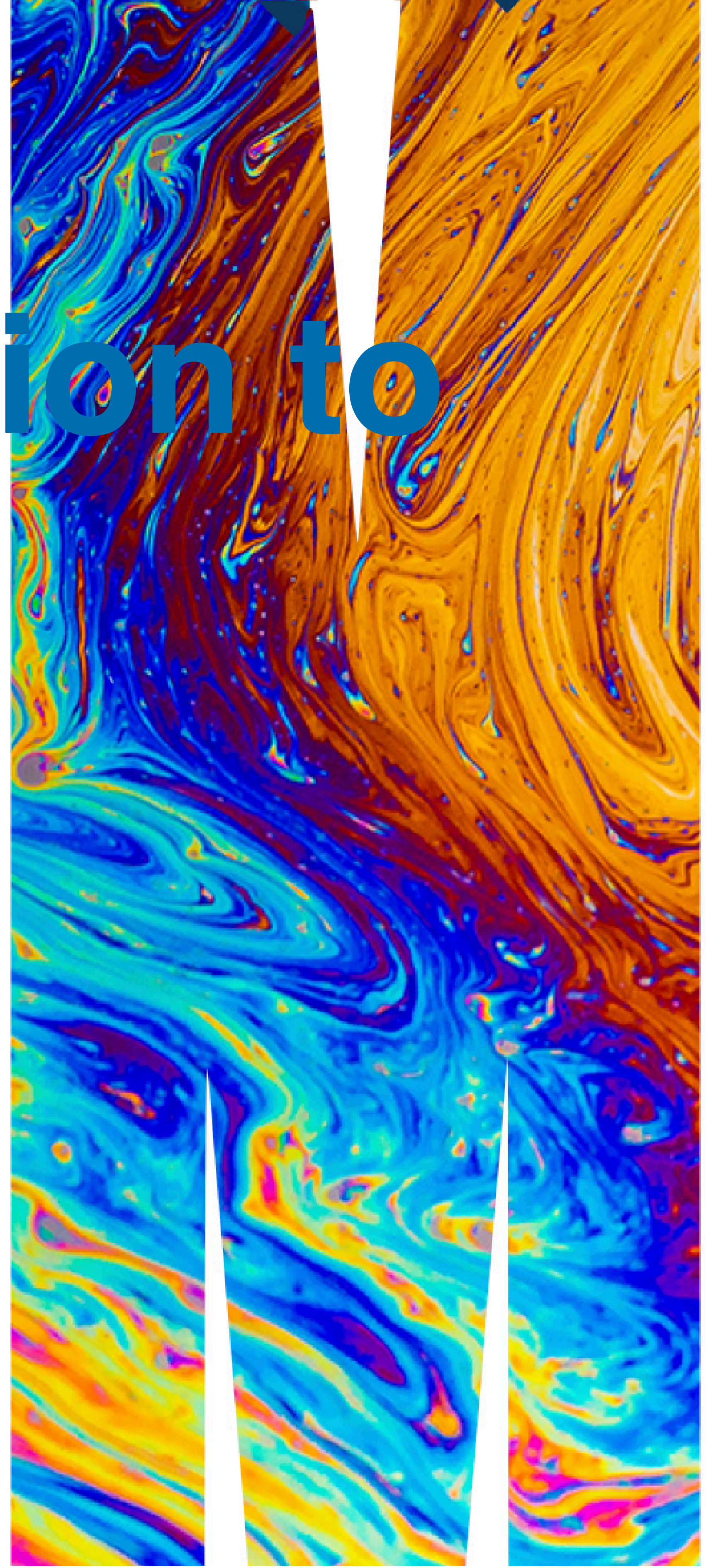
ETC3250/5250 Introduction to Machine Learning

Week 1: Foundations of machine learning

Professor Di Cook

etc3250.clayton-x@monash.edu

Department of Econometrics and Business Statistics



Welcome! Meet the teaching team

Chief examiner: Professor Dianne Cook

Communication: All questions need to be communicated through the Discussion forum. Any of a private matter can be addressed to etc3250.clayton-x@monash.edu or through a private message on the **edstem** forum. **Emails should never be sent directly to tutors or the instructor.**

Tutors:

- Patrick: 3rd year PhD student working on computer vision for reading residual plots
- Harriet: 2nd year PhD student working on visualisation of uncertainty
- Jayani: 2nd year PhD student working on methods for understanding how non-linear dimension reduction warps your data
- Krisanat: MBAt graduate, aspiring to be PhD student in 2025

What this course is about

- select and develop appropriate models for clustering, prediction or classification.
- estimate and simulate from a variety of statistical models.
- measure the uncertainty of a prediction or classification using resampling methods.
- apply business analytic tools to produce innovative solutions in finance, marketing, economics and related areas.
- manage very large data sets in a modern software environment.
- explain and interpret the analyses undertaken clearly and effectively.

Assessment

- Weekly learning quizzes: 3% DUE: Mondays 9am
- Assignment 1: 9%
- Assignment 2: 9%
- Assignment 3: 9%
- Project: 10%
- Final exam: 60%

How to do well

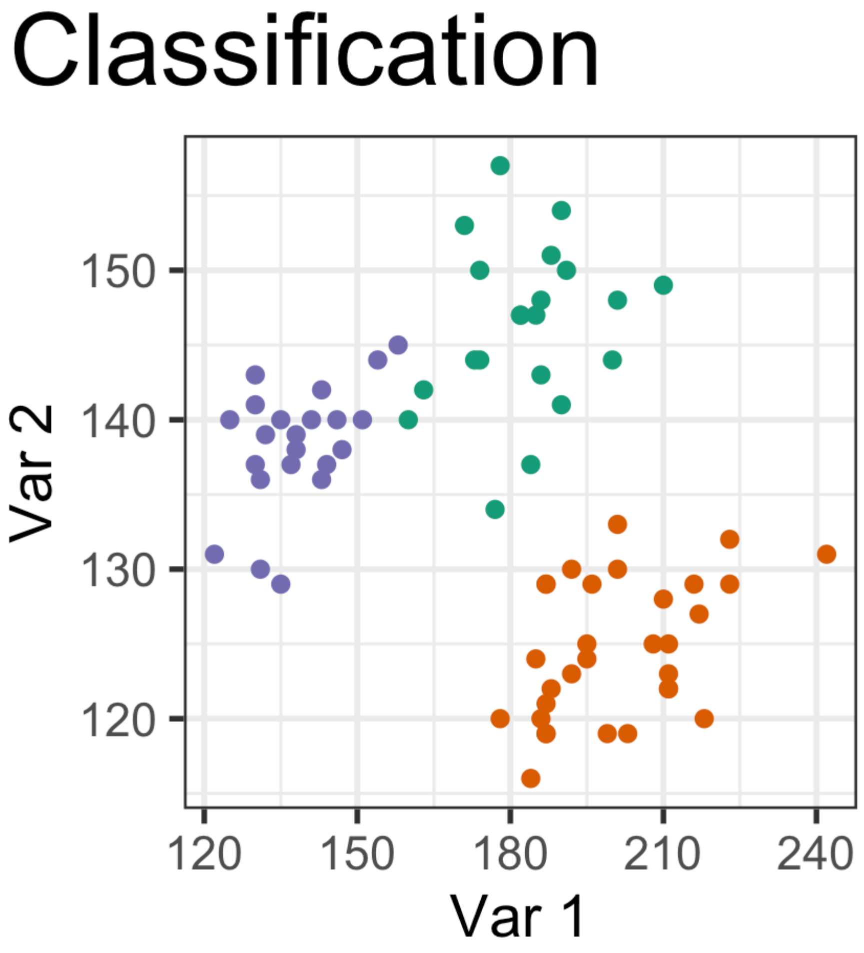
- **Keep up-to-date** with content:
  - **participate** in the lecture each week
  - **attend tutorials**
  - complete **weekly learning quiz** to check your understanding
  - **read** the relevant sections of the resource material
  - **run the code** from lectures in the `qmd` files
- Begin assessments **early**, when posted, map out a plan to complete it on time
- **Ask** questions

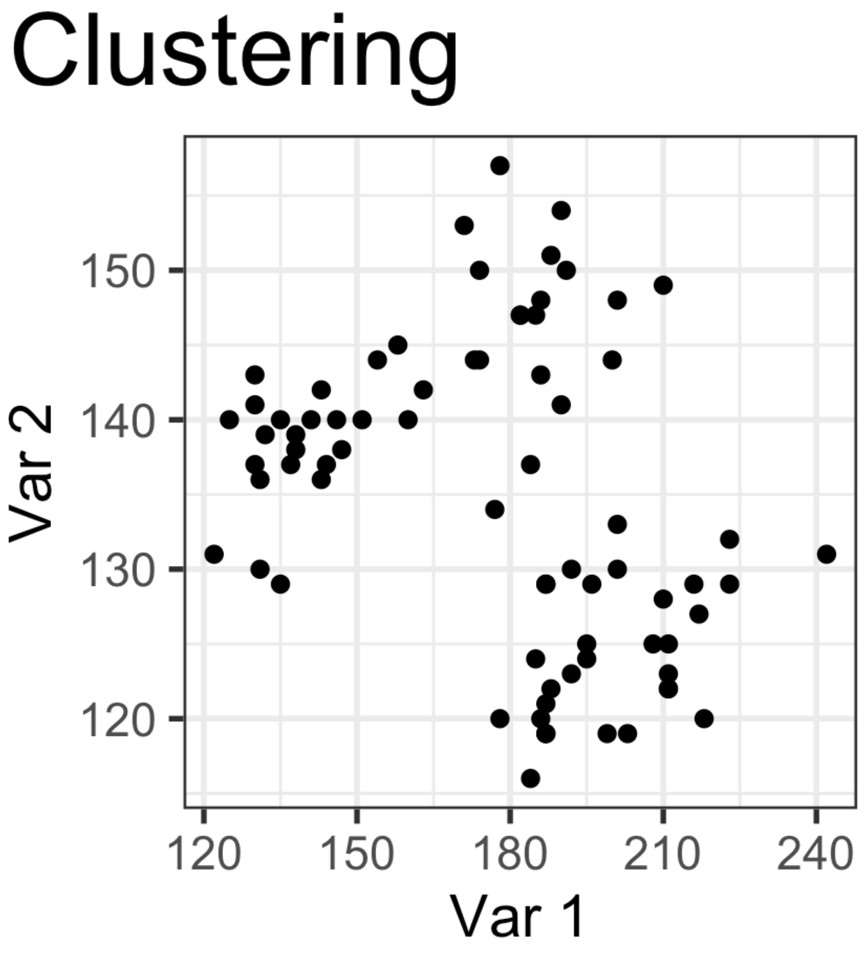
Machine learning is a big, big area. This semester is like the tip of the iceberg, there's a lot more, and interesting methods and problems, than what we can cover. Take this as a challenge to get you started, and become hungry to learn more!

