

# Machine Learning for biology

V. Monbet



UFR de Mathématiques  
Université de Rennes

# Outline

## Supervised learning

### Introduction

Linear model (I)

Linear model (II)

# Outline

Supervised learning

Introduction

# Supervised learning

- Problem: prediction of a variable  $Y \in \mathcal{Y}$  given inputs  $\mathbf{X} = (X_1, \dots, X_p) \in \mathcal{X}$
- In practice,  $n$  observations of  $Y \in \mathcal{Y}$  and  $\mathbf{X} \in \mathcal{X}$  are available to learn the prediction "model".  $\mathcal{S}_n = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$

Response (or output)

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

Inputs

$$\mathbf{x} = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & & \vdots \\ x_{j1} & \cdots & x_{jp} \\ \vdots & & \vdots \\ x_{n1} & \cdots & x_{np} \end{pmatrix}$$

- The response  $Y$  can be quantitative (ex: Ozone concentration, El Niño index, temperature, ...) or qualitative (ex: yes/no, wet/dry, ...).
- Problem: find a mapping  $f$  such that

$$Y = f(\mathbf{X}) + \epsilon$$

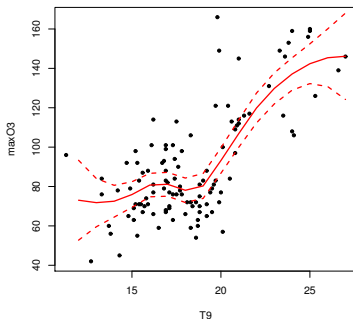
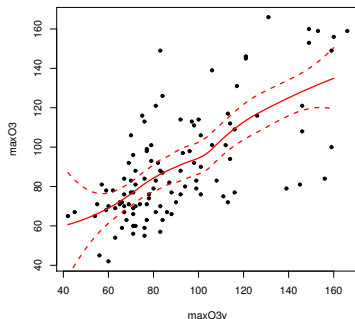
where  $\epsilon$  denotes an error.

## Supervised learning, example (regression)

Regression = prediction of a continuous variable.

- ▶ Example: **predict maximum Ozone (O3) concentration** on day D given maxO3 concentration and meteorological variables of the day before (temperature, nebulosity, West-East wind intensity, wind direction, rain yes/no) at times (9:00, 12:00, 15:00).
- ▶ Example: prediction of maxO3 given maxO3 the day before (left) and the temperature at 9:00 (right).

Black dots correspond to observations, red solid lines to prediction and red dotted lines to a 95% confidence interval for the prediction.



# Supervised learning, example (classification)

Classification = prediction of a categorical variable.



► Approximate the mapping :  $f(\text{image}) = \text{Tomate}$

$$f : \mathbb{R}^{128 \times 128} \rightarrow \{\text{Apple, Pear, Tomato, Cow, Dog, Horse}\}$$

# Supervised learning

- ▶ Supervised learning methods can be mainly gathered in two groups: purely data driven approaches and methods based on a parametric model.
- ▶ **Data driven approaches** (non parametric)
  - **Analogs** (or **k**-nearest neighbors) consists in learning from similar (or dissimilar) observations.
  - **Regression trees** allow to split the space  $\mathcal{X}$  of  $\mathbf{x}$  into small regions with constant  $y$ .
- ▶ **Modeling approaches** (parametric)
  - **Linear regression** uses a linear combination of inputs for predicting a quantitative variable
  - **Artificial Neural Network** and **Deep learning** stack (small) linear models and non linear layers for predicting a quantitative variable or qualitative variable/generating processes or objects/etc...
  - **Support Vector Machines** transform the data in order to be able to use linear models.
- ▶ **Aggregation models:** Some of the afore mentionned methods or models can be combined to take advantage of the best features of each of them.
  - **Bagging** and random forest combines regression trees by computing a mean of small trees calibrated independantly
  - **Boosting**, **Gradient Boosting** stack regression trees so that the current tree improve the previous ones.

