Applied Data Science

L2. Learning from Data. Cross-validation. Data pre-processing

Irina Mohorianu

Head of Bioinformatics/ Scientific Computing @CSCI

We've got data. Find the signal! Data (pre-processing)

Finding patterns in the data

i.e. the technical characteristics of the data are less important than the pattern (model)

- Input: some data
- Methods: statistical analysis, machine learning, programming
- Output: a robust, reproducible, generalizable model

Input	Pre-processing	ML model
Raw data	[1] **Accuracy** – how well does the data reflect a real object/ event [2] **Completeness** – are all mandatory/ necessary features present? [3] **Consistency** – repeated entries/ consistent labels? [4] **Invariance in time** – is the data outdated or a mix of old & new? [5] **Believability** – the presented values are trustworthy [6] **Interpretability** – the data and conclusions can be interpreted	Parametric/ non-parametric Can handle continuous/ categorical vars?

Data (pre-processing). Feature engineering Overview of concepts.

Input	Pre-processing	ML model

Raw data

[1] **Accuracy**
[2] **Completeness**
[3] **Consistency**
[4] **Invariance in time**
[5] **Believability**
[6] **Interpretability**

[a] **Understanding the structure of the data**
- type of features: continuous, categorical
- ranges of features: [min, max], number of categories
- missing information [labels, features]
- discriminative power of features (redundancy)

[b] **Adjusting data without tampering with signal**
- Expression ranges and One-hot encoding
- Standardisation vs scaling
- Near zero variance
- Multi-collinearity
- Dimensionality reduction

[c] **creating robust models – cross-validation. Bias/ variance**
- Training/ Validation/ Test splitting
- Cross-validation

Parametric/
non-parametric

Can handle continuous/
categorical vars?

Feature selection

Data pre-processing. Feature engineering
Understanding the structure of the data

[a] **Understanding the structure of the data**

type of features: continuous, categorical
ranges of features: [min, max], number of categories
missing information [labels, features]
discriminative power of features (redundancy)

```
!pip install seaborn
import seaborn as sns

# Load the Palmer's penguin dataset
penguins = sns.load_dataset('penguins')
```

```
penguins.head()
```

Continuous features

Categorical features

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	Male
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	Female
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	Female
3	Adelie	Torgersen	NaN	NaN	NaN	NaN	NaN
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	Female

Missing data

Horst AM, Hill AP, Gorman KB (2020). palmerpenguins: Palmer Archipelago (Antarctica) penguin data. R package version 0.1.0. https://allisonhorst.github.io/palmerpenguins/. doi:10.5281/zenodo.3960218.

Data pre-processing. Feature engineering
Understanding the structure of the data

Knowing your dataset is essential – **explore**, **clean**, **visualize** (part of) your data

Examine several rows of data

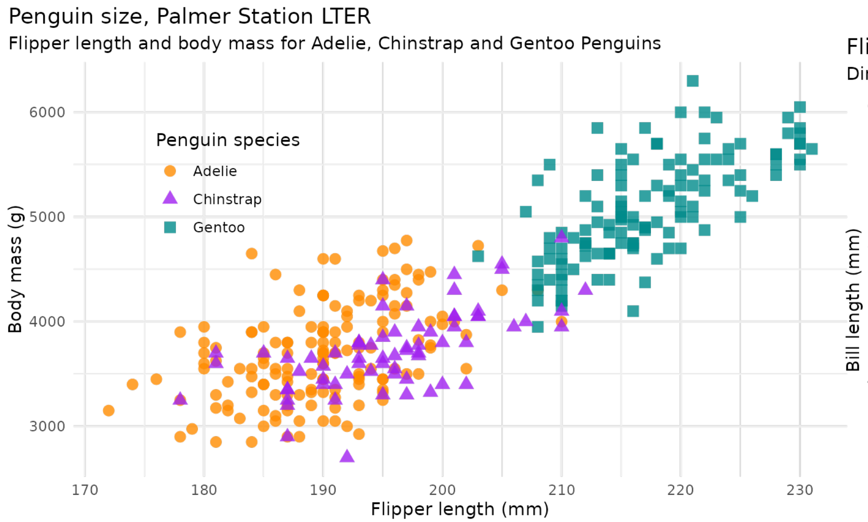
Check basic statistics. Evaluate data types

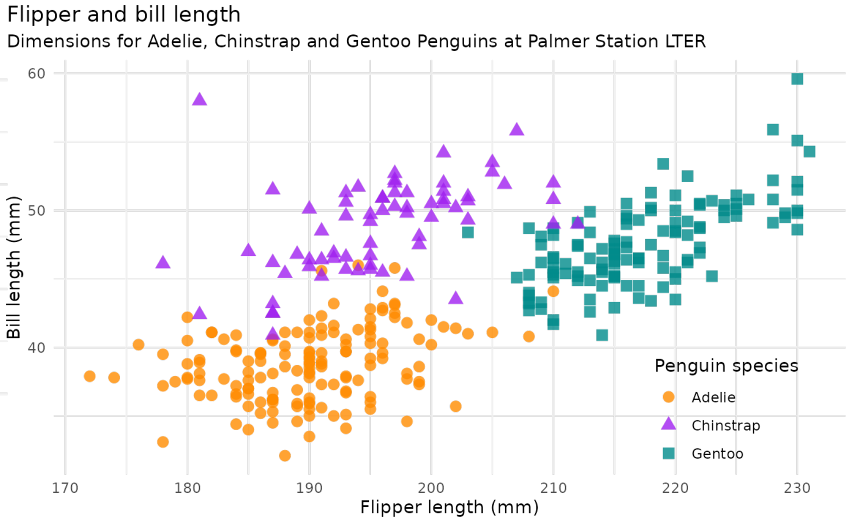
Assess missing entries

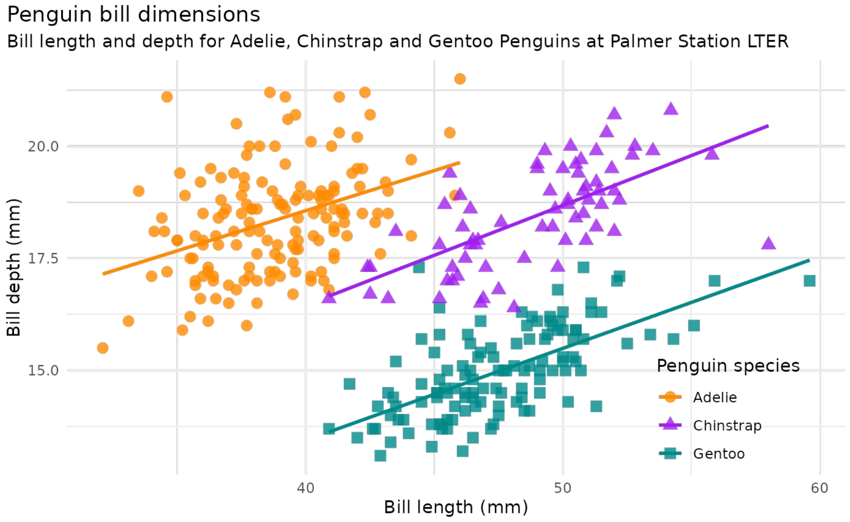
```
import pandas as pd

# Get summmary of numeric and non-numeric features
numeric_summary = penguins.describe(include=[float, int])
non_numeric_summary = penguins.describe(include=[object])

# Combine the summaries
summary = pd.concat([numeric_summary, non_numeric_summary], axis=0)

print(summary)
```

```
        bill_length_mm  bill_depth_mm  flipper_length_mm  body_mass_g species
count       342.000000     342.000000         342.000000   342.000000     NaN
mean         43.921930      17.151170         200.915205  4201.754386     NaN
std           5.459584       1.974793          14.061714   801.954536     NaN
min          32.100000      13.100000         172.000000  2700.000000     NaN
25%          39.225000      15.600000         190.000000  3550.000000     NaN
50%          44.450000      17.300000         197.000000  4050.000000     NaN
75%          48.500000      18.700000         213.000000  4750.000000     NaN
max          59.600000      21.500000         231.000000  6300.000000     NaN
count              NaN            NaN                NaN          NaN     344
unique             NaN            NaN                NaN          NaN       3
top                NaN            NaN                NaN          NaN  Adelie
freq               NaN            NaN                NaN          NaN     152

         island   sex
count       NaN   NaN
mean        NaN   NaN
std         NaN   NaN
min         NaN   NaN
25%         NaN   NaN
50%         NaN   NaN
75%         NaN   NaN
max         NaN   NaN
count       344   333
unique        3     2
top      Biscoe  Male
freq        168   168
```

Data pre-processing. Feature engineering
Understanding the structure of the data





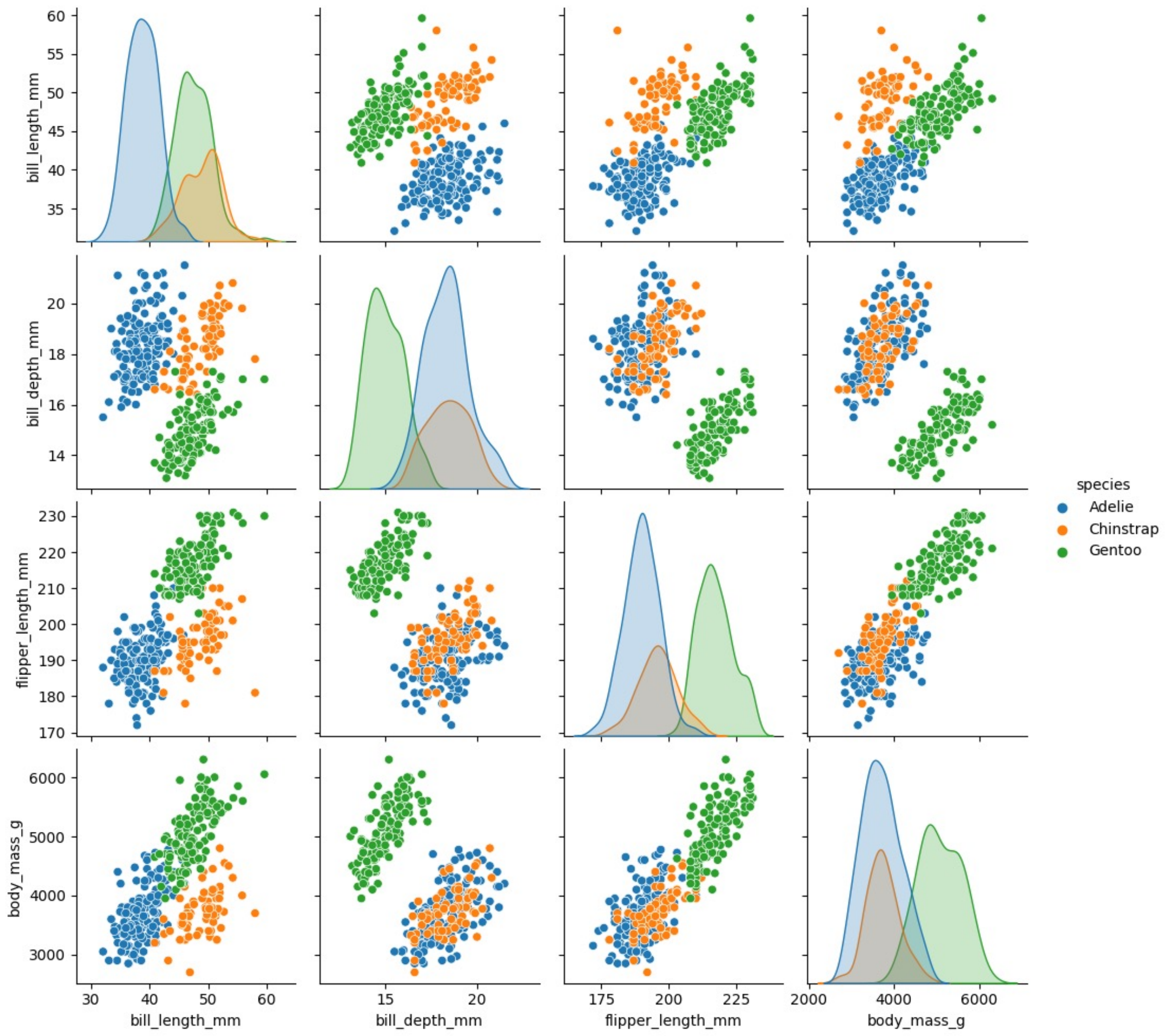


The body mass is highly correlated to flipper length (top left)

