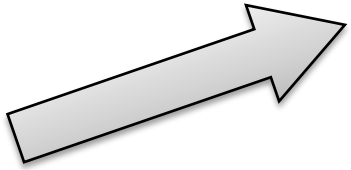
- Classification:
  - prédit la catégorie/classe d'un objet
  - construit un modèle basé sur un jeu d'apprentissage et des valeurs (nom des catégories) et l'utilise pour classer des données nouvelles
- Prédiction:
  - modélise des données numériques pour prédire des valeurs inconnues ou manquantes
- Applications
  - diagnostique médical
  - séquences codantes
  - structure secondaire, tertiaire
  - …

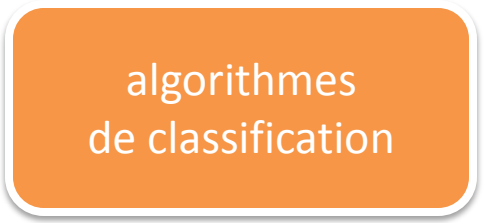
- Méthodes
  - arbre de décision
  - classificateur bayésien
  - analyse discriminante
  - réseau de neurones
  - plus proches voisins
  - règles d'association
  - algorithme génétique
  - machine à vecteur de support
  - modèle de Markov
  - …
- Evaluation des performances

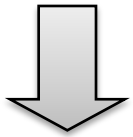
Classification : 2 étapes

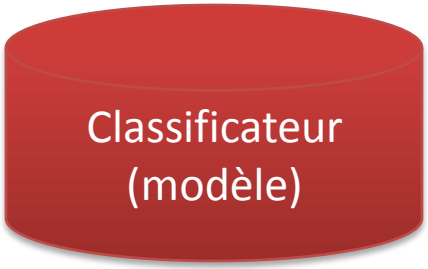
- Construction du modèle
  - chaque objet appartient à une classe connue
  - jeu de données d'apprentissage : ensemble des objets utilisés pour la construction du modèle
- Utilisation du modèle pour classer des objets nouveaux ou inconnus
  - estimation de la précision du modèle
    - les classes connues du jeu d'apprentissage sont comparées à celles prédites
    - précision : pourcentage d'objets de jeu de test correctement classés
    - le jeu de test est indépendant du jeu d'apprentissage sinon risque de biais

Classification : construction du modèle





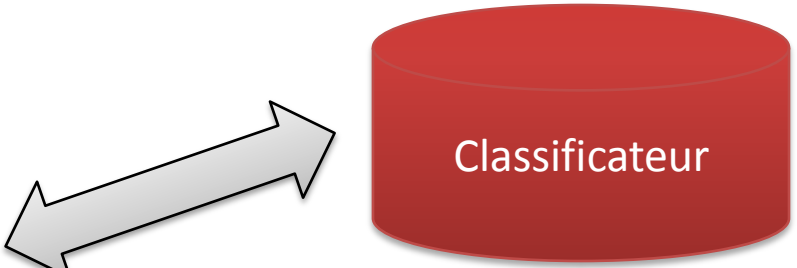




NOM	GRADE	ANC.	TITULAIRE
Mike	MdC	3	non
Mary	MdC	7	oui
Bill	Prof	2	oui
Jim	MdC	7	oui
Dave	MdC	6	non
Anne	MdC	3	non

IF grade= 'Prof'
OR anc. > 6
THEN titulaire = 'oui'

Classification : utilisation du modèle



(Jeff, Prof, 4)

NOM	GRADE	ANC.	TITULAIRE
Tom	MdC	2	non
Merlisa	MdC	7	non
George	Prof	5	oui
Joseph	MdC	7	oui

Titulaire ?

