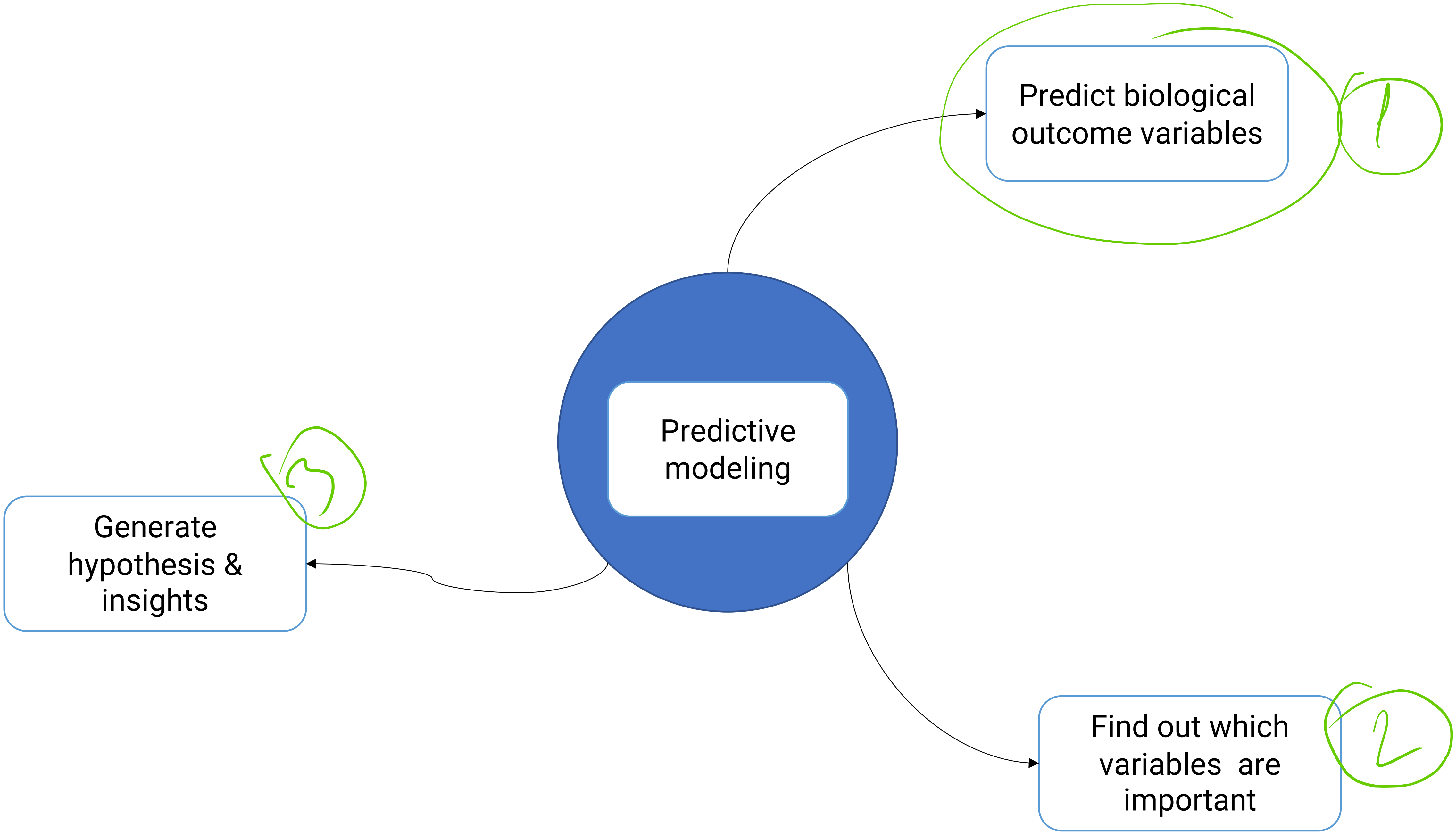
Computational genomics: hands on course

Predictive modeling with supervised learning



Predictive modeling

Predict biological outcome variables

Find out which variables are important

Generate hypothesis & insights

How are machine learning models fit ?

- Define a prediction function or method $\boxed{f(X)}$ → LR SVM RF

- Devise a loss/cost function: such as $\sum(\underset{\text{org}}{Y}-\underset{\text{pred.}}{f(X)})^2$

- Apply optimization & find best parameters for $\underbrace{\sum(Y-f(X))^2}$

$\boxed{f(x)}$

$\boxed{X-WH}$

k-means

cost ∝ $\boxed{TSS}$ cluster

Steps for supervised learning

1) Pre-processing data → transform
→ filter
→ normalize

2) Data split → Training & test

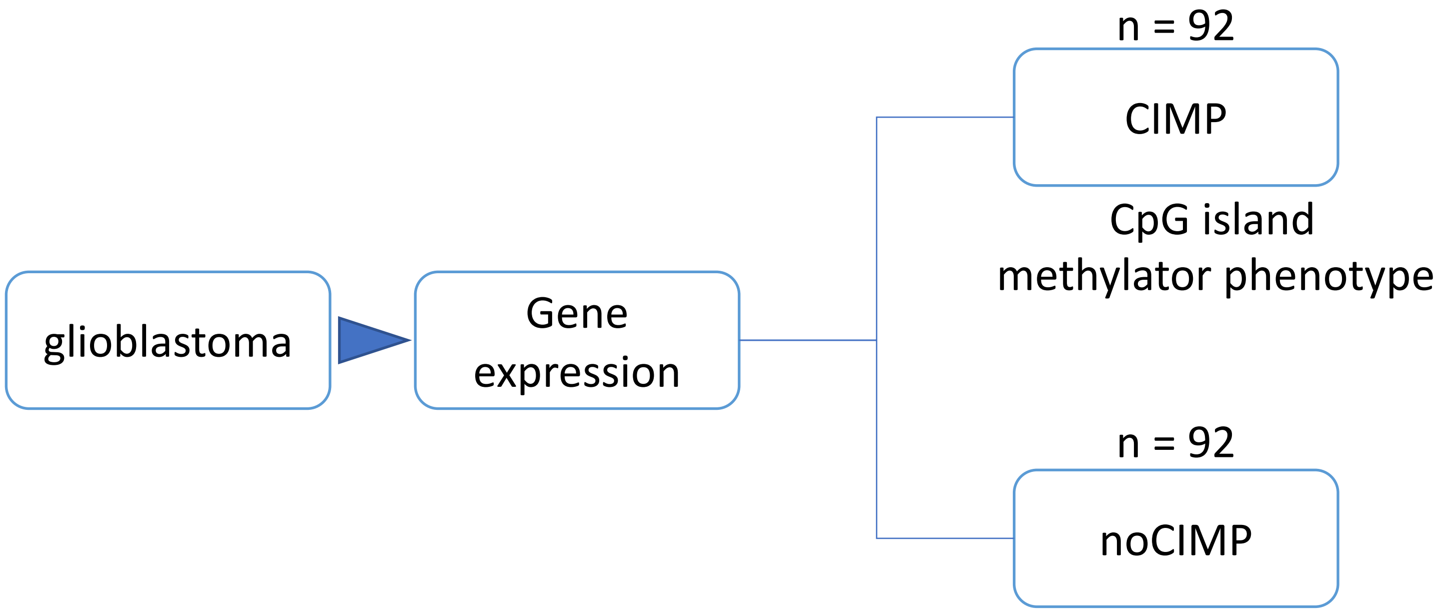
3) Training → specific model / algo.

