

AI for Genomics: from CNNs and LSTMs to Transformers

Nikolay Oskolkov, Group Leader (PI) at LIOS, Riga, Latvia

Physalia course, 09.09.2025

Session 1ab: Introduction to machine learning, comparison with statistics



@NikolayOskolkov



@oskolkov.bsky.social



Personal homepage:
<https://nikolay-oskolkov.com>

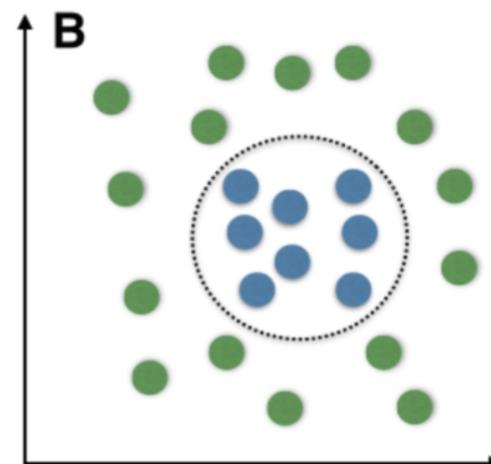
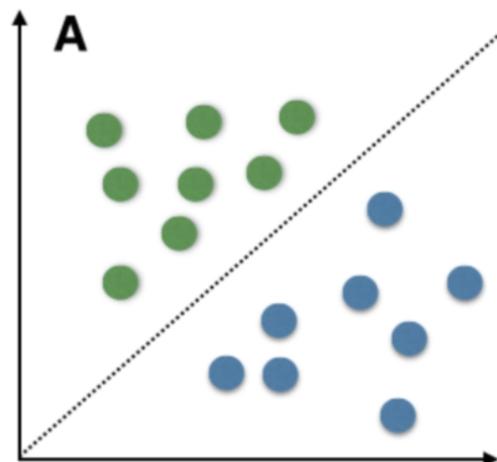
Topics we'll cover in this session:

- 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)

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

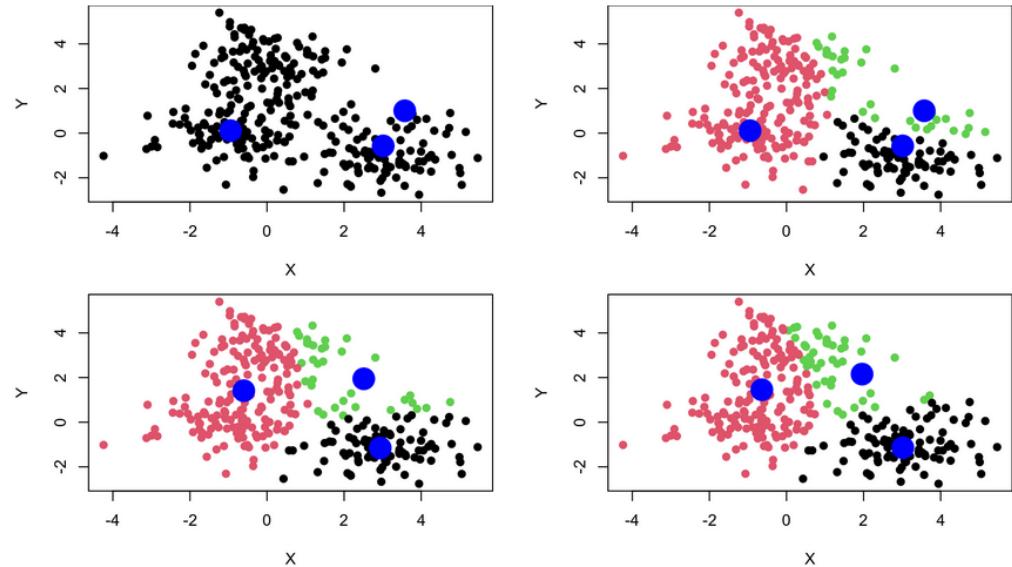
$$\frac{\partial L(x_i | \mu, \sigma^2)}{\partial \mu} = 0; \quad \frac{\partial L(x_i | \mu, \sigma^2)}{\partial \sigma^2} = 0$$

