# Data Analysis

## Réduction de dimensions

## ACP

- $X = \begin{bmatrix} & x_{ij} & \\ & \downarrow i & \\ & j & \end{bmatrix}$ $\quad y = [\overline{x_1} \dots \overline{x_j}]$ $\quad \overline{x_j} = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$
  $\overline{x_j} \in \mathbb{R}^p$

- centrer les données $X_c = \left[ x_j - \overline{x_j} \right] \in \mathbb{R}^{n \times p}$

- matrice de variance-covariance : $\Sigma = \frac{1}{n} X_c^T X_c$

- mesure de corrélation : $cor(X,Y) = \frac{cov(X,Y)}{\sqrt{var(X)\, var(Y)}}$

$$cov(C_j, C_\ell) = E\left( (C_j - E(C_j))(C_\ell - E(C_\ell)) \right)$$
$$= \frac{1}{n}\sum_{i=1}^{n} (x_{ij} - \overline{x_j})(x_{i\ell} - \overline{x_\ell})$$

$$var(C_j) = \frac{1}{n}\sum_{i=1}^{n} (x_{ij} - \overline{x_j})^2$$

$$\Sigma = \frac{1}{N-1} X_c^T X_c$$

$$\Sigma = \begin{bmatrix} var X & cov(X,Y) \\ cov(X,Y) & var Y \end{bmatrix}$$

$-1 \leq cor(X,Y) \leq 1$

- $0 \to$ X, Y indép, info à garder
- $\pm 1 \to$ X, Y dép lin.

- but ACP : trouver q composantes principales $C_1, \dots, C_q$ avec $q \ll p$ comme des nouvelles variables combi. lin. des var d'origine $x_1, \dots, x_p$ tq les $C_\ell$ soient 2 à 2 non corrélées, de variance maximale, d'importance décroissante.
  ↳ maximiser la dispersion pour garder les caract.

exo :

| Personne | Capteur 1 | Capteur 2 |
|---|---|---|
| Ind 1 | 0 | 2 |
| Ind 2 | -2 | -1 |
| Ind 3 | 1 | 0 |
| Ind 4 | 1 | -1 |

1) Existe-t-il une dépendance entre ces 2 capteurs ?

$$cor(C_1, C_2) = \frac{\frac{1}{4}\left[(0-0)(2-0) \dots (1-0)(-1-0)\right]}{\sqrt{\frac{(0-0)^2 + (-2-0)^2 + (1-0)^2 + (1-0)^2}{4}} \times \frac{\dots}{4}} = \frac{1/4}{\sqrt{\frac{6}{4}\cdot\frac{6}{4}}} = \frac{1}{6}$$

$\exists$ dép non lin.

2) Calculer le 1er axe princip de ces pts.

$$X = \begin{bmatrix} 0 & 2 \\ -2 & -1 \\ 1 & 0 \\ 1 & -1 \end{bmatrix} = X_c \qquad \Sigma = \frac{1}{4} X_c^T X_c = \frac{1}{4}\begin{bmatrix} 6 & 1 \\ 1 & 6 \end{bmatrix}$$

$$\chi_{4\Sigma}(\lambda) = \det(4\Sigma - \lambda I_2) = \begin{bmatrix} 6-\lambda & 1 \\ 1 & 6-\lambda \end{bmatrix} = (6-\lambda)^2 - 1 = (7-\lambda)(5-\lambda)$$

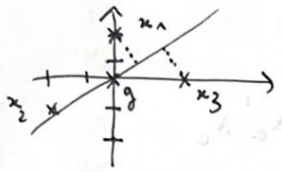
↳ $\lambda_1 = 7$, $\lambda_2 = 5$
↑ + grande VP

Soit $X_1 = \begin{bmatrix} x \\ y \end{bmatrix}$ $\vec{vp}$ de $\Sigma$ associé à $d_1 = 7$.