Types of problems

Framing the problem

1. **Supervised classification**: categorical y_i is available for all x_i
2. **Unsupervised learning**: y_i unavailable for all x_i





What type of problem is this? (1/3)

Food servers' tips in restaurants may be influenced by many factors, including the nature of the restaurant, size of the party, and table locations in the restaurant. Restaurant managers need to know which factors matter when they assign tables to food servers. For the sake of staff morale, they usually want to avoid either the substance or the appearance of unfair treatment of the servers, for whom tips (at least in restaurants in the United States) are a major component of pay.

In one restaurant, a food server recorded the following data on all customers they served during an interval of two and a half months in early 1990. The restaurant, located in a suburban shopping mall, was part of a national chain and served a varied menu. In observance of local law the restaurant offered seating in a non-smoking section to patrons who requested it. Each record includes a day and time, and taken together, they show the server's work schedule.

What is y? What is x?

What type of problem is this? (2/3)

Every person monitored their email for a week and recorded information about each email message; for example, whether it was spam, and what day of the week and time of day the email arrived. We want to use this information to build a spam filter, a classifier that will catch spam with high probability but will never classify good email as spam.

What is y? What is x?

What type of problem is this? (3/3)

A health insurance company collected the following information about households:

- Total number of doctor visits per year
- Total household size
- Total number of hospital visits per year
- Average age of household members
- Total number of gym memberships
- Use of physiotherapy and chiropractic services
- Total number of optometrist visits

The health insurance company wants to provide a small range of products, containing different bundles of services and for different levels of cover, to market to customers.

What is y? What is x?

Math and computing

Data: math

n number of observations or sample points

p number of variables or the dimension of the data

A data matrix is denoted as:

$$\mathbf{X}_{n\times p} = (\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_p) = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix}$$

This is also considered the matrix of predictors, or explanatory or independent variables, features, attributes, or input.

Data: computing

```r
library(mvtnorm)
vc <- matrix(c(1, 0.5, 0.2,
               0.5, 1, -0.3,
               0.2, -0.3, 1),
             ncol=3, byrow=TRUE)
set.seed(449)
x <- rmvnorm(5,
             mean = c(-0.2, 0, 0.3),
             sigma = vc)
x
```

```
       [,1]   [,2]   [,3]
[1,] -0.423  1.011 -0.645
[2,] -0.829 -0.988 -0.370
[3,] -0.981  0.660 -0.133
[4,] -0.129 -0.252  0.567
[5,] -0.366  0.554  1.370
```

What’s the dimension of the data?

```r
library(palmerpenguins)
p_tidy <- penguins |>
  select(species, bill_length_mm:body_mass_g) |>
  rename(bl=bill_length_mm,
         bd=bill_depth_mm,
         fl=flipper_length_mm,
         bm=body_mass_g)
p_tidy |> slice_head(n=10)
```

```
# A tibble: 10 × 5
   species    bl    bd    fl    bm
   <fct>   <dbl> <dbl> <int> <int>
 1 Adelie   39.1  18.7   181  3750
 2 Adelie   39.5  17.4   186  3800
 3 Adelie   40.3  18     195  3250
 4 Adelie   NA    NA      NA    NA
 5 Adelie   36.7  19.3   193  3450
 6 Adelie   39.3  20.6   190  3650
 7 Adelie   38.9  17.8   181  3625
 8 Adelie   39.2  19.6   195  4675
 9 Adelie   34.1  18.1   193  3475
10 Adelie   42    20.2   190  4250
```

What’s the dimension of the data?

Observations and variables: math

The i^{th} observation is denoted as

$$x_i = \begin{pmatrix} x_{i1} & x_{i2} & \dots & x_{ip} \end{pmatrix}$$

The j^{th} variable is denoted as

$$x_j = \begin{pmatrix} x_{1j} \\ x_{2j} \\ \vdots \\ x_{nj} \end{pmatrix}$$

Observations and variables: computing

Observations - rows

```r
x[2,]
```

```
[1] -0.829 -0.988 -0.370
```

```r
p_tidy |> slice_sample()
```

```
# A tibble: 1 × 5
  species      bl    bd    fl    bm
  <fct>     <dbl> <dbl> <int> <int>
1 Chinstrap  51.5  18.7   187  3250
```

Variables - columns

```r
x[,1]
```

```
[1] -0.423 -0.829 -0.981 -0.129 -0.366
```

```r
p_tidy |> pull(fl)
```

```
  [1] 181 186 195  NA 193 190 181 195 193 190 186 180 182
 [14] 191 198 185 195 197 184 194 174 180 189 185 180 187
 [27] 183 187 172 180 178 178 188 184 195 196 190 180 181
 [40] 184 182 195 186 196 185 190 182 179 190 191 186 188
 [53] 190 200 187 191 186 193 181 194 185 195 185 192 184
 [66] 192 195 188 190 198 190 190 196 197 190 195 191 184
 [79] 187 195 189 196 187 193 191 194 190 189 189 190 202
 [92] 205 185 186 187 208 190 196 178 192 192 203 183 190
[105] 193 184 199 190 181 197 198 191 193 197 191 196 188
[118] 199 189 189 187 198 176 202 186 199 191 195 191 210
[131] 190 197 193 199 187 190 191 200 185 193 193 187 188
[144] 190 192 185 190 184 195 193 187 201 211 230 210 218
[157] 215 210 211 219 209 215 214 216 214 213 210 217 210
[170] 221 209 222 218 215 213 215 215 215 216 215 210 220
[183] 222 209 207 230 220 220 213 219 208 208 208 225 210
```

Response: math

The response variable (or target variable, output, outcome measurement), when it exists, is denoted as:

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

An observation can also be written as

$$\mathcal{D} = \{(y_i, x_i)\}_{i=1}^n = \{(y_1, x_1), (y_2, x_2), \dots, (y_n, x_n)\}$$

where x_i is a vector with p elements.

Response: computing

`species` is the response variable, and it is categorical.

```r
set.seed(424)
p_tidy |> slice_sample(n=10)
```

```
# A tibble: 10 × 5
   species      bl    bd    fl    bm
   <fct>     <dbl> <dbl> <int> <int>
 1 Gentoo     47.4  14.6   212  4725
 2 Gentoo     46.2  14.5   209  4800
 3 Chinstrap  45.6  19.4   194  3525
 4 Adelie     38.8  17.6   191  3275
 5 Gentoo     50    15.3   220  5550
 6 Chinstrap  46    18.9   195  4150
 7 Adelie     38.9  18.8   190  3600
 8 Gentoo     46.5  14.5   213  4400
 9 Gentoo     46.8  14.3   215  4850
10 Gentoo     45.7  13.9   214  4400
```

A binary matrix format is sometimes useful.

```r
set.seed(424)
model.matrix(~ 0 + species, data = p_tidy) |>
  as_tibble() |>
  slice_sample(n=10)
```

```
# A tibble: 10 × 3
   speciesAdelie speciesChinstrap speciesGentoo
           <dbl>            <dbl>         <dbl>
 1             0                0             1
 2             0                0             1
 3             0                1             0
 4             1                0             0
 5             0                0             1
 6             0                1             0
 7             1                0             0
 8             0                0             1
 9             0                0             1
10             0                0             1
```

Linear algebra

A transposed data matrix is denoted as

$$\mathbf{X}^\top = \begin{pmatrix} x_{11} & x_{21} & \dots & x_{n1} \\ x_{12} & x_{22} & \dots & x_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1p} & x_{2p} & \dots & x_{np} \end{pmatrix}_{p\times n}$$

```r
x
```

```
       [,1]   [,2]   [,3]
[1,] -0.423  1.011 -0.645
[2,] -0.829 -0.988 -0.370
[3,] -0.981  0.660 -0.133
[4,] -0.129 -0.252  0.567
[5,] -0.366  0.554  1.370
```

```r
t(x)
```

```
       [,1]   [,2]   [,3]   [,4]   [,5]
[1,] -0.423 -0.829 -0.981 -0.129 -0.366
[2,]  1.011 -0.988  0.660 -0.252  0.554
[3,] -0.645 -0.370 -0.133  0.567  1.370
```

Matrix multiplication: math

$$\mathbf{A}_{2\times3} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix}$$

$$\mathbf{B}_{3\times4} = \begin{pmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \end{pmatrix}$$

then

$$\mathbf{AB}_{2\times4} = \begin{pmatrix} \sum_{j=1}^3 a_{1j}b_{j1} & \sum_{j=1}^3 a_{1j}b_{j2} & \sum_{j=1}^3 a_{1j}b_{j3} & \sum_{j=1}^3 a_{1j}b_{j4} \\ \sum_{j=1}^3 a_{2j}b_{j1} & \sum_{j=1}^3 a_{2j}b_{j2} & \sum_{j=1}^3 a_{2j}b_{j3} & \sum_{j=1}^3 a_{2j}b_{j4} \end{pmatrix}$$

Pour the rows into the columns. Note: You can't do $\mathbf{BA}$!