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

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

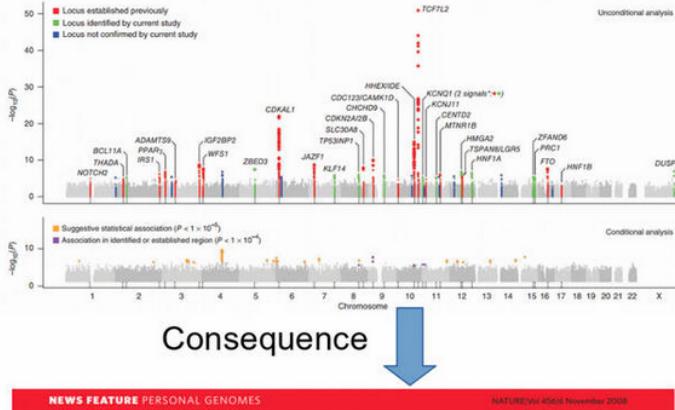
- Machine Learning is more algorithmic (ex. K-means)

```
1 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))
2 plot(X, xlab = "X", ylab = "Y", pch = 19); points(c, col = "blue", cex = 3, pch = 19)
3 for(t in 1:3)
4 {
5   l <- vector()
6   for(i in 1:dim(X)[1])
7   {
8     d <- vector(); for(j in 1:K){d[j] <- sqrt((X[i,1]-c[j,1])^2 + (X[i,2]-c[j,2])^2)}
9     l[i] <- which.min(d)
10   }
11   plot(X, xlab="X", ylab="Y", col=l, pch=19); points(c, col="blue", cex=3, pch=19)
12   s = list(); for(i in unique(l)){s[[i]] <- colMeans(X[l==i,]); c = Reduce("rbind", s)
13 }
```



Statistics vs. machine learning: prediction

Statistics searches for candidates



Consequence

NEWS FEATURE PERSONAL GENOMES NATURE Vol 456 | November 2008



The case of the missing heritability

Machine Learning optimizes prediction



A clinically applicable approach to continuous prediction of future acute kidney injury

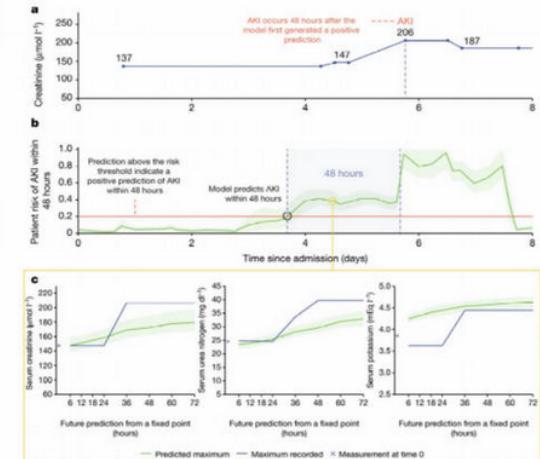
Nemat Temam, Koenraad Cnossen, Shaiur Mohamed

Nature 372, 116–119 (2010) Download Citation

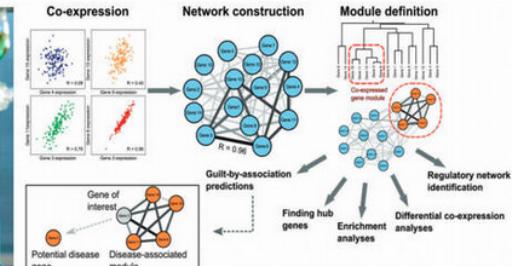
Abstract

The early prediction of deterioration could have an important role in supporting healthcare professionals, as an estimated 11% of deaths in hospital follow a failure to promptly recognize and treat deteriorating patients¹. To achieve this goal requires predictions of patient risk that are continuously updated and accurate, and delivered at an individual level with sufficient context and enough time to act. Here we develop a deep learning approach for the continuous risk prediction of future deterioration in patients, building on recent work that models adverse events from electronic health records^{2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18} and using acute kidney injury – a common and potentially life-threatening condition¹⁹ – as an exemplar. Our model was developed on a large, longitudinal dataset of electronic health records that cover diverse