OUI

Apprentissage supervisé ou non

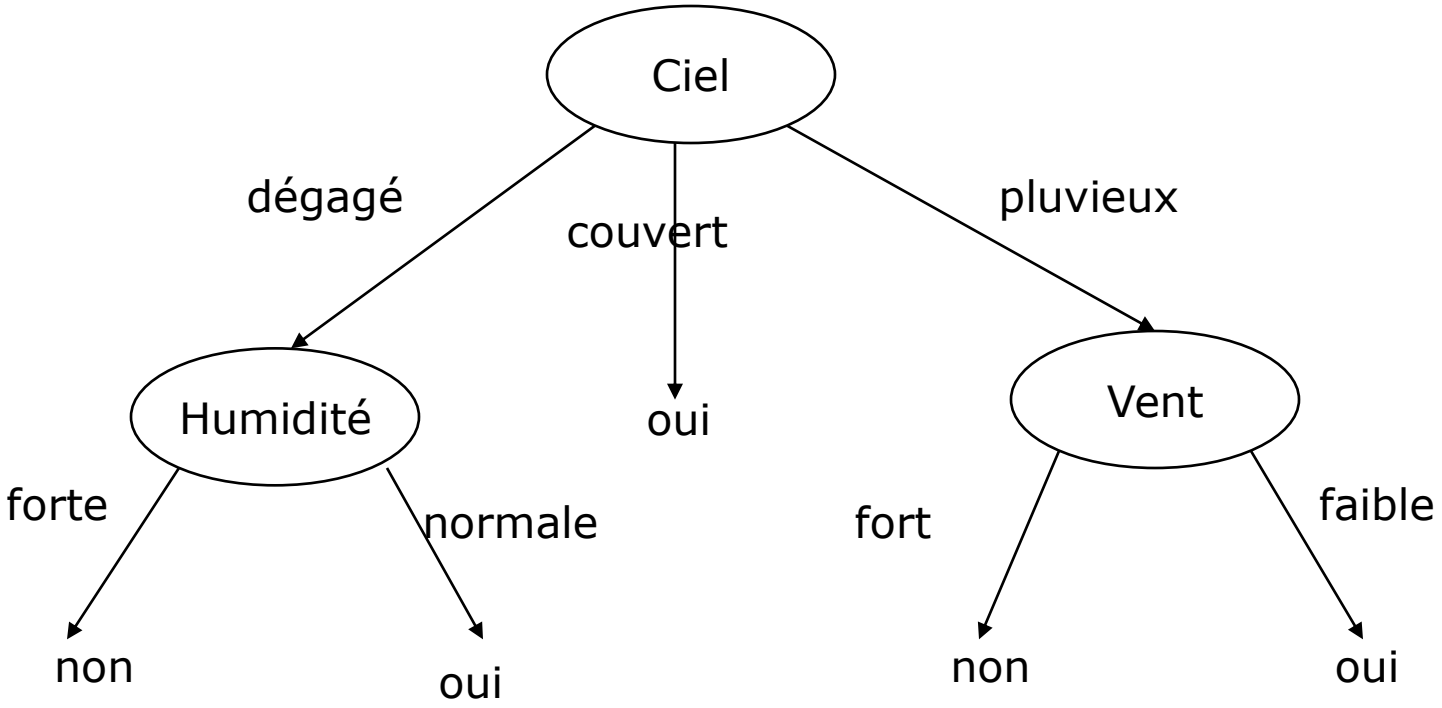
- Apprentissage supervisé (classification)
  - supervision : le jeu de données d'apprentissage fournit les classes des objets
  - les nouveaux objets sont classés en fonction du jeu d'apprentissage

- Apprentissage non supervisé (clustering)
  - Pas de classes définies
  - Étant donné un ensemble de mesures, d'observations, etc., essayer d'établir l'existence de classes ou de clusters dans les données

Considérations pour la classification

- Nettoyage des données
  - pré-traiter les données pour réduire le bruit et gérer les valeurs manquantes
- Analyse de pertinence
  - supprimer les attributs non pertinents ou redondants
- Transformation des données
  - généraliser ou normaliser les données
- Précision de la prédiction
- Efficacité et mise à l'échelle
  - pour construire le modèle
  - pour l'utiliser
- Robustesse
  - tolérance au bruit et aux données manquantes
- Interprétabilité
  - compréhension des données via le modèle
- Qualité des règles
  - taille de l'arbre de décision
  - règles de classification compactes