Matrix multiplication: computing

```r
x
```

```
       [,1]   [,2]   [,3]
[1,] -0.423  1.011 -0.645
[2,] -0.829 -0.988 -0.370
[3,] -0.981  0.660 -0.133
[4,] -0.129 -0.252  0.567
[5,] -0.366  0.554  1.370
```

```r
proj <- matrix(c(1/sqrt(2), 1/sqrt(2), 0,
                 0, 0, 1), ncol=2, byrow=FALSE)
proj
```

```
      [,1] [,2]
[1,] 0.707    0
[2,] 0.707    0
[3,] 0.000    1
```

```r
x %*% proj
```

```
       [,1]   [,2]
[1,]  0.416 -0.645
[2,] -1.285 -0.370
[3,] -0.227 -0.133
[4,] -0.269  0.567
[5,]  0.133  1.370
```

Try this:

```r
t(x) %*% proj
```

It produces an error because it can't be done

```
Error in t(x) %*% proj : non-conformable arguments
```

Notice: %*% uses a $*$ so it is NOT the tidyverse pipe.

Identity matrix

$$I = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & 0 & & 1 \end{pmatrix}_{p \times p}$$

```r
diag(1, 8, 8)
```

```
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
[1,]    1    0    0    0    0    0    0    0
[2,]    0    1    0    0    0    0    0    0
[3,]    0    0    1    0    0    0    0    0
[4,]    0    0    0    1    0    0    0    0
[5,]    0    0    0    0    1    0    0    0
[6,]    0    0    0    0    0    1    0    0
[7,]    0    0    0    0    0    0    1    0
[8,]    0    0    0    0    0    0    0    1
```

Inverting a matrix: math

Suppose that $\mathbf{A}$ is square

$$\mathbf{A}_{2\times 2} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

then the inverse is (if $ad - bc \neq 0$)

$$\mathbf{A}^{-1}_{2\times 2} = \frac{1}{ad - bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

and $\mathbf{A}\mathbf{A}^{-1} = I$ where

$$\mathbf{I}_{2\times 2} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

If $AB = I$, then $B = A^{-1}$.

```r
vc
```

```
     [,1] [,2] [,3]
[1,]  1.0  0.5  0.2
[2,]  0.5  1.0 -0.3
[3,]  0.2 -0.3  1.0
```

```r
vc_i <- solve(vc)
vc_i
```

```
       [,1]   [,2]   [,3]
[1,]  1.625 -1.000 -0.625
[2,] -1.000  1.714  0.714
[3,] -0.625  0.714  1.339
```

```r
vc %*% vc_i
```

```
          [,1]     [,2]      [,3]
[1,]  1.00e+00 7.45e-17 -7.34e-18
[2,] -1.96e-16 1.00e+00  4.82e-17
[3,]  0.00e+00 0.00e+00  1.00e+00
```

Projections

$d(\leq p)$ is used to denote the number of variables in a lower dimensional space, usually by taking a projection.

A is a $p \times d$ orthonormal basis, $A^\top A = I_d$ ($A'A = I_d$).

The projection of $\mathbf{x}_i$ onto A is $A^\top \mathbf{x}_i$.

```r
proj
```

```
      [,1] [,2]
[1,] 0.707    0
[2,] 0.707    0
[3,] 0.000    1
```

```r
sum(proj[,1]^2)
```

```
[1] 1
```

```r
sum(proj[,2]^2)
```

```
[1] 1
```

```r
sum(proj[,1]*proj[,2])
```

```
[1] 0
```

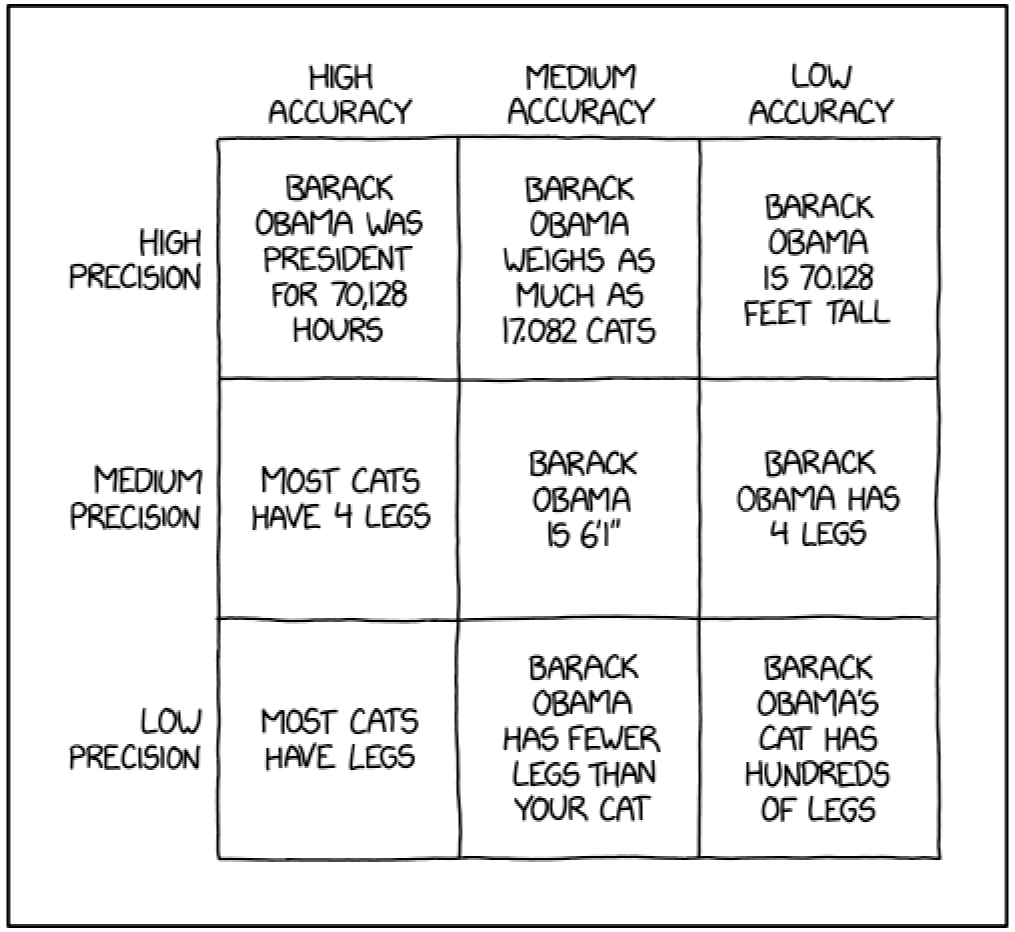
`proj` would be considered to be a orthonormal projection matrix.

Conceptual framework

Accuracy vs interpretability

Predictive accuracy

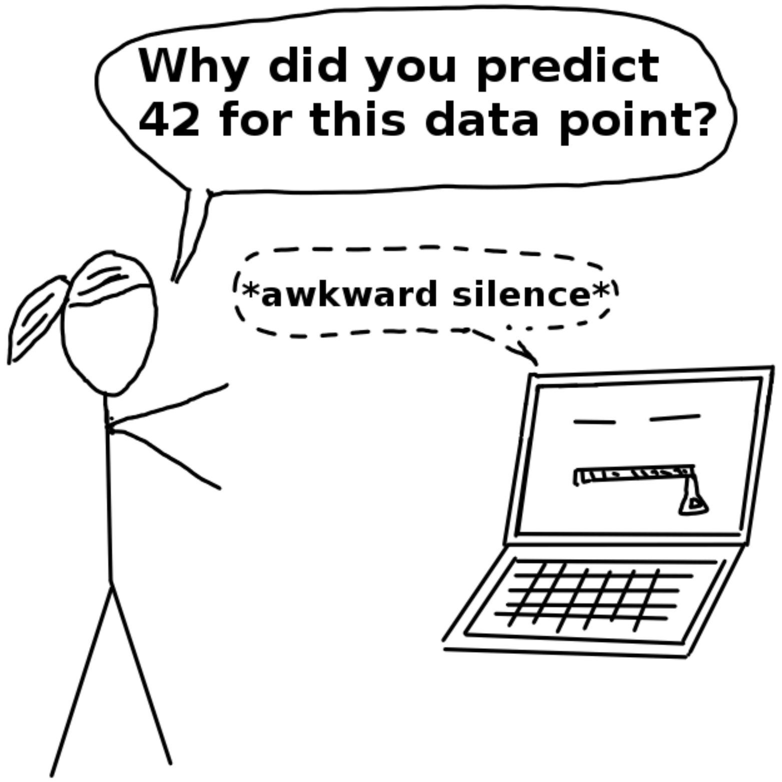
The primary purpose is to be able to predict $\widehat{Y}$ for new data. And we'd like to do that well! That is, accurately.



From XKCD

Interpretability

Almost equally important is that we want to understand the relationship between $\mathbf{X}$ and Y. The simpler model that is (almost) as accurate is the one we choose, always.



From Interpretable Machine Learning

Training vs test splits

> When data are reused for multiple tasks, instead of carefully *spent* from the finite data budget, certain risks increase, such as the risk of accentuating bias or compounding effects from methodological errors. *Julia Silge*

- Training set: Used to fit the model, might be also broken into a validation set for frequent assessment of fit.
- Test set: Purely used to assess final models performance on future data.