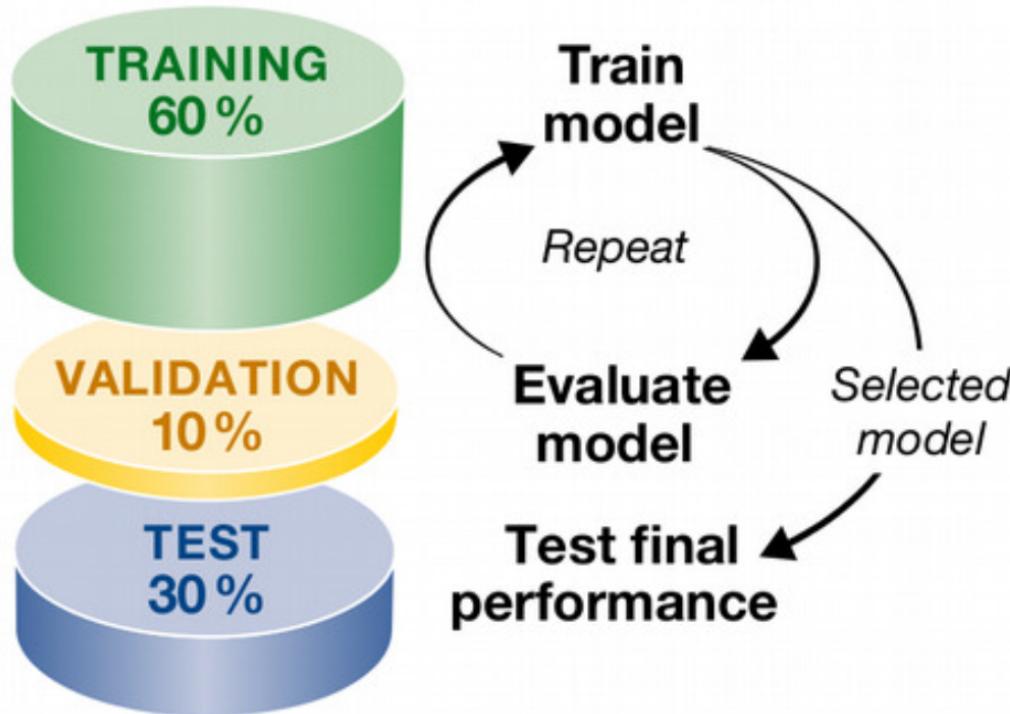
From: A clinically applicable approach to continuous prediction of future acute kidney injury



Consequence



How does machine learning work?

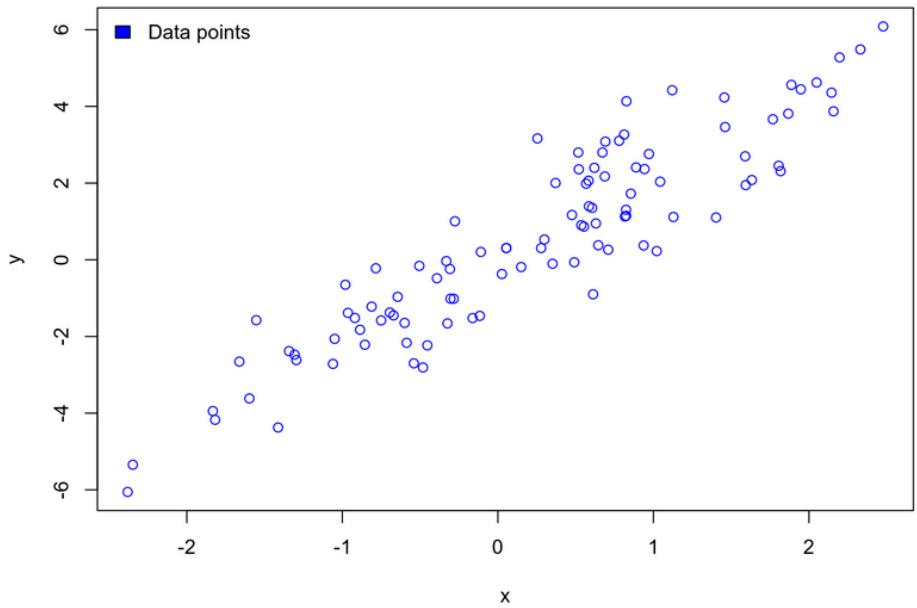


Machine Learning typically involves five basic steps:

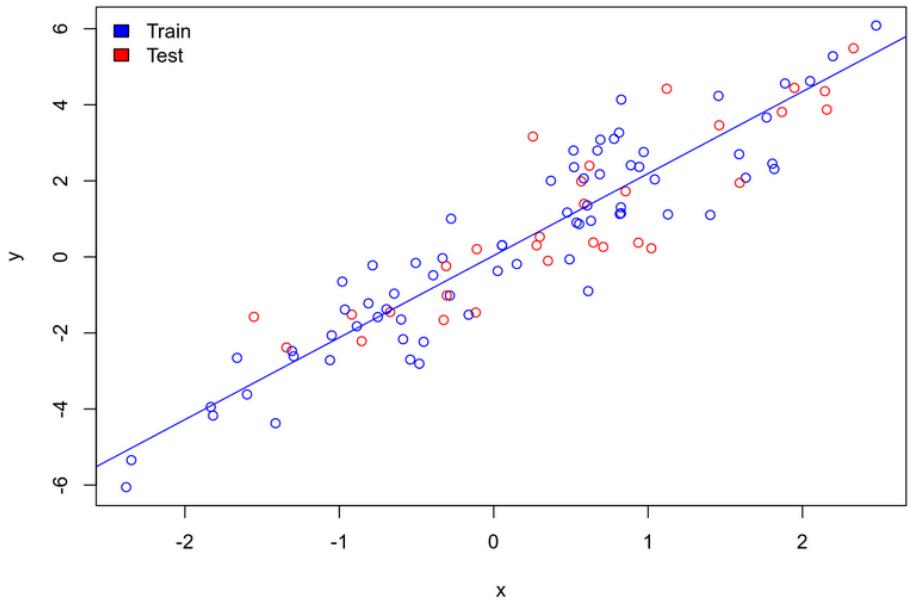
1. Split data set into train, validation and test subsets
2. Fit the model on the train subset
3. Validate your model on the validation subset
4. Repeat train - validation split many times and tune hyperparameters
5. Test the accuracy of the optimized model on the test subset.

Toy example of machine learning

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

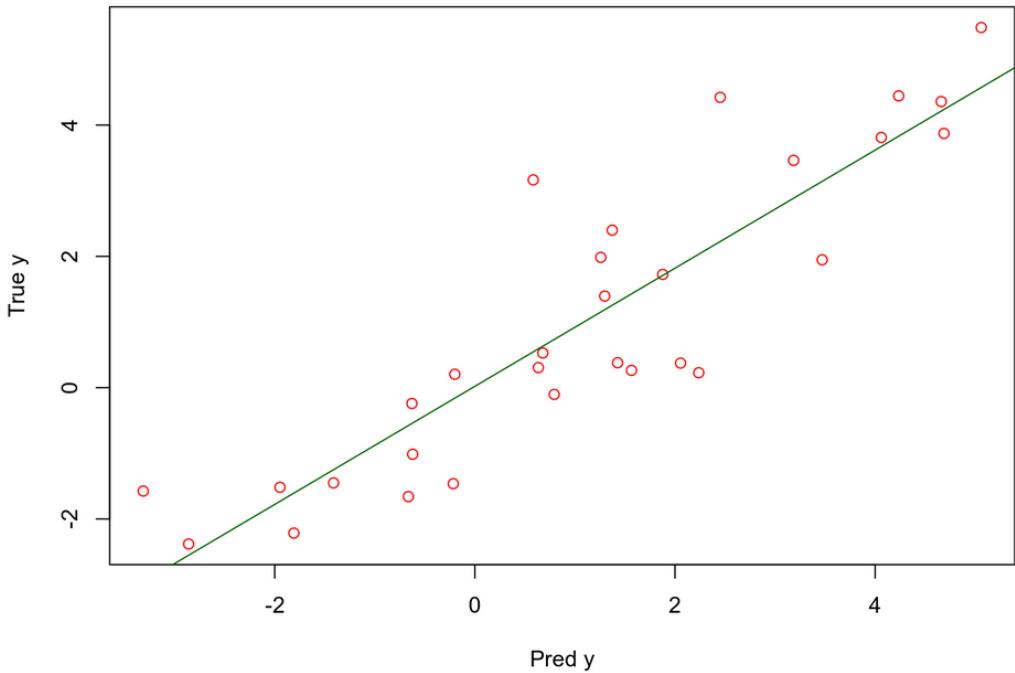


```
1 train <- df[sample(1:dim(df)[1], 0.7 * dim(df)[1]), ]
2 test <- df[!rownames(df) %in% rownames(train), ]
3 df$col <- ifelse(rownames(df) %in% rownames(test), "red", "blue")
4 plot(y ~ x, data = df, col = df$col)
5 legend("topleft", c("Train", "Test"), fill=c("blue", "red"), bty="n")
6 abline(lm(y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



Toy example: model validation

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



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

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

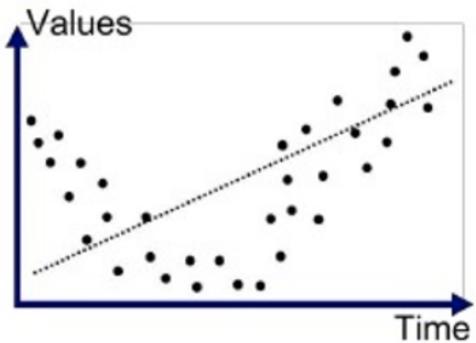
Residuals:
    Min      1Q  Median      3Q     Max 
-1.80597 -0.78005  0.07636  0.52330  2.61924 

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

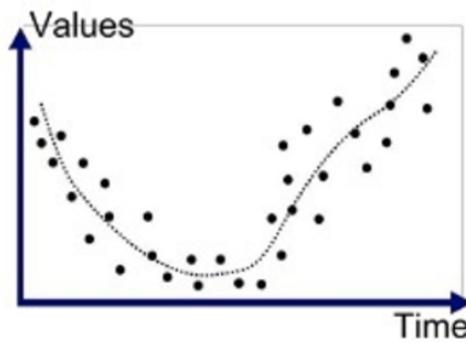
Residual standard error: 1.053 on 28 degrees of freedom
Multiple R-squared:  0.7933,    Adjusted R-squared:
0.7859
F-statistic: 107.4 on 1 and 28 DF,  p-value: 4.329e-11
```