We note linear separability between classes

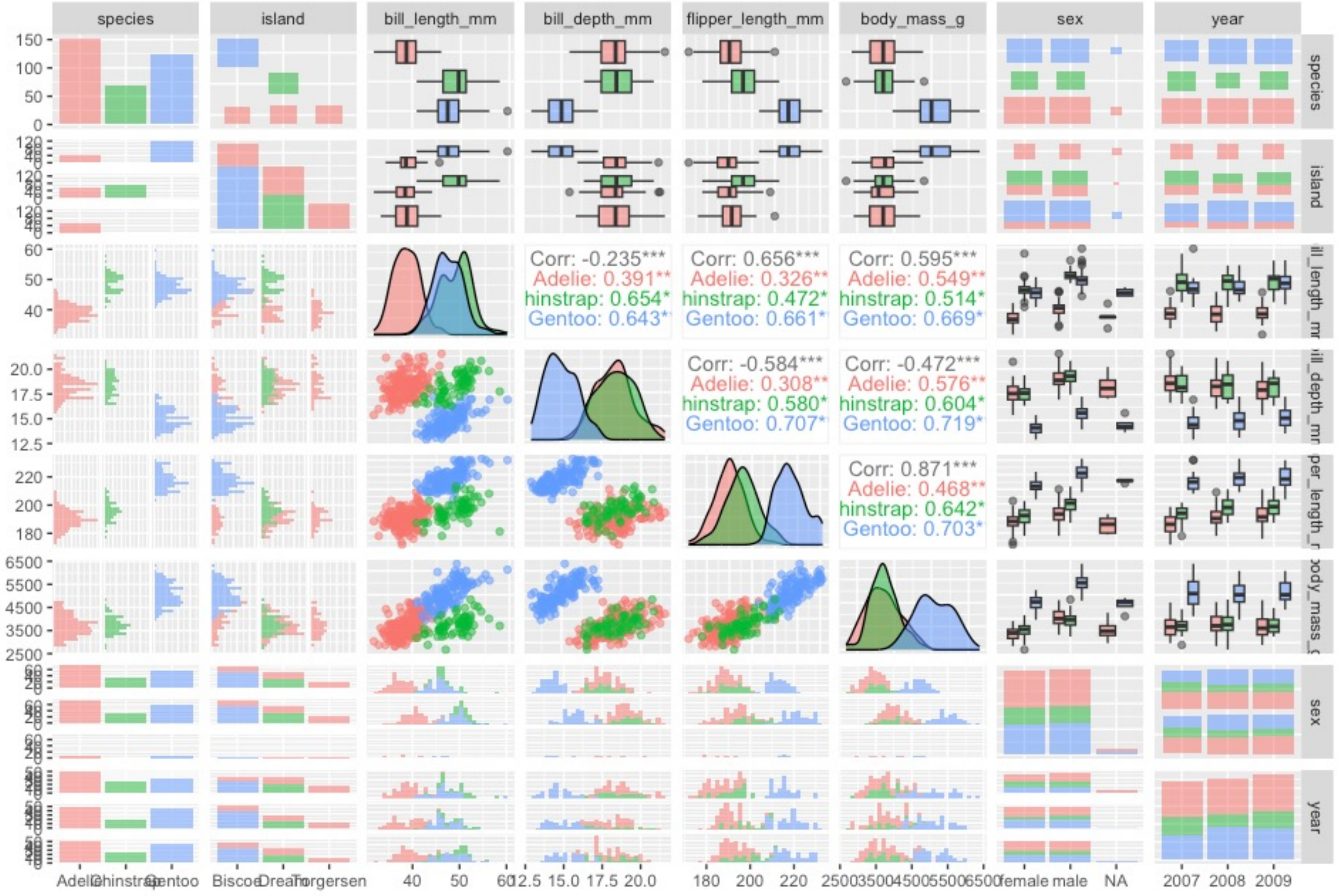
The colinearity differs per class (bottom left).

Data pre-processing. Feature engineering
Understanding the structure of the data




```python
# Create pairplot
sns.pairplot(penguins, hue='species', diag_kind='kde')
```

Data pre-processing. Feature engineering
Understanding the structure of the data



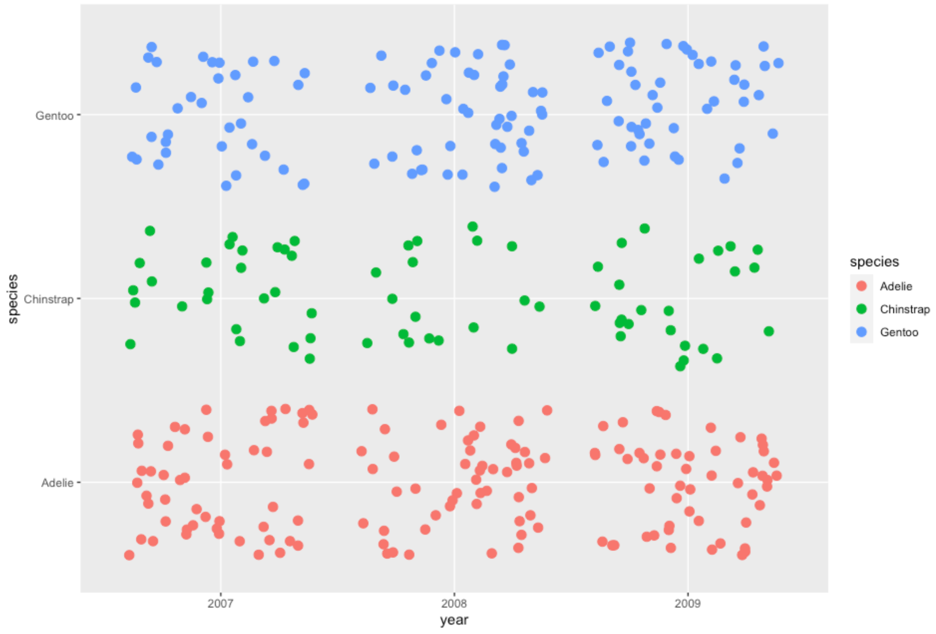

```
ggpairs(penguins, ggplot2::aes(colour
= species, alpha = 0.4))
```

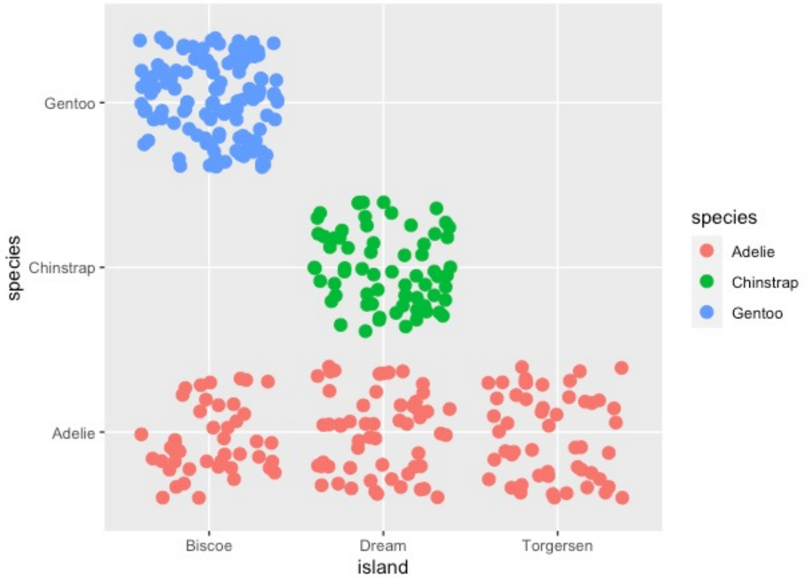
Simpson's paradox is observed in probability and statistics; a trend appears in several groups of data but disappears or reverses when the groups are combined.

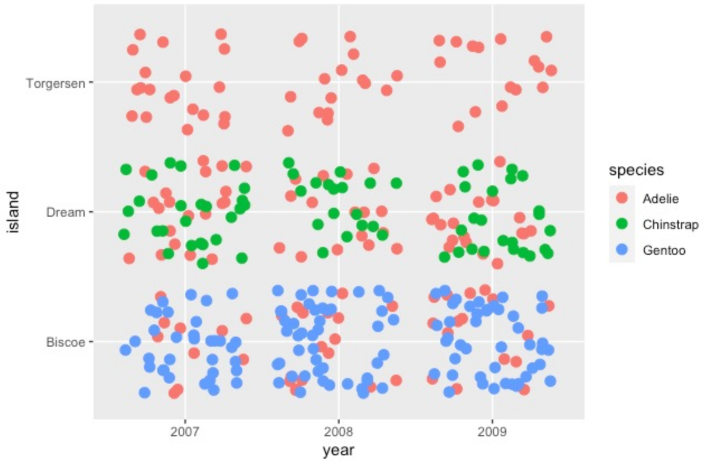
Data pre-processing. Feature engineering
Understanding the structure of the data

```
table(penguins$species, penguins$island)
```

	Biscoe	Dream	Torgersen
Adelie	44	56	52
Chinstrap	0	68	0
Gentoo	124	0	0







Data pre-processing. Feature engineering
Adjusting data without tampering with signal

Adjusting data without tampering with signal

- **Expression ranges and One-hot encoding**
- Standardisation vs scaling
- Near zero variance
- Multi-collinearity
- Dimensionality reduction

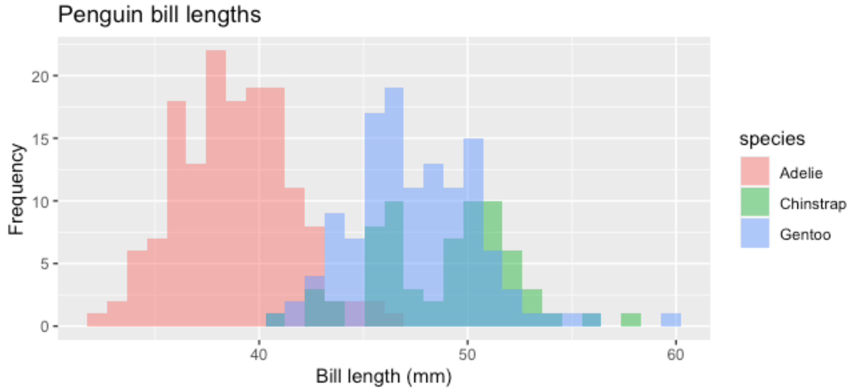
```
summary(penguins)
      species          island    bill_length_mm  bill_depth_mm  
 Adelie   :152   Biscoe   :168   Min.   :32.10   Min.   :13.10  
 Chinstrap: 68   Dream    :124   1st Qu.:39.23   1st Qu.:15.60  
 Gentoo   :124   Torgersen: 52   Median :44.45   Median :17.30  
                                 Mean   :43.92   Mean   :17.15  
                                 3rd Qu.:48.50   3rd Qu.:18.70  
                                 Max.   :59.60   Max.   :21.50  
                                 NA's   :2       NA's   :2      
 flipper_length_mm  body_mass_g       sex           year     
 Min.   :172.0     Min.   :2700   female:165   Min.   :2007  
 1st Qu.:190.0     1st Qu.:3550   male  :168   1st Qu.:2007  
 Median :197.0     Median :4050   NA's  : 11   Median :2008  
 Mean   :200.9     Mean   :4202                Mean   :2008  
 3rd Qu.:213.0     3rd Qu.:4750                3rd Qu.:2009  
 Max.   :231.0     Max.   :6300                Max.   :2009  
 NA's   :2         NA's   :2                                 
```