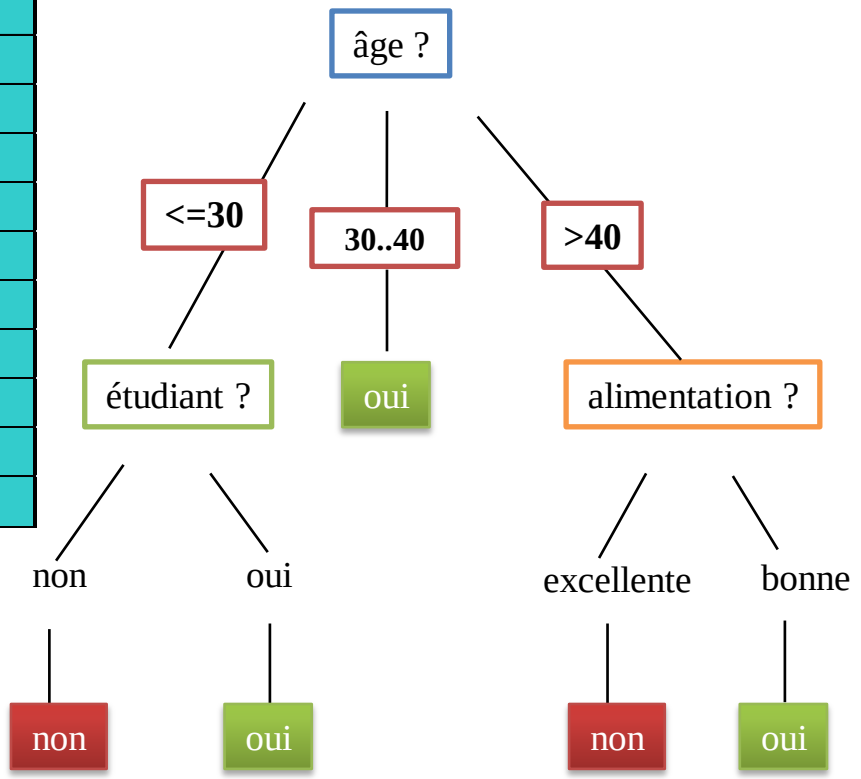
- Arbre de décision
  - nœuds internes : test sur un attribut
  - branches : résultat d'un test / valeur de l'attribut
  - feuilles : classe



- Génération de l'arbre en 2 étapes
  - Construction
    - au départ, tous les exemples du jeu d'apprentissage sont à la racine
    - partitionne récursivement les exemple en sélectionnant des attributs
  - Élagage
    - identification et suppression des branches correspondant à des exceptions ou du bruit
- Utilisation de l'arbre
  - teste les valeurs des attributs avec l'arbre de décision

Exemple : pratique la méditation ?

âge	stress	étudiant	alimentation	médite
<=30	élevé	non	bonne	non
<=30	élevé	non	excellente	non
31…40	élevé	non	bonne	oui
>40	moyen	non	bonne	oui
>40	bas	oui	bonne	oui
>40	bas	oui	excellente	non
31…40	bas	oui	excellente	oui
<=30	moyen	non	bonne	non
<=30	bas	oui	bonne	oui
>40	moyen	oui	bonne	oui
<=30	moyen	oui	excellente	oui
31…40	moyen	non	excellente	oui
31…40	élevé	oui	bonne	oui
>40	moyen	non	excellente	non



Algorithme pour l'induction d'arbre de décision

- Algorithme glouton
  - approche descendante récursive *diviser pour régner*
  - au départ, tous les objets sont à la racine
  - attributs catégoriels (les valeurs continues sont discrétisées à l'avance)
  - les exemples sont partitionnés récursivement par la sélection d'attribut
  - les attributs sont sélectionnés sur la base d'une heuristique ou d'une mesure statistique

- Conditions d'arrêt
  - tous les exemples pour un nœud appartiennent à la même classe
  - plus d'attribut pour partitionner, dans ce cas la classe attribuées correspond à celle la plus représentée
  - plus d'exemple à classer

Mesure pour la sélection d'attribut. Exemple : gain d'information (ID3 et c4.5)

- Sélectionne l'attribut qui a le gain le plus élevé
- Soient 2 classes P et N
  - Soit un jeu d'apprentissage S qui contient p objets de classe P et n objets de classe N
  - La quantité d'information nécessaire pour décider si un objet de S appartient à P ou N est définie comme

$$I(p,n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

- Les valeurs de A partitionnent S en {S1, …, Sv}
  - si S_i contient p_i exemples de P et n_i exemple de N, l'entropie ou l'information attendue nécessaire pour classer les objets dans tous les sous-arbres S_i est