Training vs test splits

```r
d_bal <- tibble(y=c(rep("A", 6), rep("B", 6)),
                x=c(runif(12)))
d_bal$y
```

```
[1] "A" "A" "A" "A" "A" "A" "B" "B" "B" "B" "B" "B"
```

```r
set.seed(130)
d_bal_split <- initial_split(d_bal, prop = 0.7
training(d_bal_split)$y
```

```
[1] "A" "A" "B" "A" "B" "A" "B" "A"
```

```r
testing(d_bal_split)$y
```

```
[1] "A" "B" "B" "B"
```

```r
d_unb <- tibble(y=c(rep("A", 2), rep("B", 10))
                x=c(runif(12)))
d_unb$y
```

```
[1] "A" "A" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B"
```

```r
set.seed(132)
d_unb_split <- initial_split(d_unb, prop = 0.7
training(d_unb_split)$y
```

```
[1] "B" "B" "A" "B" "B" "A" "B" "B"
```

```r
testing(d_unb_split)$y
```

```
[1] "B" "B" "B" "B"
```

Always stratify splitting by sub-groups, especially response variable classes.

```r
d_unb_strata <- initial_split(d_unb, prop = 0.
training(d_unb_strata)$y
```

```
[1] "A" "B" "B" "B" "B" "B" "B" "B"
```

```r
testing(d_unb_strata)$y
```

```
[1] "A" "B" "B" "B"
```

Measuring accuracy for categorical response

Compute $\hat{y}$ from training data, $\{(y_i, \mathbf{x}_i)\}_{i=1}^n$. The error rate (fraction of misclassifications) to get the Training Error Rate

$$\text{Error rate} = \frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}(\mathbf{x}_i))$$

A better estimate of future accuracy is obtained using test data to get the Test Error Rate.

> Training error will usually be smaller than test error. When it is much smaller, it indicates that the model is too well fitted to the training data to be accurate on future data (over-fitted).

Confusion (misclassification) matrix

		predicted	
		1	0
true	1	*a*	*b*
	0	*c*	*d*

Consider 1=positive (P), 0=negative (N).

- True positive (TP): a
- True negative (TN): d
- False positive (FP): c (Type I error)
- False negative (FN): b (Type II error)

- Sensitivity, recall, hit rate, or true positive rate (TPR): TP/P = a/(a+b)
- Specificity, selectivity or true negative rate (TNR): TN/N = d/(c+d)
- Prevalence: P/(P+N) = (a+b)/(a+b+c+d)
- Accuracy: (TP+TN)/(P+N) = (a+d)/(a+b+c+d)
- Balanced accuracy: (TPR + TNR)/2 (or average class errors)

Confusion (misclassification) matrix: computing

Two classes

```
# Write out the confusion matrix in standard f
#| label: oconfusion-matrix-tidy
cm <- a2 %>% count(y, pred) |>
  group_by(y) |>
  mutate(cl_err = n[pred==y]/sum(n))
cm |>
  pivot_wider(names_from = pred,
              values_from = n) |>
  select(y, bilby, quokka, cl_err)
```

```
# A tibble: 2 × 4
# Groups:   y [2]
  y      bilby quokka cl_err
  <fct>  <int>  <int>  <dbl>
1 bilby      9      3  0.75
2 quokka     5     10  0.667
```

```
accuracy(a2, y, pred) |> pull(.estimate)
```

```
[1] 0.704
```

```
bal_accuracy(a2, y, pred) |> pull(.estimate)
```

```
[1] 0.708
```

```
sens(a2, y, pred) |> pull(.estimate)
```

```
[1] 0.75
```

```
specificity(a2, y, pred) |> pull(.estimate)
```

```
[1] 0.667
```

More than two classes

```
# Write out the confusion matrix in standard f
cm3 <- a3 %>% count(y, pred) |>
  group_by(y) |>
  mutate(cl_err = n[pred==y]/sum(n))
cm3 |>
  pivot_wider(names_from = pred,
              values_from = n, values_fill=0)
  select(y, bilby, quokka, numbat, cl_err)
```

```
# A tibble: 3 × 5
# Groups:   y [3]
  y      bilby quokka numbat cl_err
  <fct>  <int>  <int>  <int>  <dbl>
1 bilby      9      3      0  0.75
2 numbat     0      2      6  0.75
3 quokka     5     10      0  0.667
```

```
accuracy(a3, y, pred) |> pull(.estimate)
```

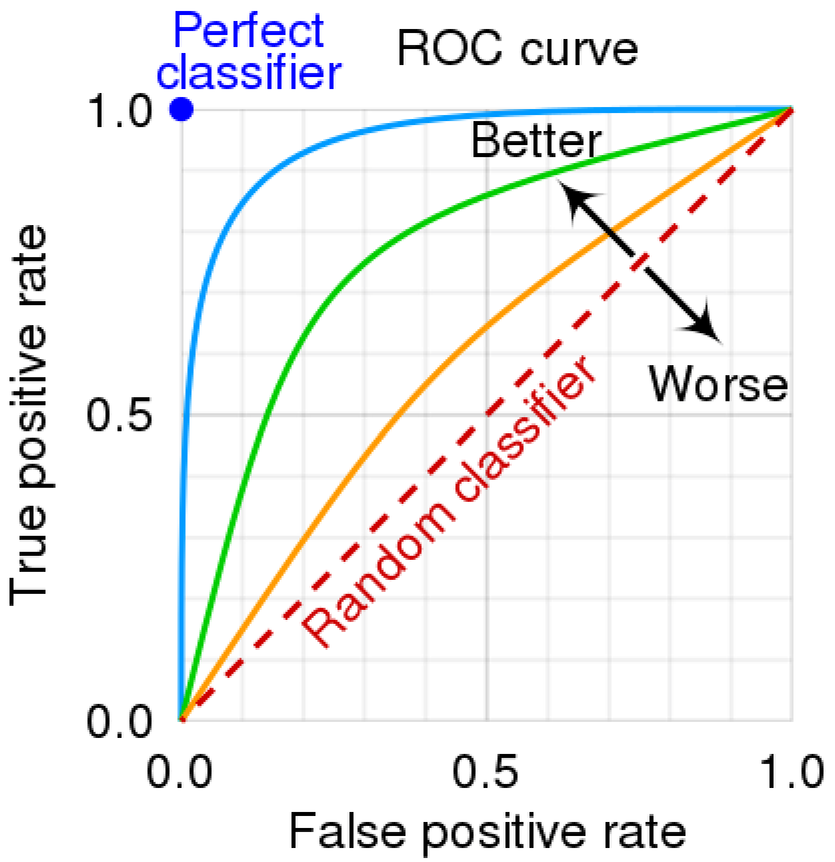
```
[1] 0.714
```

```
bal_accuracy(a3, y, pred) |> pull(.estimate)
```

```
[1] 0.783
```

Receiver Operator Curves (ROC)

The balance of getting it right, without predicting everything as positive.



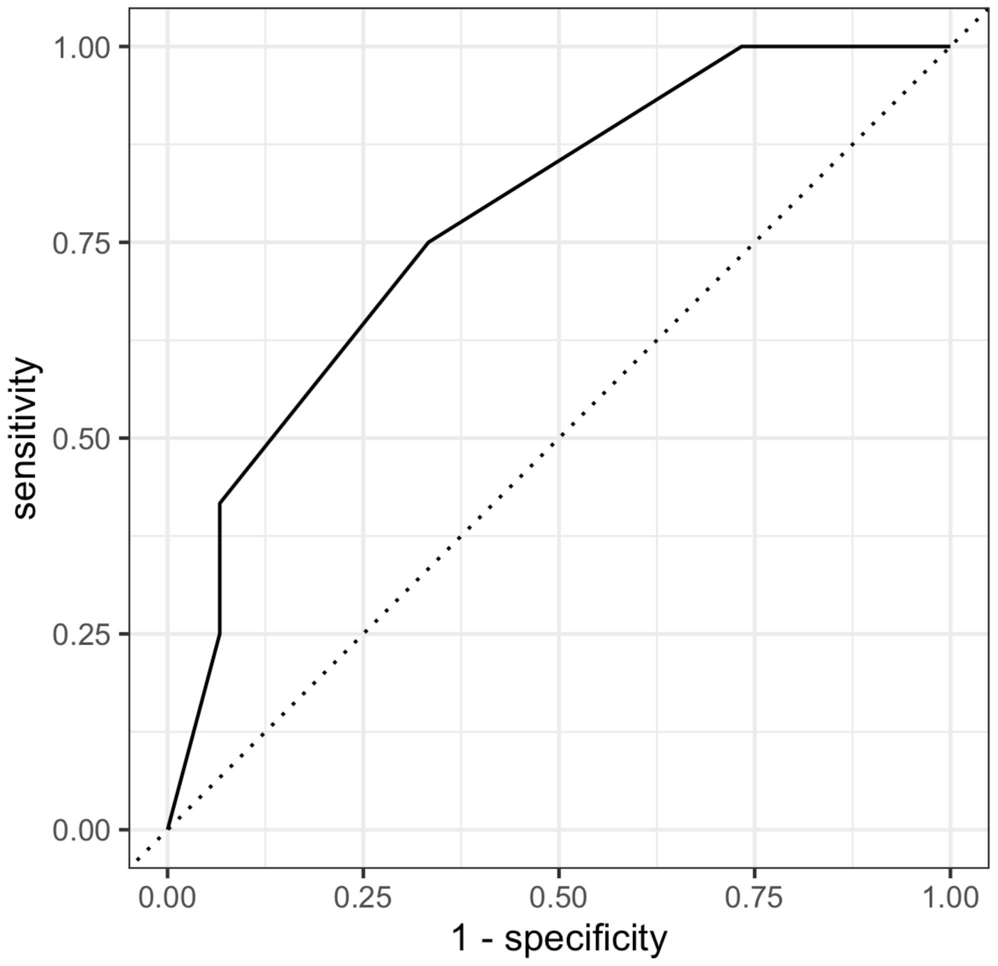
From wikipedia

Need predictive probabilities, probability of being each class.

```r
a2 |> slice_head(n=3)
```

```
# A tibble: 3 × 4
  y     pred  bilby quokka
  <fct> <fct> <dbl>  <dbl>
1 bilby bilby   0.9    0.1
2 bilby bilby   0.8    0.2
3 bilby bilby   0.9    0.1
```

```r
roc_curve(a2, y, bilby) |>
  autoplot()
```