4) Performance estimation → error

5) Model tuning → to further optimize parameters

R packages: mlr and caret

Use case: Disease subtype from genomics data

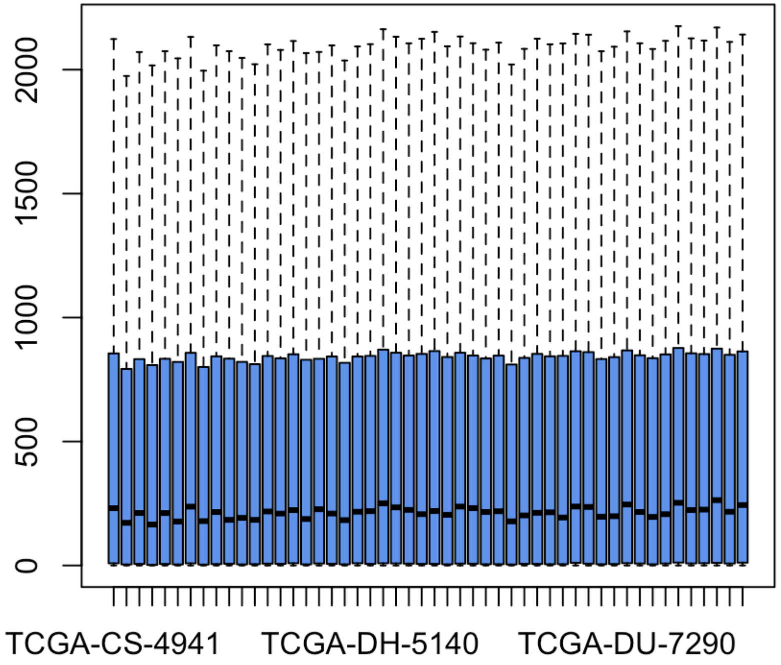


Data pre-processing

Data transformation

Systematic differences between samples and outliers are a problem for fitting ML models

Gene expression values from glioblastoma samples



Data pre-processing

Data transformation

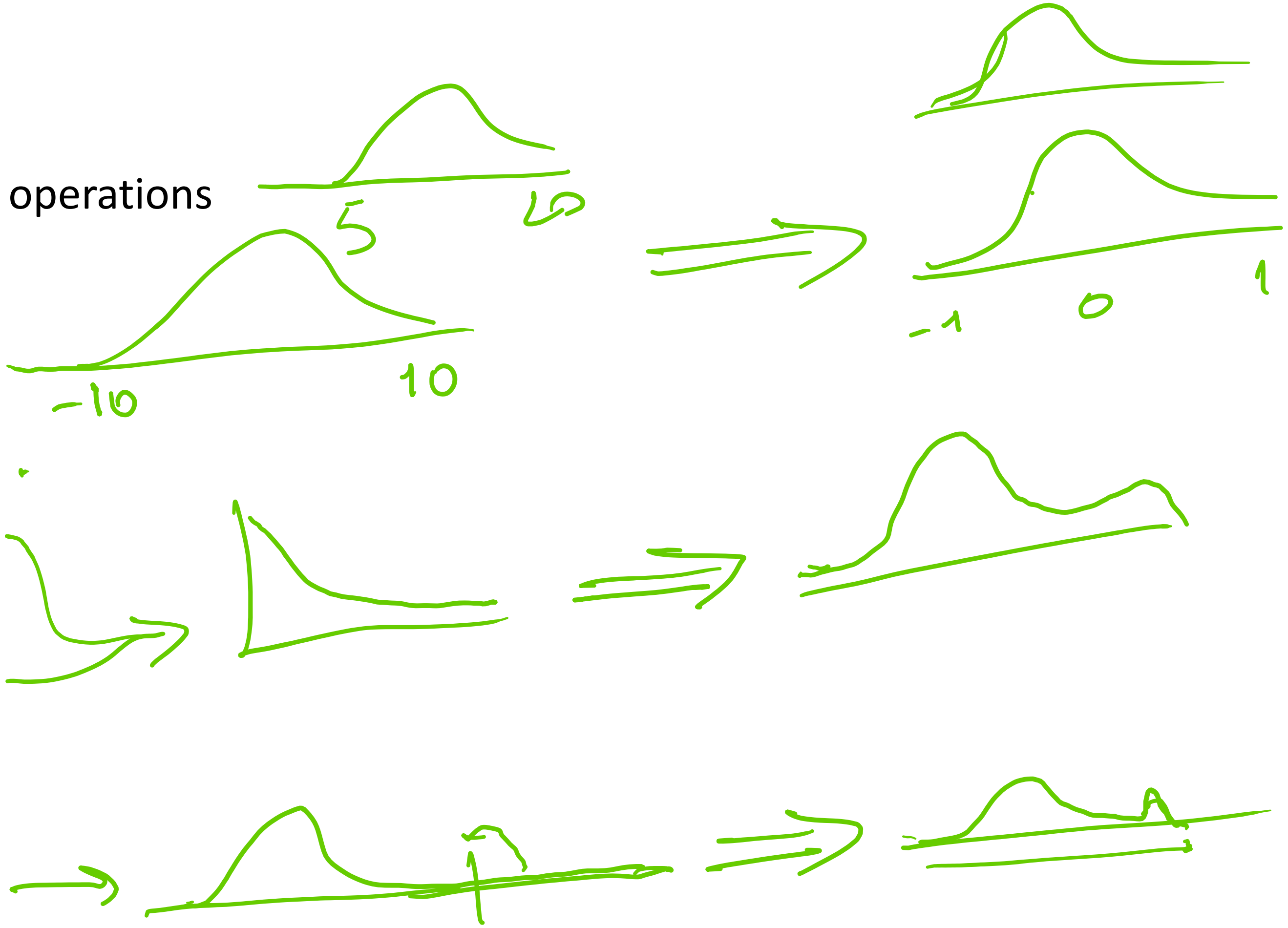
Some of the data transformation operations