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p+n} I(p_i, n_i)$$

- Le gain d'information de l'attribut A est $Gain(A) = I(p,n) - E(A)$

f(x) = -x log x

- si f1 = 0 et f2 = 1

  I = 0

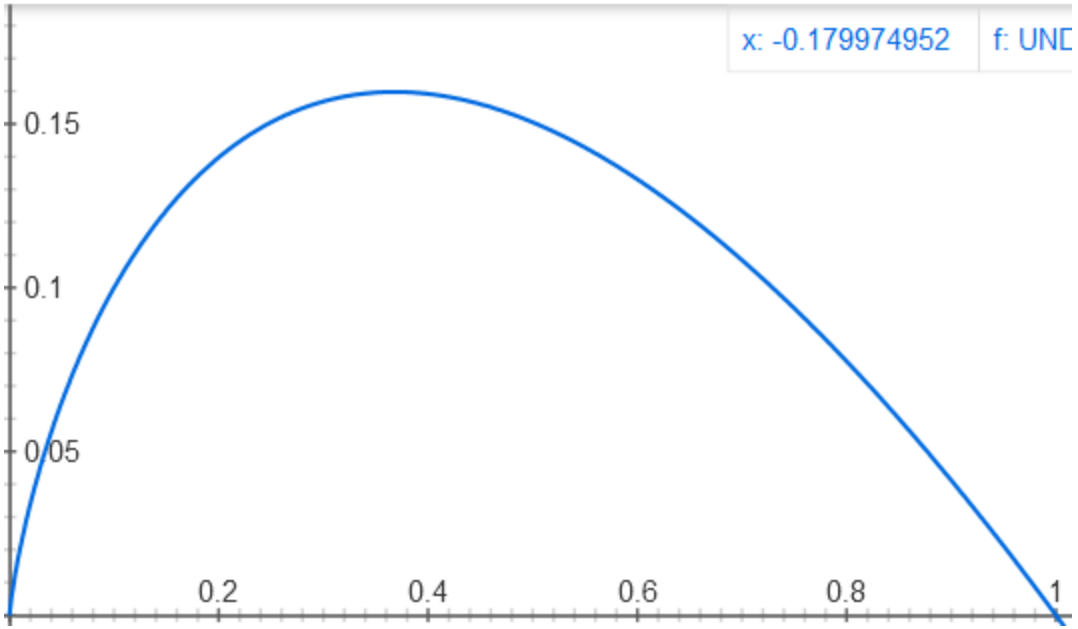
- si f1 = f2 = 0.5

  I = 0.3

- si f1 = 0.4 et

  f2 = 0.6

  I = 0.29



- si f1 = f2 = f3 = f4, I = 0.6

Exemple : pratique la méditation ?

- Classe P : médite = oui
- Classe N : médite = non

âge	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0,971
30…40	4	0	0
>40	3	2	0,971