Continuous features: bill length, depth, flipper length, body mass
Categorical features: island, sex, year
Output: species (categorical i.e. classification problem)

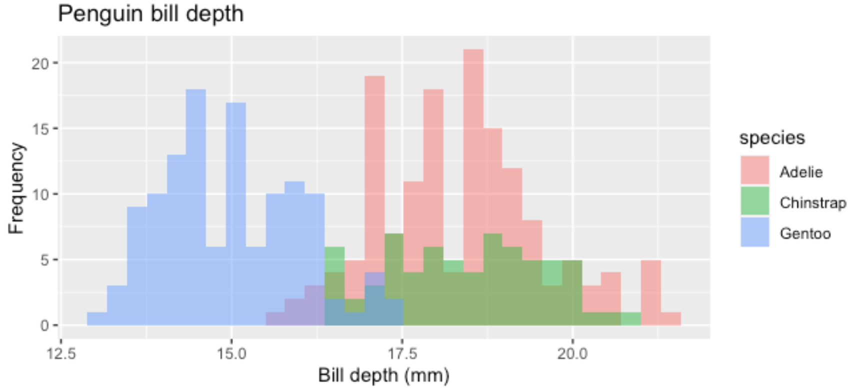
Data pre-processing. Feature engineering
Adjusting data without tampering with signal

Adjusting data without tampering with signal

- **Expression ranges and One-hot encoding**
- Standardisation vs scaling
- Near zero variance
- Multi-collinearity
- Dimensionality reduction




```
bill_length_mm   bill_depth_mm    flipper_length_mm   body_mass_g
 Min.   :32.10    Min.   :13.10    Min.   :172.0       Min.   :2700
 1st Qu.:39.23    1st Qu.:15.60    1st Qu.:190.0       1st Qu.:3550
 Median :44.45    Median :17.30    Median :197.0       Median :4050
 Mean   :43.92    Mean   :17.15    Mean   :200.9       Mean   :4202
 3rd Qu.:48.50    3rd Qu.:18.70    3rd Qu.:213.0       3rd Qu.:4750
 Max.   :59.60    Max.   :21.50    Max.   :231.0       Max.   :6300
 NA's   :2        NA's   :2        NA's   :2           NA's   :2
```



The ranges are not comparable.

Standardization vs scaling

- [min, max] scaling to a predefined range
- Robust scaling – the transformation is performed on the IQR
- Z transformation (on mean, standard deviation or median and MAD)

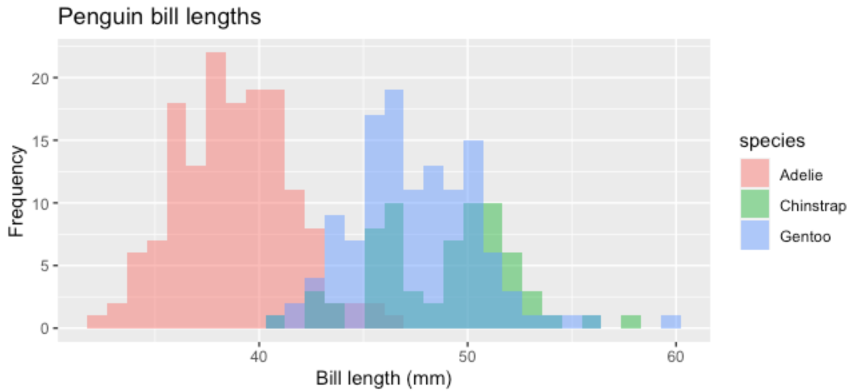
Data pre-processing. Feature engineering
Adjusting data without tampering with signal

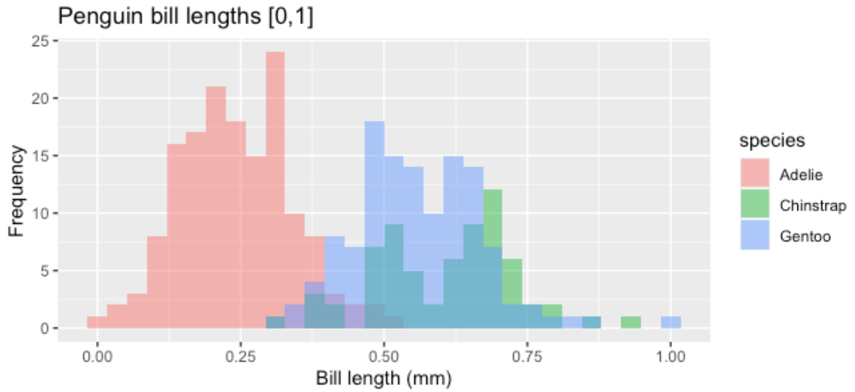
Adjusting data without tampering with signal

- Expression ranges and One-hot encoding
- **Standardisation vs scaling**
- Near zero variance
- Multi-collinearity
- Dimensionality reduction

[min, max] scaling to a predefined range

```
[min, max] --> [0,1]

[min, max] - min --> [min - min, max - min]
[min - min, max - min] / (max - min) --> [0,1]

[min, max] --> [0,1] --> [a,b]
Use a linear transformation f(x) = mx + n
f(0) = n => a = n
f(1) = m + n => b = m + n => m = b - a

f(x) = (b - a) * x + a
```

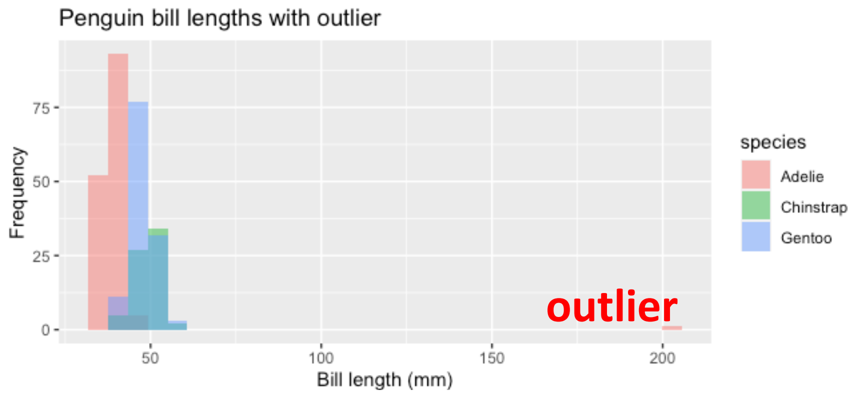




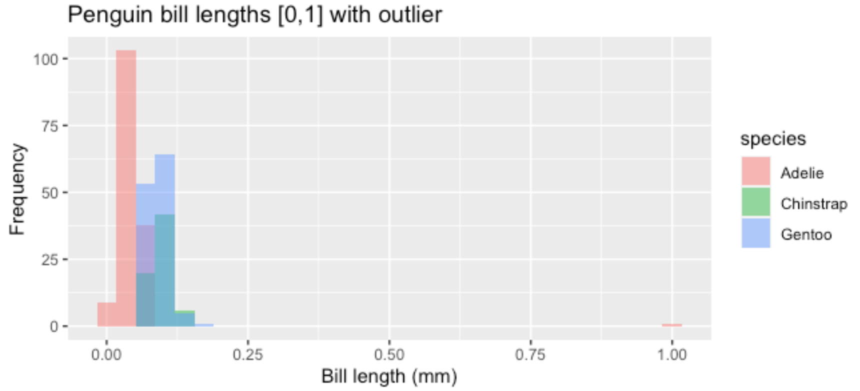
Data pre-processing. Feature engineering
Adjusting data without tampering with signal

Adjusting data without tampering with signal

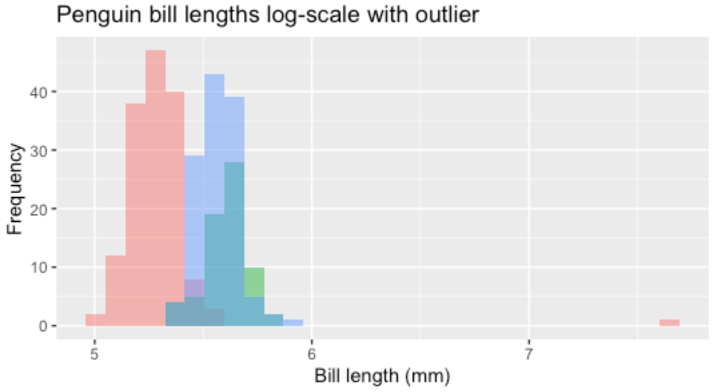
- Expression ranges and One-hot encoding
- **Standardisation vs scaling**
- Near zero variance
- Multi-collinearity
- Dimensionality reduction



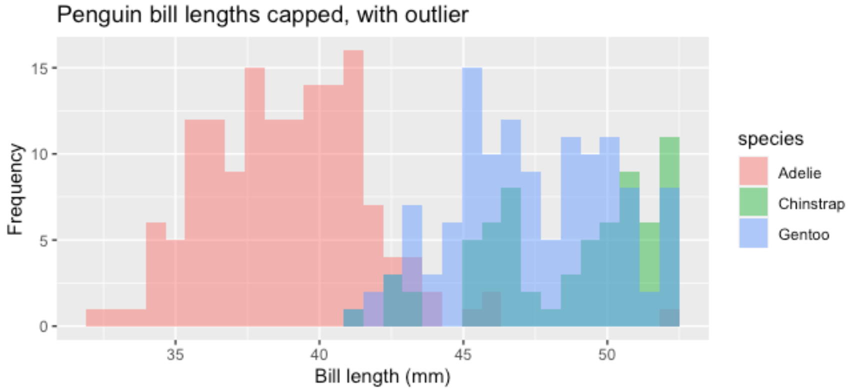
Outliers can skew the data and lead to misinterpretations.



The proposed scaling does not change the distribution.



Log$_2$ transformation of bill lengths



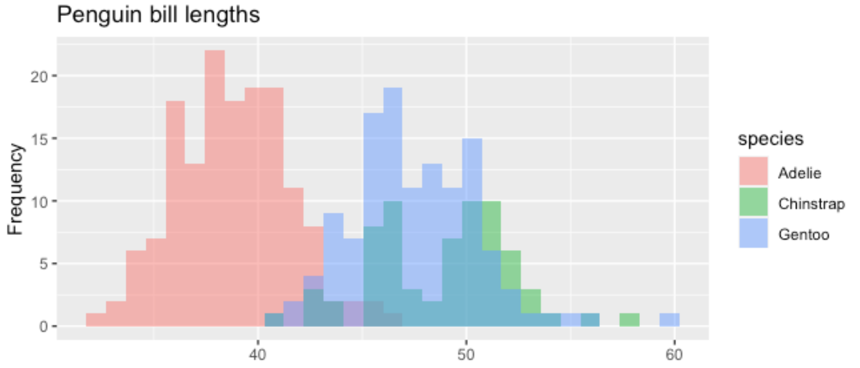
Capped values of bill lengths

Data pre-processing. Feature engineering

Adjusting data without tampering with signal

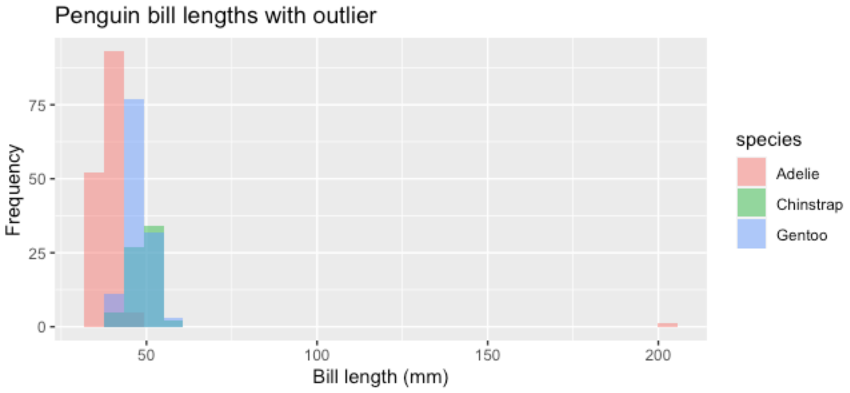
Adjusting data without tampering with signal

