Machine Learning for Computational Biology

Nikolay Oskolkov, Lund University, NBIS SciLifeLab, Sweden

AFI course, 16.02.2026

Session 2: From Statistics to Machine Learning

@NikolayOskolkov

GitHub

github.com/NikolayOskolkov

Image generated at www.canva.com

Session content

Topics we'll cover in this session:

1) How statistics is different from machine learning

2) K-means clustering as fingerprint of machine learning

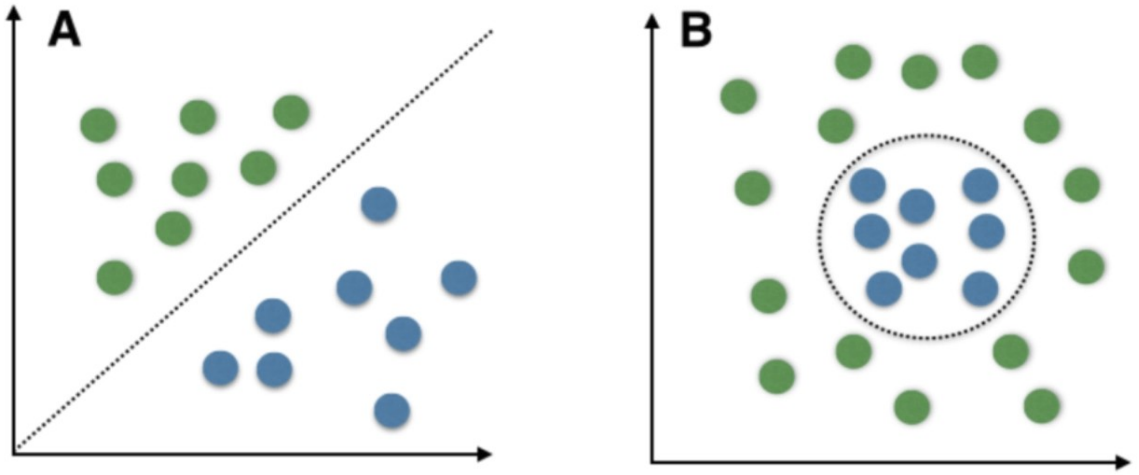
3) Basics of machine learning mentality

4) Linear machine learning: train, validation and test split

5) Overfitting vs. underfitting, ROC-curves, KNN, SVM

Y = f (X), where X is input (data) and Y is output (response)

Y is present – supervised machine learning
Y is absent – unsupervised machine learning



A: Linearly Separable Data B: Non-Linearly Separable Data

Moving from statistics to machine learning

- Statistics is more **analytical** (pen & paper)

$$\mathrm{L}\left(\mathrm{x_i} \mid \mu, \sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp^{-\frac{\sum_{\mathrm{i}=1}^{\mathrm{N}} (\mathrm{x_i} - \mu)^2}{2\sigma^2}}$$

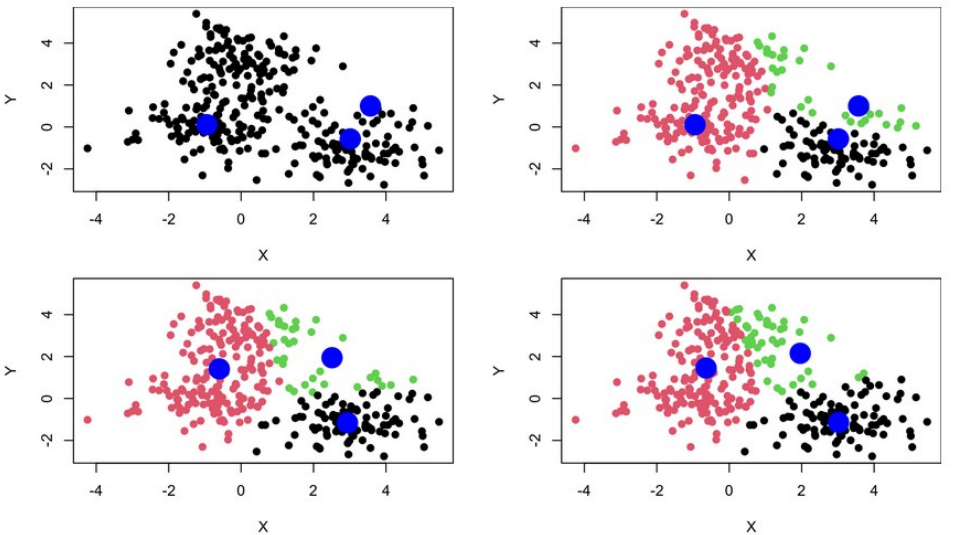
$$\frac{\partial \mathrm{L}\left(\mathrm{x_i} \mid \mu, \sigma^2\right)}{\partial \mu} = 0; \quad \frac{\partial \mathrm{L}\left(\mathrm{x_i} \mid \mu, \sigma^2\right)}{\partial \sigma^2} = 0$$