$4\Sigma X_1 = d_1 X_1 \iff \begin{bmatrix} 6 & 1 \\ 1 & 6 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 7 \begin{bmatrix} x \\ y \end{bmatrix}$

$\iff \begin{cases} 6x + y = 7x \\ x + 6y = 7y \end{cases} \iff x = y$

$X_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$




$g = [0\ 0]$

1ère Composante Principale :
(coeff de projection des données sur l'axe principal)

$C = XX_1 = \begin{bmatrix} 0 & 2 \\ -2 & -1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ -3 \\ 1 \\ 0 \end{bmatrix}$

## Analyse de données temporelles

**DTW** : déformation temporelle dynamique, permet de mesurer la similarité entre 2 suites qui peuvent varier au cours du temps

- basé sur le principe du + court chemin de Dijkstra
  + effet temporel pris en compte
- évaluer la dissemblance entre 2 séq → détermine un chemin

(pptés : coïncidence extrémités + monotonie & croissance temporelle + continuité)

Input : . 2 séries temporelles A et B de supports temporels resp. $I = [\![1, N]\!]$ et $J = [\![1, M]\!]$
· distance locale d

Algo :

1. Initialisat° : $\omega_0 \leftarrow 1$, $\omega_1 \leftarrow 2$, $\omega_2 \leftarrow 1$, $g(0,0) \leftarrow 0$
2. Pour $j \in \{1 \dots J\}$
   (a) $g(0,j) \leftarrow +\infty$
   (b) Pour $i \in [1,..N]$
   $g(i,0) \leftarrow +\infty$
   Pour $j \in \{1 \dots M\}$
   Recherche du chemin minimal
   $g(i,j) \leftarrow \min \begin{cases} g(i-1,j) + \omega_0 * d(i,j) \\ g(i-1,j-1) + \omega_1 * d(i,j) \\ g(i,j-1) + \omega_2 * d(i,j) \end{cases}$
3. calcul du score d'alignement
   $S = g(N,M) \,/\, (N+M)$

Output : score S et chemin optimal C.

exo :

$d(i,j) = x_i - x_j$

| g | 0 | 1 | 2 | -1 |
|---|---|---|---|---|
| | 0 | + | ∞ | |
| 0 | 0 → | 1 → | 3 → | 4 |
| 2 | 8 ↓ 2 | ↘ 2 | ↘ 1 | 4 |
| 4 | + 6 | 5 | 3 → | 8 |

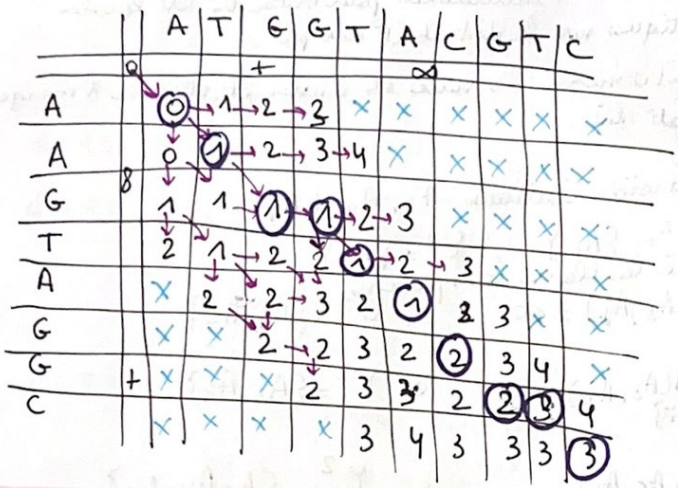
Score = $\frac{8}{3+4} = \frac{8}{7}$

exo Séquences ADN . $d(L_i, L_j) = \begin{cases} 0 & \text{si } L_i = L_j \\ 1 & \text{sinon} \end{cases}$

contraintes locales : $(\omega_0, \omega_1, \omega_2) = (1,1,1)$

globales : les cases à + de 4 cases de la diago ne sont pas calculées.

Matrice de coût :




$S = \frac{3}{10+8} = \frac{3}{18}$

## Classification non supervisée

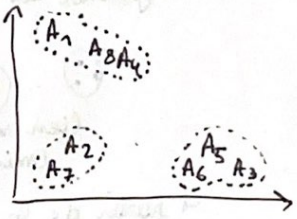
* Approches par partitionnement :

K-ppv : classification non supervisé
k - plus proche voisins

exo : $A_1 = (2,10)$ $A_2 = (2,5)$ $A_3 = (8,4)$ $A_4 = (5,8)$
$A_5 = (7,5)$ $A_6 = (6,4)$ $A_7 = (1,2)$ $A_8 = (4,9)$

- Matrice des distances euclidiennes au carré :

| | $A_1$ | $A_2$ | $A_3$ | $A_4$ | $A_5$ | $A_6$ | $A_7$ | $A_8$ |
|---|---|---|---|---|---|---|---|---|
| $A_1$ | 0 | 25 | 72 | 13 | 50 | 52 | 65 | 5 |
| $A_2$ | | 0 | 37 | 18 | 25 | 17 | 10 | 20 |
| $A_3$ | | | 0 | 25 | 2 | 4 | 53 | 41 |
| $A_4$ | | | | 0 | 13 | 17 | 52 | 2 |
| $A_5$ | | | | | 0 | 2 | 45 | 25 |
| $A_6$ | | | | | | 0 | 29 | 29 |
| $A_7$ | | | | | | | 0 | 58 |
| $A_8$ | | | | | | | | 0 |