- Expression ranges and One-hot encoding
- **Standardisation vs scaling**
- Near zero variance
- Multi-collinearity
- Dimensionality reduction

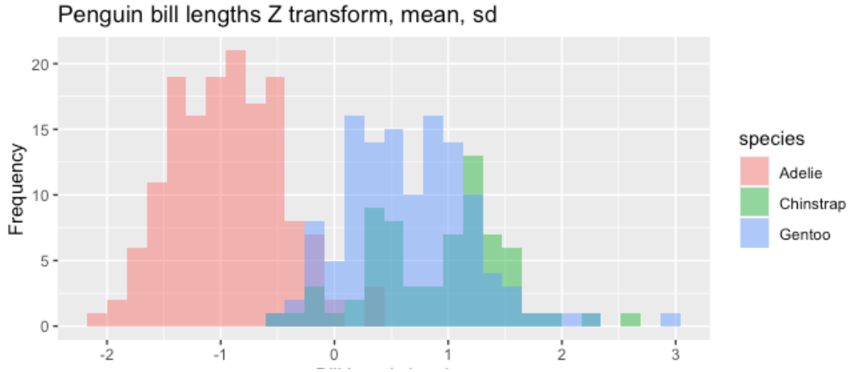


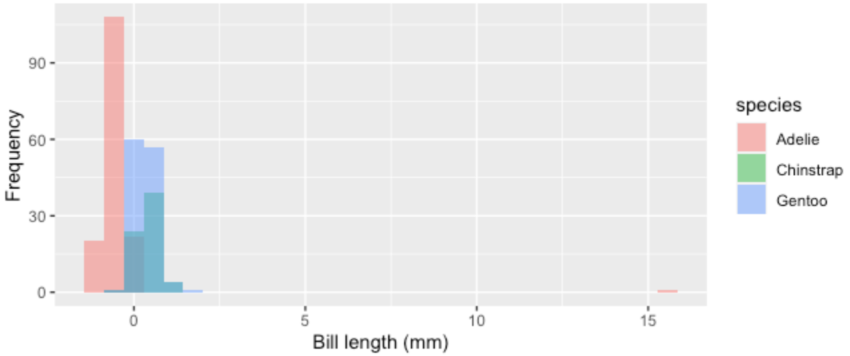
mean = 43.92
sd = 5.49

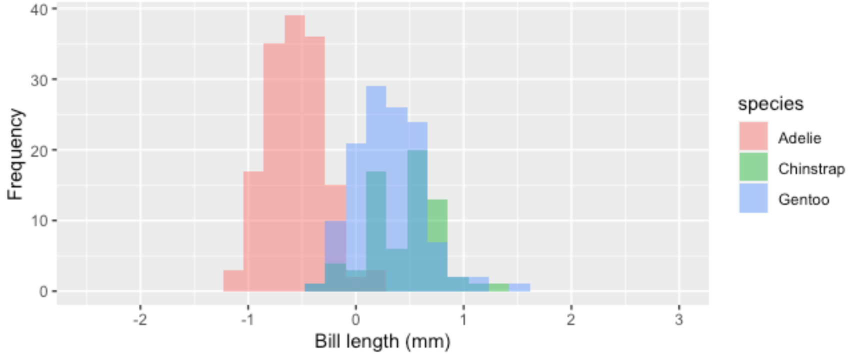
$$Z = \frac{x_i - \mu}{\sigma}$$



mean = 44.39
sd = 10.04







Data pre-processing. Feature engineering
Adjusting data without tampering with signal

Adjusting data without tampering with signal

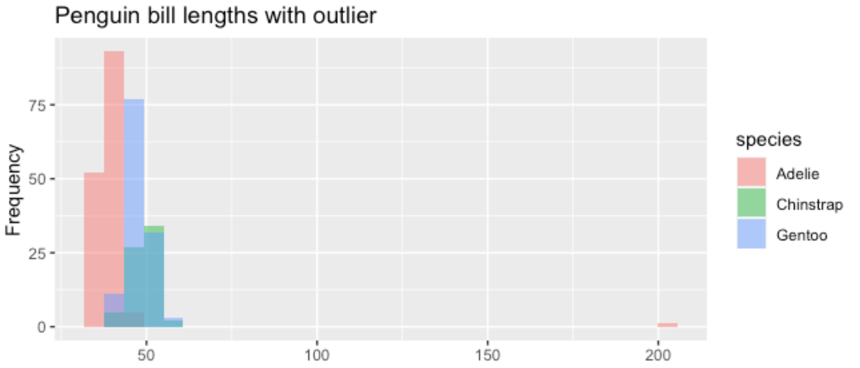
- Expression ranges and One-hot encoding
- **Standardisation vs scaling**
- Near zero variance
- Multi-collinearity
- Dimensionality reduction

$$Z = \frac{x_i - \mu}{\sigma}$$

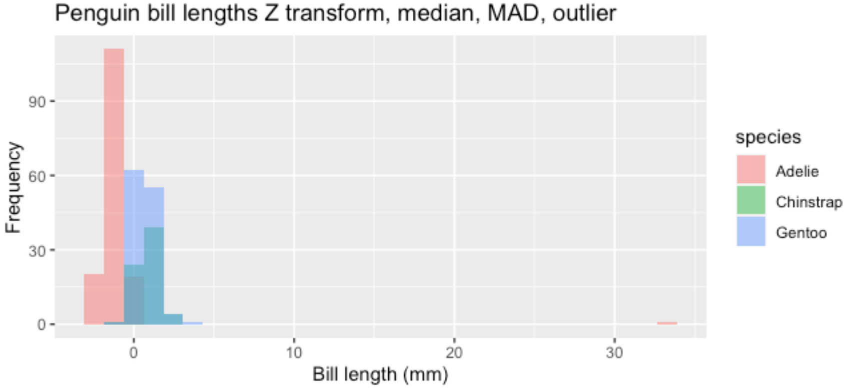
Mean and sd can be influenced by outliers.
Median, MAD are more robust
MAD = median absolute deviation

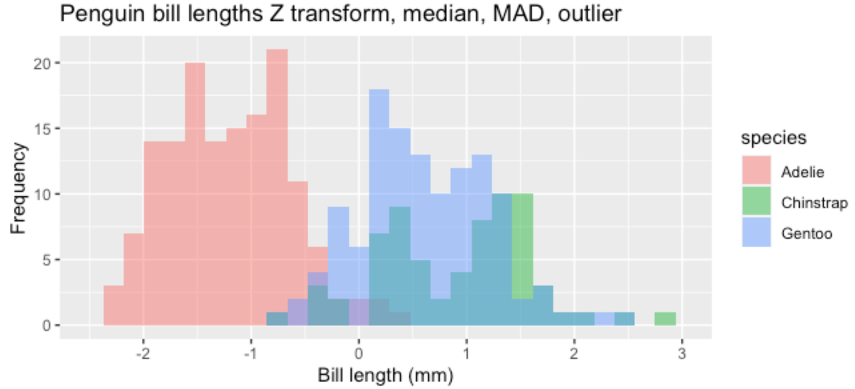
$$Z_{med} = \frac{x_i - median}{MAD}$$

$$MAD = median(|x_i - median|)$$




```
mean = 43.92
sd   = 5.49

mean_o = 44.39
sd_o   = 10.04

median   = 44.45
median_o = 44.5
MAD      = 4.7
```





Data pre-processing. Feature engineering
Adjusting data without tampering with signal

Adjusting data without tampering with signal

- **Expression ranges and One-hot encoding**
- Standardisation vs scaling
- Near zero variance
- Multi-collinearity
- Dimensionality reduction

```
 species          island         sex        year
 Adelie   :152   Biscoe   :168   female:165  2007:110
 Chinstrap: 68   Dream    :124   male  :168  2008:114
 Gentoo   :124   Torgersen: 52   NA's  : 11  2009:120
```

Briscoe	1		Briscoe	100
Dream	2	→	Dream	010
Torgersen	3		Torgersen	001

The island feature is difficult to handle in a numerical setting.

[longitude and latitude]

[distance from the POI]

Do we want to compare classes or use them in a static way?

Hamming (Edit) distances

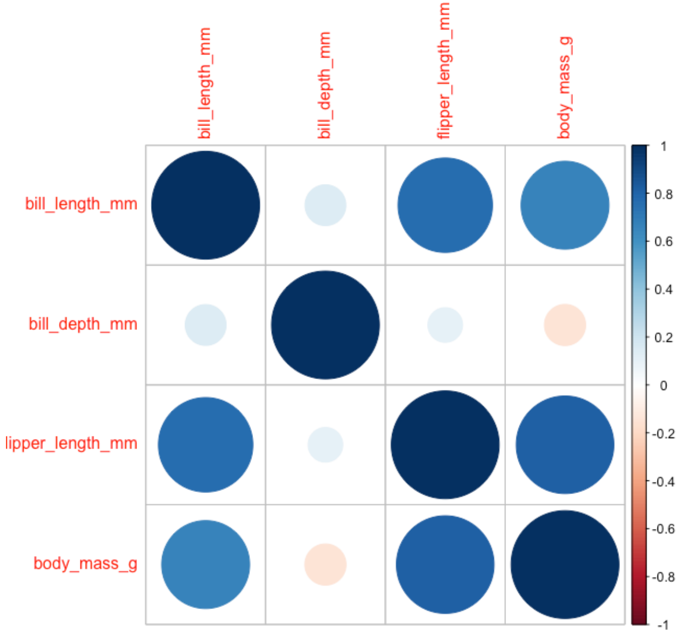
Briscoe	100
Dream	010

Briscoe	100
Torgersen	001

Data pre-processing. Feature engineering
Adjusting data without tampering with signal

Adjusting data without tampering with signal

- Expression ranges and One-hot encoding
- Standardisation vs scaling
- **Near zero variance**
- **Multi-collinearity**
- Dimensionality reduction



The near zero variance protects against constant features.

	freqRatio	percentUnique	zeroVar	nzv
species	1.226891	0.9009009	FALSE	FALSE
flipper_length_mm	1.235294	16.2162162	FALSE	FALSE
body_mass_g	1.200000	27.9279279	FALSE	FALSE
sex.female	1.018182	0.6006006	FALSE	FALSE
sex.male	1.018182	0.6006006	FALSE	FALSE

There are no issues on near zero variance.

The maximum correlation is 0.76 between bill and flipper length.

