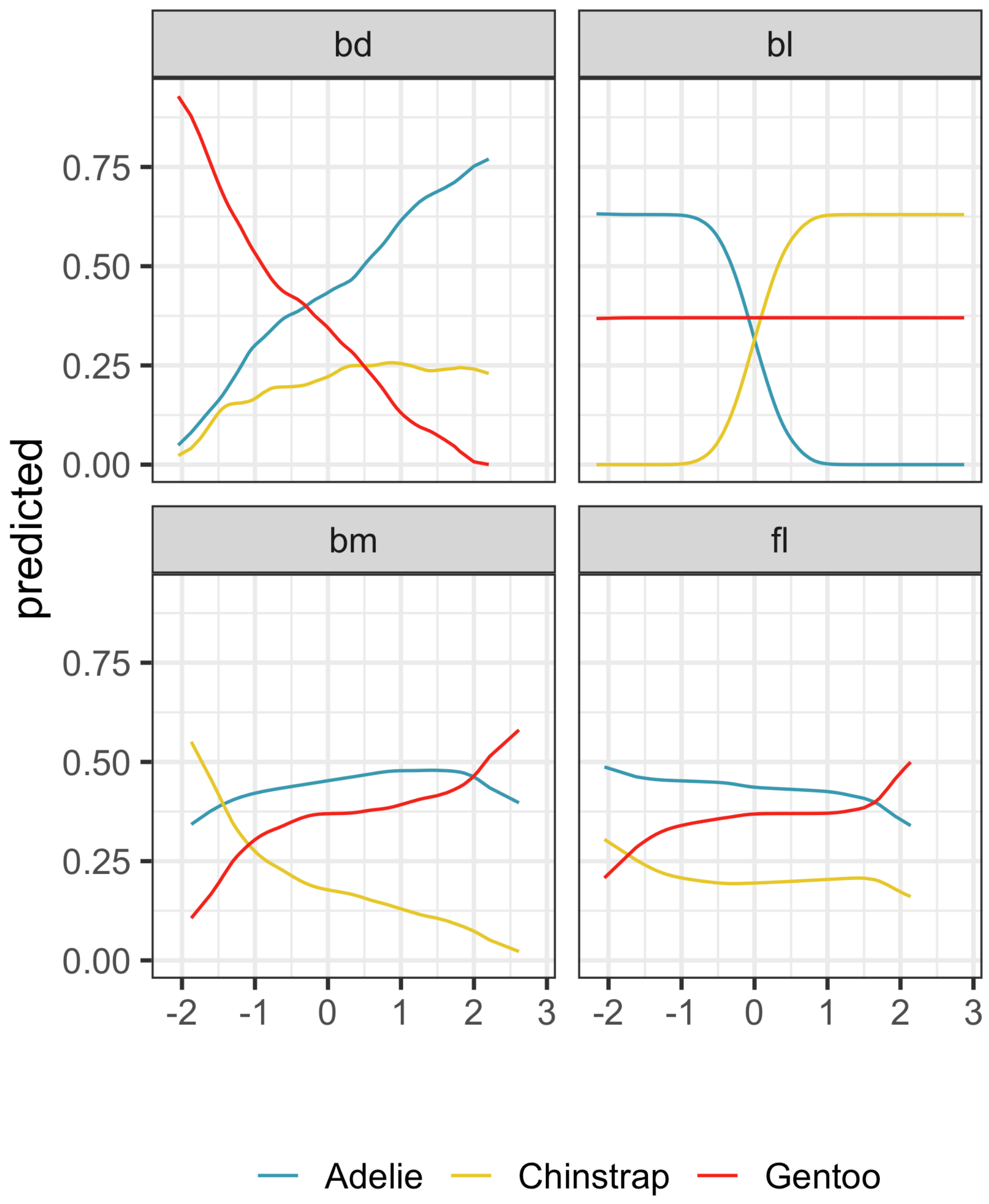
```r
# With DALEX
pdp_lda <- model_profile(
             explainer_lda,
             N=100)
```

Shows what the model sees.