$$\mu = \frac{1}{N} \sum_{i=0}^{N} x_i \; - \text{mean estimator}$$

$$\sigma^2 = \frac{1}{N} \sum_{i=0}^{N} (x_i - \mu)^2 \; - \text{variance estimator}$$

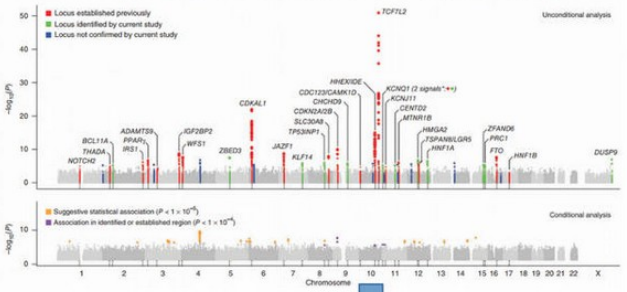
- Machine Learning is more **algorithmic** (ex. K-means)

```r
K = 3; set.seed(123); c = X[sample(1:dim(X)[1],K),]; par(mfrow=c(2,2),mai=c(0.8,1,0,0))
plot(X, xlab = "X", ylab = "Y", pch = 19); points(c, col = "blue", cex = 3, pch = 19)
for(t in 1:3)
{
  l <- vector()
  for(i in 1:dim(X)[1])
  {
    d <- vector(); for(j in 1:K){d[j] <- sqrt((X[i,1]-c[j,1])^2 + (X[i,2]-c[j,2])^2)}
    l[i] <- which.min(d)
  }
  plot(X, xlab="X", ylab="Y", col=l, pch=19); points(c, col="blue", cex=3, pch=19)
  s = list(); for(i in unique(l)){s[[i]] <- colMeans(X[l==i,])}; c = Reduce("rbind", s)
}
```



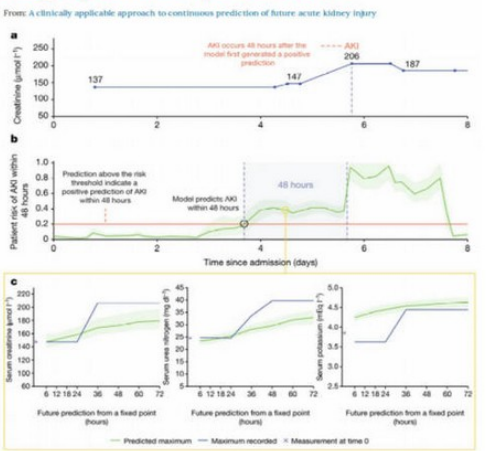
Statistics vs. machine learning: prediction