Highly correlated features:

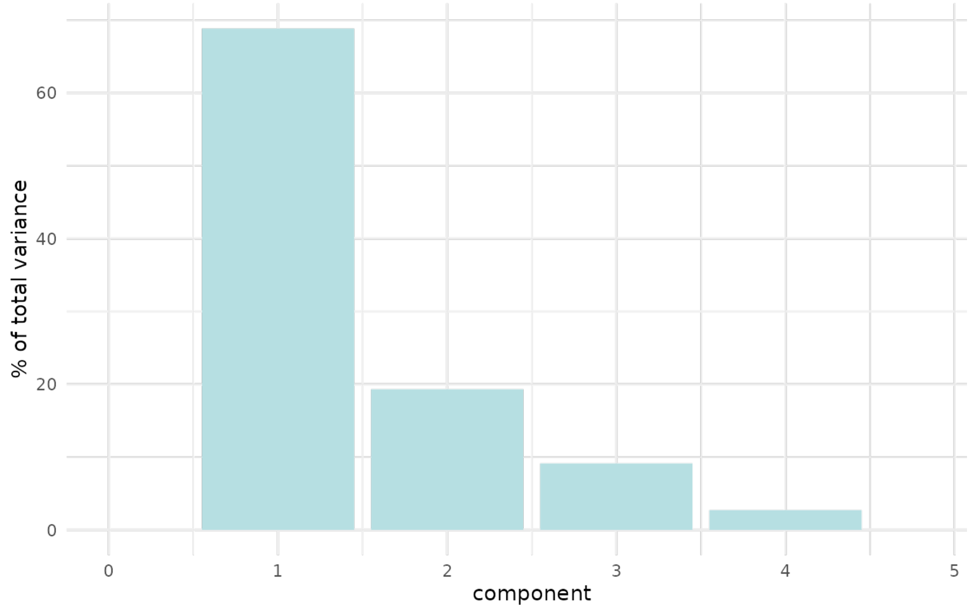
- [a] could be excluded
- [b] could be replaced with
  - a representative
  - a weighted summary

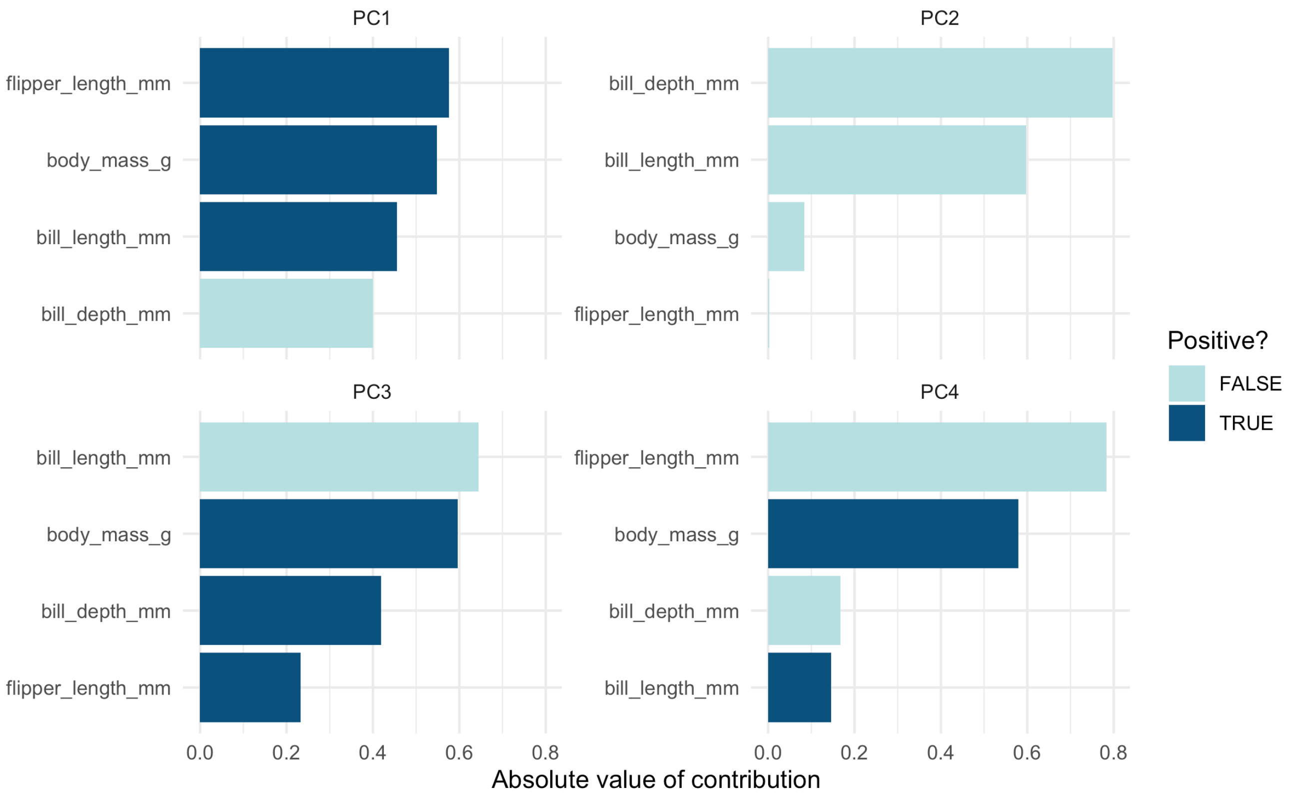
Data pre-processing. Feature engineering
Adjusting data without tampering with signal

Adjusting data without tampering with signal

- Expression ranges and One-hot encoding
- Standardisation vs scaling
- Near zero variance
- Multi-collinearity
- **Dimensionality reduction**

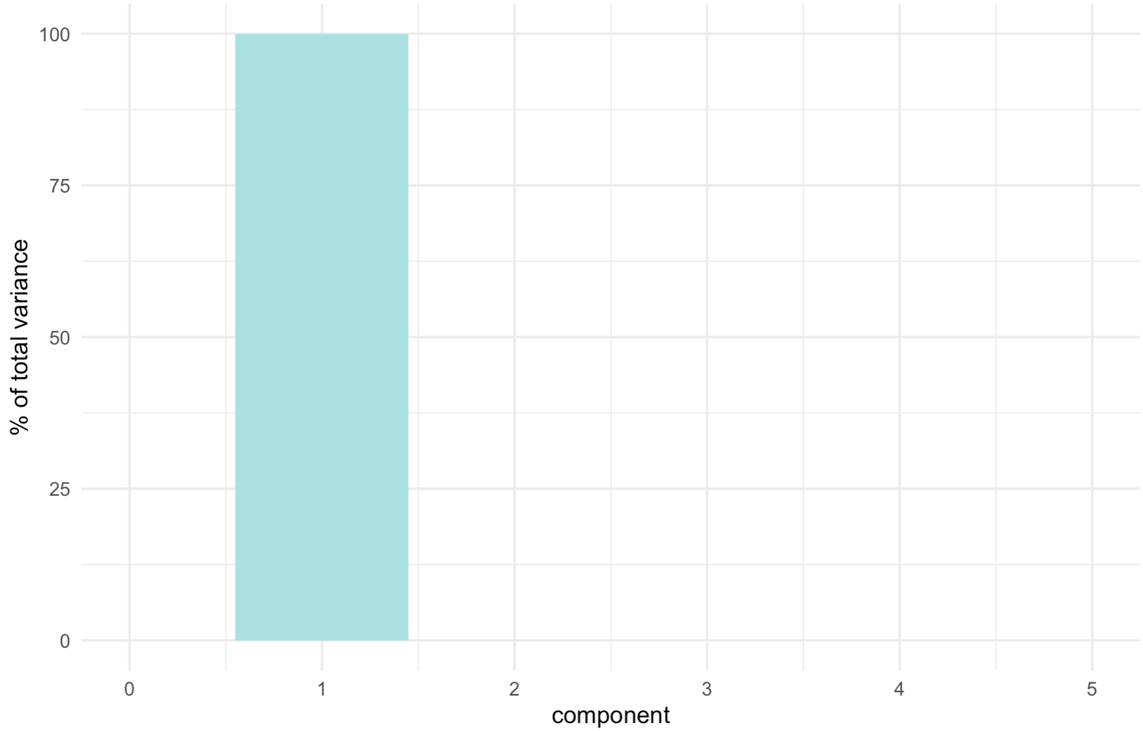
Yet another angle of linear combination of features is:
Principal Component Analysis (very briefly presented)

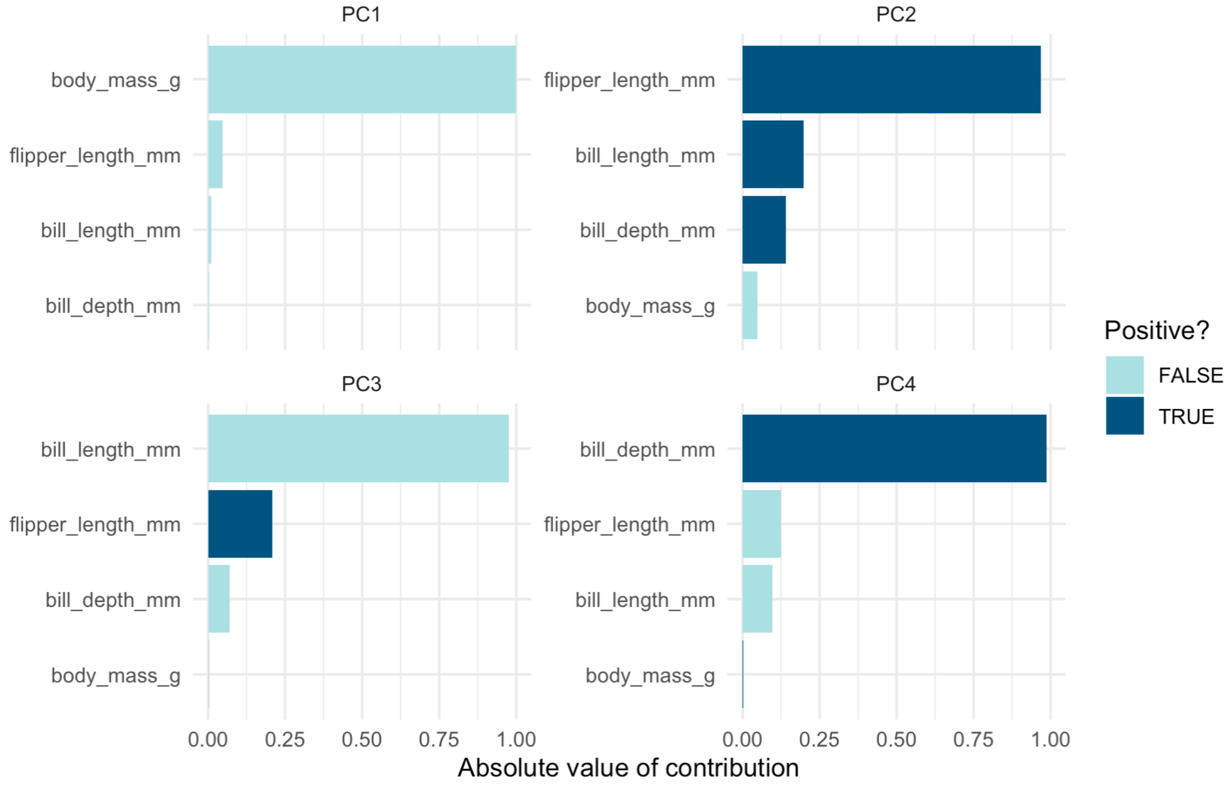




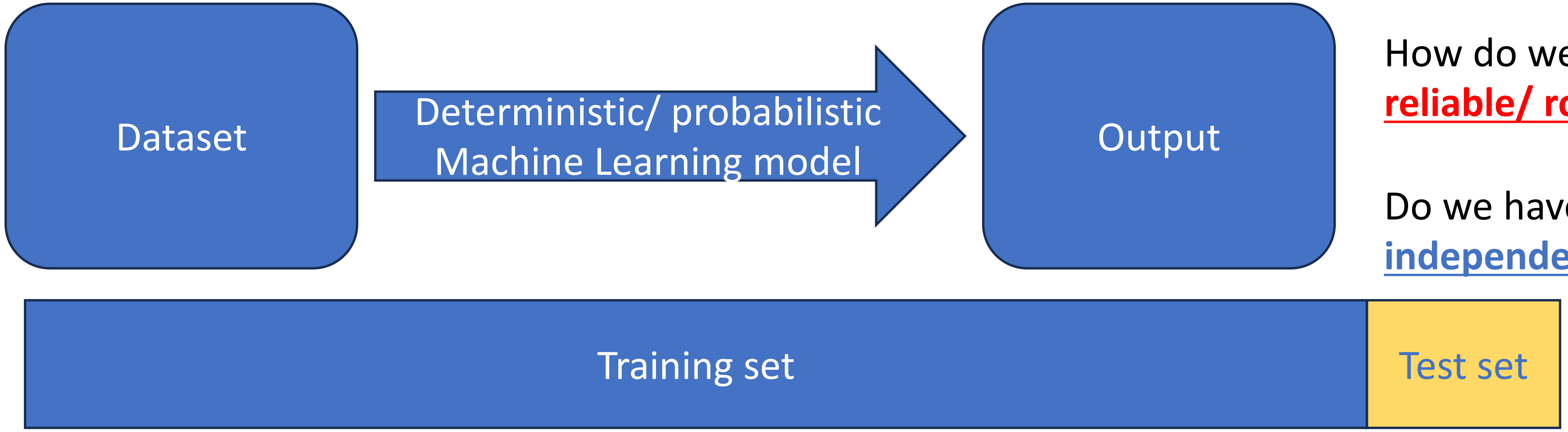
Data pre-processing. Feature engineering
Adjusting data without tampering with signal

The PCAs rely on computing eigenvalues and the respective eigenvectors.
The PCs are linear combinations of features.
Not scaling and centering the features only underlines the magnitude of the features.