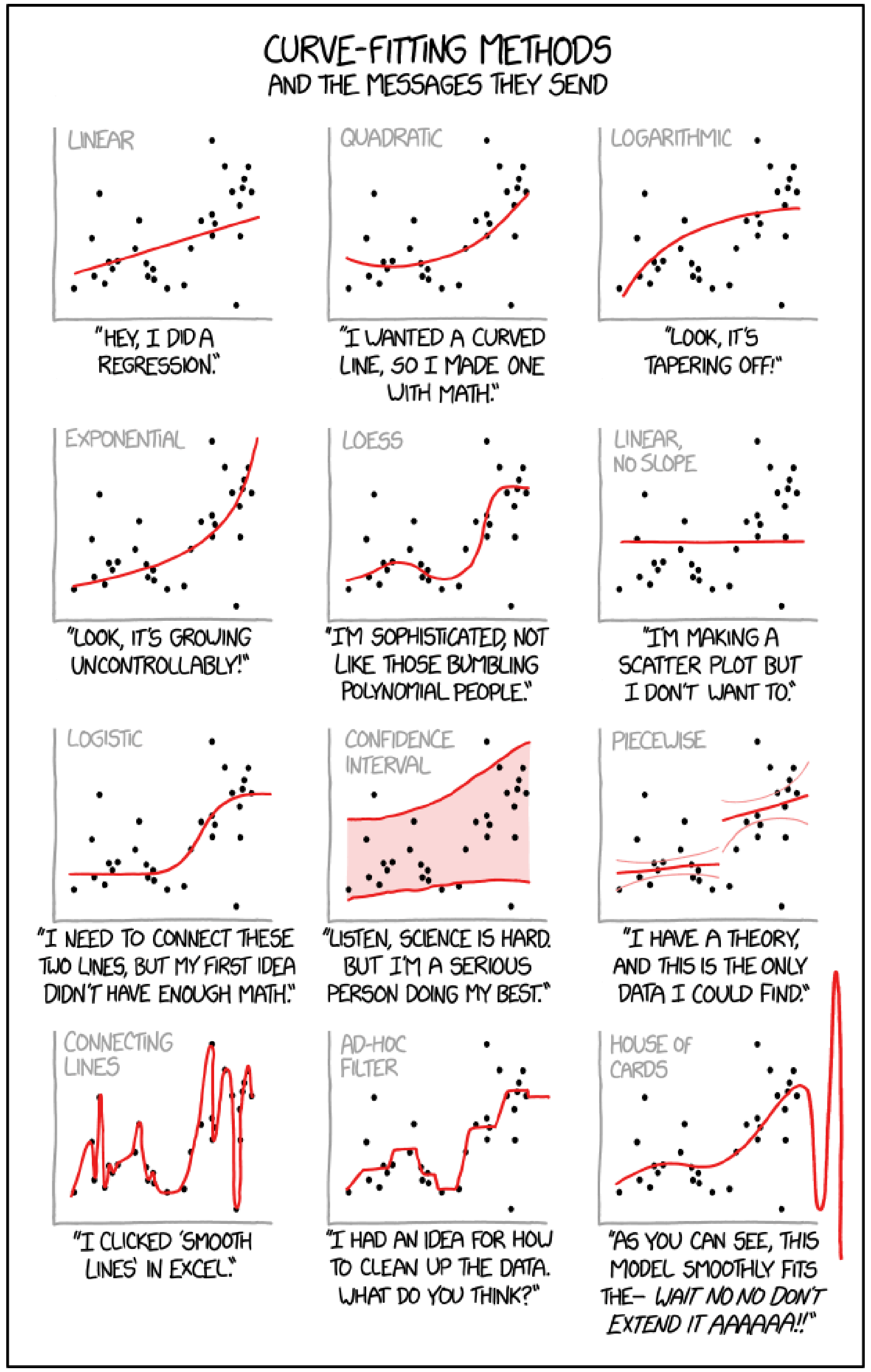
Parametric vs non-parametric

Parametric methods

- Assume that the model takes a specific form
- Fitting then is a matter of estimating the parameters of the model
- Generally considered to be less flexible
- If assumptions are wrong, performance likely to be poor

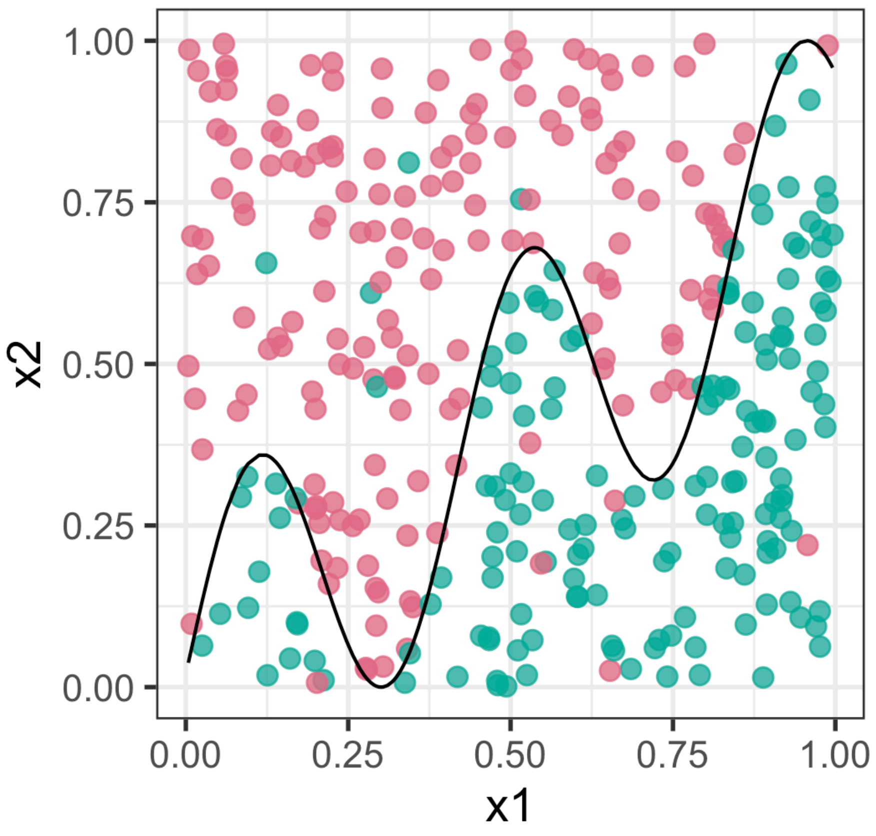
Non-parametric methods

- No specific assumptions
- Allow the data to specify the model form, without being too rough or wiggly
- More flexible
- Generally needs more observations, and not too many variables
- Easier to over-fit



From XKCD

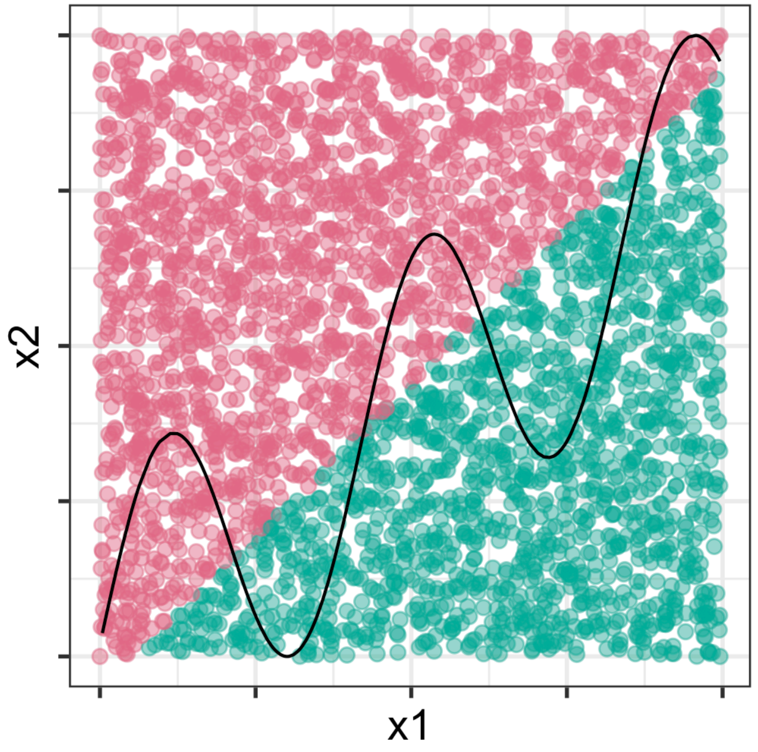
Observed data



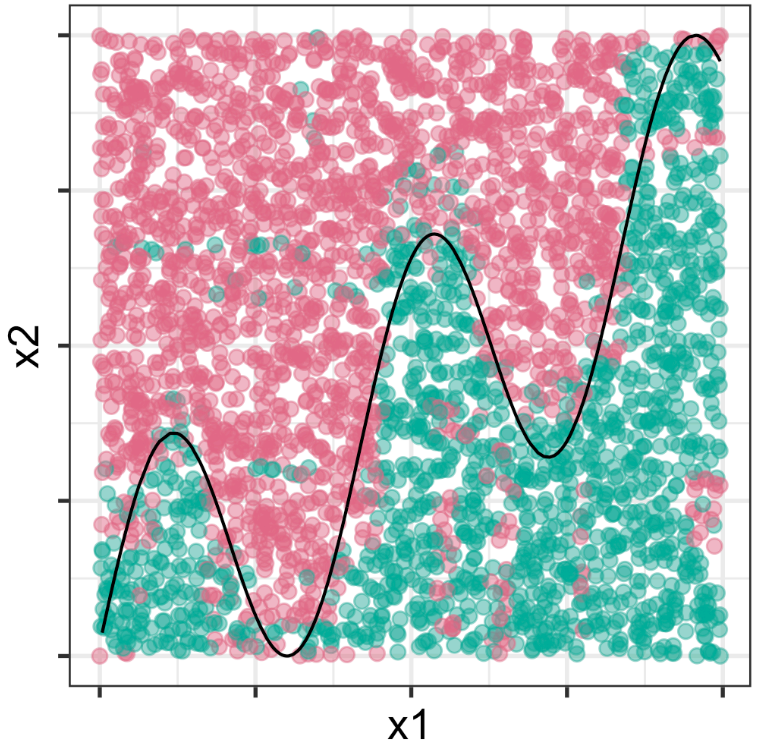
Black line is true boundary.