Normalization /scaling

Log transform

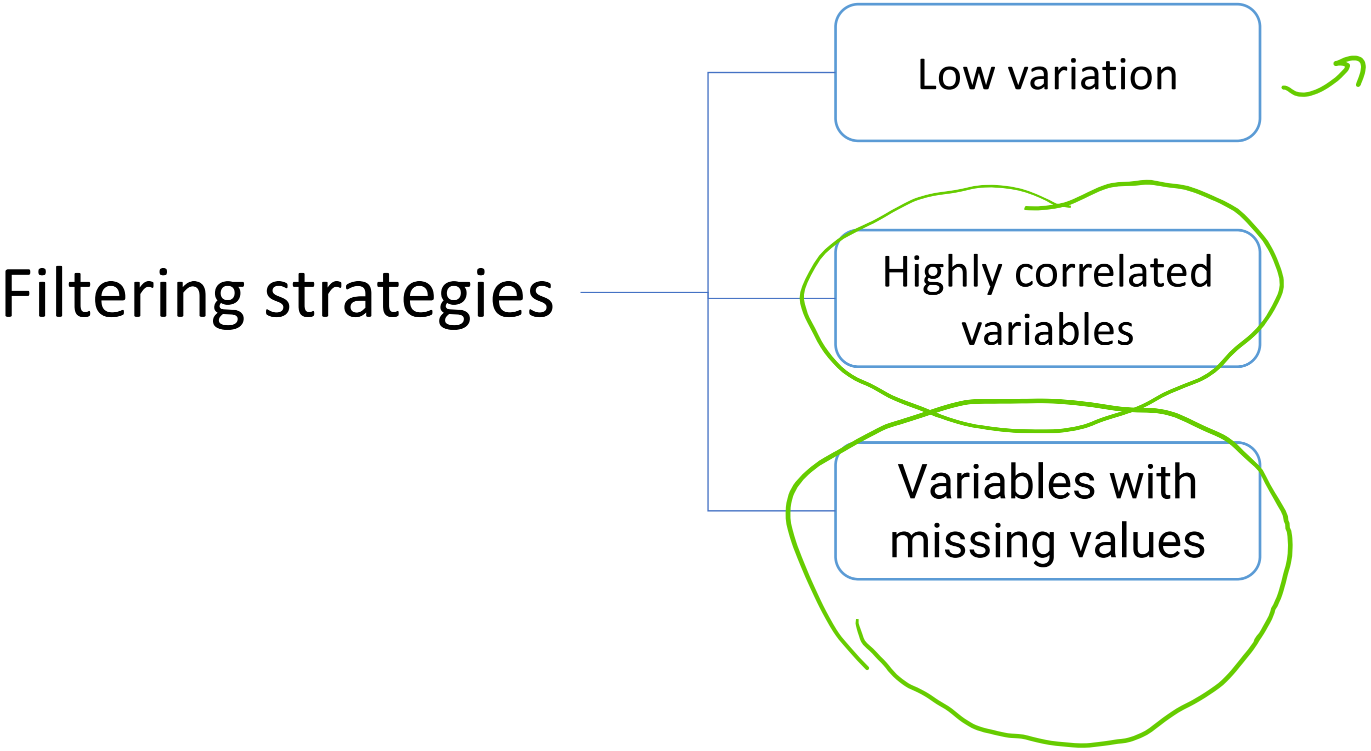
Box Cox transform

winsorizing



Data pre-processing

Data filtering

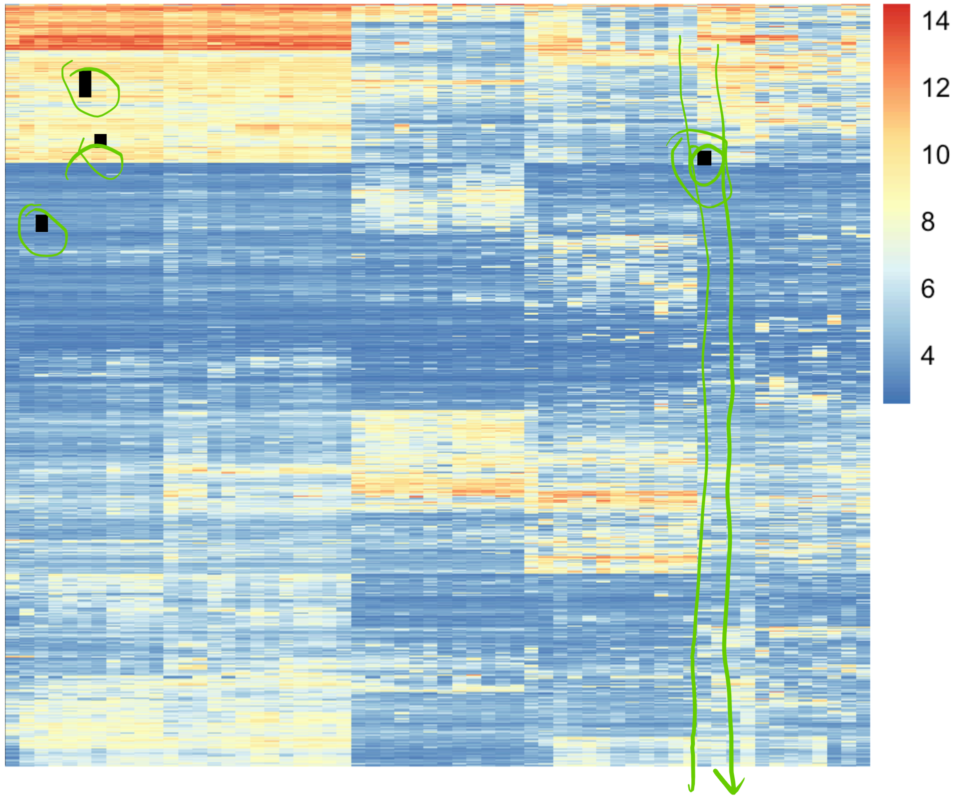


Filtering strategies

- Low variation → Remove
- Highly correlated variables
- Variables with missing values

Data pre-processing

Dealing with missing values



Choices:

1) Remove samples/variables with missing values

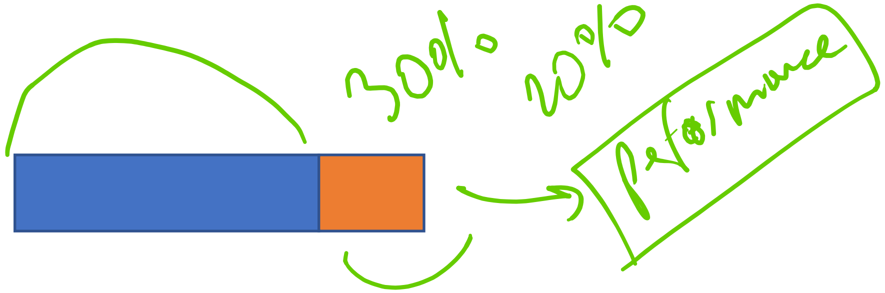
2) Assign the mean/median value

3) Try to predict missing values

```
knnImpute=preProcess(missing_tgexp,method="knnImpute")
```