- K ppv avec le seuil $t = 16$ et $k = 1$ :

Pour $A_1$ : $d(A1, A8) = \min_{j \in \{2 \dots 8\}} d(A_1, A_j) = 5 < t$ donc $\mathcal{C}^1 = \{A_1 ; A_8\}$

Pour $A_2$ : $d(A2, A7) = \min_{\substack{j \in \{1 \dots 8\} \\ j \neq 2}} d(A_2, A_j) = 10 < t$ donc $\mathcal{C}^2 = \{A_2 ; A_7\}$

$\mathcal{C}^1 = \{A_1 ; A_4 ; A8\}$ $\mathcal{C}^2 = \{A_2 ; A_7\}$ $\mathcal{C}^3 = \{A_3 ; A_5 ; A_6\}$

**Kmeans** : Partitionner un ensemble de données en k classes représentées par les k centres, notés $\mathcal{C} = \{\mathcal{C}^1 \dots \mathcal{C}^k\}$
On associe alors à chq pt/donnée le centre le + proche au sens d'une certaine distance.

⚠ ◦ Bien choisir les centres à l'initialisat° pour éviter les CV locales
↳ Heuristiques sur la distribut° des pts

Nb de classes à déterminer → varie nb classes et étudier l'énergie
Méthode de séparat° lin.

exo K-means avec centroïdes initiaux $A_1$, $A_4$ et $A_7$.

$\mathcal{C}^1 = \{A_1\}$ $\mathcal{C}^2 = \{A_4\}$ $\mathcal{C}^3 = \{A_7\}$

◦ Assigner à chaque donnée la classe la + proche

$d(A_2, A_7) = \min_{j \in \{1,4,7\}} d(A_2, A_j) = 10 \rightarrow \mathcal{C}^3 = \{A_2, A_7\}$

$d(A_3, A_4) = \min_{j \in \{1,4,7\}} d(A_3, A_j) \rightarrow \mathcal{C}^2 = \{A_3, A_4\}$

$d(A_5, A_4) = \min_{j \in \{1,4,7\}} d(A_5, A_j) \rightarrow \mathcal{C}^2 = \{A_3, A_4, A_5\}$

$d(A_6, A_4) = \min_{j \in \{1,4,7\}} d(A_6, A_j) \rightarrow \mathcal{C}^2 = \{A_3, A_4, A_5, A_6\}$

$d(A_8, A_4) = \min_{j \in \{1,4,7\}} d(A_8, A_j) \rightarrow \mathcal{C}^2 = \{A_3, A_4, A_5, A_6, A_8\}$

◦ Mise à jour des centres : $m_1 = (2, 10)$

$m_2 = \left(\frac{1}{5}(8+5+7+6+4), \frac{1}{5}(4+8+5+4+9)\right)$
$= (6, 6)$

$m_3 = \left(\frac{2+1}{2}, \frac{5+2}{2}\right) = (1{,}5 ; 3{,}5)$

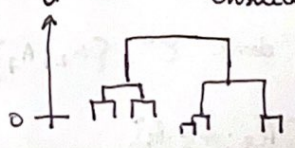
## Classification hiérarchique

↳ pas de cluster prédéfini + pas de config initiale + bonne interprétabilité

↳ fournir la mesure de dissimilarité pour comparer les groupes d'obs.


↳ partir de la granularité la + fine ( 1 classe = 1 donnée) puis stratégie ascendante / agglomératives.

↳ Avantage : visualiser comment les groupes sont créés (explicabilité)
ensuite à nous de voir où c'est le mieux de couper (n° classes)
↳ similaire à l'arbre de décision