Statistics searches for candidates

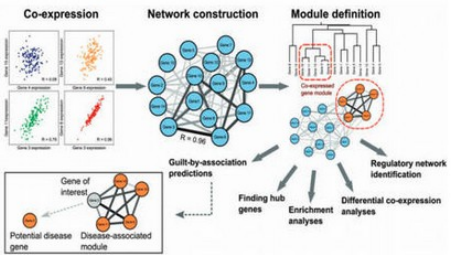


Consequence

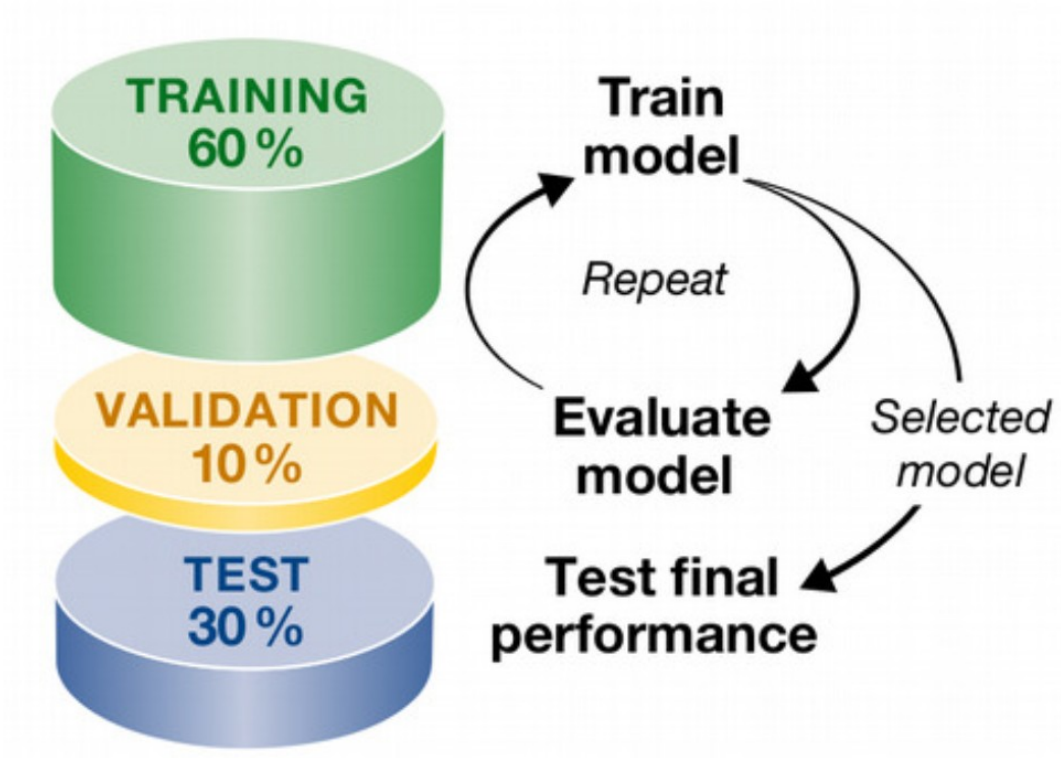
Machine Learning optimizes prediction



Consequence



How does machine learning work?



Machine Learning typically involves five basic steps:

1. Split data set into **train**, **validation** and **test** subsets

2. Fit the model on the train subset

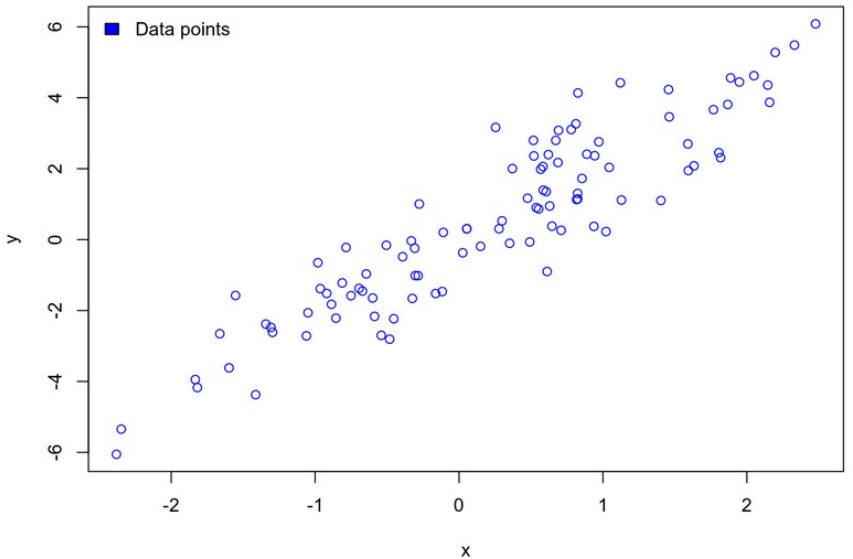
3. Validate your model on the validation subset