Data split

Holdout test dataset

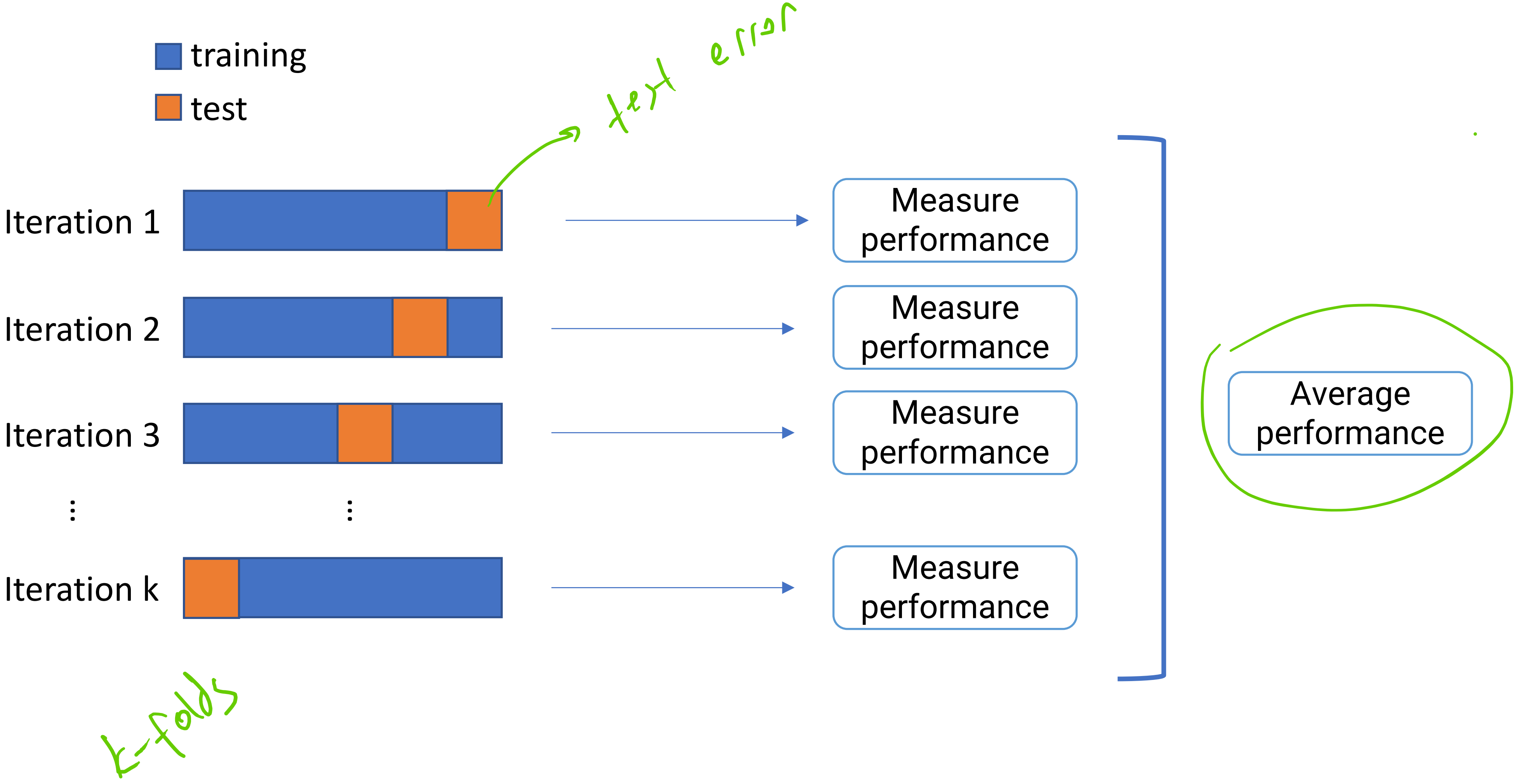


training

Holdout test

Data split

cross-validation



🟦 training

🟧 test

test error

Iteration 1

Iteration 2

Iteration 3

⋮

Iteration k

Measure performance

Measure performance

Measure performance

Measure performance

Average performance

k-folds

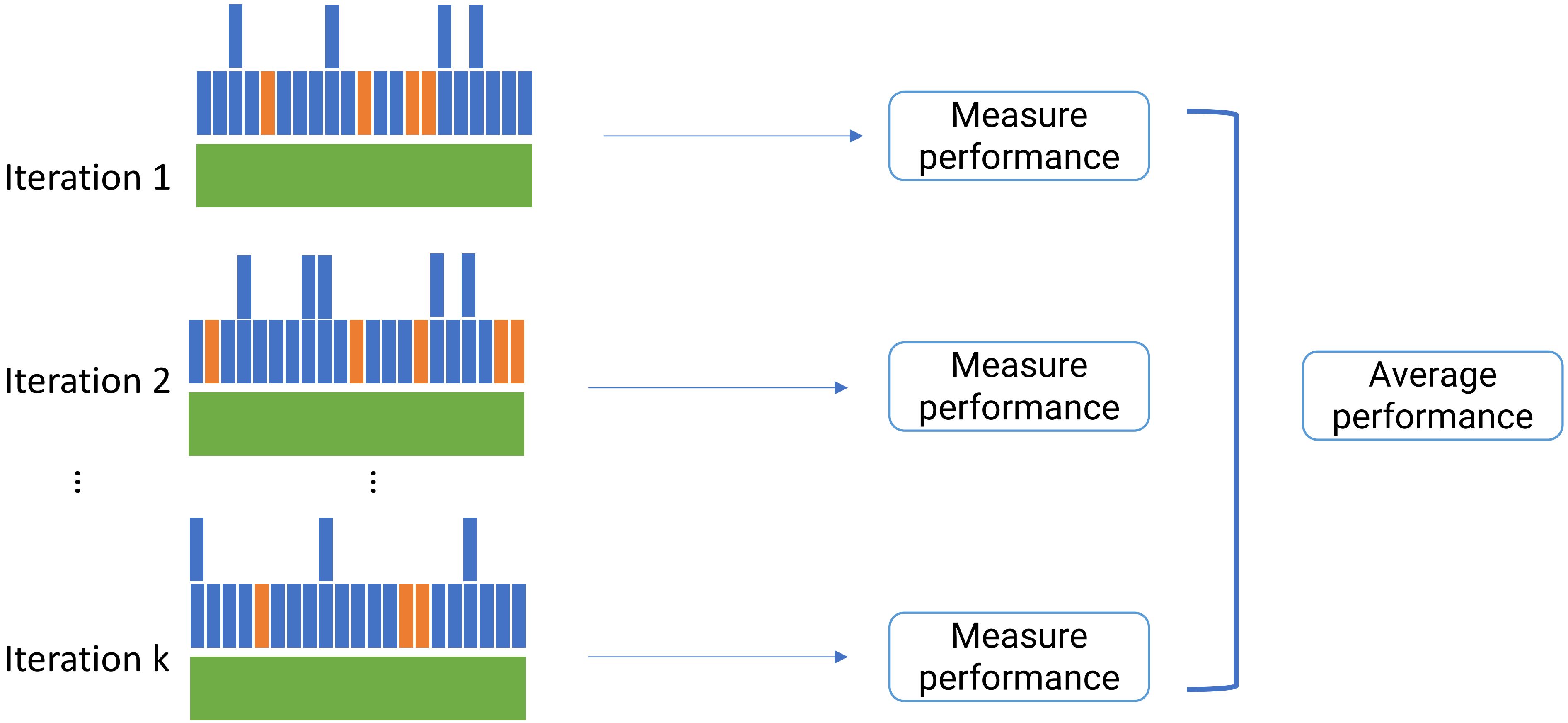
Data split

bootstrap resampling

🟦 Training (bootstrap sample)

🟧 Test: out-of-the-bag (OOB)

🟩 All the data



Iteration 1

Iteration 2

⋮

Iteration k

Measure performance

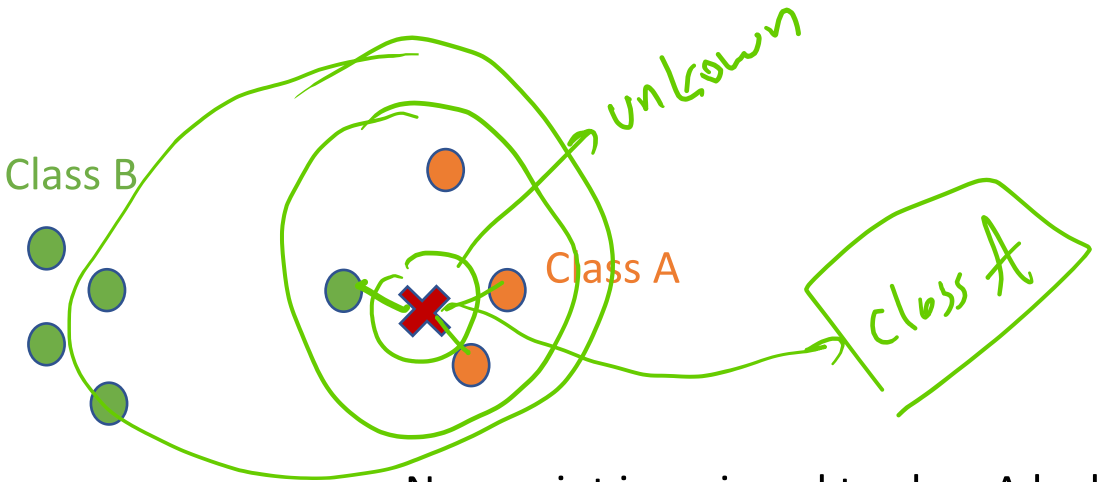
Measure performance

Measure performance

Average performance

Predicting the subtype with k-nearest neighbors

k-NN in a nutshell: find similar patients and use their labels

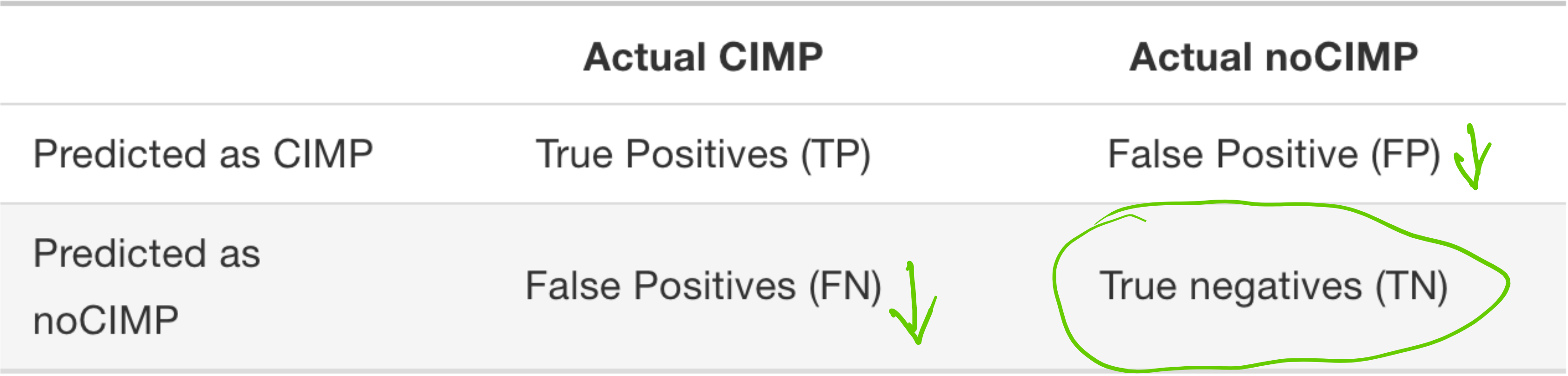


Class B

Class A

New point is assigned to class A by k-NN where k=3

Assessing the performance of our model



	Actual CIMP	Actual noCIMP
Predicted as CIMP	True Positives (TP)	False Positive (FP) ↓
Predicted as noCIMP	False Positives (FN) ↓	True negatives (TN)