- I(p,n) = I(9,5) = 0.940

- Calcul de l'entropie

$$E(\hat{a}ge) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = 0.69$$

- Gain d'information

$$Gain(\hat{a}ge) = I(p,n) - E(\hat{a}ge)$$

- De même

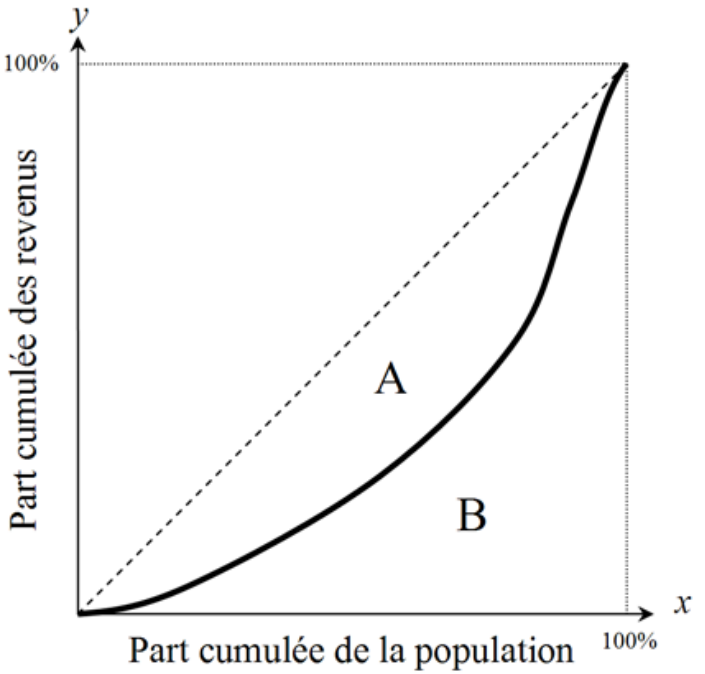
$$Gain(stress) = 0.029$$

$$Gain(étudiant) = 0.151$$

$$Gain(alimentation) = 0.048$$

- Coefficient de Gini
  - mesure la dispersion d'une distribution dans une population

    0 : uniforme, 1 : biaisée
  - utilisé pour mesurer l'inégalité de revenus
  - G = A / (A+B)
- χ^2 : indépendance entre classe et modalités d'un attribut ?



Éviter de trop modéliser le jeu d'apprentissage (overfitting)

- L'arbre généré risque de trop refléter le jeu d'apprentissage
  - trop de branches, certaines peuvent représenter des anomalies
  - précision faible pour des données nouvelles
- 2 approches
  - pré-élagage : arrêter la construction de l'arbre tôt = ne pas partitionner un nœud si la mesure de qualité dépasse un seuil
    - difficulté de fixer le seuil
  - post-élagage : supprimer des branches d'un arbre totalement construit = obtenir une séquence d'arbres progressivement élagués
    - utiliser un jeu de données différents pour décider du meilleur arbre élagué
    - principe MDL (minimum description length)
      - codage : règles associées à l'arbre + exceptions
      - supprimer la règle qui réduit le plus le codage tant que le codage diminue en taille

Améliorations de l'algorithme

- Attributs définis sur des valeurs continues
  - définir dynamiquement les valeurs pour partitionner les données
- Tolérance aux données manquantes
  - attribuer la valeur la plus fréquente
  - attribuer une probabilité pour chaque valeur possible

- Principal défaut : sensibilité et sur-apprentissage
- Bagging (**b**ootstrap **agg**regat**ing**)
  - bootstrapping
  - aggregating
- Forêts aléatoires
  - échantillonage des attributs pour décorréler les arbres construits
  - pas d'élagage : sur-apprentissage compensé par le bagging

Classification Bayésienne

- Apprentissage probabiliste : calcule explicitement les probabilités des hypothèses, une des approches les plus pragmatiques pour certains types d'apprentissage

- Incrémental : chaque exemple met à jour la probabilité qu'une hypothèse est correcte. Des connaissances *a priori* peuvent être combinées avec des données d'observation.

- Prédiction probabiliste : prédit plusieurs hypothèses, pondérées par leur probabilité

- Le problème de classification peut être formulé en utilisant les probabilités *a posteriori* :
  - P(C|X) = probabilité que l'objet X = $<x_1,\ldots,x_k>$ est de la classe C
- Exemple : P(non|ciel=dégagé,vent=fort,…)

- Idée : attribuer à l'objet X la classe qui maximise P(C|X)

Estimation des probabilités *a posteriori*

- Théorème de Bayes
  - $P(C|X) = P(X|C) \cdot P(C) / P(X)$

  P(classe | observations) = P(observations | classe) . P(classe) / P(observations)
- Principe : attribuer la classe la plus probable
  - C_i tel que $P(C_i|X)$ est maximum

    C_i tel que $P(X|C_i) \cdot P(C_i) / P(X)$ est maximum
- P(X) est constant pour toutes les classes :

  pas besoin de la calculer
- P(C) = fréquence de la classe C

- Problème : pas faisable en pratique !