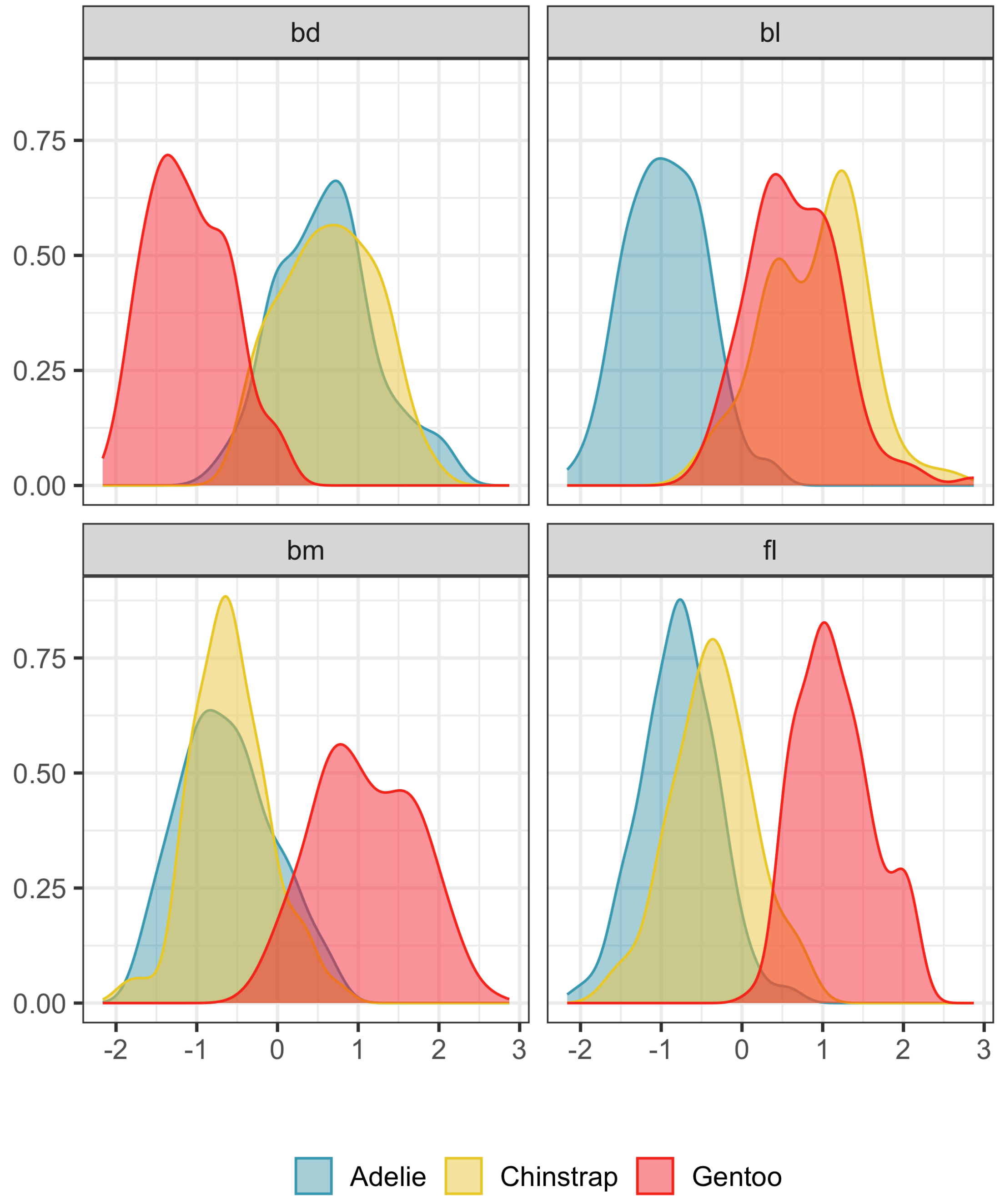


Partial dependence profiles (1/2)