# Outline

## Supervised learning

### Linear model (I)

- Generalities, prediction of a continuous variable

- Cross-validation

- Prediction of a categorical variable

### Linear model (II)

# Outline

## Linear model (I)

Generalities, prediction of a continuous variable

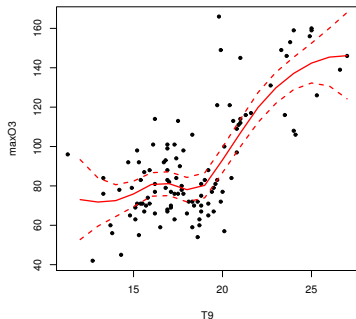
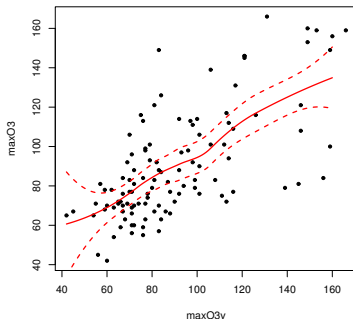
Cross-validation

Prediction of a categorical variable

# Supervised learning, example (regression)

Regression = prediction of a continuous variable.

- Example: **predict maximum Ozone (O3) concentration** on day D given maxO3 concentration and meteorological variables of the day before (temperature, nebulosity, West-East wind intensity, wind direction, rain yes/no) at times (9:00, 12:00, 15:00). Black dots correspond to observations, red solid lines to prediction and red dotted lines to a 95% confidence interval for the prediction.

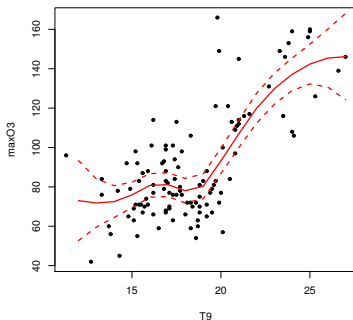
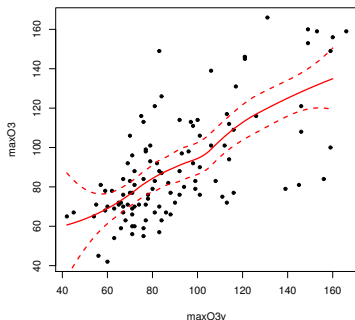


- Remark : the prediction (red line) materializes a local mean. This local mean will be denoted  $E(Y|X = x)$  in the sequel and named **Conditional expectation of Y given x**.

## Supervised learning, example (regression)

Regression = prediction of a continuous variable.

Black dots correspond to observations, red solid lines to prediction and red dotted lines to a 95% confidence interval for the prediction.



Remark : the prediction (red line) materializes a local mean.  
This local mean will be denoted  $E(Y|X = \mathbf{x})$  in the sequel.

Standard machine learning methods only focus on the local mean (and give no information about the uncertainty around the mean).

# Linear regression models

- ▶ We have an input vector  $\mathbf{X} \in \mathcal{X}$ , and want to predict a real-valued output  $Y \in \mathcal{Y} \subseteq \mathbb{R}$ .
- ▶ The linear regression model has the form

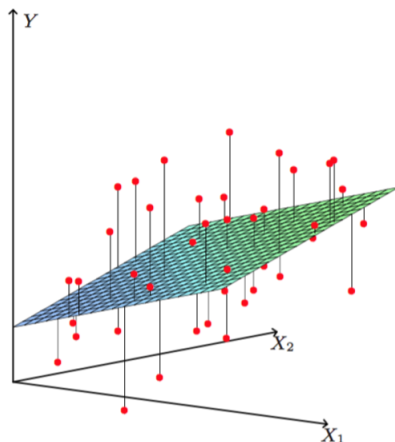
$$f(\mathbf{X}) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p = \mathbf{X}\beta \quad (1)$$

why  $\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p = \mathbf{X}\beta$ ?

- ▶ Here the  $\beta_j$ 's are unknown parameters or coefficients, and the variables  $X_j$  can come from different sources (numerical, categorical).
- ▶ The most popular **estimation method** is **least squares** in which we pick the coefficients  $\beta = (\beta_0, \beta_1, \dots, \beta_p)^T$  to minimize the residual sum of squares (sum of squared residuals = **mean square error (MSE)**)

$$MSE(\beta) = \sum_{i=1}^n \left( y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2$$

given a learning dataset  $\mathcal{S} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$



**FIGURE 3.1.** *Linear least squares fitting with  $X \in \mathbb{R}^2$ . We seek the linear function of  $X$  that minimizes the sum of squared residuals from  $Y$ .*

Figure from Hastie's book.