4. Repeat train - validation split many times and tune **hyperparameters**

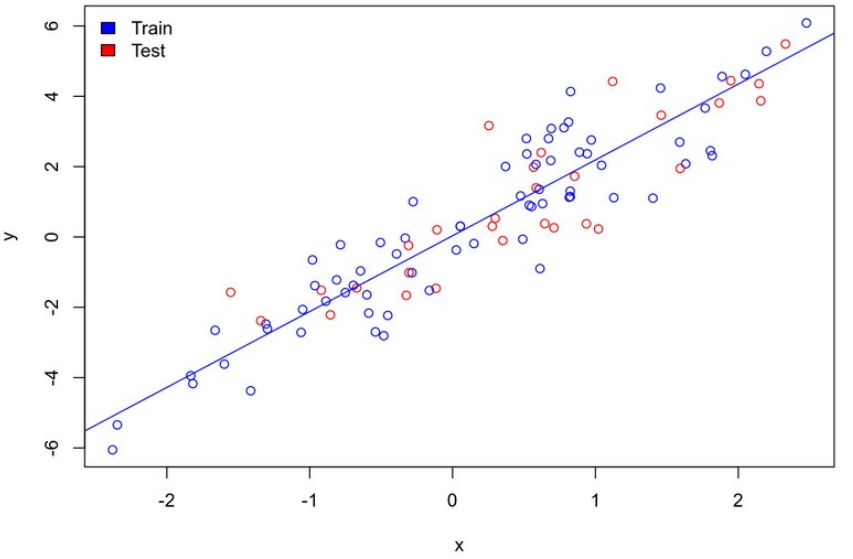
5. Test the accuracy of the optimized model on the test subset.

Toy example of machine learning

```r
N <- 100
x <- rnorm(N)
y <- 2 * x + rnorm(N)
df <- data.frame(x, y)
plot(y ~ x, data = df, col = "blue")
legend("topleft", "Data points", fill = "blue", bty = "n")
```

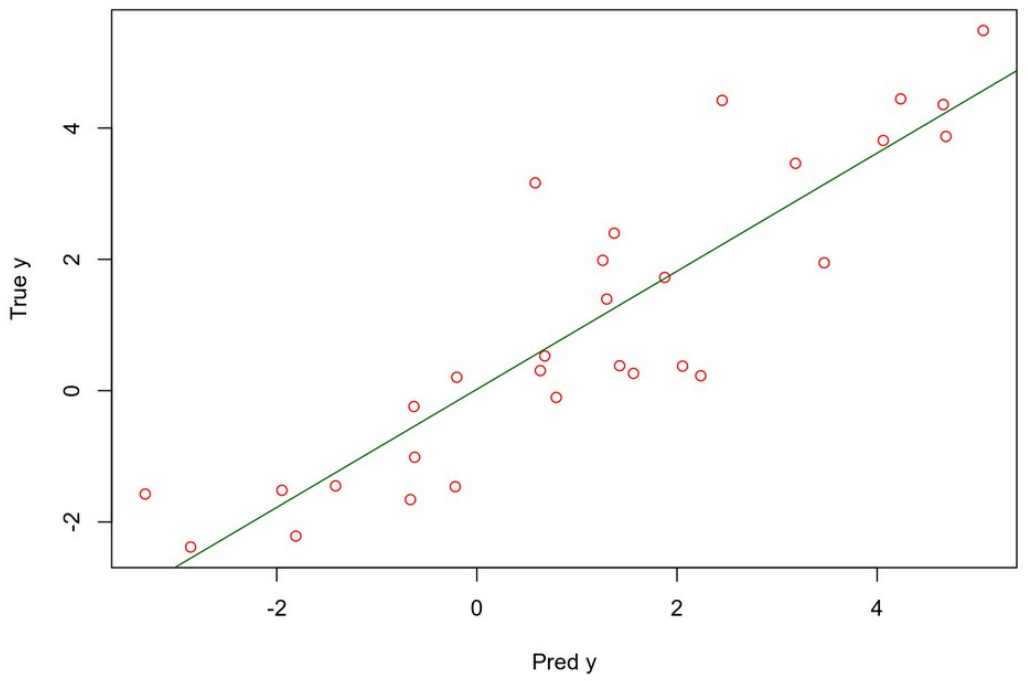



```r
train <- df[sample(1:dim(df)[1], 0.7 * dim(df)[1]), ]
test <- df[!rownames(df) %in% rownames(train), ]
df$col <- ifelse(rownames(df) %in% rownames(test), "red", "blue")
plot(y ~ x, data = df, col = df$col)
legend("topleft", c("Train","Test"), fill=c("blue","red"), bty="n")
abline(lm(y ~ x, data = train), col = "blue")
```