PDP suggests LDA sees

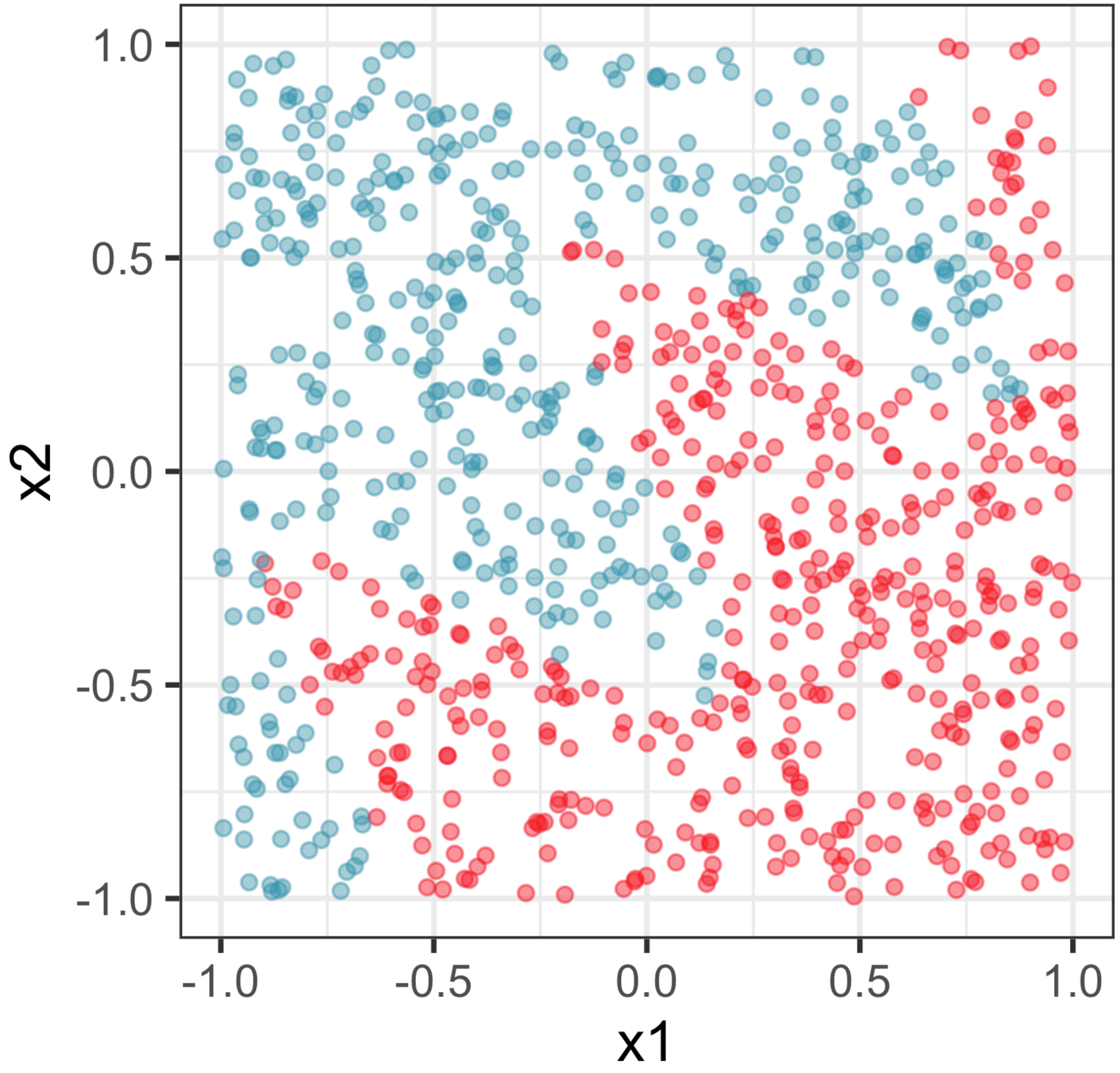


What do we see?



Local explainability

Linear vs non-linear separation

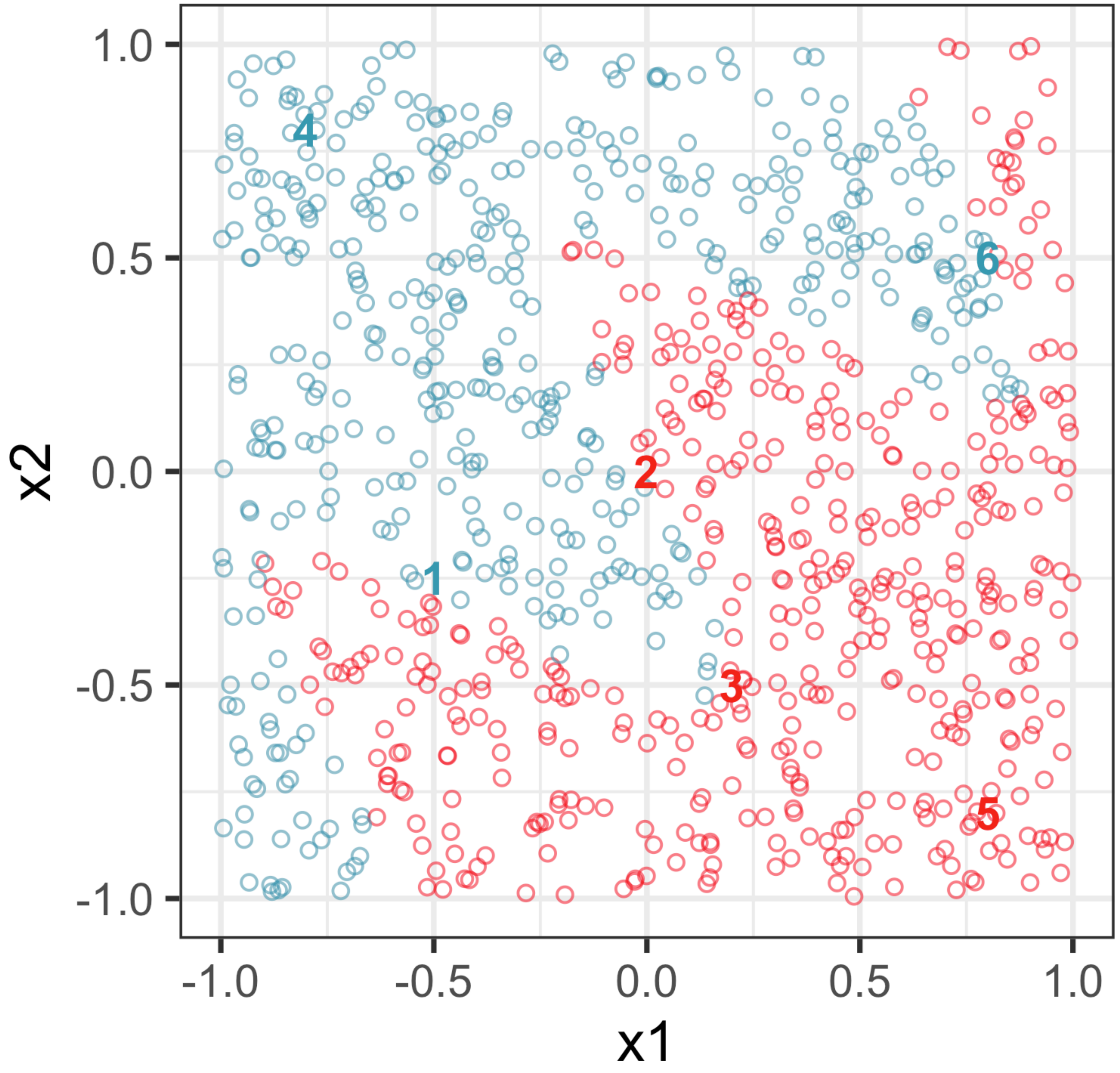


When the difference between classes is non-linear, variable importance changes locally.