## Minimization of RSS and prediction

- ▶ One can rewrite **RSS** in a matricial form with  $\mathbf{x} = [\mathbf{1}, \mathbf{x}]$

$$RSS(\beta) = (y - \mathbf{x}\beta)^T (y - \mathbf{x}\beta).$$

- ▶ This is a quadratic function in the  $p$  parameters vector  $\beta$ . Differentiating with respect to  $\beta$  Differentiation of  $(y - \mathbf{x}\beta)^T (y - \mathbf{x}\beta)$ ? and setting the derivatives to zero, one obtains **estimators**

$$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T y$$

This is an exact formula easy to compute (under some conditions).

- ▶ For **prediction**,  $\beta$  is substitute by  $\hat{\beta}$  in Eq. (1)

$$\hat{y} = \mathbf{x}\hat{\beta} = \mathbf{x} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T y$$

- ▶ Model **predictive power** is often measured by the percentage of explained variance

$$R^2 = 1 - \frac{RSS}{TSS} = \frac{\sum_i (y_i - \bar{y})(\hat{y}_i - \bar{y})}{(\sum_i (y_i - \bar{y})^2 \sum_i (\hat{y}_i - \bar{y})^2)^{1/2}}$$

$R^2 = 1$  means that  $\hat{y} = y$  and  $R^2 = 0$  means that  $\hat{\beta} = (\bar{y}, 0, \dots, 0)$

## Important remarks/summary

- ▶ As many machine learning methods, the linear model is fitted by **minimizing the mean square error**. Note that it is a global criteria ie all the observations are involved in the criteria.
- ▶ The specificity of linear model is the shape constraint imposed to the regression fonction.
- ▶ **Estimators** are obtained by a close form expression

$$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T y$$

→ **no numerical optimization algorithm has to be run**

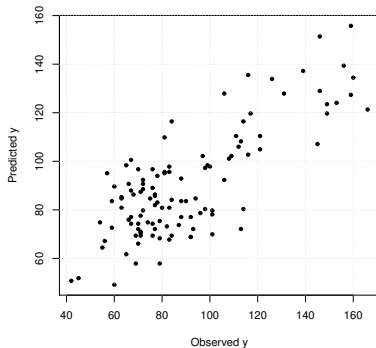
- ▶ Linear models are **elementary blocks** for other machine learning models such as neural networks, deep learning, machine support vectors.

## Validation of a regression model, helpfull plot

### Example: Ozone

To validate a regression mode the first step is to plot the prediction with respect to the observation. It is a qualitative validation.

- Model:  $\text{max03} = \beta_0 + \beta_1 T12$   
 $R^2 = 0.61$ ,  $R^2_{aj} = 0.48$



Question : what do you think about this plot?

# Validation of a regression model, quantitative measures

Quantitative measures provide more objective criteria, usefull for comparing models.

- ▶ The quality of the model is usually measured by the square root of the mean/average of the square of all of the errors (**root mean square error**).

$$RMSE = \sqrt{\sum_{i \in \mathcal{I}} (\hat{y}_i - y_i)^2}$$

where  $\mathcal{I}$  denotes a set of validation individuals (see below).

- ✓ RMSE is probably the most used score in regression tasks.
- ✗ But, take care, RMSE is very sensible to few extreme errors.
- ▶ Some other loss functions may be defined, as Mean Absolute Error

$$MAE = \sum_{i \in \mathcal{I}} |\hat{y}_i - y_i|$$

- ▶ Correlation between observed and predicted output.
- ▶  $R^2$  measure the part of variance explained by the model.

# Linear regression models

- ▶ A linear regression model assumes that the regression function  $f(x) = E(Y|X = x)$  is linear in the inputs  $X_1, \dots, X_p$ .

$$\begin{aligned} f(\mathbf{X}) &= \beta_0 + \beta_1 X_1 + \dots + \beta_p X_p \\ &= \mathbf{X}\beta \end{aligned}$$

- ✓ They are simple and often provide an adequate and interpretable description of how the inputs affect the output.
- ✓ For prediction purposes they can sometimes outperform fancier nonlinear models, especially in situations with small numbers of training cases, low signal-to-noise ratio or sparse data.
- ✓ Linear methods can be applied to transformations of the inputs and this considerably expands their scope.
- ✗ They may be too simple and not flexible enough for describing the observed processes.
- ✓ Linear models are used as small brick in neural networks.