← dendrogramme

exo :

| | A | B | C | D |
|---|---|---|---|---|
| A | 0 | 1 | 4 | 5 |
| B | | 0 | 2 | 6 |
| C | | | 0 | 3 |
| D | | | | 0 |

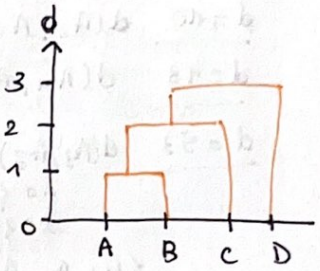
1) Lien simple

$d=0$ $\{A\}$ $\{B\}$ $\{C\}$ $\{D\}$

$d=1$ $d(A,B)=1$ $\{A,B\}$ $\{C\}$ $\{D\}$

$d=2$ $d(B,C)=2$ $\{A,B,C\}$ $\{D\}$

$d=3$ $d(C,D)=3$ $\{A,B,C,D\}$




2) Lien complet

$d=0$ $\{A\}$ $\{B\}$ $\{C\}$ $\{D\}$

$d=1$ $d(A,B)=1$ $\{A,B\}$ $\{C\}$ $\{D\}$ → pour les singletons prendre le min

$d=2$ $d(B,C)=2 < d(C,A)=4$

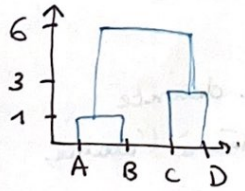
prendre la + grande distance

↳ il faudra regarder si C fusionne avec $\{A,B\}$ à $d=4$

$d=3$ $d(C,D)=3$ $\{A,B\}$ $\{C,D\}$

on va fusionner que pour la + grande distance ie pour $d(B,D)=6$

$d=6$ $\{A,B,C,D\}$




exo avec les $A_i$ $1 \le i \le 8$

1) Lien simple

$d=0$ $\{A_1\}$ $\{A_2\}$ ... $\{A_8\}$

$d=2$ $d(A_3,A_5)=d(A_5,A_6)=d(A_4,A_8)=2$

$\{A_1\}$ $\{A_2\}$ $\{A_3,A_5,A_6\}$ $\{A_4,A_8\}$ $\{A_7\}$

$d=4$ $d(A_3,A_6)=4$

$d=5$ $d(A_1,A_8)=5$ $\{A_1,A_4,A_8\}$ $\{A_2\}$ $\{A_3,A_5,A_6\}$ $\{A_7\}$

$d=10$ $d(A_2,A_7)=10$ $\{A_1,A_4,A_8\}$ $\{A_2,A_7\}$ $\{A_3,A_5,A_6\}$

$d=13$ $d(A_1,A_4)=d(A_4,A_5)=13$ $\{A_1,A_3,A_4,A_5,A_6\}$ $\{A_2,A_7\}$

$d=17$ $d(A_2,A_6)=d(A_4,A_6)=17$

2) Lien complet

$d = 0$ $\{A_1\}$ $\{A_2\}$ ⋯ $\{A_8\}$

$d = 2$ $d(A_3, A_5) = d(A_5, A_6) = d(A_4, A_8) = 2$

$\{A_1\}$ $\{A_2\}$ $\{A_3, A_5, A_6\}$ $\{A_4, A_8\}$ $\{A_7\}$

$d = 4$ $d(A_3, A_6) = 4$

$d = 5$ $d(A_1, A_8) = 5 < d(A_4, A_1) = 13$

$d = 10$ $d(A_2, A_7) = 10$ $\{A_1\}$ $\{A_2, A_7\}$ $\{A_3, A_5, A_6\}$ $\{A_4, A_8\}$

$d = 13$ $d(A_4, A_1) = 13$ $\{A_1, A_4, A_8\}$ $\{A_2, A_7\}$ $\{A_3, A_5, A_6\}$

$d = 53$ $d(A_3, A_7) = \max d(A_i, A_j)$

$i = \{3, 5, 6\}$

$j = \{2, 7\}$

$\{A_2, A_3, A_5, A_6, A_7\}$ $\{A_1, A_4, A_8\}$

72 ($\max = d(A_1, A_3)$)

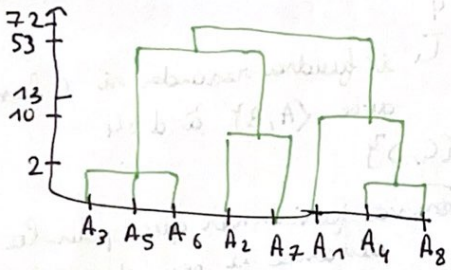
65 ($\max = d(A_1, A_7)$)

53 ($\max = d(A_3, A_7)$)

$d = 72$ $d(A_1, A_3) = \max d(A_i, A_j)$

$i = \{2, 3, 5, 6, 7\}$

$j = \{1, 4, 8\}$




## DBSCAN

↳ 1 classe associée à 1 densité

↳ classes non déterminées à l'avance

↳ 2 param : · rayon pour voir $\varepsilon$
· nb de pts min pour 1 cluster (densité)