Toy example: model validation

```r
test_predicted <- as.numeric(predict(lm(y ~ x, data = train), newdata = test))
plot(test$y ~ test_predicted, ylab = "True y", xlab = "Pred y", col = "red")
abline(lm(test$y ~ test_predicted), col = "darkgreen")
```

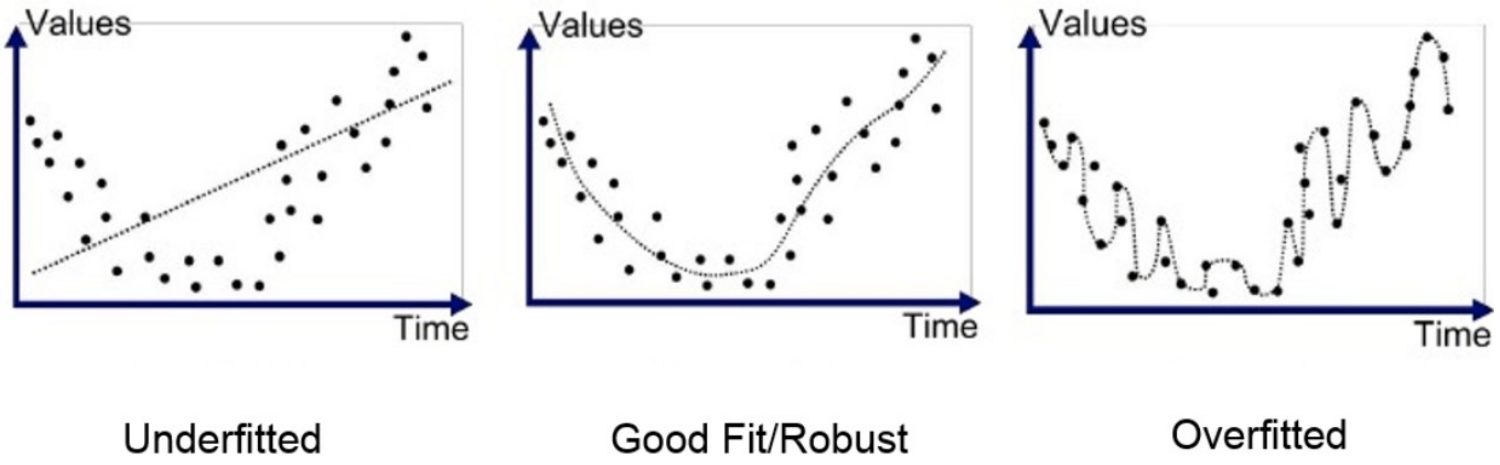



```r
summary(lm(test$y ~ test_predicted))
```

```
Call:
lm(formula = test$y ~ test_predicted)

Residuals:
     Min       1Q   Median       3Q      Max
-1.80597 -0.78005  0.07636  0.52330  2.61924

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)     0.02058    0.21588   0.095    0.925
test_predicted  0.89953    0.08678  10.366 4.33e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1

Residual standard error: 1.053 on 28 degrees of freedom
Multiple R-squared:  0.7933,    Adjusted R-squared:
0.7859
F-statistic: 107.4 on 1 and 28 DF,  p-value: 4.329e-11
```

Thus the model explains 79% of variation on the test subset.