# Outline

## Linear model (I)

Generalities, prediction of a continuous variable

### Cross-validation

Prediction of a categorical variable

# Cross-validation

- ▶ **Cross-validation** is used for more careful model validation ; it helps to evaluate the **generalization** capacity of the model.  
ie how the model performs for sample outside of the learning data set.
- ▶ The principle of cross validation is split the initial dataset into two independant subsets.  
One is used for calibrating the model (ie estimating the  $\beta$  parameters)  
The other one it used to test the capacity of the model to predict the right ouput for a new data.
- ▶ In practice a partition of the dataset into 2 subsets is generated randomly.
- ▶ It can be repeated several times to obtain a distribution of the prediction error associated to the model.

# Cross-validation, algorithms

## K-fold CV

**index = sample  $n$  indices in**

**$1, 2, \dots, n$  without replacement**

$n_K = n/K$

**for  $k = 1$  to  $K$ ,**

**$train = index[k * n_K + (1 : n_K)]$**

**$test = \{1, \dots, n\} - \{train\}$**

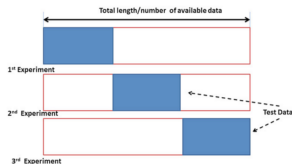
**Calibrate the model  $M_k$  on  $train$**

**Predict  $\hat{Y}_i = M_k(\mathbf{x}_i), \forall i \in test$**

$$MSE(k) = \frac{1}{card(test)} \sum_{i \in test} (\hat{Y}_i - y_i)^2$$

**end for**

$$RMSE = \sqrt{\frac{1}{K} \sum_{k=1}^K MSE(k)}$$



## Monte Carlo CV

$n_{train} = 2/3 * n$

$B = 100$

**for  $b = 1$  to  $B$ ,**

**$train = sample\ n_{train}$  indices in**

**$1, 2, \dots, n$  without replacement**

**$test = \{1, \dots, n\} - \{train\}$**

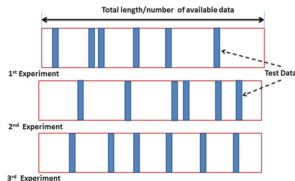
**Calibrate the model  $M_k$  on  $train$**

**Predict  $\hat{Y}_i = M_k(\mathbf{x}_i), \forall i \in test$**

$$MSE(b) = \frac{1}{card(test)} \sum_{i \in test} (\hat{Y}_i - y_i)^2$$

**end for**

$$RMSE = \sqrt{\frac{1}{B} \sum_{b=1}^B MSE(b)}$$



## k-fold CV vs. Monte Carlo CV

- ▶ Under **k-fold cross validation**, each point gets tested exactly once, which seems fair. However, cross-validation only explores a few of the possible ways that your data could have been partitioned.
- ▶ Averaging the results of a  $k$ -fold cross validation run gets you a (nearly) unbiased estimate of the algorithm's performance, but with high variance (as you'd expect from having only 5 or 10 data points).
- ▶ **Monte Carlo cross validation** lets you explore somewhat more possible partitions, though you're unlikely to get all of them.
- ▶ Since you can, in principle, run it for as long as you want/can afford, Monte Carlo cross validation can give you a less variable, but more biased estimate.
- ▶ Best choice : **Monte Carlo cross validation**

## Take home message

Machine learning model validation when prediction a quantitative variables (Regression task)

- ▶ **Usually use cross-validation !!!**
- ▶ When using cross validation, one part of the data is used for the calibration (learning) of the model ; the other part is used for evaluating the performances of the model.
- ▶ Usual **scores** : **RMSE**, **MAE**,  $R^2$
- ▶ **Plots are useful for identifying the pro/con of the model.**

# Outline

## Linear model (I)

Generalities, prediction of a continuous variable

Cross-validation

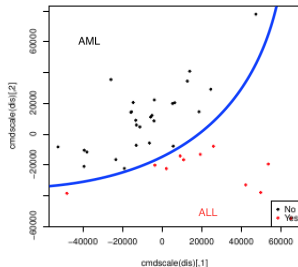
Prediction of a categorical variable

## Extensions of linear model for categorical variable

- ▶ When  $Y$  is a **categorical variable** the prediction problem is equivalent to search for boundaries between classes.
- ▶ There are several different (generalized) linear <sup>a</sup> approaches to model the boundaries
- ▶ The most common methods are linear discriminant analysis or generalized linear models (ex: logistic regression).

<sup>a</sup>**linear** means that the decision rule can be expressed as  $g(\beta_0 + \beta_1 X_1 + \dots + \beta_p X_p)$

Here, the problem is to predict a leucemia type from microarray data. The data have been projected on a 2 dimensions space by MDS, after a preselection of the most informative genes.