Classificateur Bayésien naïf

- Assomption naïve : indépendance des attributs
  - $P(x_1,\ldots,x_k|C) = P(x_1|C)\cdot\ldots\cdot P(x_k|C)$

- si le i-ième attribut est catégoriel :

  $P(x_i|C)$ est estimée comme la fréquence des échantillons qui ont pour valeur x_i et qui sont de classe C

- si le i-ième attribut est continu :

  $P(x_i|C)$ est estimée avec une gaussienne

- calcul facile

Exemple

Ciel	Température	Humidité	Vent	Classe
dégagé	chaud	forte	faux	N
dégagé	chaud	forte	vrai	N
couvert	chaud	forte	faux	P
pluvieux	moyenne	forte	vrai	P
pluvieux	frais	normale	faux	P
pluvieux	frais	normale	vrai	N
couvert	frais	normale	vrai	P
dégagé	moyenne	forte	faux	N
dégagé	frais	normale	faux	P
pluvieux	moyenne	normale	faux	P
dégagé	moyenne	normale	vrai	P
couvert	moyenne	forte	vrai	P
couvert	chaud	normale	faux	P
pluvieux	moyenne	forte	vrai	N

Ciel	
P(dégagé|p) = 2/9	**P(dégagé|n) = 3/5**
P(couvert|p) = 4/9	**P(couvert|n) = 0**
P(pluvieux|p) = 3/9	**P(pluvieux|n) = 2/5**
température	
P(chaud|p) = 2/9	**P(chaud|n) = 2/5**
P(moyenne|p) = 4/9	**P(moyenne|n) = 2/5**
P(frais|p) = 3/9	**P(frais|n) = 1/5**
humidité	
P(forte|p) = 3/9	**P(forte|n) = 4/5**
P(normale|p) = 6/9	**P(normale|n) = 2/5**
vent	
P(vrai|p) = 3/9	**P(vrai|n) = 3/5**
P(faux|p) = 6/9	**P(faux|n) = 2/5**

P(p) = 9/14
P(n) = 5/14

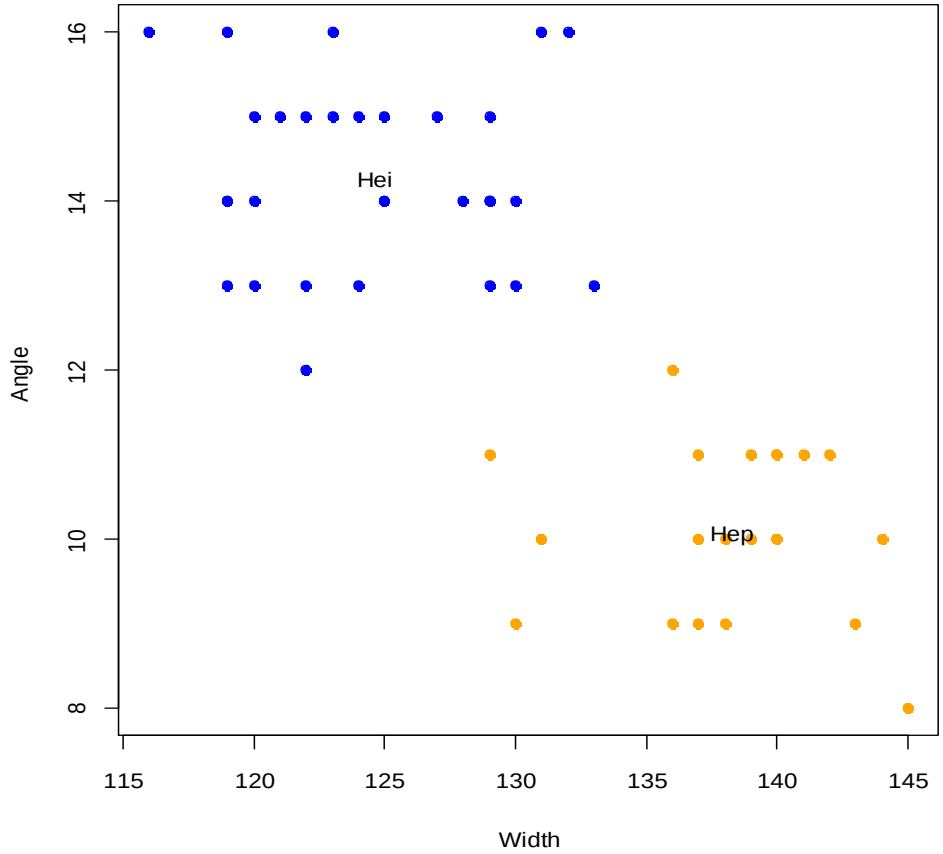
- Un objet X = <pluvieux, chaud, forte, faux>
- P(X|p).P(p) =
  P(pluvieux|p)·P(chaud|p)·P(forte|p)·P(faux|p)·P(p) = 3/9·2/9·3/9·6/9·9/14 = 0.010582
- P(X|n)·P(n) =
  P(pluvieux|n)·P(chaud|n)·P(forte|n)·P(faux|n)·P(n) = 2/5·2/5·4/5·2/5·5/14 = **0.018286**
- X est classé comme n