Data pre-processing. Feature engineering
Model robustness.



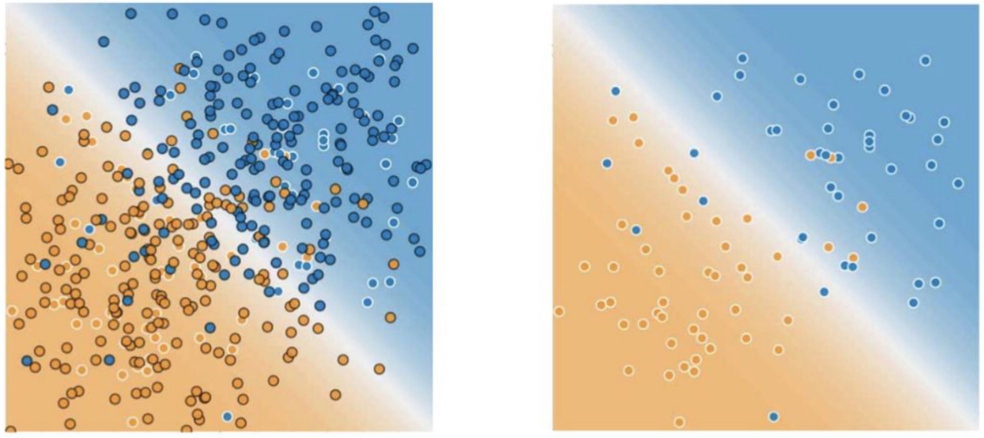
How do we ensure the model is **reliable/ robust**?

Do we have access to **independent** testing data?

Proportion of training/ test split

To ensure robustness (and optimize the model) – we perform this split several times.

The test set should act an independent evaluation of the model



Training Data   Test Data

Data pre-processing. Feature engineering
Model robustness.

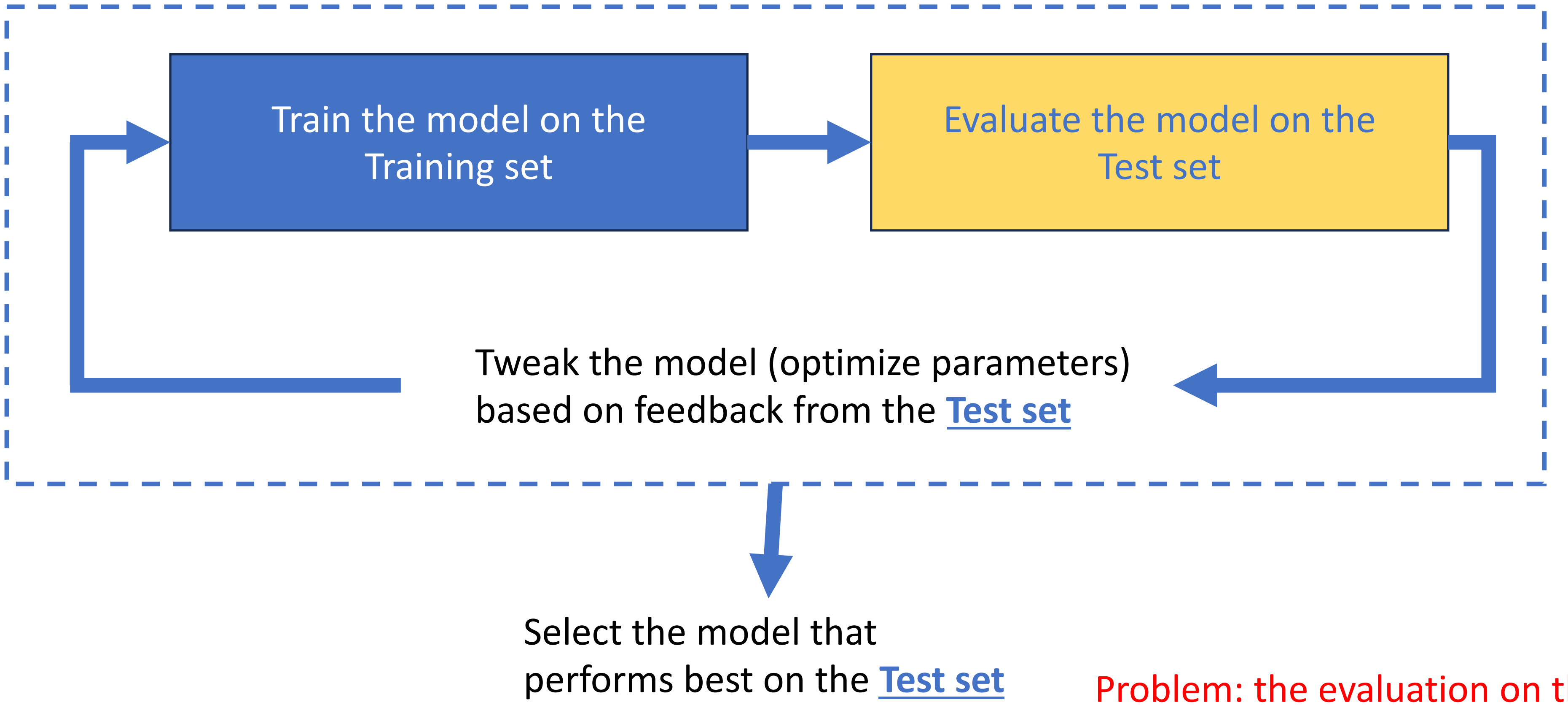
Assumptions for the training/ validation/ test sets

[a] we draw the samples **independently and identically** (*iid*) at random from the distribution

[b] the sets are **disjunct** partitions of the original distribution
i.e. no entries from the the training set will be found in the test set and vice versa

[c] the size of the validation and test sets should be comparable (if not identical)
The validation set should be large enough to detect differences between models
If classifier A has an accuracy of 95% and classifier B has an accuracy 95.1% then a validation set of 100 entries would not be sufficient/ able to detect the 0.1% difference.
A validation set with 1000 – 10000 entries might detect the improvement of 0.1%

Data pre-processing. Feature engineering
Model robustness.



Problem: the evaluation on the test set, after the model was optimized, is no longer unbiased (self-fulfilling prophecy).

Data pre-processing. Feature engineering Model robustness.

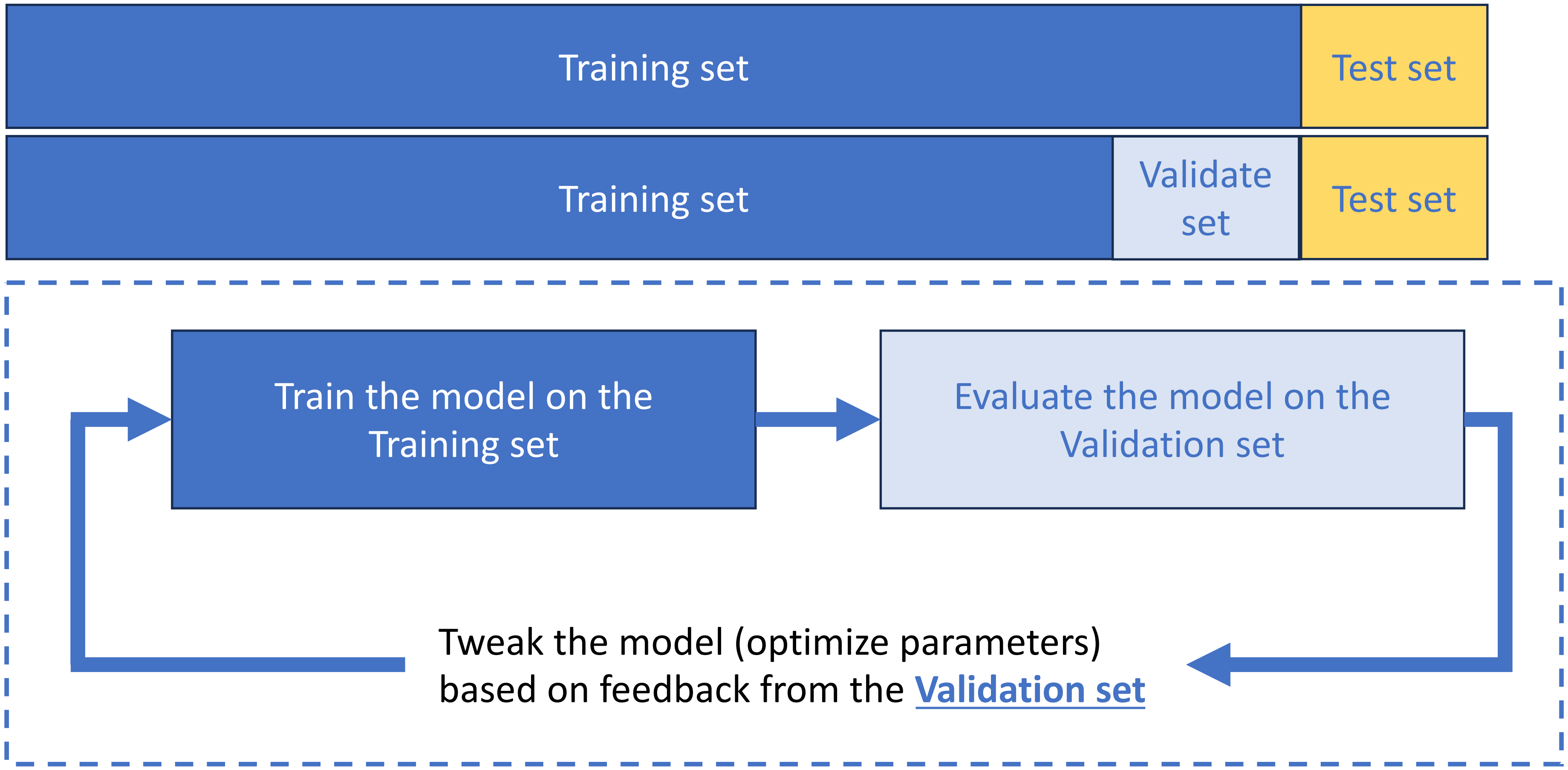
Assumptions for the training/ validation/ test sets

[d] on the test set, the error between the prediction and the actual label is the **test error**

[e] the **objective function** of the algorithm minimizes the test errors by **parameter tuning**

[f] Models are further evaluated for **Bias** and **Variance** (assessment of overfitting/ underfitting)