Mark a point where `x1` is most important in distinguishing the classes.

Mark a point where `x2` is most important in distinguishing the classes.

Selected points to use for illustration



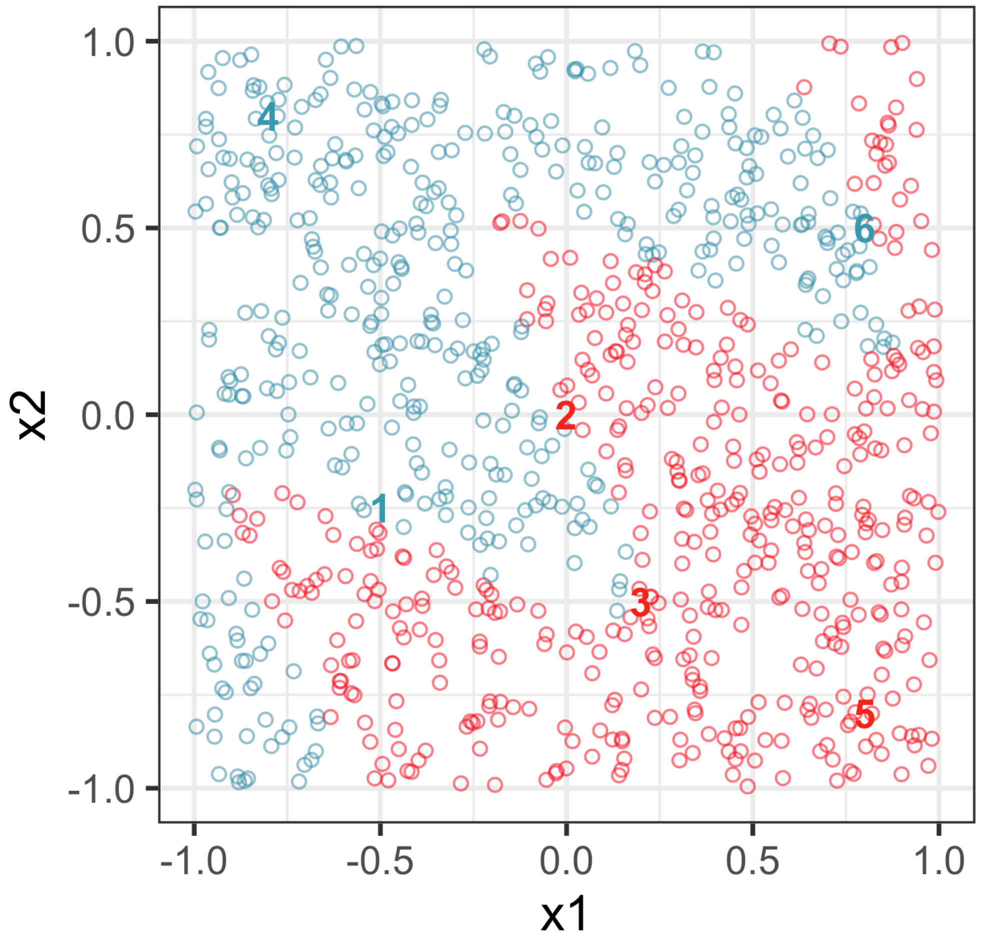
Which variable is most important?

obs	expect
1	x1
2	x2
3	x2 ?
4	x1, x2
5	x1, x2
6	x2

LIME

Fit a linear regression in the local neighbourhood of observation of interest.

```r
library(DALEXtra)
library(lime)
w_rf <- randomForest(cl~., data=w)
w_rf_exp <- DALEX::explain(model = w_rf,
                          data = w[, 1:2],
                          y = w$cl == "A")
model_type.dalex_explainer <-
  DALEXtra::model_type.dalex_explainer
predict_model.dalex_explainer <-
  DALEXtra::predict_model.dalex_explainer
w_lime <- predict_surrogate(
  explainer = w_rf_exp,
              new_observation = w_new,
              n_features = 2,
              n_permutations = 100,
              type = "lime")
```




```
# A tibble: 6 × 4
  case  model_intercept     x1       x2
  <chr>           <dbl>  <dbl>    <dbl>
1 1               0.485 -0.419  0.189
2 2               0.643 -0.184 -0.00800
3 3               0.474  0.141  0.347
4 4               0.695 -0.354 -0.441
5 5               0.466  0.212  0.387
6 6               0.498  0.339 -0.379
```