# Logistic regression

- ▶ **Logistic regression** is a direct and straightforward extension of linear regression.
- ▶  $Y \in \{0, 1\}$  is binary.
- ▶ The logistic model does not directly predict  $Y$  but  $P(Y = 1 | \mathbf{X} = \mathbf{x})$
- ▶ Remind that  $P(Y = 1 | \mathbf{X} = \mathbf{x}) \in [0, 1]$  as a probability.
- ▶ A link function  $g : \mathbf{R} \mapsto [0, 1]$  is then introduced to transform the linear part into a quantity in  $[0, 1]$ .

$$P(Y = 1 | X = x) = g \left( \beta_0 + \sum_{j=1}^p \beta_j x_j \right)$$

In the logistic model,

$$g(x) = \exp(x) / (1 + \exp(x)).$$

- ▶ Parameter  $\beta$  is estimated by maximum likelihood.

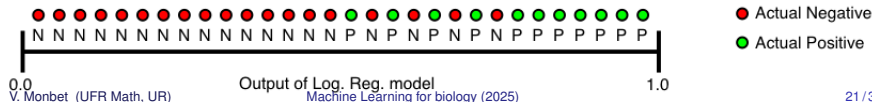
## Confusion Matrix for a 2 classes problem

- "No pollution" is a negative class (equivalent to  $Y = 0$ ).

We can summarize our "pollution-prediction" model using a  $2 \times 2$  confusion matrix that depicts all four possible outcomes:

<b>True Positive (TP):</b> - Reality : the river is polluted - Model prediction: pollution	<b>False Positive (FP):</b> - Reality : the river is not polluted - Model prediction: pollution
<b>False Negative (FN):</b> - Reality : the river is polluted - Model prediction: no pollution	<b>True Negative (TN):</b> - Reality : the river is not polluted - Model prediction: no pollution

- ▶ A **true positive** is an outcome where the model correctly predicts the positive class. Similarly, a **true negative** is an outcome where the model correctly predicts the negative class.
- ▶ A **false positive** is an outcome where the model incorrectly predicts the positive class. And a **false negative** is an outcome where the model incorrectly predicts the negative class.



## Computation of TP, TN, FP, FN

Remind that, in a classification problem, the fitted model predicts

$$P(Y = 1|X = x)$$

and not directly the classes.

We need a **decision rule**!

Alternative possibilities

**Rule 1** For a given  $x$ , if  $P(Y = 1|X = x) > P(Y = 0|X = x)$  then sample  $x$  belongs to class 1, else it belongs to class 0.

**Rule 2** For a given  $x$ , if  $P(Y = 1|X = x) > s$  then  $x$  belongs to class 1, else it belongs to class 0, where  $s$  is a chosen threshold

Remarks

1. in a two classes problem, if  $s = 1/2$  then rule 2 is equivalent to rule 1.
2. Softwares usually return the predicted classes (by default) ; they uses the first rule.
- ✗ Rule 1 may not be appropriate when classes are unbalanced in the population.

## Accuracy score

**Accuracy** computes how many times a model made a correct prediction across the entire dataset. Formally, accuracy has the following definition:

$$\text{accuracy} = \frac{\text{number of correct prediction}}{\text{total of number prediction}}$$

For binary classification, accuracy can also be calculated in terms of positives and negatives as follows:

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy can be a reliable metric only if the dataset is class-balanced; that is, each class of the dataset has more or less the same number of samples.

## Precision, Recall and F1-score

**F1 score** is an alternative machine learning evaluation metric that assesses the predictive skill of a model by elaborating on its class-wise performance rather than an overall performance as done by accuracy.

**Precision** measures how many of the “positive” predictions made by the model were correct.

$$\text{precision} = \frac{TP}{TP + FP}$$

**Recall** measures how many of the positive class samples present in the dataset were correctly identified by the model.

$$\text{recall} = \frac{TP}{TP + FN} = \text{sensitivity}$$

The **F1 score** combines precision and recall using their harmonic mean, and maximizing the F1 score implies simultaneously maximizing both precision and recall.

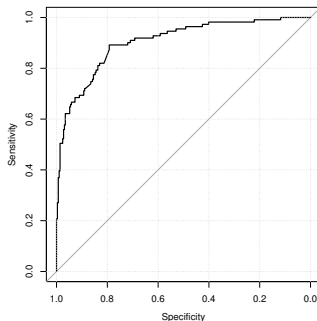
$$\text{F1-score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

## Sensitivity, specificity, ROC curve

An **ROC curve** (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters:

- True Positive Rate  $TPR = \frac{TP}{TP+FN} = \text{sensitivity}$
- False Positive Rate  $FPR = \frac{FP}{FP+TN} = \text{specificity}$ .