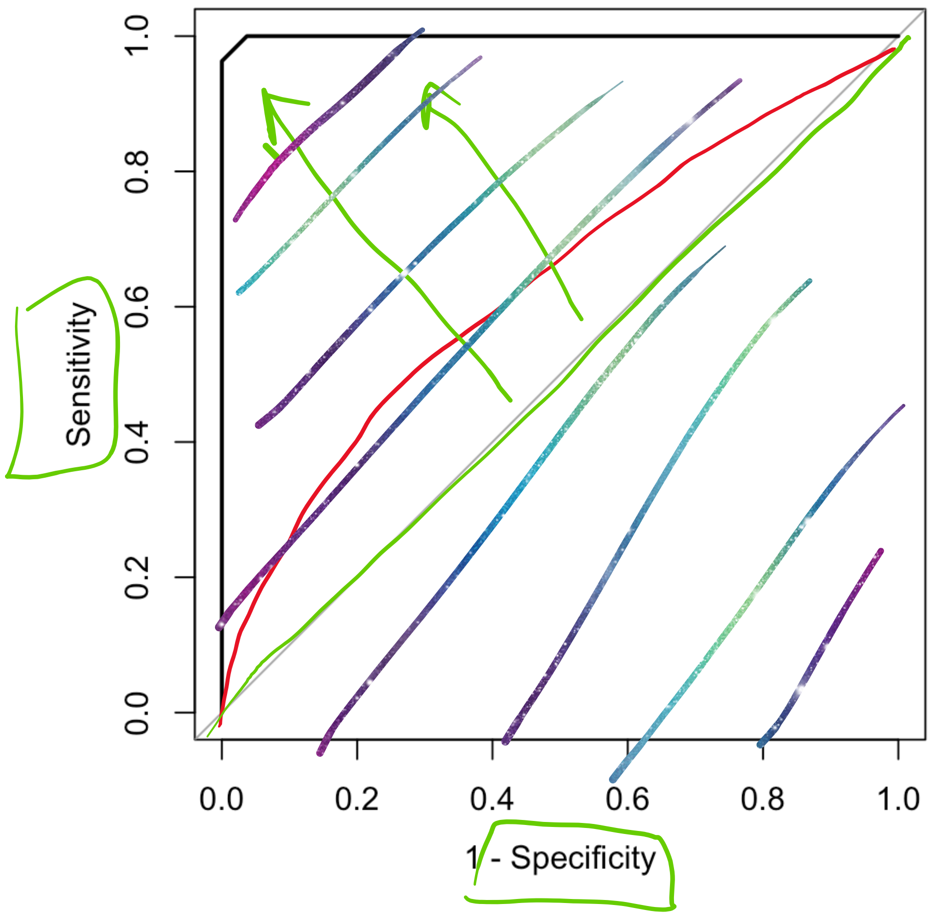
1 Precision, $TP/(TP+FP)$

2 Sensitivity, $TP/(TP+FN)$ ↓

3 Specificity, $TN/(TN+FP)$ ↓

Assessing the performance of our model

Receiver Operating Characteristic (ROC) Curves



(0,5) = class CIMP

[0,1]

↓

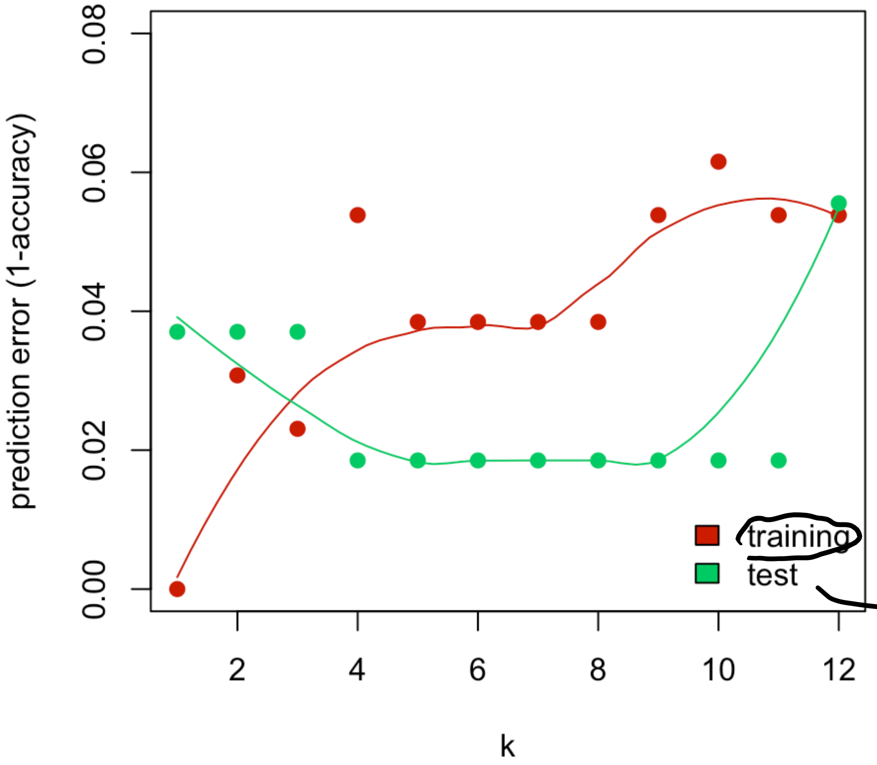
0,5

0,5 →

↓

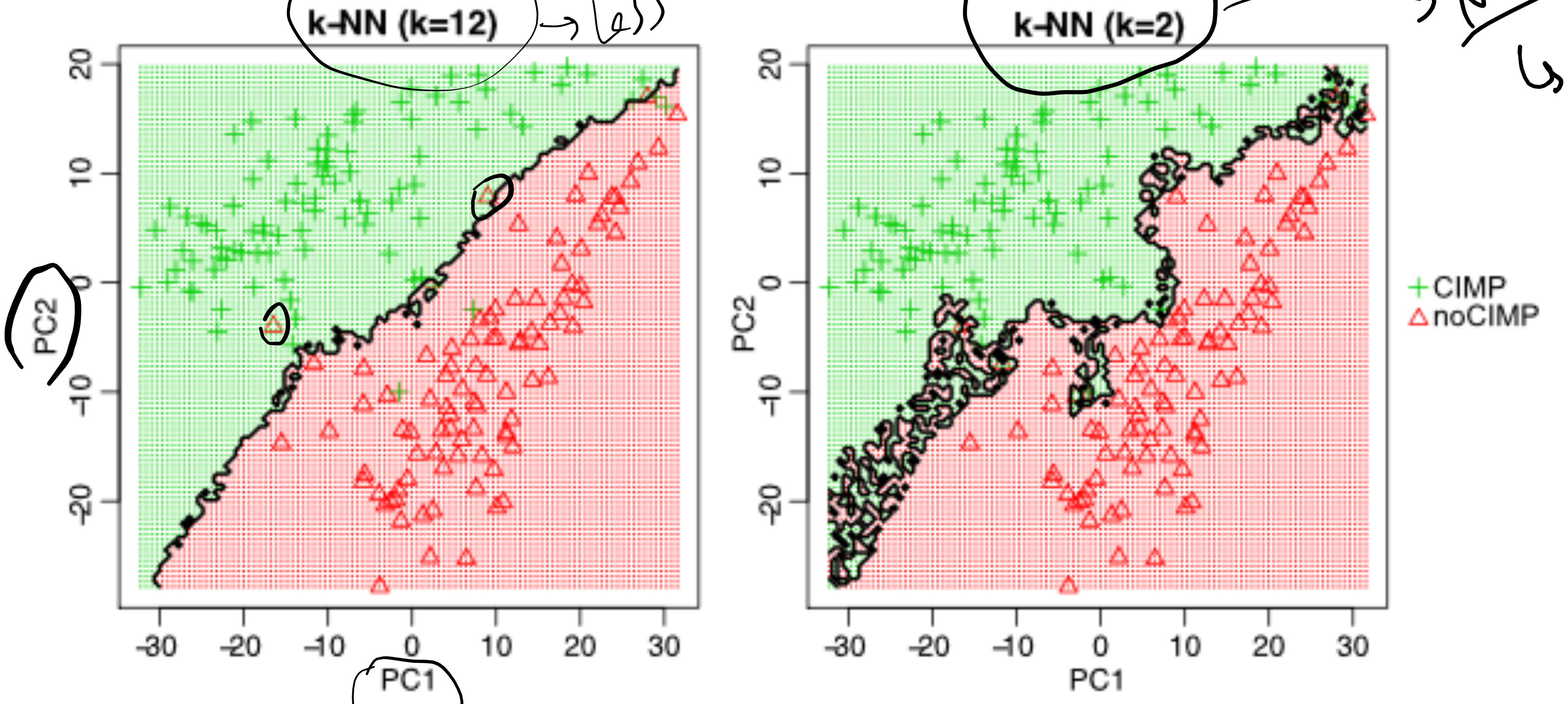
0,3

Model tuning and avoiding overfitting



Holdout

Model tuning and avoiding overfitting



less flex.