Grids (right) show boundaries for two different models.

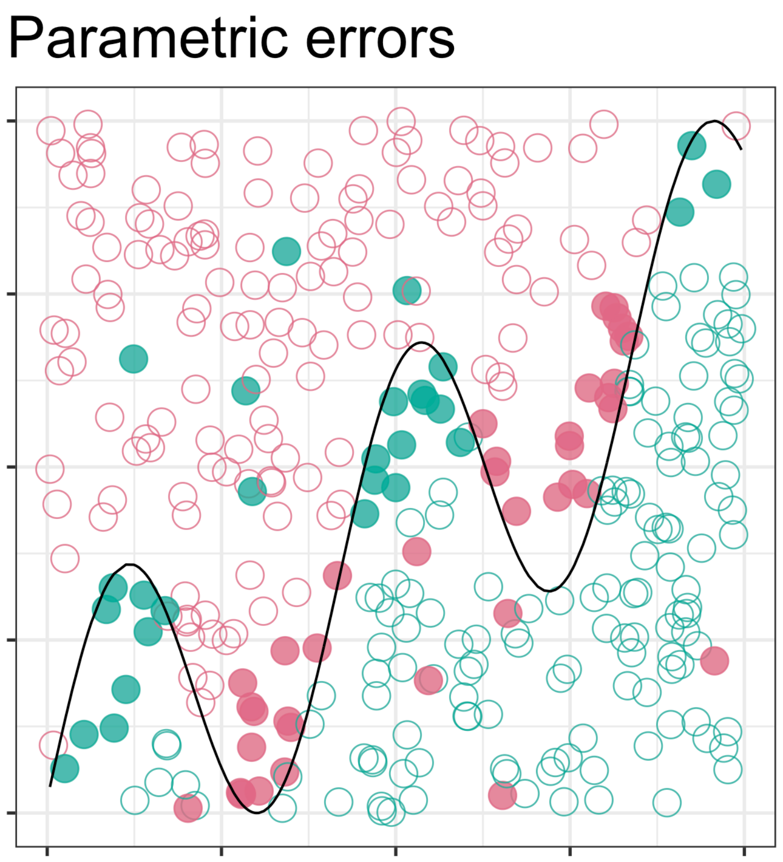
Parametric

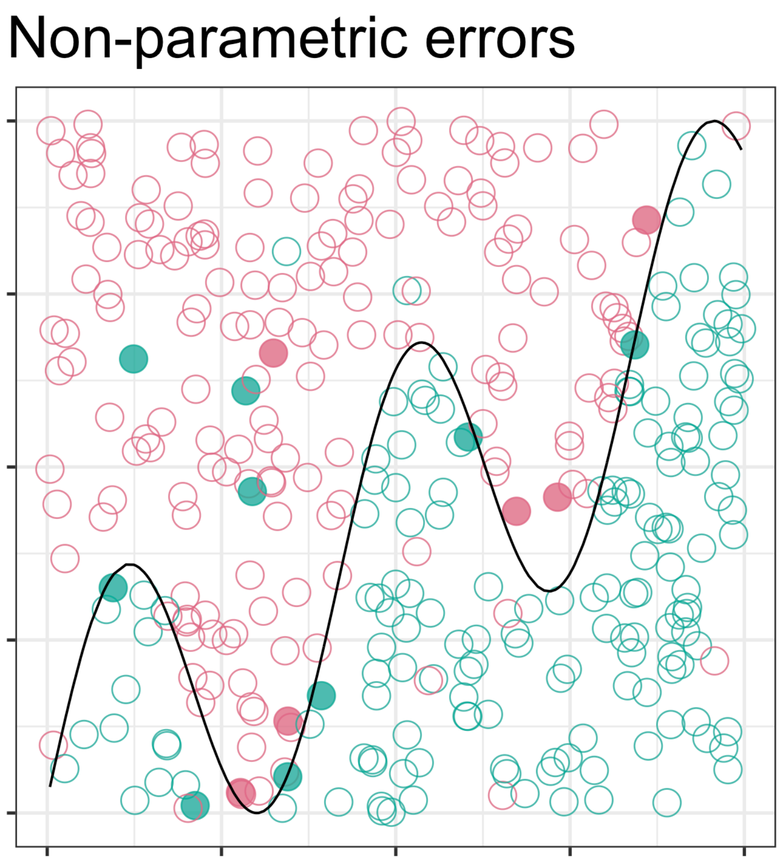


Non-parametric



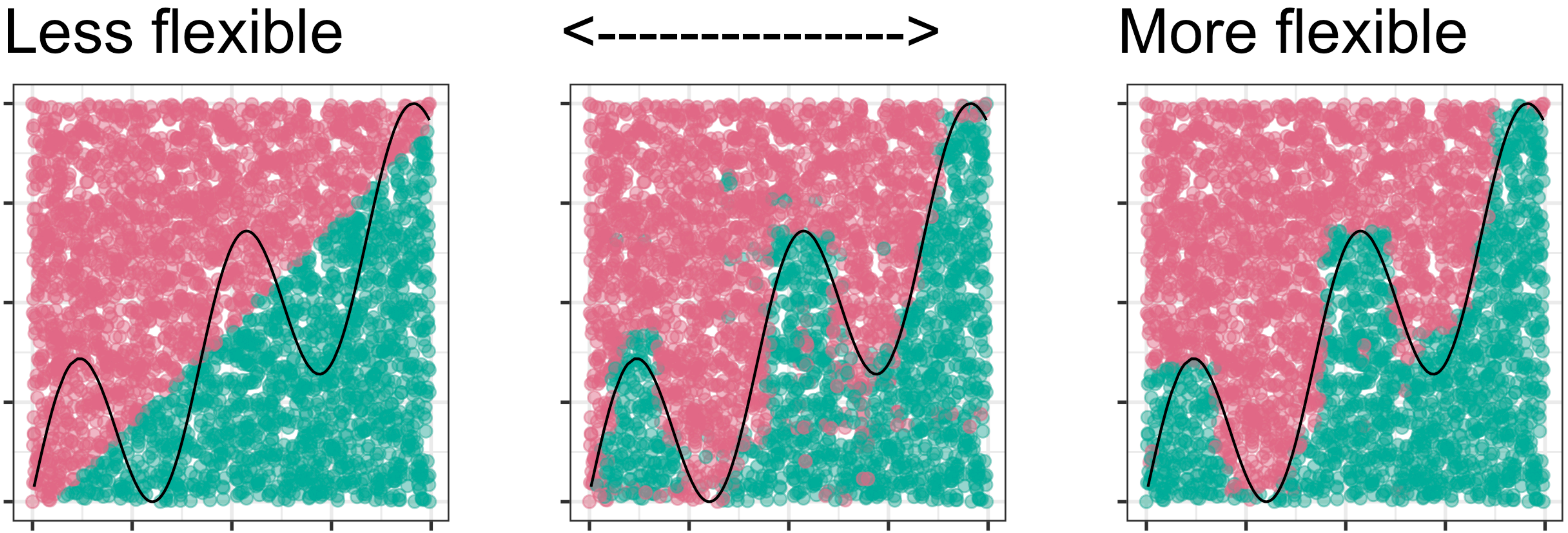
Reducible vs irreducible error





If the model form is incorrect, the error (solid circles) may arise from wrong shape, and is thus reducible. Irreducible means that we have got the right model and mistakes (solid circles) are random noise.

Flexible vs inflexible



Parametric models tend to be less flexible but non-parametric models can be flexible or less flexible depending on parameter settings.

Bias vs variance

Bias is the error that is introduced by modeling a complicated problem by a simpler problem.

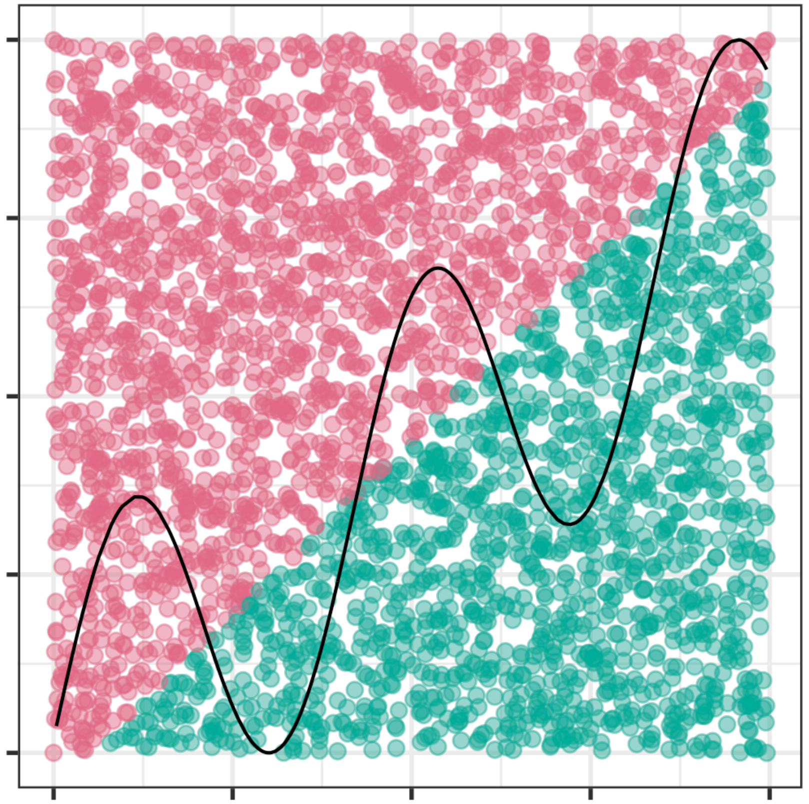
- For example, linear regression assumes a linear relationship and perhaps the relationship is not exactly linear.
- In general, but not always, the more flexible a method is, the less bias it will have because it can fit a complex shape better.