↳ pts mon assignés à une classe = bruit
++ filtrage

glouton


exo : • $\varepsilon = 4$ , MinPts = 2

$\{A_1\}$ $\{A_2\}$ $\{A_3, A_5, A_6\}$ $\{A_4, A_8\}$ $\{A_7\}$

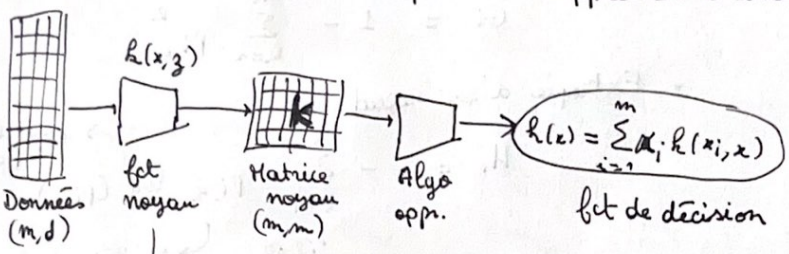
$d(A_3, A_5) = d(A_5, A_6) = 2$

$\forall j \neq i$ $i \in (1, 2, 7\}$ $d(A_i, A_j) \geq 5$ | $d(A_4, A_8) = 2$

• $\varepsilon = 10$ $\{A_2, A_7\}$ $\{A_3, A_5, A_6\}$ $\{A_1, A_4, A_8\}$

## Approche par noyaux

↳ au lieu de séparer avec des hyperplans comme k-means, augmenter la dimension pour se rapprocher d'une séparat° non lin




↓ fct sym pos

| | |
|---|---|
| polynomial kernel | $k(a,b) = (a.b+c)^d$ |
| gaussian ——— | $k(a,b) = \exp(-\|\|a-b\|\|^2 / 2\sigma^2)$ |
| sigmoid ——— | $k(a,b) = \tanh(c.(a.b)+\theta)$ |

## Graphe et modularité

...

## Critères d'évaluation en classification non supervisée

↳ basé sur 2 notions : cohésion & séparation

↳ SSE ou SSW : variance intra-classe → cohésion → données proches (SSE)

↳ BSS : variance inter-classe → séparation → à quel point la dist. est grande (BSS)

↳ mix de cohésion et séparation : silhouette ☺

$$s(x) = \begin{cases} \dfrac{b(x) - a(x)}{\max(a(x), b(x))} & \text{si } |C_i| > 1 \\ 0 & \text{si } |C_i| = 1 \end{cases}$$

>0 : séparat° élevée entre cluster
<0 : supperpost° de classes
0 : distrib uniforme

## Apprentissage supervisé

Modèles :
- arbres de décision
- apprentissage d'ensemble : forêts aléatoires
- réseaux de neurones
- SVM

Méthodes d'évaluation :
- validation croisée
- matrice de confusion
- précision, rappel, F-mesure
- courbe ROC

# Arbres de décision

↳ mesures d'impureté

- Indice de Gini d'un noeud $i$ comportant $n$ éléments :

$$G_i = 1 - \sum_{k=1}^{n} P_{i,k}^2$$

- Entropie d'un noeud $i$ :

$$H_i = - \sum_{\substack{k=1 \\ P_{i,k} \neq 0}}^{n} P_{i,k} \log(P_{i,k})$$

→ part d'info de la classe (faible qd redondante)

↳ % d'obs de la classe $k$ parmi toutes les obs d'entraînement i^e noeud. Si c'est redondant, très fort

↳ Si $G_i$ ou $H_i = 0$ → noeud pur ⇒ qu'une seule classe

· Gini est plus rapide à calculer

· Gini a tendance à isoler les classes les + fréquentes / Entropie produit des arbres + équilibrés

↳ crée des frontières de décision + donne des probabilités d'appartenance (interprétabilité)

↳ Algorithme CART

1) Séparation entraînement en 2
2) Choix du couple $(k, t_k)$ qui produit les sous-ensembles les + purs.
3) Fonction de coût J à minimiser $J(k, t_k) = \frac{m_G}{m} G_{gauche} + \frac{m_D}{m} G_{droite}$
4) Récursion
5) Critère d'arrêt : profondeur max atteinte (hyperparam)
noeuds purs → éviter overfitting