→ less variable.

→ flexible ~ complexity

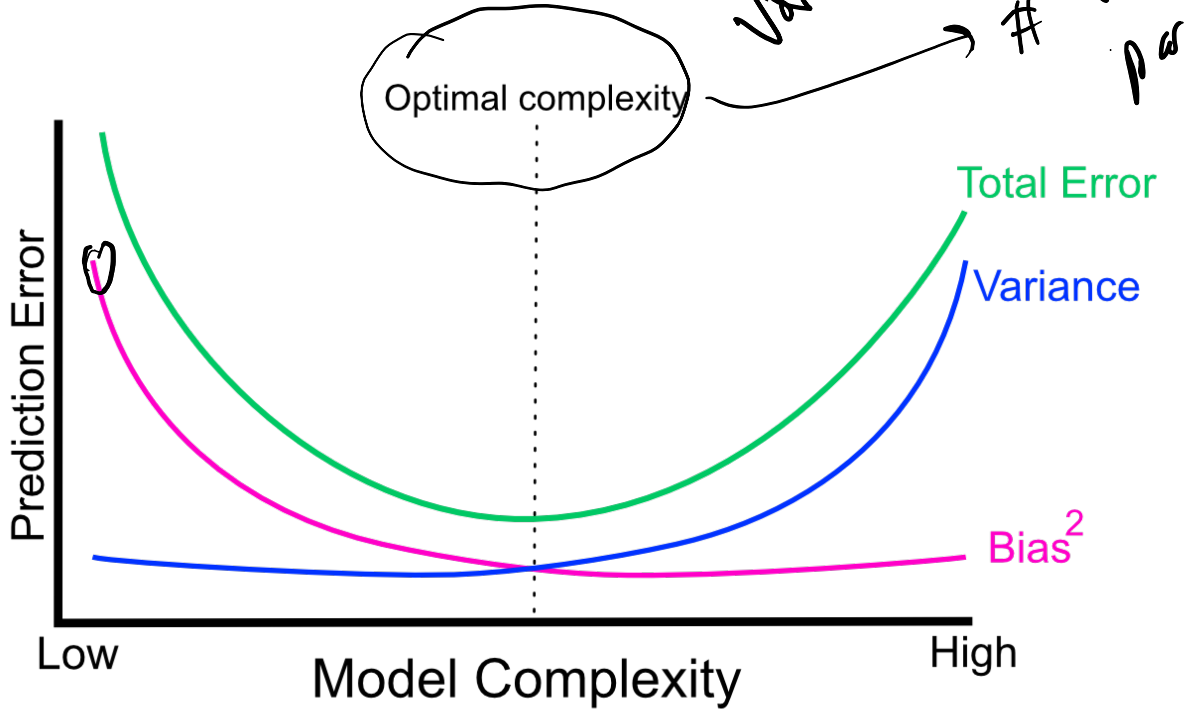
variable

↳ overfit

Model tuning and avoiding overfitting

↓ decrease complexity

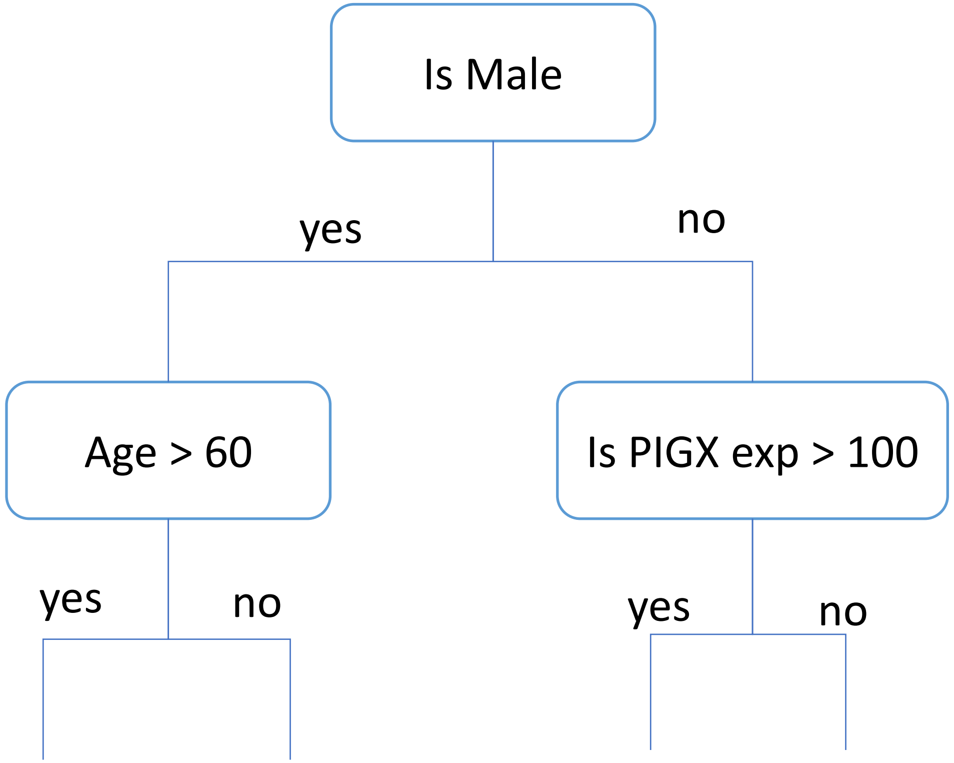
$$E[Error] = \underbrace{Bias^2}_{} + \underbrace{Variance}_{Var(prediction)} + \sigma_e^2$$