Variance refers to how much your estimate would change if you had different training data. Its measuring how much your model depends on the data you have, to the neglect of future data.

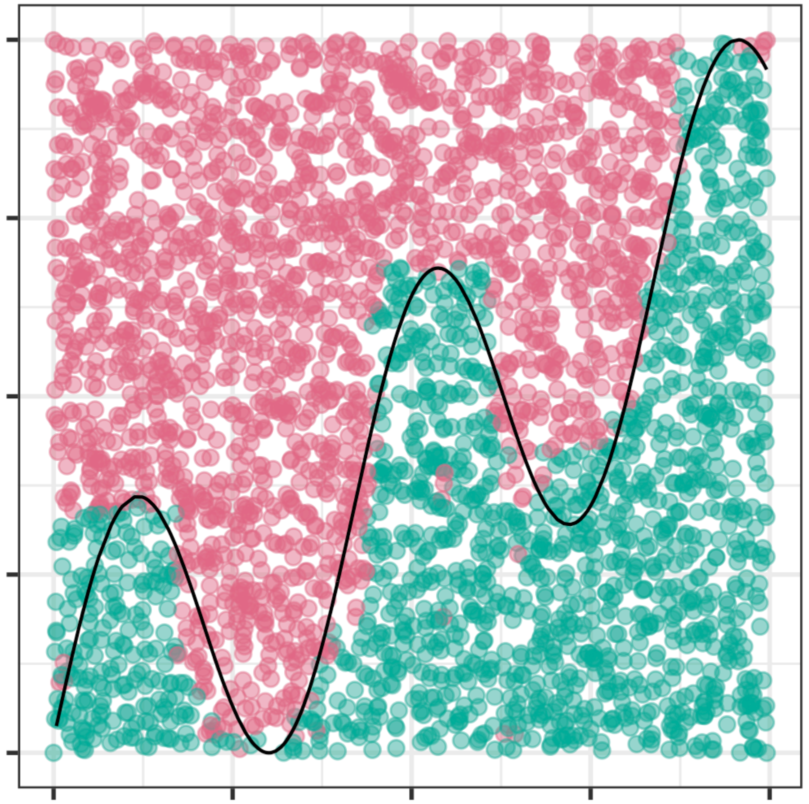
- In general, the more flexible a method is, the more variance it has.
- The size of the training data can impact on the variance.

Bias

Large bias



Small bias

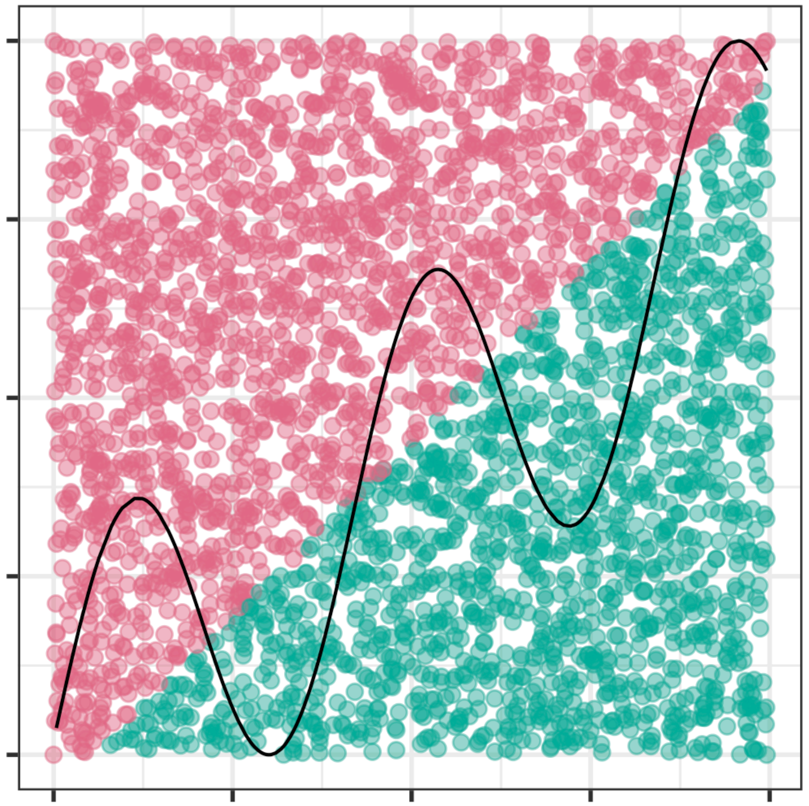


When you impose too many assumptions with a parametric model, or use an inadequate non-parametric model, such as not letting an algorithm converge fully.

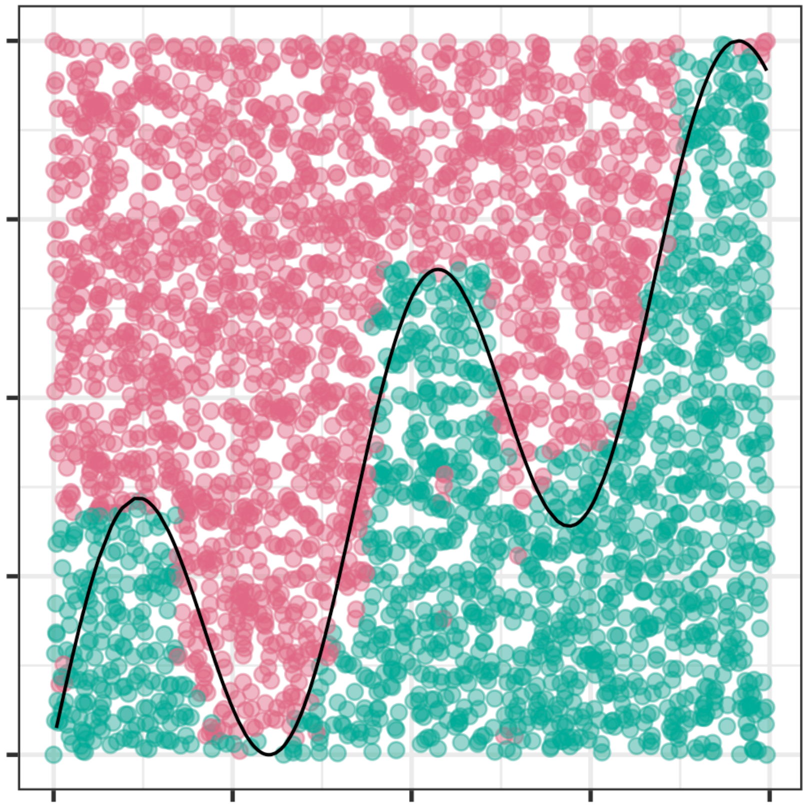
When the model closely captures the true shape, with a parametric model or a flexible model.

Variance

Small variance



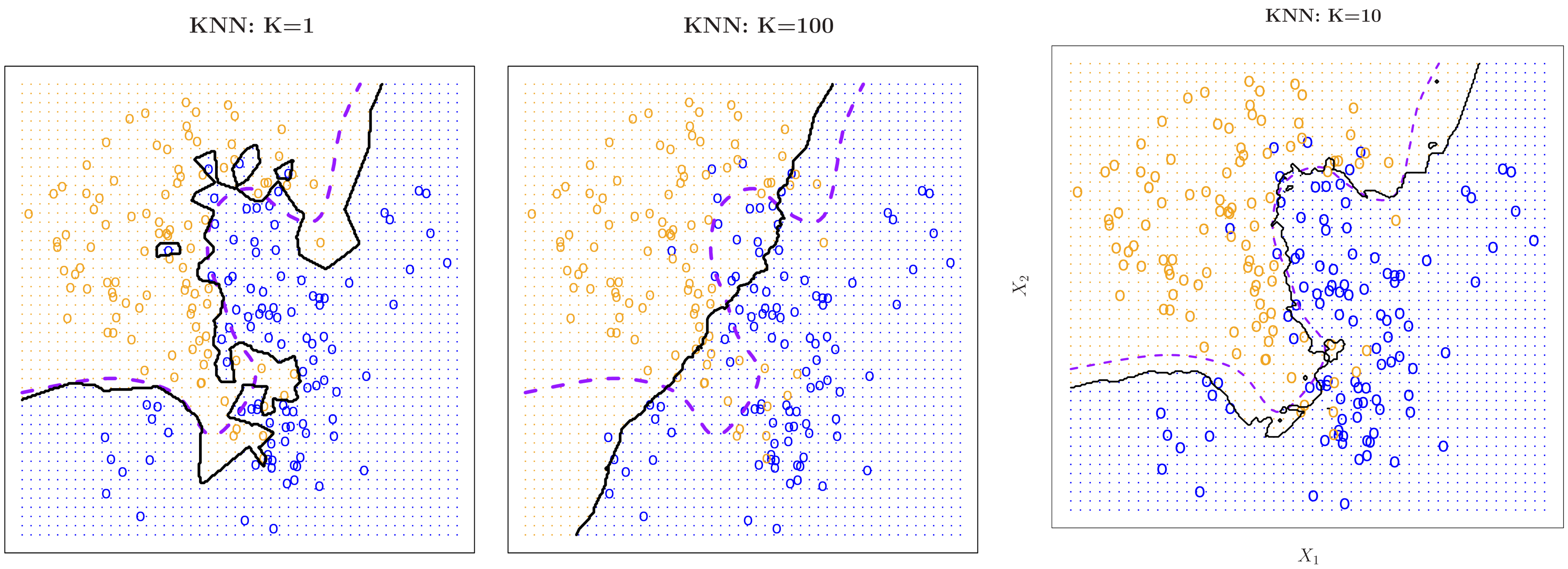
Large variance



This fit will be virtually identical even if we had a different training sample.