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

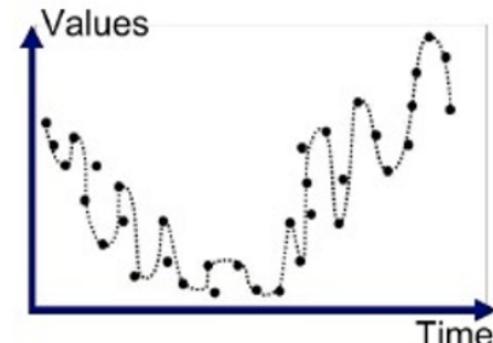
▼ Underfitting vs. Overfitting



Underfitted



Good Fit/Robust



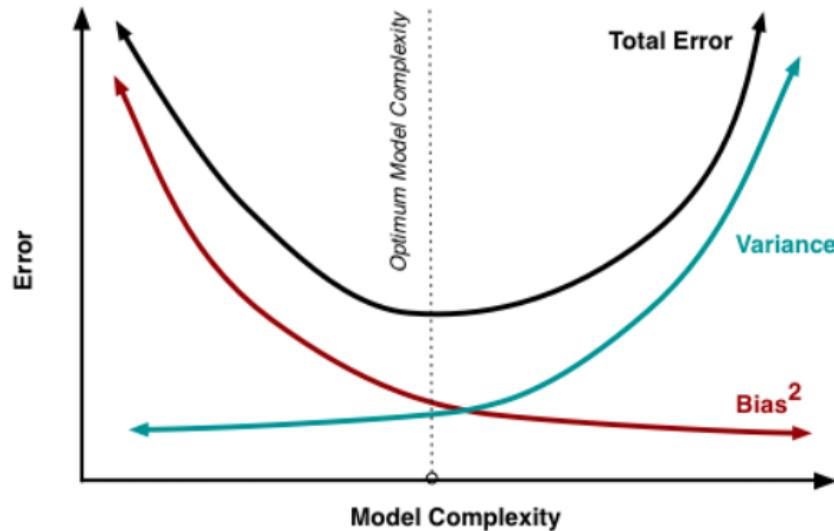
Overfitted

- Akaike Information Criterion (AIC):