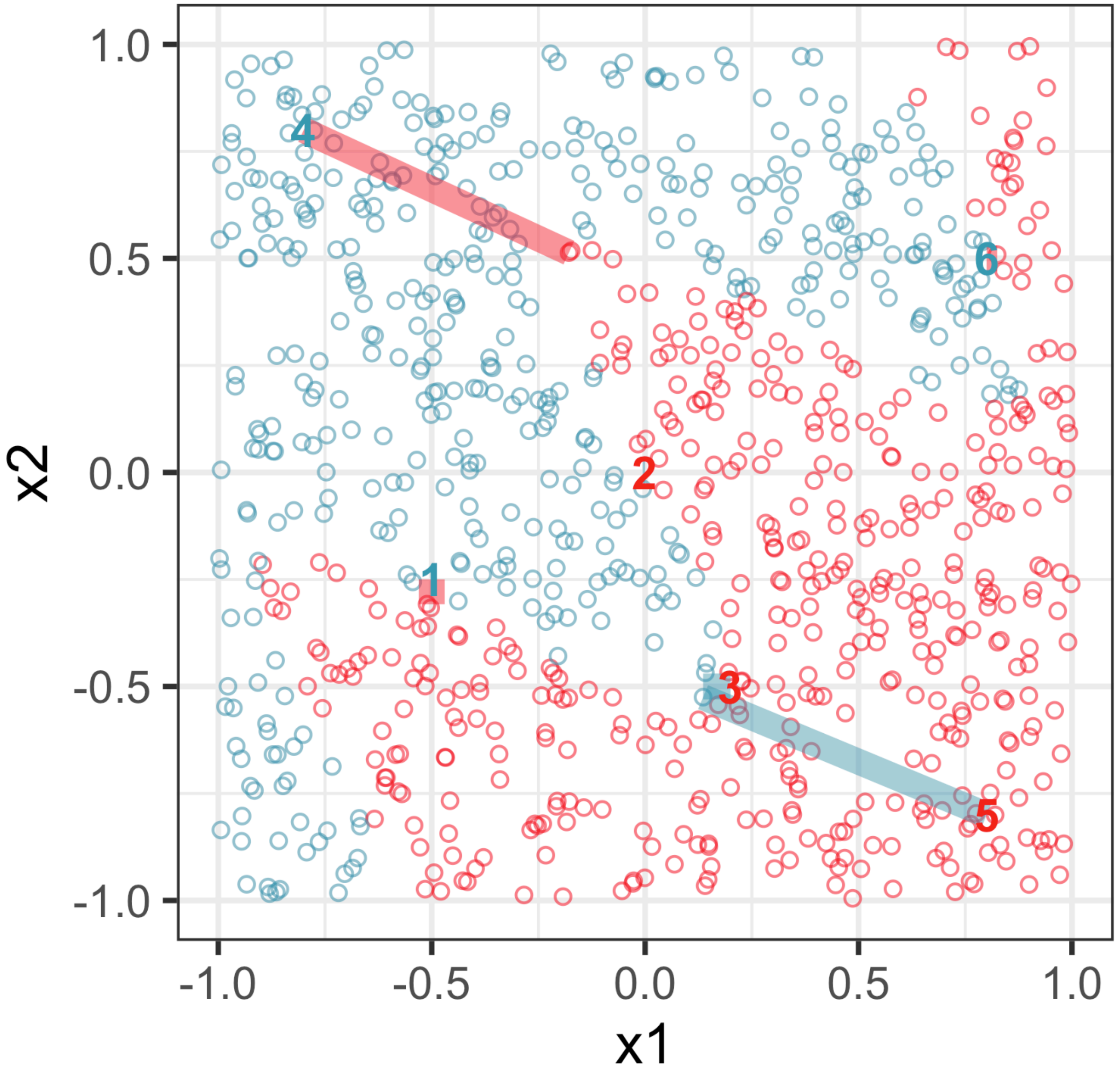
Counterfactuals

Find the closest observation (counterfactual) that has the different class. What values of the variables would you need to change to change the observation of interest into the counterfactual.

```r
library(iml)
# devtools::install_github("dandls/counterfactuals")
library(counterfactuals)
predictor_rf = iml::Predictor$new(w_rf,
                                  type = "prob")
# predictor_rf$predict(w_new[1,])
w_classif <- counterfactuals::NICEClassif$new(
  predictor_rf)

w_new_cf <- w_new
w_new_cf$cl <- ifelse(w_new[,3]=="A",
                      "B", "A")
for (i in 1:nrow(w_new)) {
  w_cf = w_classif$find_counterfactuals(
    x_interest = w_new[i,],
    desired_class = w_new_cf[i,3],
                desired_prob = c(0.5, 1)
  )
  w_new_cf[i, 1] <- w_cf$data$x1
  w_new_cf[i, 2] <- w_cf$data$x2
}
```

	x1o	x2o	clo	x1	x2	cl
1	-0.5	-0.25	A	-0.5000	-0.31	B
2	0.0	0.00	B	-0.0057	0.00	A
3	0.2	-0.50	B	0.1358	-0.50	A
4	-0.8	0.80	A	-0.1785	0.51	B
5	0.8	-0.80	B	0.1358	-0.52	A
6	0.8	0.50	A	0.8249	0.50	B