An ROC curve plots TPR vs. FPR at different classification thresholds.



## Area under the ROC curve (AUC)

**AUC** is the area under the ROC curve.

- ▶  $0 \leq AUC \leq 1$ . A model whose predictions are 100% wrong has an AUC of 0.0; one whose predictions are 100% correct has an AUC of 1.0.
- ▶ The largest the best.
- ✓ AUC is scale invariant. It measures how well predictions ( $\widehat{P(Y=1)}$ ) are ranked, rather than their absolute values.
- ✓ AUC is threshold invariant. It measures the quality of the model's predictions irrespective of what classification threshold is chosen.
- ✗ Be carrefull : AUC is defined for a 2 classes problems ; as welle as other metrics!

## QUIZZ - Question

In which of the following scenarios would a high accuracy value suggest that the ML model is doing a good job?

1. A deadly, but curable, medical condition afflicts .01% of the population. An ML model uses symptoms as features and predicts this affliction with an accuracy of 99.99%.
2. In the game of roulette, a ball is dropped on a spinning wheel and eventually lands in one of 38 slots. Using visual features (the spin of the ball, the position of the wheel when the ball was dropped, the height of the ball over the wheel), an ML model can predict the slot that the ball will land in with an accuracy of 4%.
3. An expensive robotic chicken crosses a very busy road a thousand times per day. An ML model evaluates traffic patterns and predicts when this chicken can safely cross the street with an accuracy of 99.99%.

## QUIZZ - Response

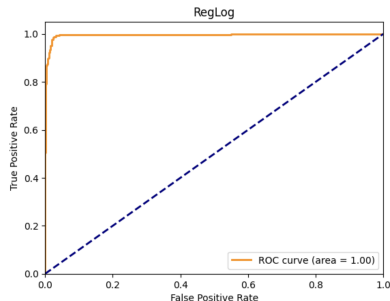
In which of the following scenarios would a high accuracy value suggest that the ML model is doing a good job?

1. A deadly, but curable, medical condition afflicts .01% of the population. An ML model uses symptoms as features and predicts this affliction with an accuracy of 99.99%.
2. In the game of roulette, a ball is dropped on a spinning wheel and eventually lands in one of 38 slots. Using visual features (the spin of the ball, the position of the wheel when the ball was dropped, the height of the ball over the wheel), an ML model can predict the slot that the ball will land in with an accuracy of 4%.  
*This ML model is making predictions far better than chance; a random guess would be correct 1/38 of the time-yielding an accuracy of 2.6%. Although the model's accuracy is "only" 4%, the benefits of success far outweigh the disadvantages of failure.*
3. An expensive robotic chicken crosses a very busy road a thousand times per day. An ML model evaluates traffic patterns and predicts when this chicken can safely cross the street with an accuracy of 99.99%.

# Logistic regression, example

- ▶ Leukemia gene expression
- ▶ Model based on the genes with highest absolute correlation with class AML/ALL ( $|\rho| > .5$ )
- ▶ n.train = 28, n.test = 10, 100 MC repetitions

Accuracy	0.966
Precision	0.964
Recall.	0.907
F1-score	0.935
AUC	0.947



code <https://colab.research.google.com/drive/1TMnolN73yYGhFi-nuoy6ptaW3HVc626n?usp=sharing>

## Multi class problems and Bayesian decision rule

- ▶ In classification problems,  $Y$  takes its values in  $\{1, \dots, K\}$  with probabilities  $\pi_1, \dots, \pi_K$ .
- ▶  $\pi_\ell = P(Y = \ell)$  correspond to the *a priori probabilities* of classes (ex: patient/control).
- ▶ Now let us suppose that the predictors  $\mathbf{X}$  have probability density functions

$$f_\ell(\mathbf{x}) = P[\mathbf{X} | Y = \ell]$$

in each class  $\ell$ .

