Méthodes de “Data Mining” et d’Intelligence Collective

Données sous forme brute

Outils d’analyse et de prévision

Tests effectués sur une large base de données conçue dans ce but :

1500 modèles de **voitures** évalués selon plusieurs **critères** et assortis d’une **note globale**

N°	Prix	Entretien	Portes	Places	Coffre	Sécurité	Note
1	élevé	bas	3	2	petit	élevée	**bonne**
2	bas	moyen	3	2	moyen	moyenne	**bonne**
3	élevé	élevé	5	6	moyen	basse	moyenne
4	moyen	moyen	4	6	large	basse	**mauvaise**
5	moyen	élevé	5	4	large	moyenne	moyenne
6	moyen	élevé	5	4	large	moyenne	moyenne
7	bas	bas	3	4	petit	élevée	moyenne
8	élevé	bas	5	4	moyen	moyenne	**bonne**
9	bas	élevé	5	6	petit	basse	**mauvaise**
10	moyen	moyen	4	4	moyen	basse	moyenne

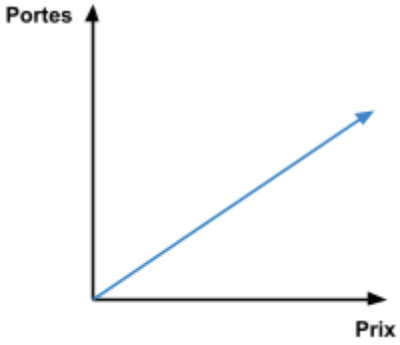
Problème : prévoir la **note probable** que des clients attribueraient à un **nouveau modèle**

I. Une première approche du problème

Méthode “naïve” par **proximité géométrique** de **vecteurs** en **dimension p**
(où p = nombre de critères)

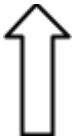
Critère : Prix

Prix	Abscisse
élevé	1
moyen	0,66
bas	0,33



Critère : Portes

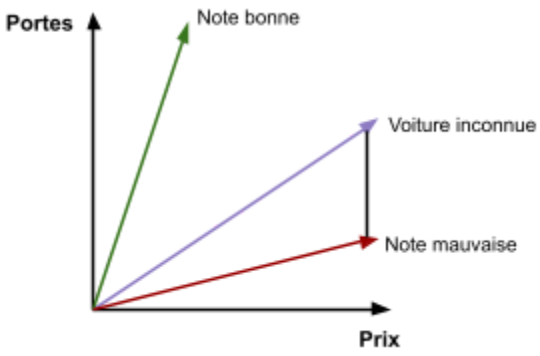
Portes	Ordonnée
5	1
4	0,66
3	0,33



élevé	élevé	4	6	moyen	basse	?

Trois **vecteurs “moyens”** établis (un pour chaque note possible)

Note d’une voiture prédite par “proximité” selon la **norme 2** dans $\Re^{p}$

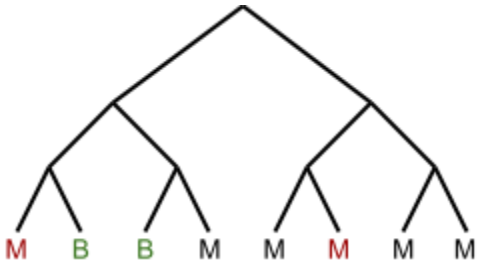


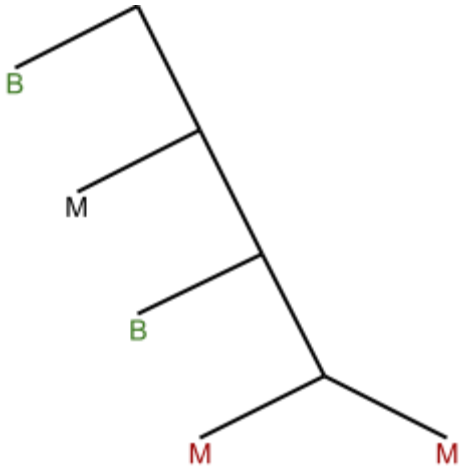
II. Une autre méthode : l'arbre de décision

Recherche d'une **classification optimale**

Méthode : division de l'ensemble grâce à des "questions" discriminantes

Problème : comment construire un arbre **efficace**?





Lequel des deux s'avère le plus **pertinent et efficace**?

Un outil pour cela : l'**entropie**

Deviner un nombre entre **1** et **100**, deux façons :

1? 2? … 100? 100 Questions au maximum.	>50? >75? >25? … 100 1 $log_2(100)$ Questions au maximum.

L'entropie est la **mesure** de la **quantité d'information** restant à déterminer.

Algorithme

On cherche à **maximiser** la quantité d'information obtenue à chaque étape.