→ # variables parameters

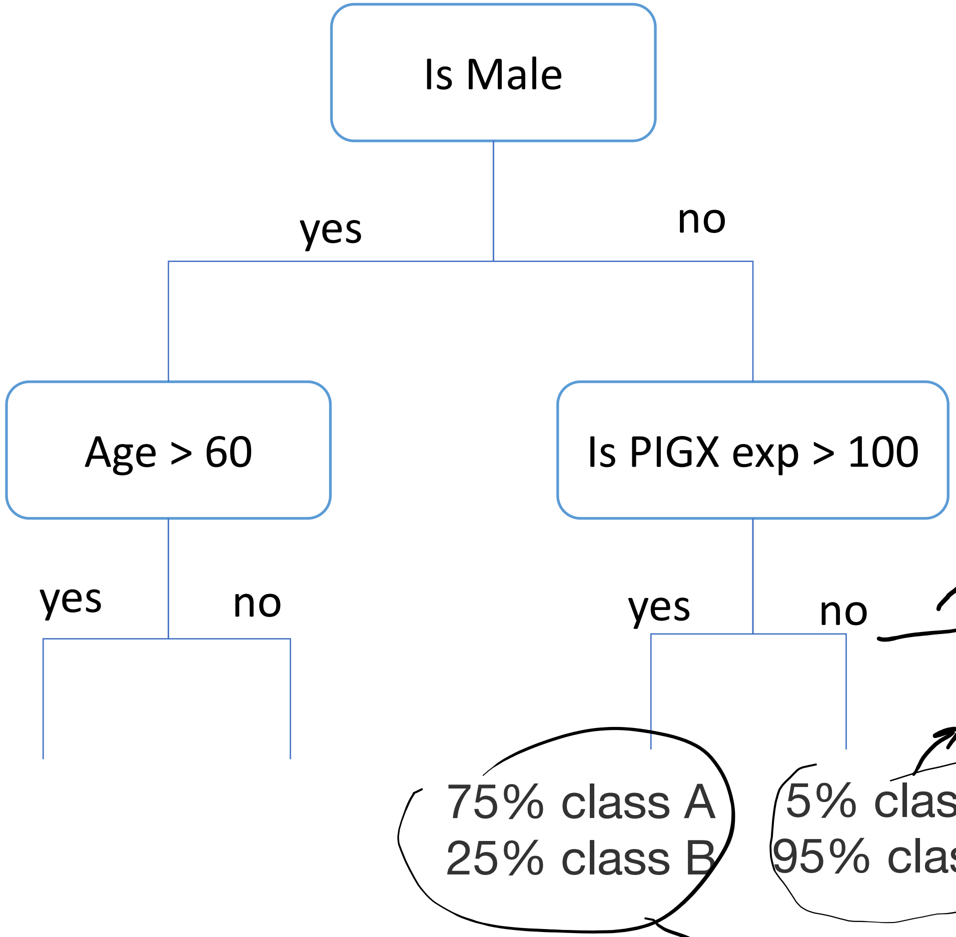
Trees and forests

Decision trees



Trees and forests

Decision trees



Gini Impurity 0 ↔ 0,5

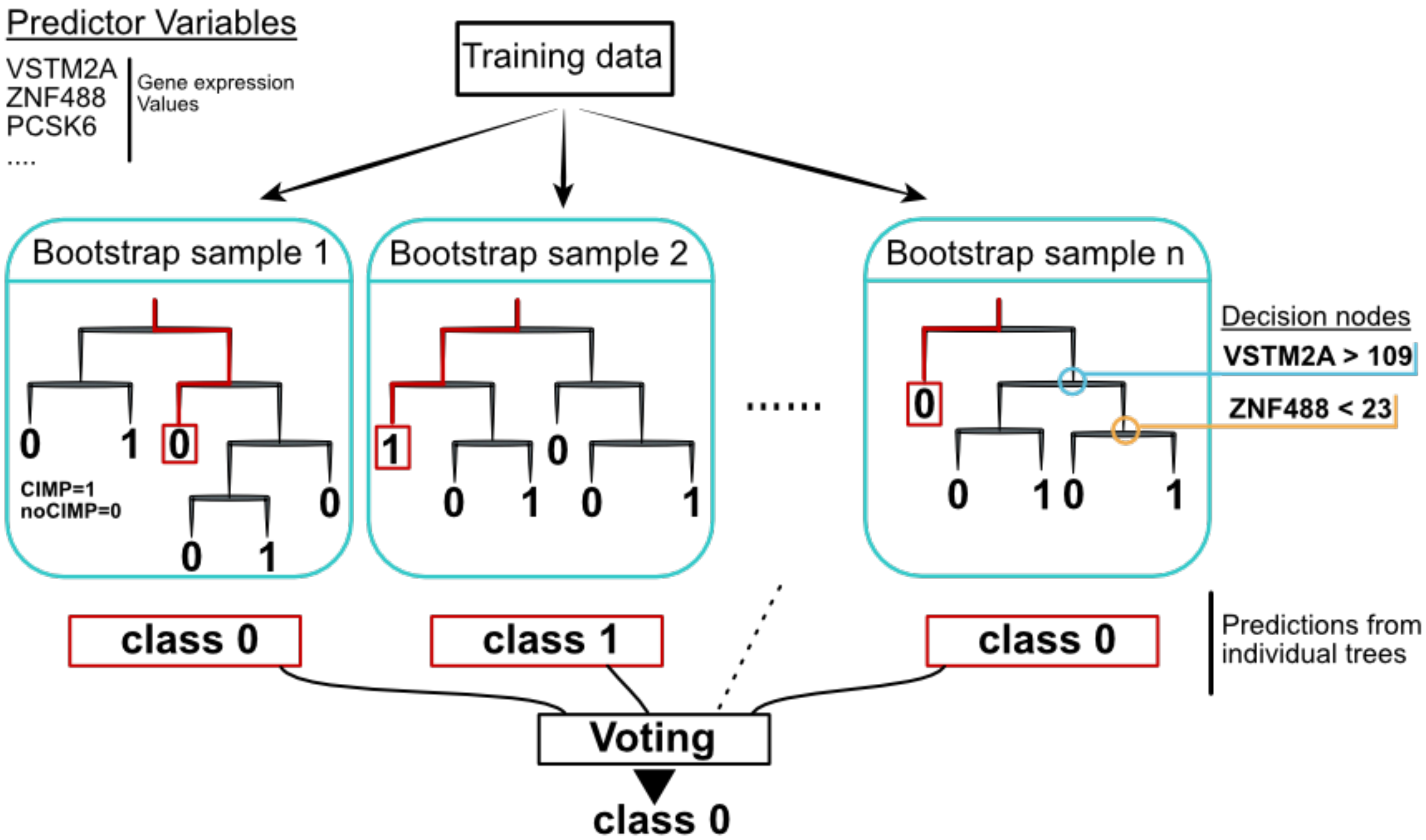
0,235

0,095

$$I_G(p) = \sum_{i=1}^{K} p_i(1-p_i) = \sum_{i=1}^{K} p_i - \sum_{i=1}^{K} {p_i}^2 = \boxed{1 - \sum_{i=1}^{K} {p_i}^2}$$

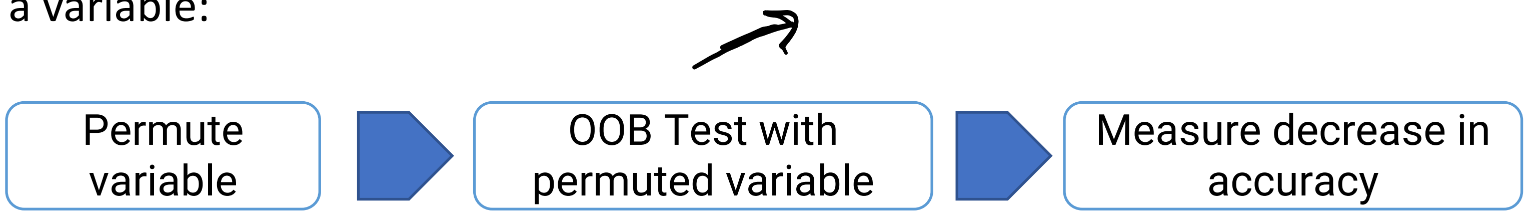
$$1 - (0{,}75^2 + 0{,}25)^2 = 0{,}375$$

Trees and forests:
Random Forests



Variable importance for RF

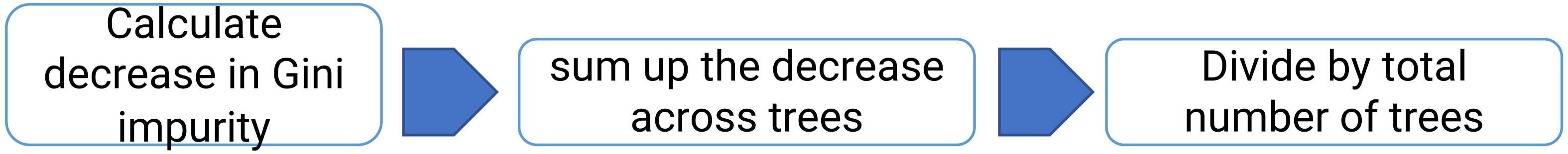
Given a variable:



Permute variable

OOB Test with permuted variable

Measure decrease in accuracy



Calculate decrease in Gini impurity

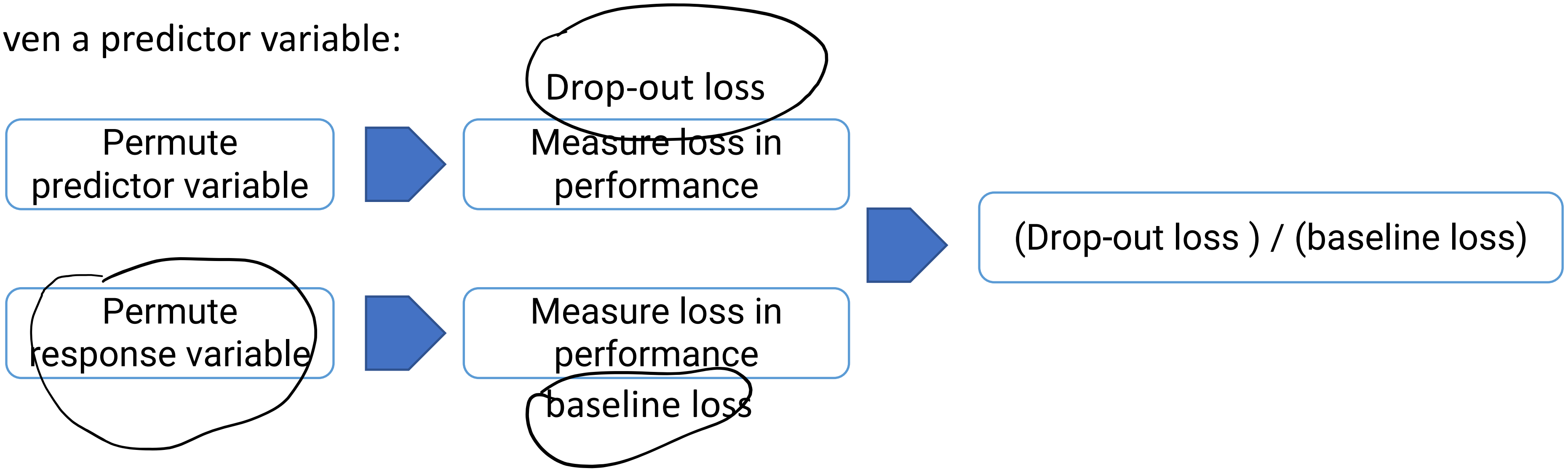
sum up the decrease across trees

Divide by total number of trees

$$I = \underbrace{G_{parent}} \underbrace{- G_{split1} - G_{split2}}$$

Variable importance
Method agnostic

Given a predictor variable:



Drop-out loss

Permute predictor variable

Measure loss in performance

Permute response variable

Measure loss in performance

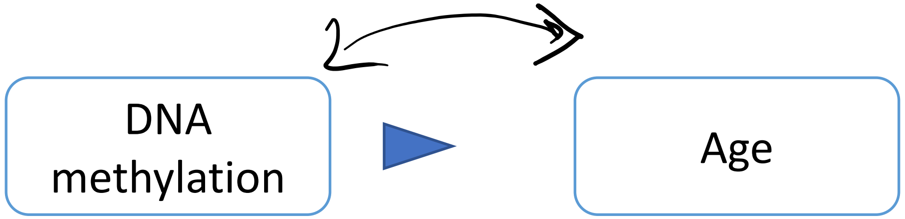
baseline loss

(Drop-out loss) / (baseline loss)

DALEX package in R implements this strategy

Regression using random forests

$$SSE = \sum (Y - f(x))^2$$

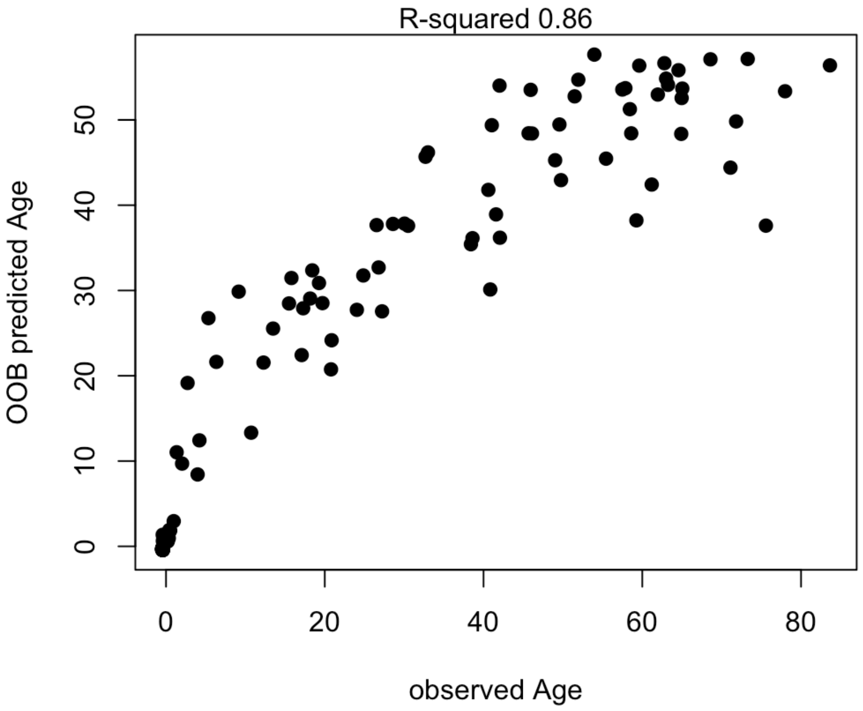


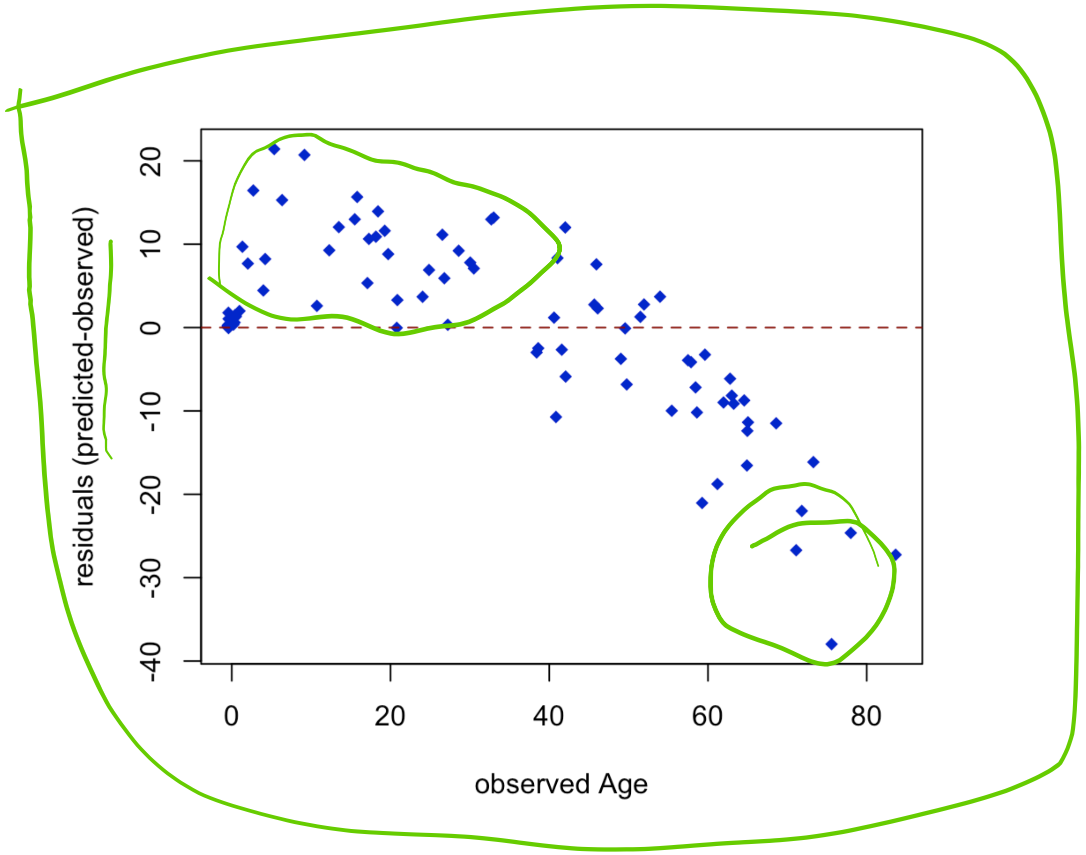
[0,1] CpG

108 subjects

~27 000 CpGs

Regression using random forests





Supervised learning for Genomics:

Recap

Key concepts:

1) Data prep → model performance

2) Overfitting → model complexity

↳ resampling test sets

3) Variable importance

4) Practical applications: classification & regression ⇒ ML