Likely to get a very different model if a different training set is used.

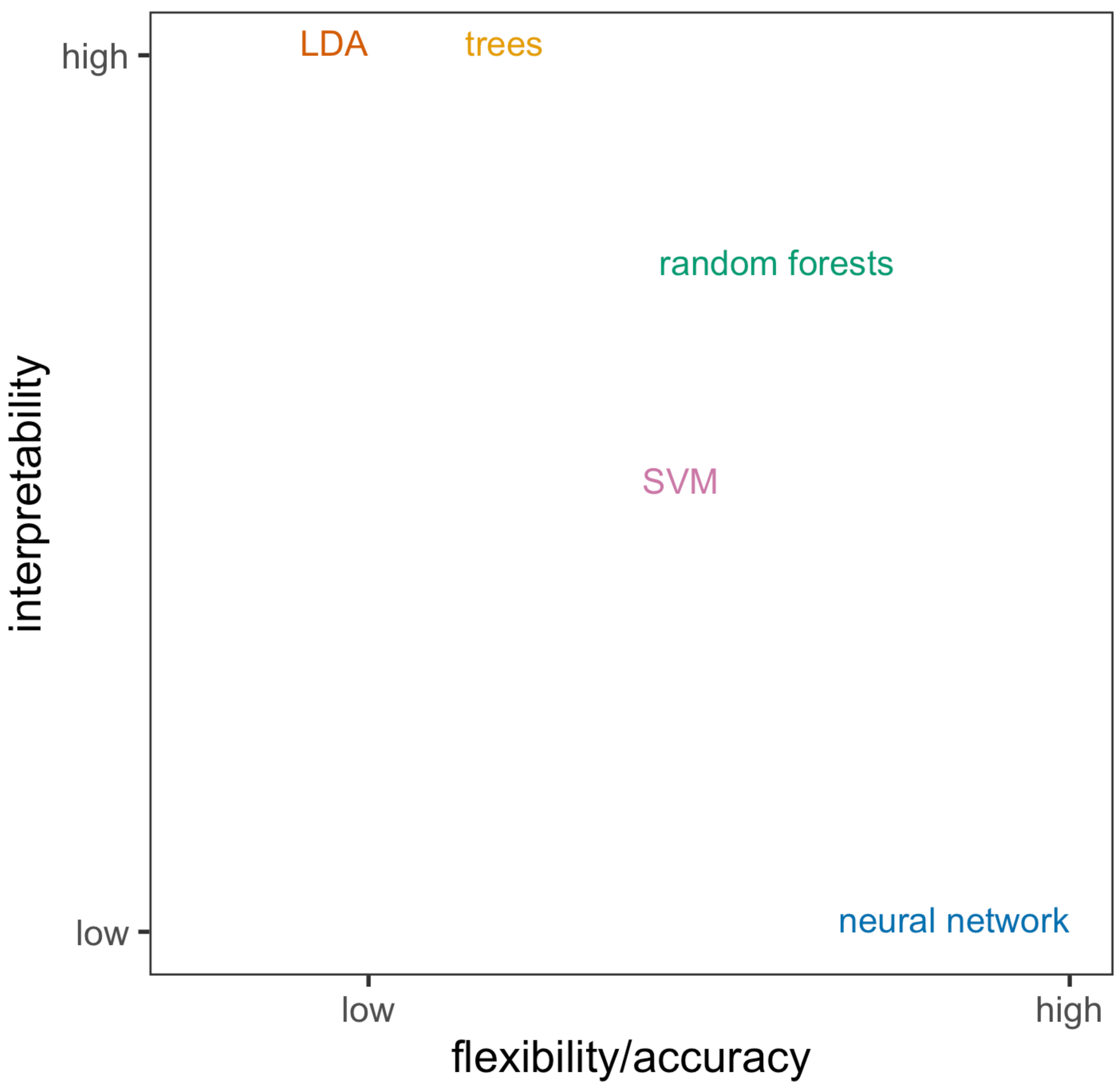
Bias-variance tradeoff



Goal: Without knowing what the true structure is, fit the signal and ignore the noise. Be flexible but not too flexible.

Images 2.16, 2.15 from ISLR

Trade-off between accuracy and interpretability



Diagnosing the fit

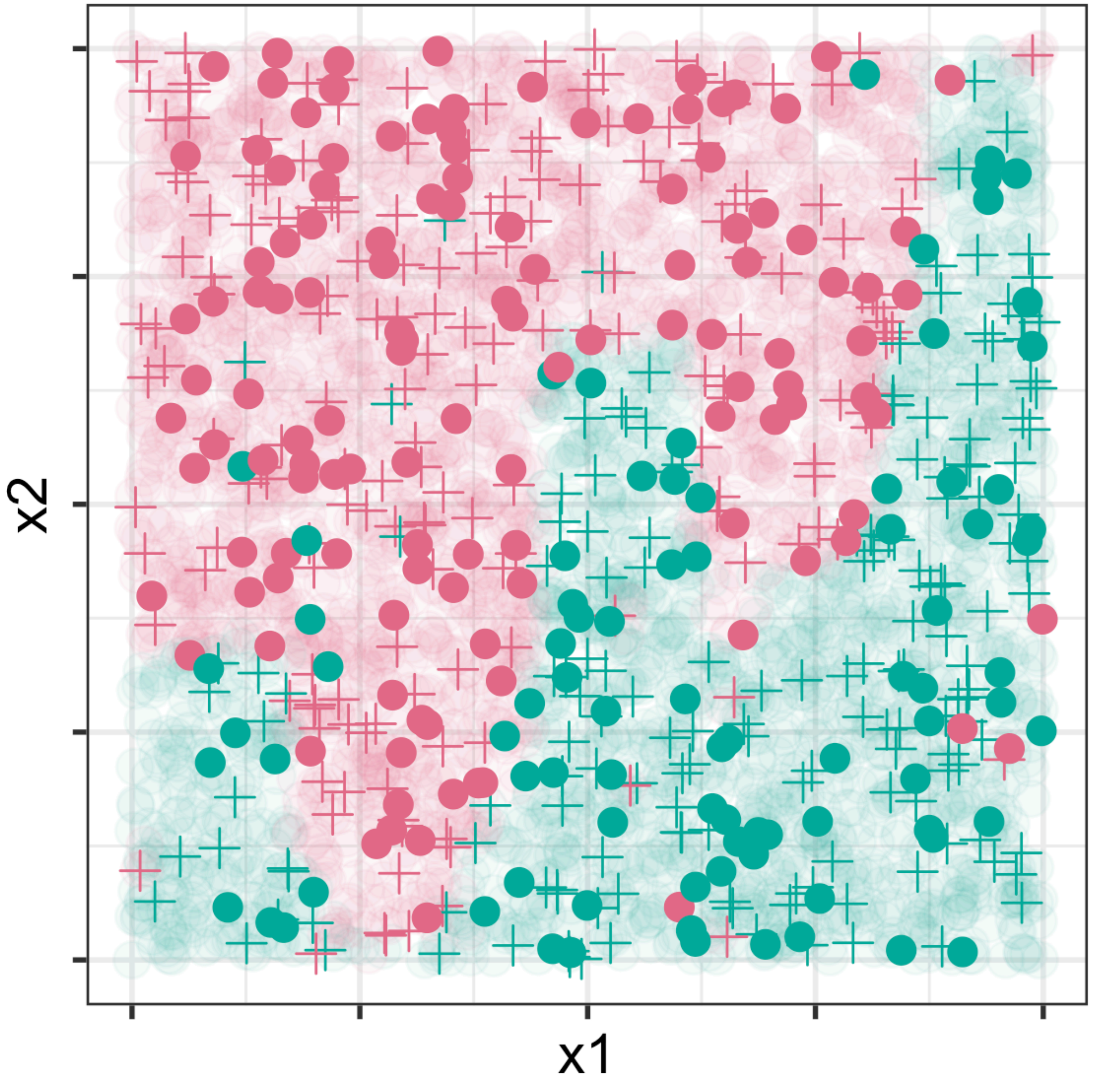
Compute and examine the usual diagnostics, some methods have more

- fit statistics: accuracy, sensitivity, specificity
- errors/misclassifications
- variable importance
- plot residuals, examine the misclassifications
- check test set is similar to training

Go beyond … Look at the data and the model together!

Wickham et al (2015) Removing the Blindfold

Training - plusses; Test - dots



Feature engineering

Creating new variables to get better fits is a special skill! Sometimes automated by the method. All are transformations of the original variables. (See `tidymodels` steps.)

- scaling, centering, sphering (`step_pca`)
- log or square root or box-cox transformation (`step_log`)
- ratio of values (`step_ratio`)
- polynomials or splines: x_1^2, x_1^3 (`step_ns`)
- dummy variables: categorical predictors expanded into multiple new binary variables (`step_dummy`)
- Convolutional Neural Networks: neural networks but with pre-processing of images to combine values of neighbouring pixels; flattening of images

The big picture

1. Know your data
    - Categorical response or no response
    - Types of predictors: quantitative, categorical
    - Independent observations
    - Do you need to handle missing values?
    - Are there anomalous observations?
2. Plot your data
    - What are the shapes (distribution and variance)?
    - Are there gaps or separations (centres)?
3. Fit a model or two
    - Compute fit statistics
    - Plot the model
    - Examine parameter estimates
4. Diagnostics
    - Which is the better model
    - Is there a simpler model?
    - Are the errors reducible or systematic?
    - Are you confident that your bias is low and variance is low?

Next: Visualisation