Entropie de Shannon :

$$e = \sum_{i=1}^{n} p_i log_2(p_i)$$

Ensemble d'apprentissage

|
v

Calcul de l'entropie initiale **e**

|
v

Division selon tous les critères possibles (Prix Bas, Coffre Large,...)

|
v

Calcul de l'entropie pondérée **e'** pour chaque division

$$\frac{e_1 \times l_1 + e_2 \times l_2}{l_1 + l_2}$$

|
v

Choix de la division réalisant le meilleur gain d'entropie g_{max}

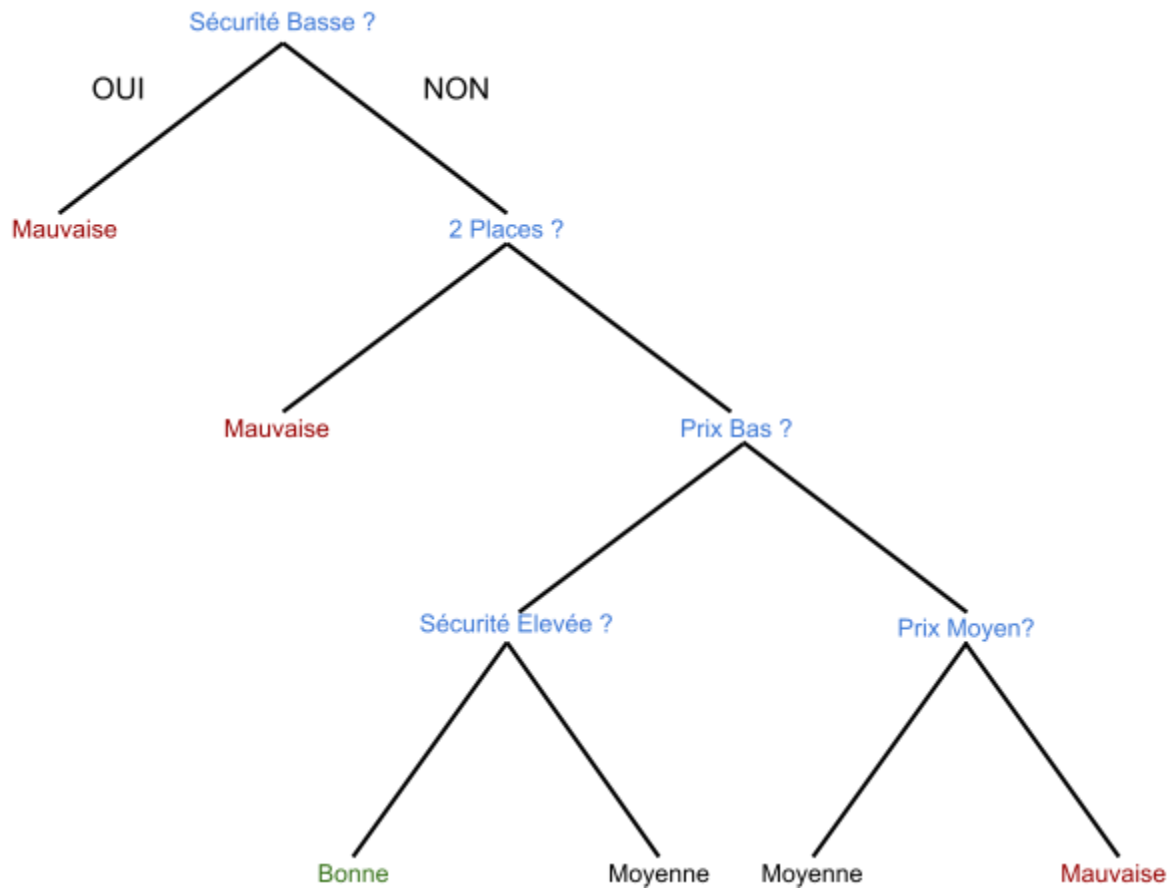
max (e - e')

|
v

Rappel récursif sur les deux sous-ensembles obtenus par cette division

Résultat et comparaison des algorithmes

Arbre obtenu après **élagage**, à partir d'un échantillon de **50** voitures de note connue



Mise en œuvre des deux algorithmes sur un échantillon d'apprentissage identique de 50 voitures

Précision des prévisions obtenues sur les **1500** autres voitures :

Vecteurs : **56%** *de prévisions justes*
Arbre : **73%** *de prévisions justes*

Autres avantages de l'arbre de décision :