- ▶ The decision rule of LDA is based on the *posterior probabilities*

$$P(Y = \ell | \mathbf{X} = \mathbf{x})$$

### ▶ Bayes rule

A sample corresponding to observation  $\mathbf{x}$  is classified in class  $\ell$  if

$$P(Y = \ell | \mathbf{X} = \mathbf{x}) \geq P(Y = k | \mathbf{X} = \mathbf{x}) \text{ for all } k \in \{1, \dots, K\}$$

- ▶ The Bayes rule is optimal for the classification error (=1-accuracy).
- ▶ A simple application of Bayes theorem Bayes formula gives us

$$P(Y = \ell | \mathbf{X} = \mathbf{x}) = \frac{\pi_\ell f_\ell(\mathbf{x})}{\sum_{k=1}^K \pi_k f_k(\mathbf{x})}$$

## Linear discriminant analysis : a Gaussian interpretation of Bayes rule

- ▶ The **linear discriminant analysis** (LDA) is obtained when the densities  $f_\ell(\cdot)$  are Gaussian with the same covariance matrix  $\Sigma$  in all classes.

$$f_\ell(\mathbf{x}) = \frac{(2\pi)^{d/2}}{\det(\Sigma)^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_\ell)^T \Sigma^{-1}(\mathbf{x} - \mu_\ell)\right)$$

- ▶ The Bayes rule leads to a linear boundaries:

$$\begin{aligned} \log \frac{P(Y = k | \mathbf{X} = \mathbf{x})}{P(Y = \ell | \mathbf{X} = \mathbf{x})} &= \log \frac{f_k(\mathbf{x})}{f_\ell(\mathbf{x})} + \log \frac{\pi_k}{\pi_\ell} \\ &= \log \frac{\pi_k}{\pi_\ell} - \frac{1}{2}(\mu_k + \mu_\ell)^T \Sigma^{-1}(\mu_k - \mu_\ell) \\ &\quad + \mathbf{x}^T \Sigma^{-1}(\mu_k - \mu_\ell). \end{aligned}$$

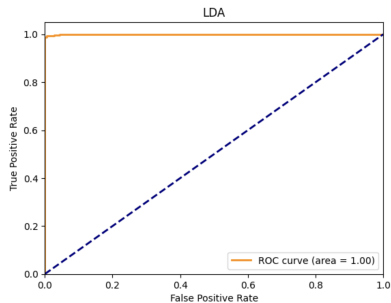
This equation is linear in  $\mathbf{x}$ .

- ▶ The **quadratic discriminant analysis** (QDA) is obtained when the densities  $f_\ell(\cdot)$  are Gaussian with the different covariance matrices  $\Sigma_\ell$  in each classe.
- ▶ **Naive Bayes** models are variants of the discriminant analysis, and assume that each of the class densities are products of marginal Gaussian densities.

# LDA, example

- ▶ Leukemia gene expression
- ▶ Model based on
- ▶ the genes with highest absolute correlation with class AML/ALL ( $|\rho| > .5$ )
- ▶ n.train = 28, n.test = 10, 100 MC repetitions

	RL	LDA
Accuracy	0.966	0.978
Precision	0.964	1.00
Recall.	0.907	0.919
F1-score	0.935	0.958
AUC	0.947	0.960



## Multiclass model validation

✗ The scores or metrics defined above are defined for two classes.

No	Actual	Predicted	Match
1	Airplane	Airplane	✓
2	Car	Boat	✗
3	Car	Car	✓
4	Car	Car	✓
5	Car	Boat	✗
6	Airplane	Boat	✗
7	Boat	Boat	✓
8	Car	Airplane	✗
9	Airplane	Airplane	✓
10	Car	Car	✓

Upon running `sklearn.metrics.classification_report`, we get the following classification report:

	precision	recall	f1-score	support
Aeroplane	0.67	0.67	0.67	3
Boat	0.25	1.00	0.40	1
Car	1.00	0.50	0.67	6
accuracy			0.60	10
macro avg	0.64	0.72	0.58	10
weighted avg	0.82	0.60	0.64	10

Per-Class F1 scores

Average F1 scores

Source <https://towardsdatascience.com/>

micro-macro-weighted-averages-of-f1-score-clearly-explained-b603420b292f/

## Multiclass model validation

- ✓ Generalisation for more than two classes under various conventions.