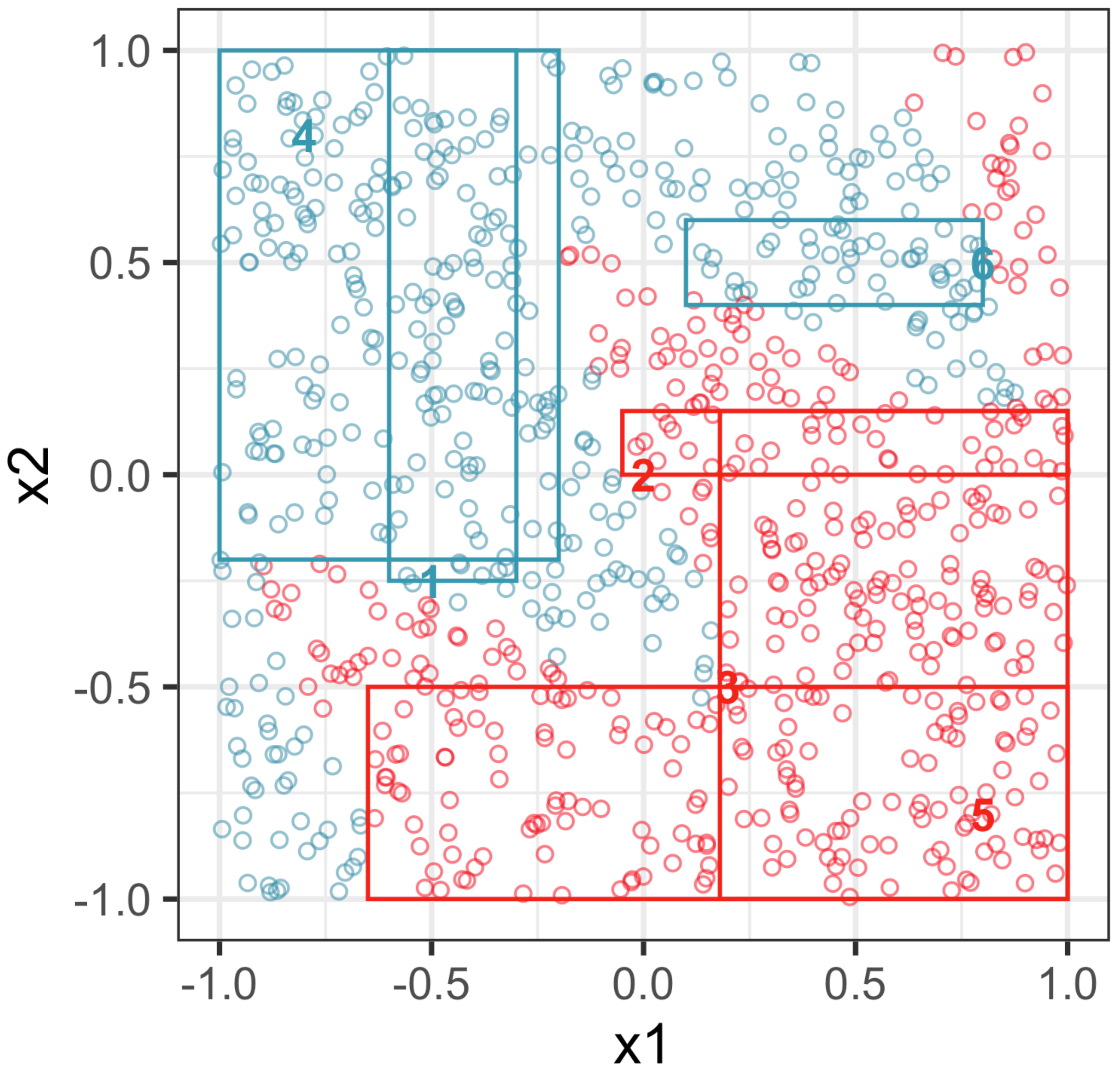


Note: If case is misclassified, the desired class needs to be the true class.

Anchors

How far can you extend from the value of the observation in each direction and still have all observations be the same class.

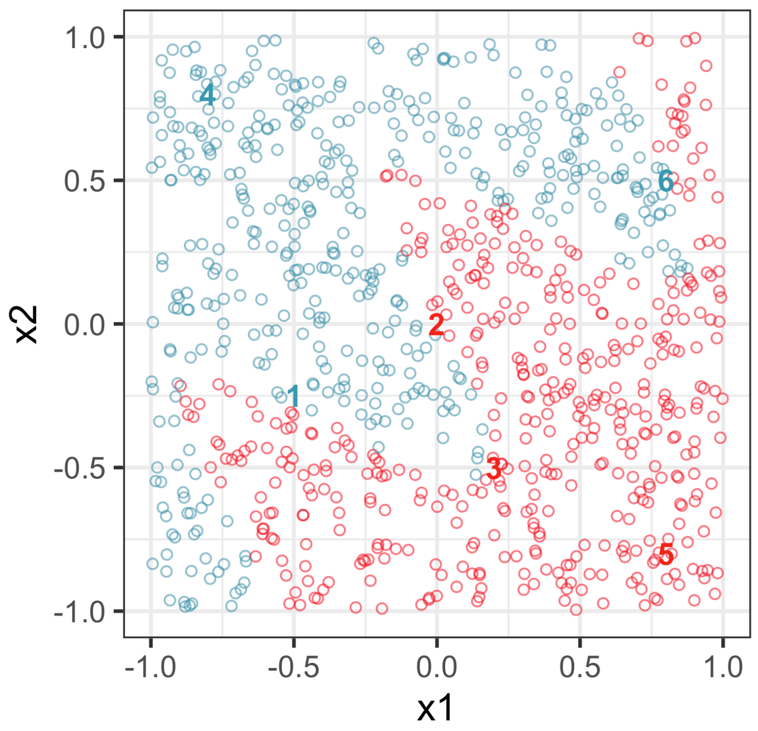
Note: No working R package to calculate these.