↳ Avantage : utiliser var quantitative et qualitative

classification + regression
(prédict° classe) (prédict° valeur)

exo

| Match à domicile | Ciel | Match préc. gagné? | Match gagné? |
|---|---|---|---|
| oui | soleil | oui | oui |
| oui | pluie | non | non |
| oui | soleil | non | oui |
| non | couvert | oui | oui |
| non | pluie | oui | oui |
| non | soleil | non | non |

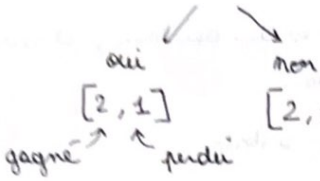
1) Gini de base : indice de Gini pour "Gagner le match" et "Perdre le Match"

$$Gini(\text{Match gagné ?}) = 1 - \left(\frac{\#\text{oui}}{\#\text{total}}\right)^2 - \left(\frac{\#\text{non}}{\#\text{total}}\right)^2 = 1 - \left(\frac{4}{6}\right)^2 - \left(\frac{2}{6}\right)^2$$

$$= 1 - \frac{4}{9} - \frac{1}{9} = \frac{4}{9}$$

2) Déterminer la variation de l'indice de Gini lorsqu'on découpe les données à l'aide des vars. En déduire celle du 1er niv. de l'arbre.

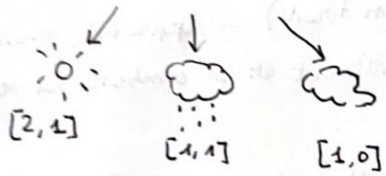
* Variable "Match à domicile"




$Gini_{oui} = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 = \frac{4}{9}$ $Gini_{non} = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 = \frac{4}{9}$

$\Rightarrow Gini(\text{"Match à domicile"}) = \frac{3}{6} Gini_{oui} + \frac{3}{6} Gini_{non} = 0.5 \times 2 \times \frac{4}{9} = \frac{4}{9}$

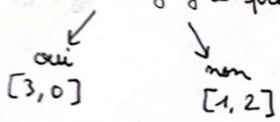
* Variable "Ciel"




$Gini_{☀} = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 = \frac{4}{9}$
$Gini_{☁☀} = 1 - \left(\frac{1}{2}\right)^2 - \left(\frac{1}{2}\right)^2 = \frac{1}{2}$
$Gini_{☁} = 1 - (1)^2 - 0 = 0$

$Gini(\text{"Ciel"}) = \frac{3}{6} Gini_{☀} + \frac{2}{6} Gini_{☁☀} + \frac{1}{6} Gini_{☁}$
$= \frac{3}{6} \cdot \frac{4}{9} + \frac{2}{6} \cdot \frac{1}{2} + \frac{1}{6} \cdot 0 = \frac{7}{18}$

* Variable "Match gagné précédent"

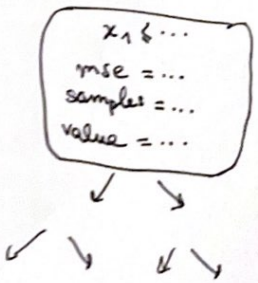



$Gini(\text{"Match préc. gagné"}) = \frac{3}{6} Gini_{oui} + \frac{3}{6} Gini_{non}$
$= \frac{3}{6}\left(1 - \left(\frac{3}{3}\right)^2 - 0\right) + \frac{3}{6}\left(1 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2\right) = \frac{3}{6} \cdot \frac{4}{9} = \frac{2}{9}$

* 1er split ?

$\underline{Gini(\text{Match préc gagné})}$ < $Gini(\text{Ciel})$ < $Gini(\text{Match domicile})$
$\frac{2}{9}$ $\frac{7}{18}$ $\frac{4}{9} = Gini(\text{base})$ ← inutile car ne minimise pas

**Régression par arbre de décision :** prédiction d'une valeur
comparer les valeurs moy → on obtient une MSE



## Apprentissage par ensembles

↳ entraînement d'un ensemble d'arbres de décision, chacun sur un sous-ensemble aléatoire du jeu de train

↳ calcul des prédictions pour chacun des arbres

↳ choisir la classe obtenant le + de vote (classificat°) / la moyenne des résultats (régression)

⇒ forêt aléatoire

↳ 3 catégories d'apprentissage par ensembles

- bagging (Random Forest) → apprend, en //, indép, des modèles de base qui le constituent et les combine en suivant un processus de moyenne
- boosting (XGBoost) → apprend séquentiellement de manière adaptative (se concentre sur les erreurs) et combine selon strat.
- stacking → apprend en // et combine les modèles de base en un méta modèle

(bagging : ⊖ de variance / boosting et stacking : ⊖ biais)