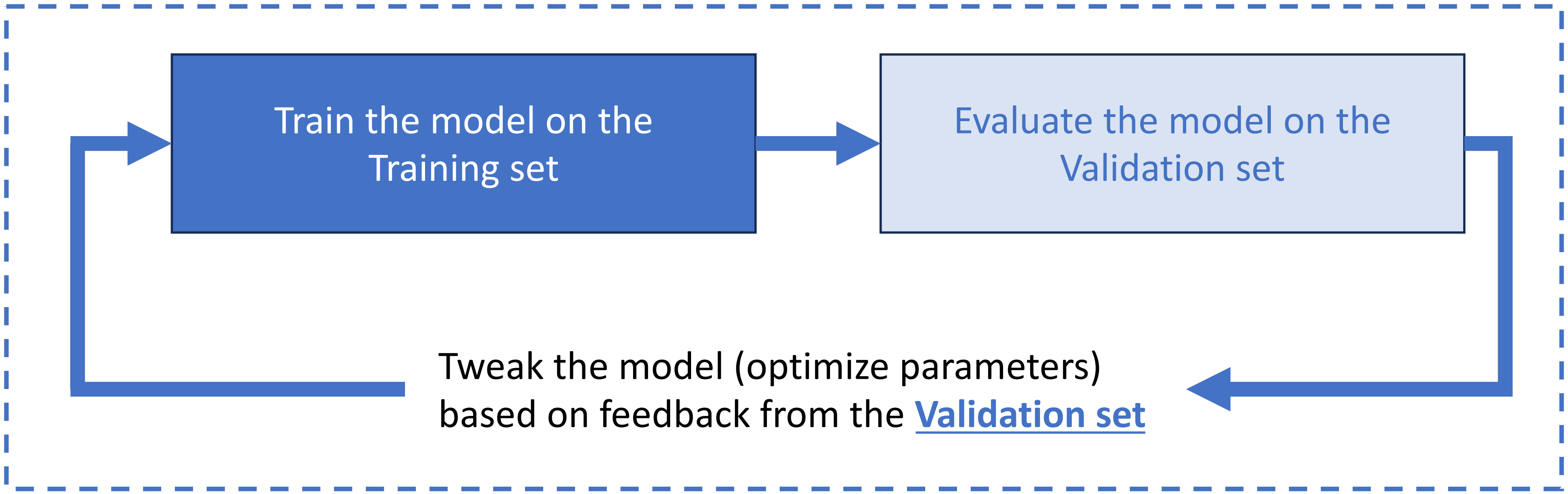
Data pre-processing. Feature engineering Model robustness.



Select the model that performs best on the **Validation set** → Confirm the model on the **Test set**

Advantage: the test set is uncorrupted and the evaluation is unbiased

Data pre-processing. Feature engineering Cross validation.



Select the model that performs best on the **Validation set** → Confirm the model on the **Test set**

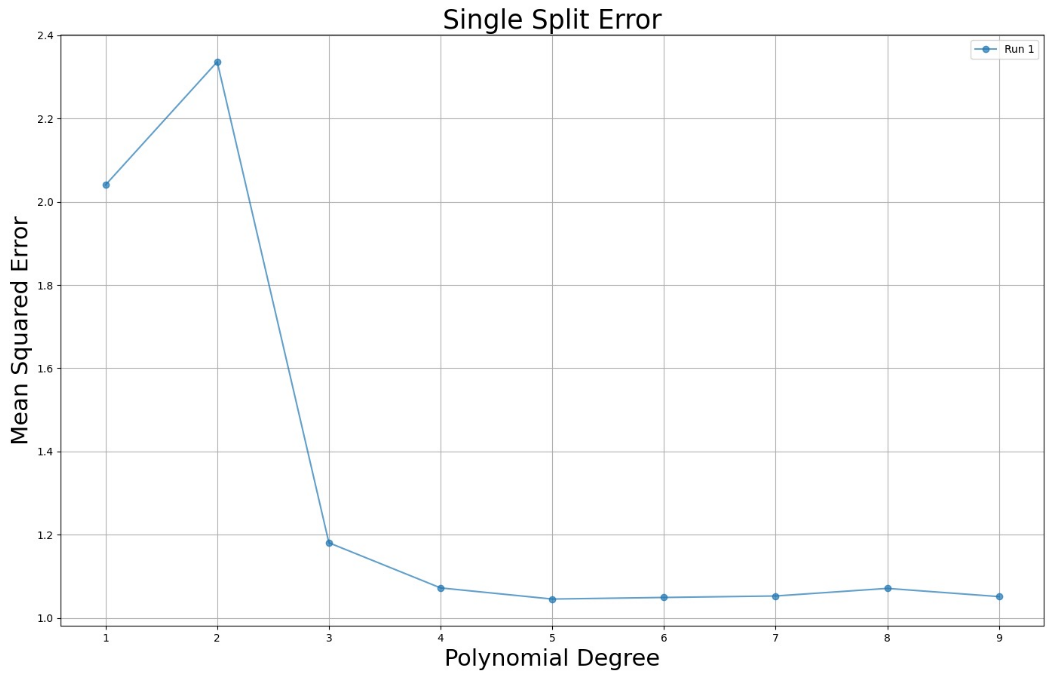
A systematic approach would rely on cross validation.

Data pre-processing. Feature engineering Cross validation.

In: 1, 2, 3, …. n

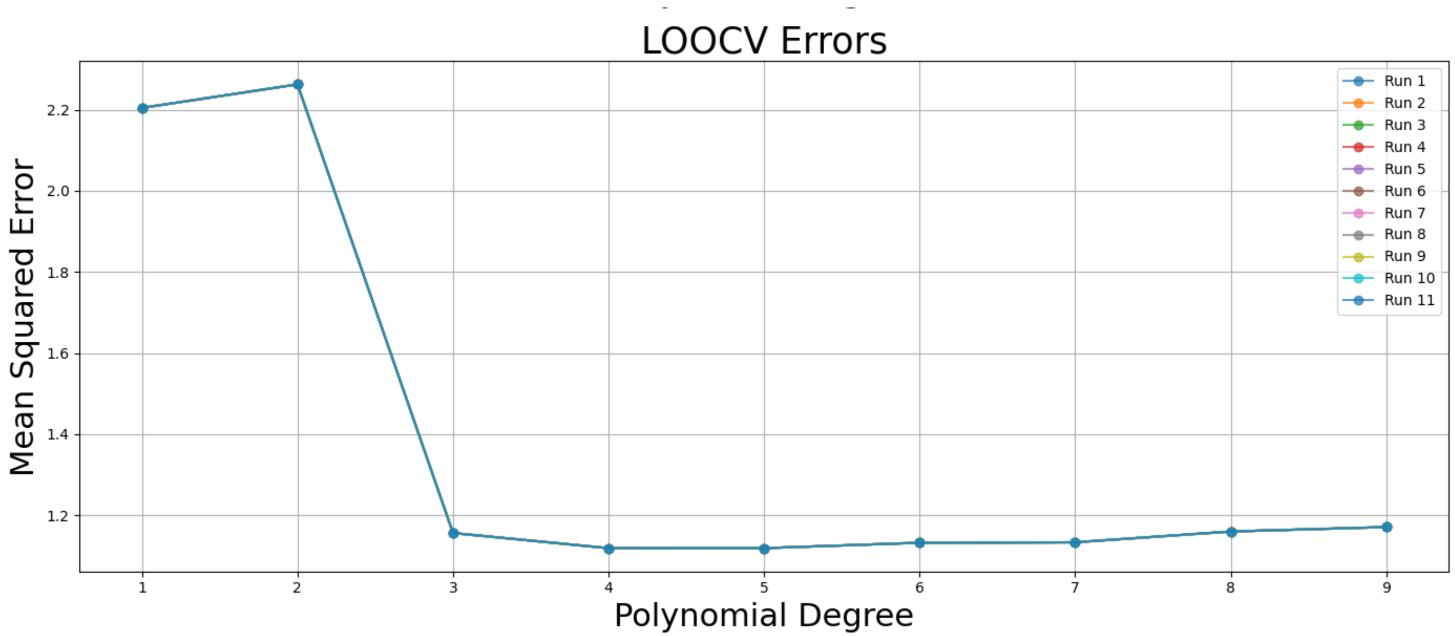
Out: 2, 9, 28, …. n

Task: infer a function that models the output wrt the input.



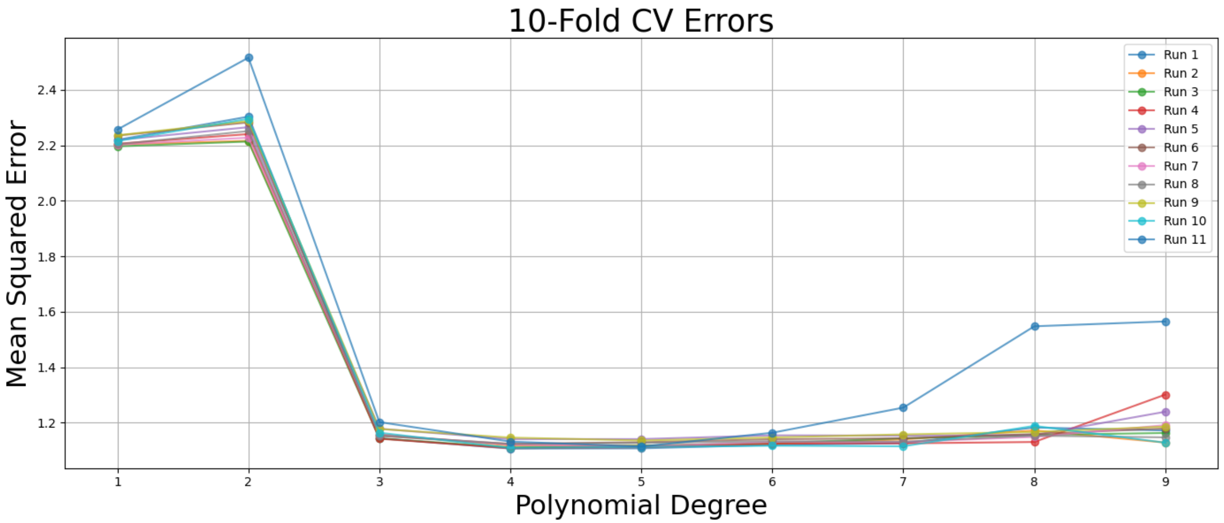
Data pre-processing. Feature engineering Cross validation.

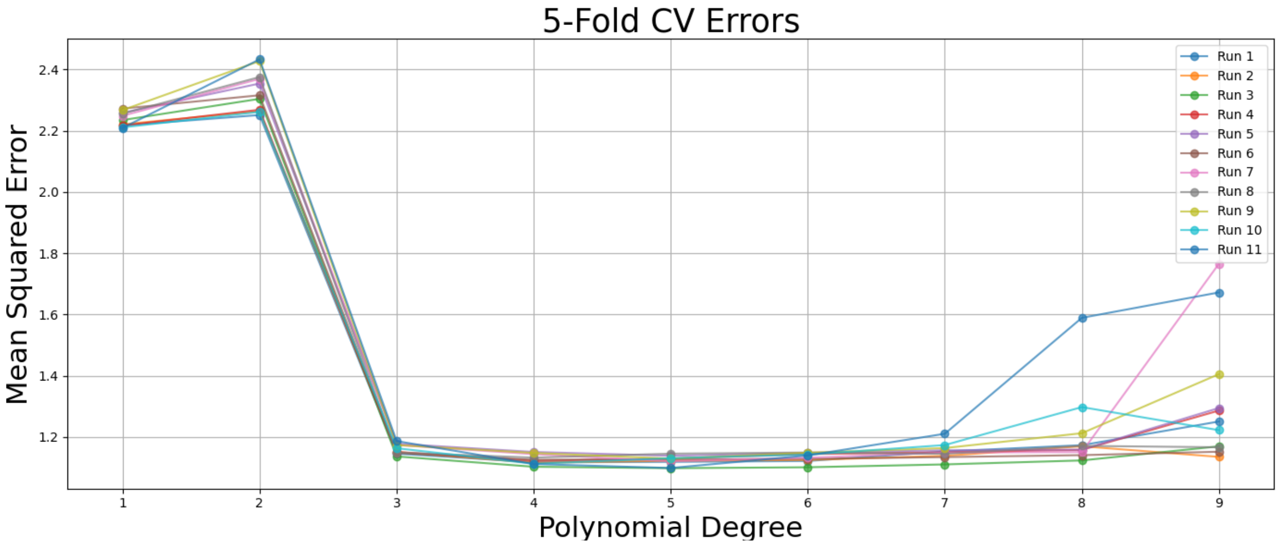
Leave on out CV separates one entry at a time.