Shapley values

A Shapley value is computed from the change in prediction when all combinations of presence or absence of other variables. In the computation, for each combination, the prediction is computed by substituting absent variables with their average value.

```
library(kernelshap)
library(shapviz)
w_explain <- kernelshap(
    w_rf,
    w_new[,1:2],
    w[,1:2],
    verbose = FALSE
  )
```



	x1	x2	cl	shapAx1	shapAx2
1	-0.5	-0.25	A	0.358	0.15
2	0.0	0.00	B	-0.236	-0.25
3	0.2	-0.50	B	-0.164	-0.32
4	-0.8	0.80	A	0.255	0.26
5	0.8	-0.80	B	-0.215	-0.27
6	0.8	0.50	A	-0.059	0.57

Summary

Which variable is most important?

obs	expect	LIME	CF	SHAP
1	x1	x1	x2	x1
2	x2	x1	x1	x1, x2
3	x2 ?	x2	x1	x2
4	x1, x2	x1, x2	x1, x2	x1, x2
5	x1, x2	x2	x1, x2	x1, x2
6	x2	x2	x1	x2

They don’t all agree.

You need good visualisation of the model in the data space to fully digest the importance of the variables.

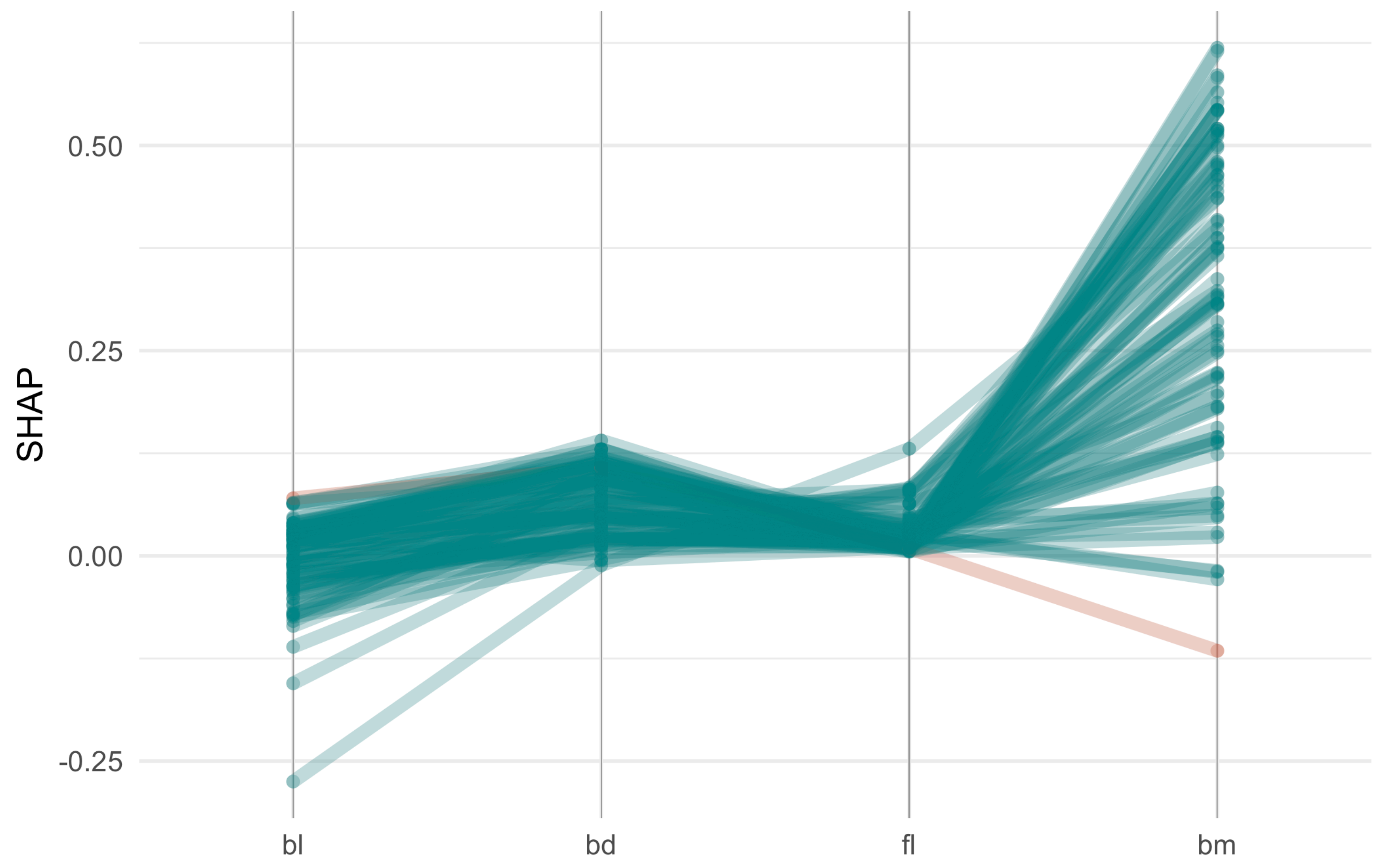
Example: penguins (1/2)