### ► Macro-average

Label	Per-Class F1-score	Macro-Avg F1-score
Airplane	0.67	$\frac{0.67+0.40+0.67}{3}$
Boat	0.40	
Car	0.67	
		<b>=0.58</b>

### ► Weighted Averaged

Label	Per-Class F1-score	Support Proportion	Macro-Avg F1-score
Airplane	0.67	0.3	$(0.67 * 0.3) +$ $(0.40 * 0.1) +$ $(0.67 * 0.6) +$ $= \mathbf{0.64}$
Boat	0.40	0.1	
Car	0.67	0.6	

### ► Micro average

Label	TP	FP	FN	Micro-avg F1-score
Airplane	2	1	1	$\frac{TP}{TP + \frac{1}{2}(FN + FP)}$
Boat	1	3	0	
Car	3	0	3	
				<b>=0.60</b>

## Recap

- ▶ **Linear model** is used to predict a quantitative variable  $Y$  by a linear combination of the  $X$  variables.  
 $X$  variables can be quantitative or qualitative.
- ▶ The estimators are easy to computed. When  $p < n$ , it usually gives a good first approximation of the relationship between  $X$  and  $Y$ .
- ▶ **If  $Y$  is qualitative**, generalized linear model are considered such as logistic regression.
- ▶ Linear discriminant analysis is a reduction dimension like technique which leads to linear frontiers between classes defined by  $Y$ .  
Quadratic discriminant analysis is an extension for quadratic frontiers.
- ▶ **Validation** is based on RMSE/MAE if  $Y$  is quantitative and classification error/confusion matrix/F1-score/AUC/ROC if  $Y$  is qualitative.
- ▶ For **multiclass prediction**, take care to the per-class metrics averaging.
- ▶ Whatever the case, cross validation has to be used for fair model validation.

# Produit matrice-vecteur

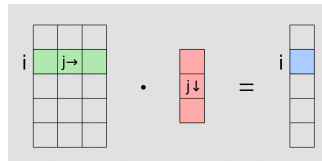
$$\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p = \mathbf{X}\beta$$

Let  $\mathbf{X}$  be defined as

$$\mathbf{X} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ \vdots & \vdots & & \vdots \\ 1 & x_{i1} & \cdots & x_{ip} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix}$$

then the line  $i$  of  $\mathbf{X}\beta$  is

$$(\mathbf{X}\beta)_i = \beta_0 + \beta_1 X_{i1} + \cdots + \beta_p X_{ip}$$



## Differentiation of the quadratic form

Firstly note that

$$(y - \mathbf{x}\beta)^T (y - \mathbf{x}\beta) = y^T y + y^T \mathbf{x}\beta + \beta^T \mathbf{x}^T y + \beta^T \mathbf{x}^T \mathbf{x}\beta$$

now because  $\mathbf{x}\beta$  and  $y$  are vectors, one has

$$y^T \mathbf{x}\beta = \beta^T \mathbf{x}^T y$$

so that

$$(y - \mathbf{x}\beta)^T (y - \mathbf{x}\beta) = y^T y + 2y^T \mathbf{x}\beta + \beta^T \mathbf{x}^T \mathbf{x}\beta$$

1st left term does not depend on  $\beta$ , 2nd one depends linearly and the 3rd one is quadratic in  $\beta$ . And when, the derivative is calculated with respect to  $\beta$

$$\frac{\partial (y - \mathbf{x}\beta)^T (y - \mathbf{x}\beta)}{\partial \beta} = 2y^T \mathbf{x} + 2\mathbf{x}^T \mathbf{x}\beta$$

## Bayes formula

The general Bayes formula is

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

In the context of discriminant analysis,

$$P(Y = \ell | \mathbf{X} = \mathbf{x}) = \frac{P(Y = \ell \text{ and } \mathbf{X} = \mathbf{x})}{P(\mathbf{X} = \mathbf{x})}$$

and, by definition,  $P(Y = \ell) = \pi_\ell$ ,  $P(\mathbf{X} = \mathbf{x} | Y = \ell) = f_\ell(\mathbf{x})$

Now, by applying the Bayes formula  $P(A \cap B) = P(A|B)P(B)$ ,

$$P(Y = \ell \text{ and } \mathbf{X} = \mathbf{x}) = P(\mathbf{X} = \mathbf{x} | Y = \ell)P(Y = \ell) = f_\ell(\mathbf{x})\pi_\ell$$

And, from the total probabilities formula,

$$\begin{aligned} P(\mathbf{X} = \mathbf{x}) &= \sum_{k=1}^K P(\mathbf{X} = \mathbf{x} | Y = k)P(Y = k) \\ &= \frac{\pi_\ell f_\ell(\mathbf{x})}{\sum_{k=1}^K \pi_k f_k(\mathbf{x})} \end{aligned}$$

# Outline

Supervised learning

Linear model (I)

**Linear model (II)**