Underfitting vs. Overfitting

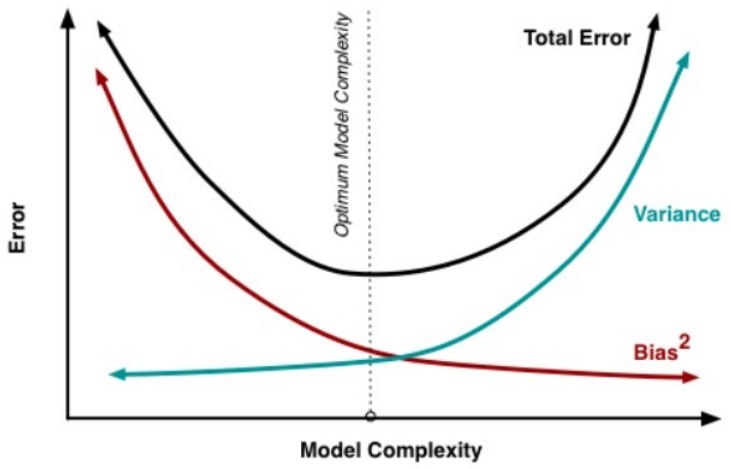


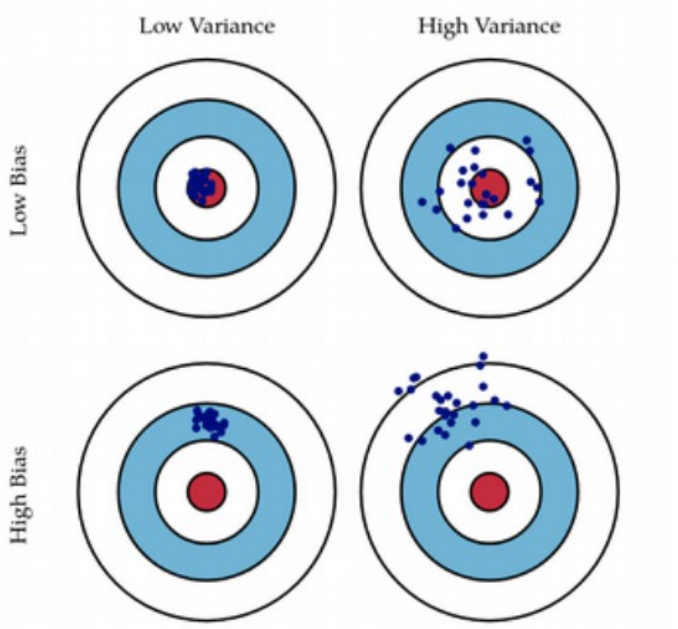
- Akaike Information Criterion (AIC):

$$\mathrm{AIC} = 2\mathrm{k} - 2\ln(\mathrm{L})$$

- Random Forest: each tree overfitted, but ensemble of trees performs very well

Bias-Variance Tradeoff





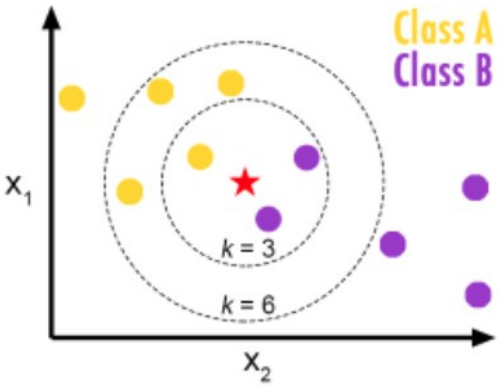
$$Y = f(X) \Longrightarrow \text{Reality}$$

$$\mathrm{Y} = \hat{\mathrm{f}}\,(\mathrm{X}) + \text{Error} \Longrightarrow \text{Model}$$