L'hypothèse d'indépendance …

- … rend le calcul possible
- … mène à des classificateurs optimaux si elle est vérifiée
- … mais c'est rarement le cas, car les attributs sont souvent corrélés
- tentative de pallier cette difficulté :
  - réseaux Bayésiens : combinent le raisonnement Bayésien avec la relation causale entre les attributs
  - arbres de décision : considère un seul attribut à la fois, en commençant par le plus important

2 familles de puces caractérisées par l'angle et la taille de leur édéage :

- Hei : Heikertingeri
- Hep : Heptapotamica

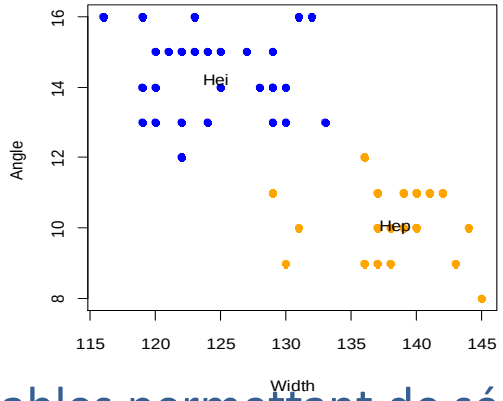


Objectif : combinaison linéaire des variables permettant de séparer au mieux les groupes dans un nouvel espace de représentation

Analyse discriminante

2 familles de puces caractérisées par l'angle et la taille de leur édéage :

- Hei : Heikertingeri
- Hep : Heptapotamica



Objectif : combinaison linéaire des variables permettant de séparer au mieux les groupes dans un nouvel espace de représentation

dispersion intra-groupe : matrice de variance-covariance W_k

avec *n* le nombre d'individus et n_k le nombre d'individu de la classe *k*

$$W = \frac{1}{n} \sum_k n_k \times W_k$$

éloignement entre les groupes : matrice de variance-covariances inter-groupes : B