$$AIC = 2k - 2\ln(L)$$

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

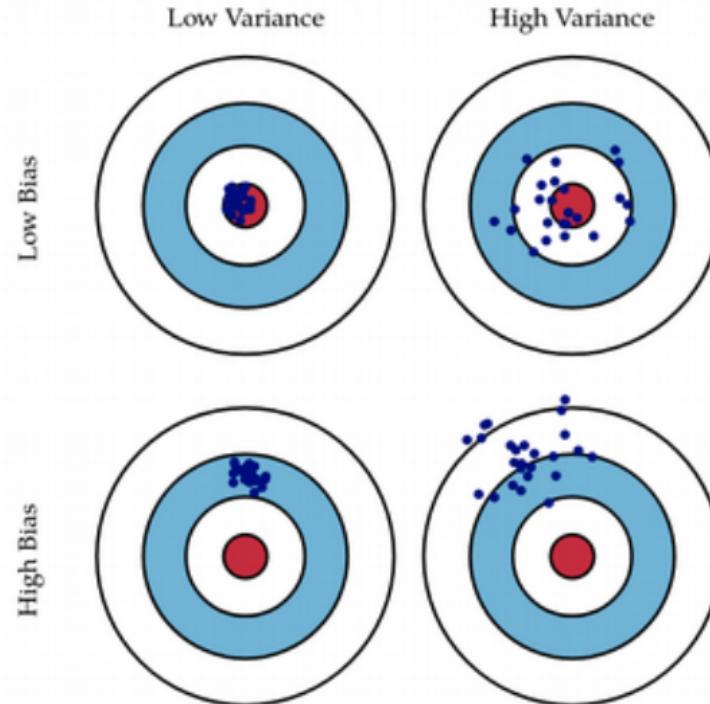
▼ Bias-Variance Tradeoff



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

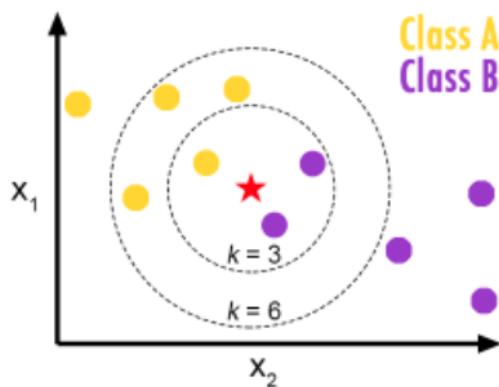
$$Y = \hat{f}(X) + \text{Error} \implies \text{Model}$$

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

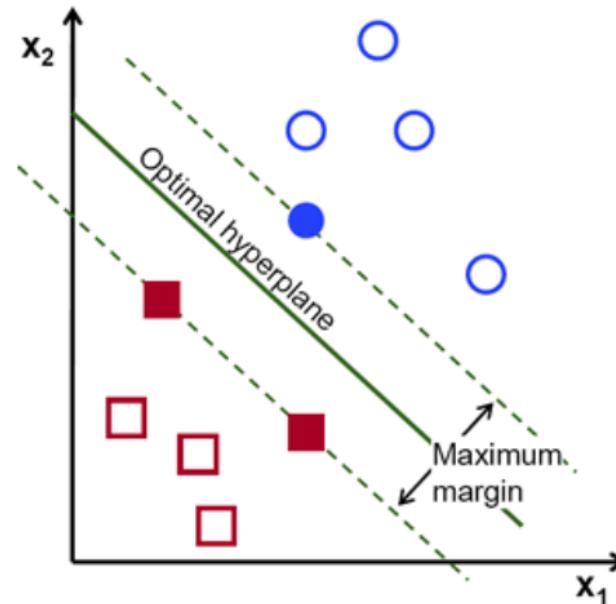


- It is mathematically proven that ensemble learning keeps the Bias the same but leads to a large decrease of Variance