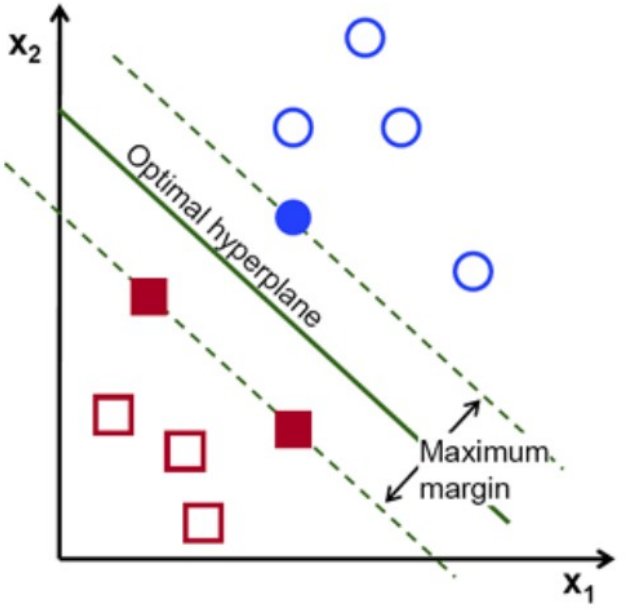
$$\text{Error}^2 = (\mathrm{Y} - \hat{\mathrm{f}}\,(\mathrm{X}))^2 = \text{Bias}^2 + \text{Variance}$$

- It is mathematically proven that ensemble learning keeps the Bias the same but leads to a large decrease of Variance

KNN and SVM



- How many out of K neighbors belong to each class
- Majority voting
- Non-linear



- Draw hyperplane that separates classes
- Maximize margins
- Can be linear and non-linear

Evaluation of model performance: ROC-curve

roc_from_scratch.R

```r
df_roc <- data.frame(ml_scores = c(0.01, 0.05, 0.13, 0.2, 0.6, 0.73, 0.8, 0.9, 0.95, 0.1, 0.3),
                     ground_truth = c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1))
par(mfrow = c(1, 2))

# ROC-curve via ROCit R package
library("ROCit")
my_roc<-rocit(df_roc$ml_scores,df_roc$ground_truth)
plot(my_roc$FPR, my_roc$TPR, col="blue", type="o", main = "ROC-curve via ROCit R package",
     ylab="SENSITIVITY (TPR)", xlab="1-SPECIFISITY (FPR)", pch=19)
lines(c(0,1), c(0,1), col="red")


# ROC-curve from scratch in plain R
df_roc <- df_roc[order(df_roc$ml_scores), ]
fpr <- c(1, 1 - cumsum(df_roc$ground_truth == 0) / sum(df_roc$ground_truth == 0))
tpr <- c(1, 1 - cumsum(df_roc$ground_truth == 1) / sum(df_roc$ground_truth == 1))
plot(tpr ~ fpr, col="blue", main = "ROC-curve from scratch in plain R",
     type="o", ylab="SENSITIVITY (TPR)", xlab="1-SPECIFISITY (FPR)", pch=19)
lines(c(0,1),c(0,1),col="red")

```

20:1 (Top Level) R Script

Console | Terminal | Background Jobs

R 4.2.3 · ~/

```
> df_roc <- data.frame(ml_scores = c(0.01, 0.05, 0.13, 0.2, 0.6, 0.73, 0.8, 0.9, 0.95, 0.1, 0.3),
+                      ground_truth = c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1))
> par(mfrow = c(1, 2))
> # ROC-curve via ROCit R package
> library("ROCit")
> my_roc<-rocit(df_roc$ml_scores,df_roc$ground_truth)
> plot(my_roc$FPR, my_roc$TPR, col="blue", type="o", main = "ROC-curve via ROCit R package",
+      ylab="SENSITIVITY (TPR)", xlab="1-SPECIFISITY (FPR)", pch=19)
> lines(c(0,1), c(0,1), col="red")
> # ROC-curve from scratch in plain R
> df_roc <- df_roc[order(df_roc$ml_scores), ]
> fpr <- c(1, 1 - cumsum(df_roc$ground_truth == 0) / sum(df_roc$ground_truth == 0))
> tpr <- c(1, 1 - cumsum(df_roc$ground_truth == 1) / sum(df_roc$ground_truth == 1))
> plot(tpr ~ fpr, col="blue", main = "ROC-curve from scratch in plain R",
+      type="o", ylab="SENSITIVITY (TPR)", xlab="1-SPECIFISITY (FPR)", pch=19)
> lines(c(0,1),c(0,1),col="red")
>
```