- Apparition **claire** des **critères pertinents** de prévision de la note

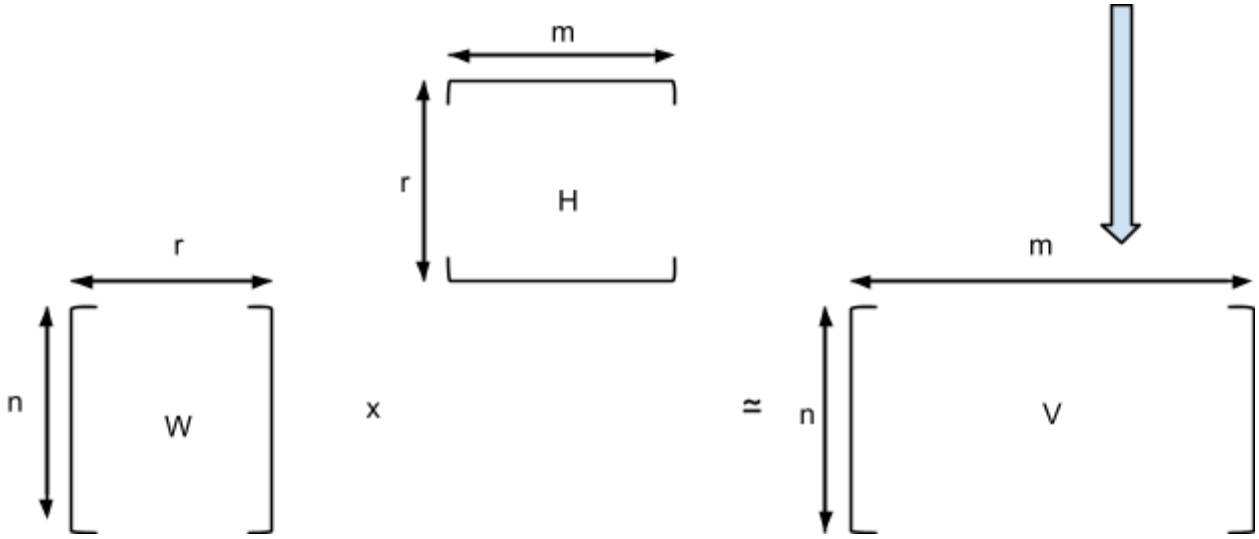
- Complexité semi-logarithmique $O(n \times log_2(n))$ contre linéaire $O(n)$ pour les vecteurs

III. Factorisation de matrices positives

Présentation des données sous forme matricielle:

Sécurité \ Note	Mauvaise	Moyenne	Bonne
Basse	41	54	10
Moyenne	304	240	63
Elevée	290	370	414

But: **factoriser** la matrice **V** en déterminant W et H telles que:



Première méthode: Annulation du gradient

Fonction à minimiser:

$$\varphi : \begin{cases} M_{n,r}(\mathbb{R}) \times M_{r,m}(\mathbb{R}) \longrightarrow \mathbb{R}_+ \\ (W, H) \longrightarrow \| WH - V \|^2 \end{cases}$$

Résolution de : $\overrightarrow{grad}(\varphi(W, H)) = 0$

D'où **r(m + n)** équations:

$$\begin{cases} \sum_{j=1}^{m} (2 w_{ik} h_{kj}^{\,2} - 2 h_{kj} v_{ij}) = 0 & \forall i \in [1; n] \\ & \forall k \in [1; r] \\ \sum_{j=1}^{m} (2 w_{ik} h_{kj}^{\,2} - 2 h_{kj} v_{ij}) = 0 & \forall j \in [1; m] \\ & \forall k \in [1; r] \end{cases}$$

Résolution sous Maple pour **m = n = 3** et **r = 2**

Deuxième méthode: algorithme de Lee-Seung:

W et **H** initialisées quelconques

|
v

Règle itérative:

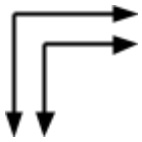
$$W_{ij} \leftarrow W_{ij} \times \left(\frac{Wn_{ij}}{Wd_{ij}}\right) \qquad avec \quad Wn = V \times {}^{t}H$$
$$Wd = W \times H \times {}^{t}H$$

$$H_{ij} \leftarrow H_{ij} \times \left(\frac{Hn_{ij}}{Hd_{ij}}\right) \qquad avec \quad Hn = {}^{t}W \times V$$
$$Hd = {}^{t}W \times W \times H$$

La quantité $||WH - V||^2$ décroît selon cette règle

Résultats :

Note \ Sécurité		Elevée	Moyenne	Basse
		9	12	**33**
		31	28	1



Bonne	1	2	41	24	10
Moyenne	1	7	304	240	63
Mauvaise	**18**	5	290	400	**414**

(**r = 2**)

Résultats en corrélation avec ceux de l'arbre de décision.

Comparaison des deux méthodes:

Distance approchée $||WH - V||^2 = 44$ dans les deux cas