Compute SHAP values for the neural network model

```r
library(keras)
p_nn_model <- load_model_tf("../data/penguins_
p_nn_model

# Explanations
# https://www.r-bloggers.com/2022/08/kernel-sh
library(kernelshap)
library(shapviz)
p_explain <- kernelshap(
    p_nn_model,
    p_train_x,
    bg_X = p_train_x,
    verbose = FALSE
  )
p_exp_sv <- shapviz(p_explain)
save(p_exp_sv, file="../data/p_exp_sv.rda")
```

Highlight SHAP values for a misclassified Gentoo penguin

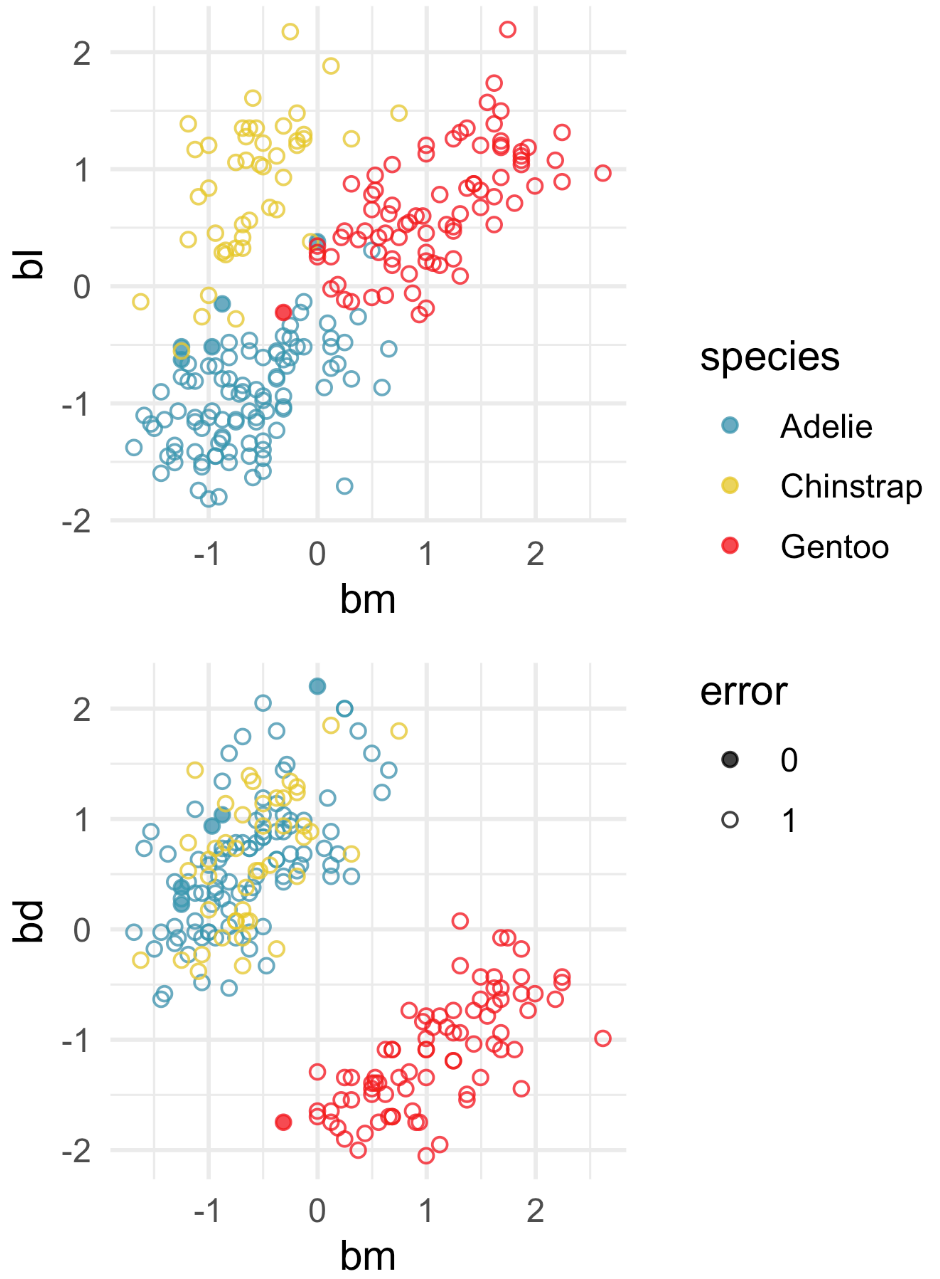


Note: the SHAP value is much lower than values for all other penguins on `bm`.

Example: penguins (2/2)

Weights from hidden layer

```
      [,1]  [,2]
[1,]  0.56  0.80
[2,]  0.17 -0.21
[3,] -0.15 -0.15
[4,] -0.80  0.54
```



Model uses mostly `bl` and `bm`.

Note: this analysis used the training set because this Gentoo penguin was misclassified as an Adelie in the training set.

```
          p_train_pred_cat
           Adelie Chinstrap Gentoo
  Adelie       95         5      0
  Chinstrap     0        45      0
  Gentoo        1         0     81
```

Next: Support vector machines and nearest neighbours