Environment | History | Files | Packages

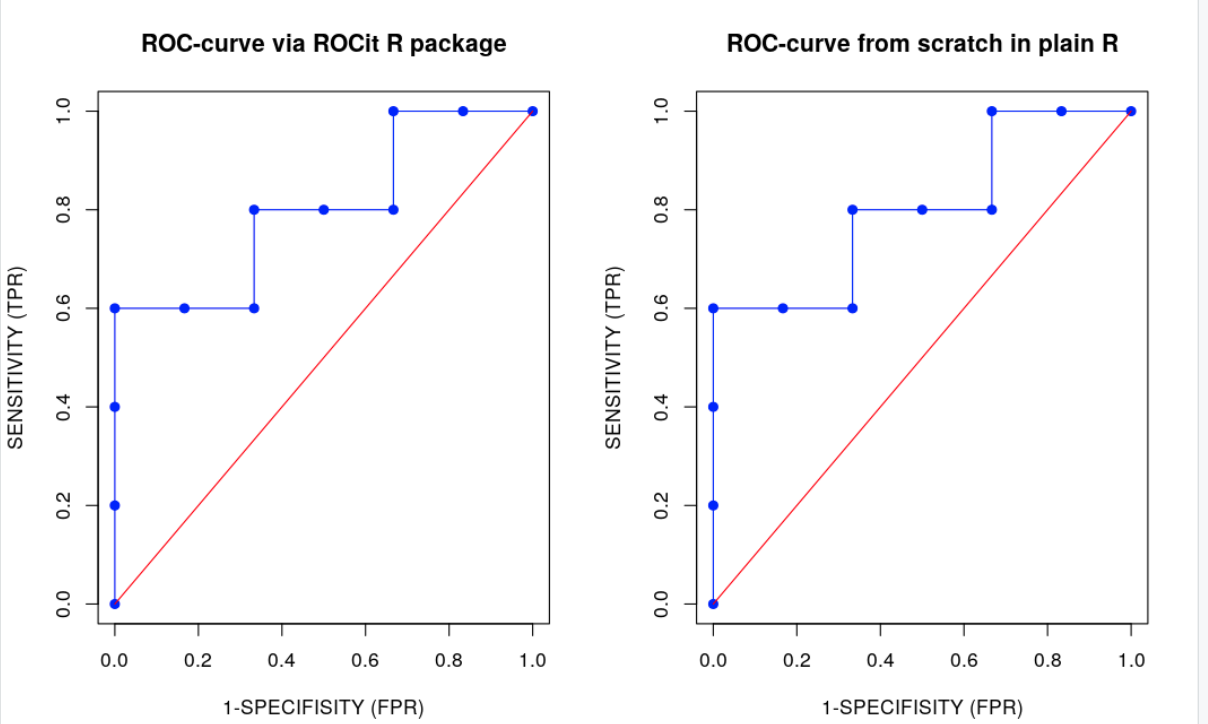
Import Dataset · 273 MiB · List

R · Global Environment

Data	
df_roc	11 obs. of 2 variables
my_roc	List of 9
Values	
fpr	num [1:12] 1 0.833 0.667 0.667 0.5 ...
tpr	num [1:12] 1 1 1 0.8 0.8 0.8 0.6 0.6 0.6 0.4 ...

Plots | Connections | Help | Viewer | Presentation

Zoom · Export · Publish



Session summary

Take home messages of the session:

1) Machine Learning is different from statistics in terms of more algorithmic approach and prediction

2) Train, validation and test are fundamental machine learning steps that aim at improving model generalizability

3) Challenge in machine learning is to find a balance between overfitting and underfitting

4) Machine learning algorithms can roughly be divided into supervised vs unsupervised, and linear vs. non-linear

National Bioinformatics Infrastructure Sweden (NBIS)

Knut och Alice Wallenbergs Stiftelse

Vetenskapsrådet

Lunds Universitet