▼ KNN and SVM



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

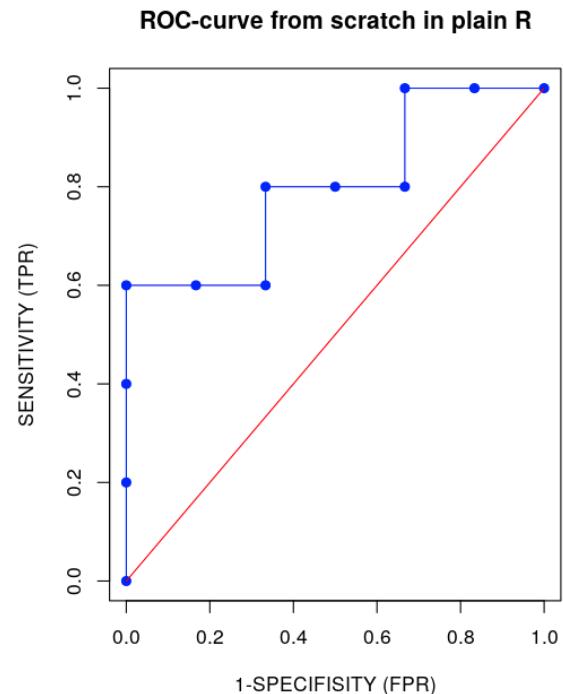
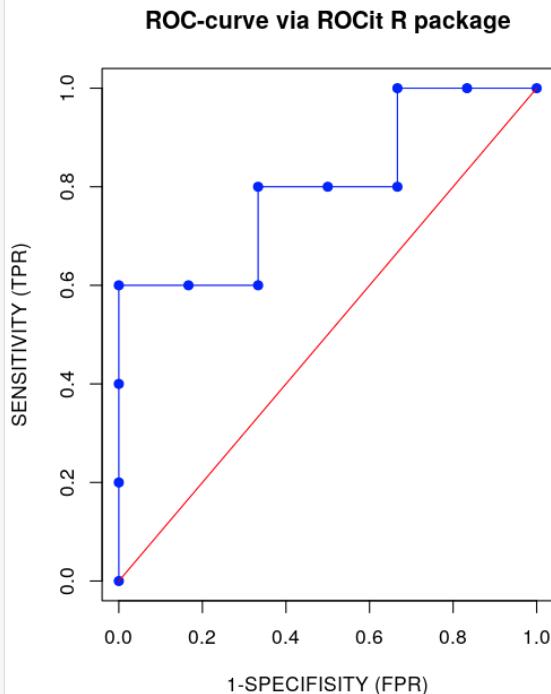
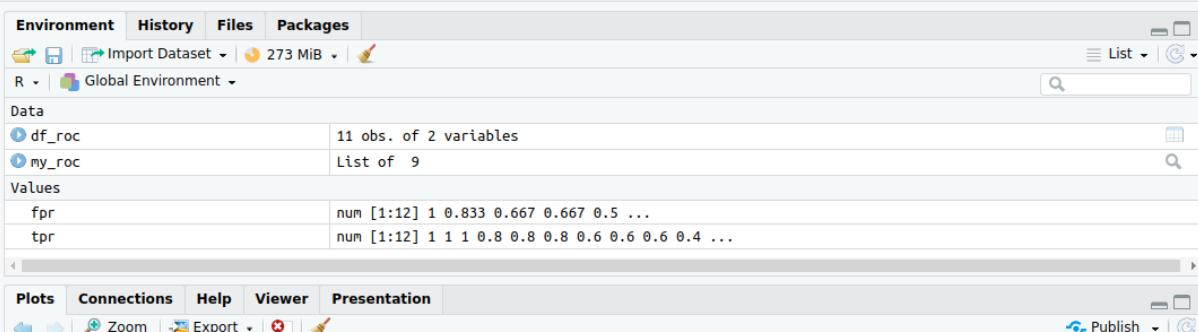


- Draw hyperplane that separates classes
- Maximize margins
- Can be linear and non-linear

Evaluation of model performance: ROC-curve

```
roc_from_scratch.R x
Source on Save Run Source
1 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),
2                         ground_truth = c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1))
3 par(mfrow = c(1, 2))
4
5 # ROC-curve via ROCit R package
6 library("ROCit")
7 my_roc<-rocit(df_roc$ml_scores,df_roc$ground_truth)
8 plot(my_roc$FPR, my_roc$TPR, col="blue", type="o", main = "ROC-curve via ROCit R package",
9      ylab="SENSITIVITY (TPR)", xlab="1-SPECIFICITY (FPR)", pch=19)
10 lines(c(0,1), c(0,1), col="red")
11
12
13 # ROC-curve from scratch in plain R
14 df_roc <- df_roc[order(df_roc$ml_scores), ]
15 fpr <- c(1, 1 - cumsum(df_roc$ground_truth == 0) / sum(df_roc$ground_truth == 0))
16 tpr <- c(1, 1 - cumsum(df_roc$ground_truth == 1) / sum(df_roc$ground_truth == 1))
17 plot(tpr ~ fpr, col="blue", main = "ROC-curve from scratch in plain R",
18      type="o", ylab="SENSITIVITY (TPR)", xlab="1-SPECIFICITY (FPR)", pch=19)
19 lines(c(0,1),c(0,1),col="red")
20 |
```

```
20:1 (Top Level) R Script
Console Terminal x Background Jobs x
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, 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-SPECIFICITY (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-SPECIFICITY (FPR)", pch=19)
> lines(c(0,1),c(0,1),col="red")
>
```



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*



**LUNDS
UNIVERSITET**