A validation on one entry is meaningless.
The validation set is too small.

Data pre-processing. Feature engineering
Cross validation.





Data pre-processing. Feature engineering Cross validation. Evaluation

		Predicted condition	
	Total population = P + N	Predicted condition **positive** (PP)	Predicted condition **negative** (PN)
Actual Condition	Actual condition positive (P)	True positive **(TP)**	False negative **(FN)** – Type II error
	Actual condition negative (N)	False positive **(FP)** – Type I error	True negative **(TN)**

Remarks

[a] we don’t always have access to all information
e.g. the true negative set might be unknown
e.g. the false negative set might be unknown

[b] always strive to assess the model from multiple angles
i.e. don’t reply solely on one value such as accuracy.

[c] multiple class classification can be simplified to a 2x2 table

Data pre-processing. Feature engineering
Cross validation. Evaluation

		Predicted condition			
	Total population = P + N	Predicted condition **positive** (PP)	Predicted condition **negative** (PN)	Bookmaker informedness (BM) $= TPR + TNR - 1$	Prevalence threshold (PT) $= \frac{\sqrt{TPR \times FPR} - FPR}{TPR - FPR}$
Actual Condition	Actual condition positive (P)	True positive **(TP)**	False negative **(FN)** – Type II error	True positive rate (TPR) – recall – sensitivity (SEN), power $= \frac{TP}{P} = 1 - FNR$	False negative rate (FNR) $= \frac{FP}{P} = 1 - TPR$
	Actual condition negative (N)	False positive **(FP)** – Type I error	True negative **(TN)**	False positive rate (FPR) $= \frac{FP}{N} = 1 - TNR$	True negative rate (TNR), specificity (SPC) $= \frac{TN}{N} = 1 - FPR$
	Prevalence $= \frac{P}{P+N}$	Positive predictive value (PPV), precision $= \frac{TP}{PP}$ $=1 - FDR$	Negative predictive value (NPV) $= \frac{TN}{PN}$	Positive likelihood ratio (LR+) $= \frac{TPR}{FPR}$	Negative likelihood ratio (LR-) $= \frac{FNR}{TNR}$
	Accuracy (ACC) $= \frac{TP+TN}{P+N}$	F1 score $= \frac{2TP}{2TP+FP+FN}$	False discover rate (FDR) $= \frac{FP}{PP}$ $=1 - PPV$		Threat score (TS), critical success index (CSI) $= \frac{TP}{TP+FN+FP}$

Data pre-processing. Feature engineering Cross validation. Evaluation

Sources of error in ML: bias and variance

Bias: the model's error rate on the training set

Variance: the model's error rate on the validation (or test) set, in addition to the bias

Training error: 0.5% [bias]
Validation error: 1% (variance = 0.5%)
Perfect model

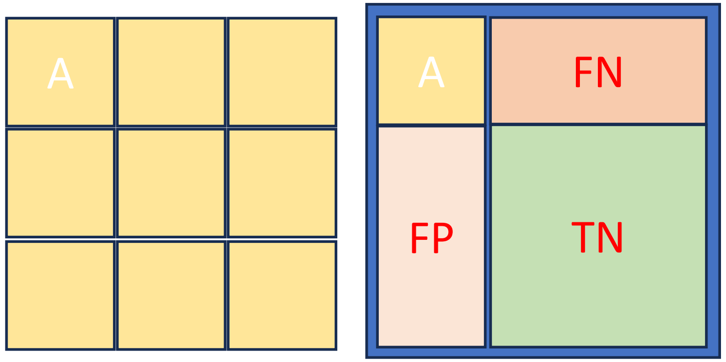
Training error: 15% [bias]
Validation error: 16% (variance = 1%)
Underfitting – learning some signal

Training error: 1% [bias]
Validation error: 11% (variance = 10%)
Overfitting – learning signal and noise

Training error: 15% [bias]
Validation error: 30% (variance = 15%)
High bias, high variance
Model unlikely suitable for the data.

Data pre-processing. Feature engineering
Evaluation. Confusion Matrices

Prediction	Reference: Adelie	Chinstrap	Gentoo
Adelie	36	10	0
Chinstrap	7	10	0
Gentoo	0	0	35



2x2 confusion matrix for species: Adelie

	Predicted Positive	Predicted Negative
Actual Positive	36	7
Actual Negative	10	45

2x2 confusion matrix for species: Chinstrap

	Predicted Positive	Predicted Negative
Actual Positive	10	10
Actual Negative	7	71

2x2 confusion matrix for species: Gentoo

	Predicted Positive	Predicted Negative
Actual Positive	35	0
Actual Negative	0	63

Data pre-processing. Feature engineering Evaluation. Confusion Matrices

	Adelie	Chinstrap	Gentoo
prevalence	0.4387755	0.2040816	0.3571429
accuracy	0.8265306	0.8265306	1
F1	0.8089888	0.5405405	1
PPV	0.7826087	0.5882353	1
NPV	0.8653846	0.8765432	1
FDR	0.7826087	0.5882353	1
TPR	0.8372093	0.5	1
FPR	0.1818182	0.08974359	0
FNR	0.2325581	0.35	0
TNR	0.8181818	0.9102564	1
BM	0.6553911	0.4102564	1
LR_pos	4.604651	5.571429	Inf
LR_neg	0.2842377	0.384507	0
PT	0.3178796	0.2975847	0
TS	0.6792453	0.3703704	1

	Reference		
Prediction	Adelie	Chinstrap	Gentoo
Adelie	36	10	0
Chinstrap	7	10	0
Gentoo	0	0	35

Note: check slide 28 for calculations

Statistics for a 2x2 confusion matrix - Adelie

		Predicted condition			
	Total population = P + N	Predicted condition **positive** (PP)	Predicted condition **negative** (PN)	Bookmaker informedness (BM) 0.66	Prevalence threshold (PT) 0.32
Actual Condition	Actual condition positive (P)	True positive **(TP)** 36	False negative **(FN)** – Type II error 7	True positive rate (TPR) – recall – sensitivity (SEN), power 0.84	False negative rate (FNR) 0.23
	Actual condition negative (N)	False positive **(FP)** – Type I error 10	True negative **(TN)** 45	False positive rate (FPR) 0.18	True negative rate (TNR), specificity (SPC) 0.82
	Prevalence 0.43	Positive predictive value (PPV), precision 0.78	Negative predictive value (NPV) 0.87	Positive likelihood ratio (LR+) 4.6	Negative likelihood ratio (LR-) 0.28
	Accuracy (ACC) 0.83	F1 score 0.81	False discover rate (FDR) 0.21		Threat score (TS), critical success index (CSI) 0.68

Data pre-processing. Unbalanced data. Sampling?

It can be a struggle to gather enough **balanced** data for a machine learning project.

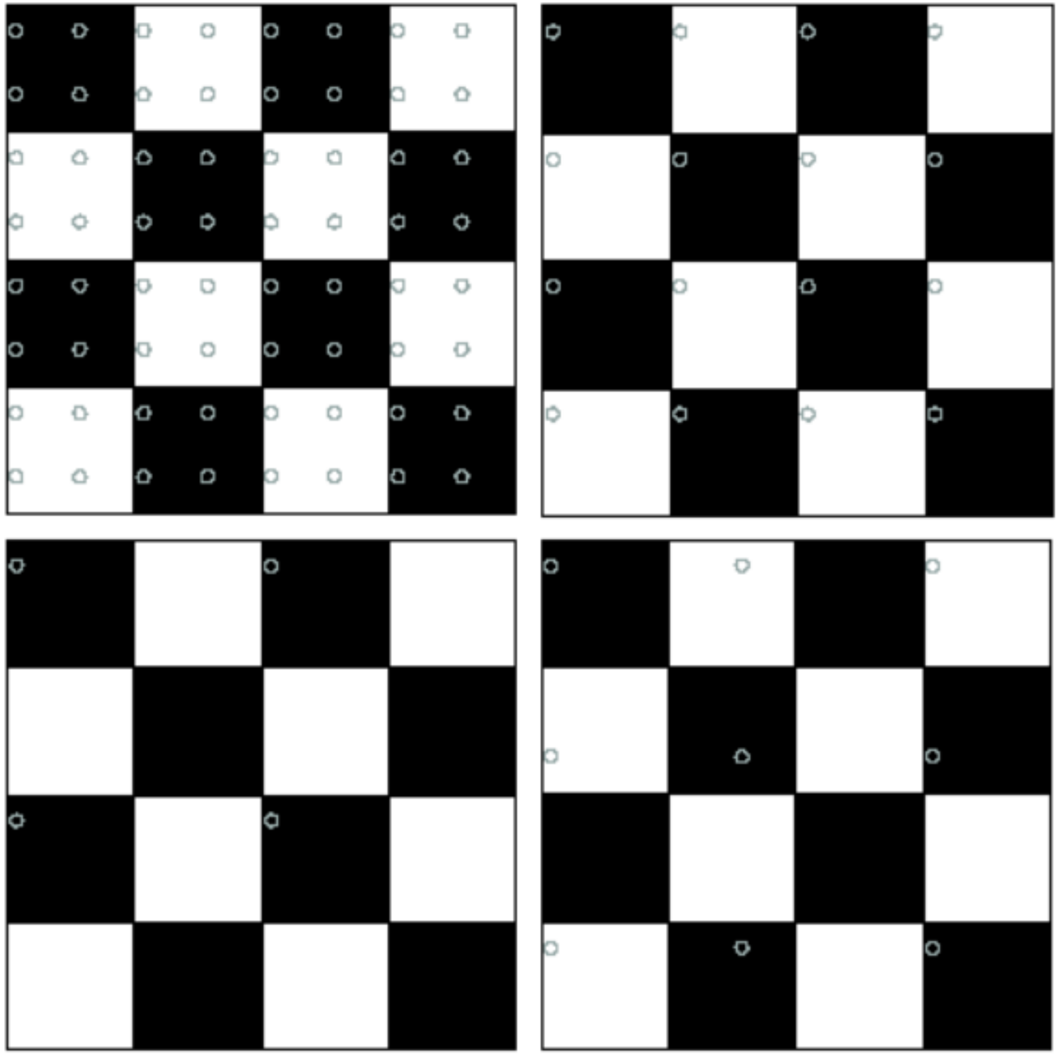
Examples of issues:
Negative examples are easier to sample than positive ones.
e.g. negative examples of galaxies outweigh the number of positive/ confirmed examples
Another angle: not all galaxies that exist were identified [*we don't know what we don't*]

The intrinsic structure of the data confounds classes
e.g. on classes A, B, C driven by the features of the data,
Class A comprises only positive examples, Class B comprises a 20/80 mix, Class C comprises a 50/50 mix

Down sampling = extracting a number of entries from the larger class
Up weighting = adding weights to entries, essentially repeating them.

Issues:
The resulting dataset must recapitulate the properties of the original dataset. [Invariance requirement]
In particular cases, a middle ground between down sampling and up weighting works better.

Data pre-processing. Unbalanced data. Sampling?



Good sampling

Bad sampling
i.e. non representative

<u>The Nyquist–Shannon sampling theorem</u> is an essential principle for digital signal processing linking the frequency range of a signal and the sample rate required to avoid a type of distortion called **<u>aliasing</u>**.
The theorem states that the sample rate must be at least twice the bandwidth of the signal to avoid aliasing distortion.
In practice, it is used to select band-limiting filters to keep aliasing distortion below an acceptable amount when an analog signal is sampled or when sample rates are changed within a digital signal processing function.

For generic distributions, test such as the Kullback-Leiber divergence (per classes or using a binning approach for continuous data) are frequently used.

Next Lecture …

Applied Data Science

L3. Data Science toolkit. Part 1