avec μ la moyenne globale et μ_k la moyenne de la classe *k*

$$B = \frac{1}{n} \sum_k n_k (\mu_k - \mu)^T (\mu_k - \mu)$$

dispersion totale : V = W + B

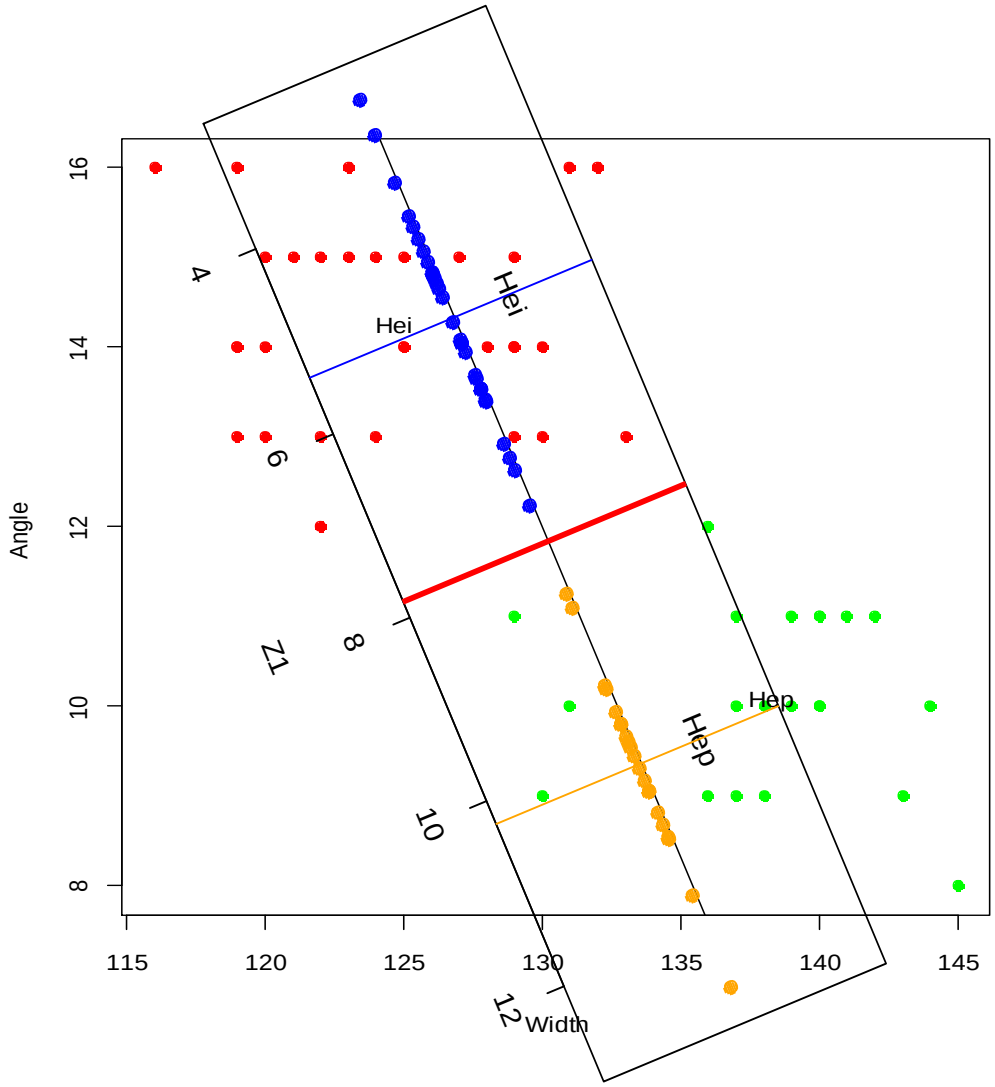
principe : trouver les axes factoriels qui maximisent l'éloignement des groupes par rapport à la dispersion totale

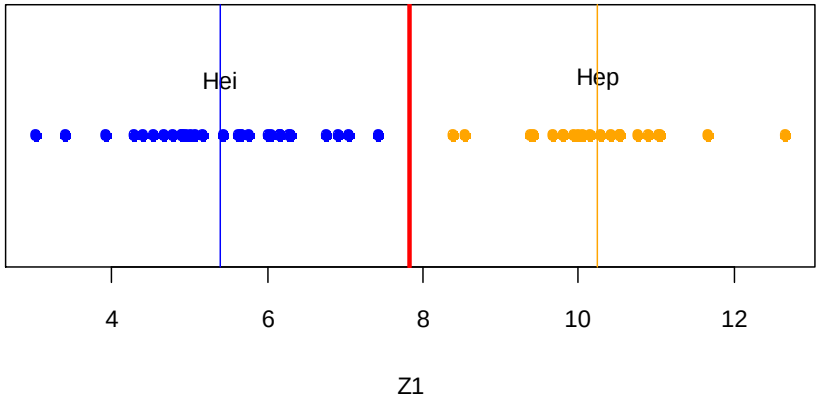
Z_1 premier axe factoriel défini par le vecteur u_1 tel que l'on maximise

$$\frac{u_1' B u_1}{u_1' V u_1}$$

Solution : résolution de $V^{-1}Bu = \lambda u$

Analyse discriminante





	u1
Width	0.1280690
Angle	-0.7393077

Règle bayesienne

$$P(C = c / X) = \frac{P(C = c).P(X / C = c)}{\sum_{i=1}^{k} P(C = c_i).(X / C = c_i)}$$

Hypothèse homoscédasticité : $W = \frac{1}{n}\sum_k n_k \times W_k$

Fonction de classement linéaire :

$$score\ (C = c, X) = -(X - \mu_c)^T W^{-1} (X - \mu_c)$$

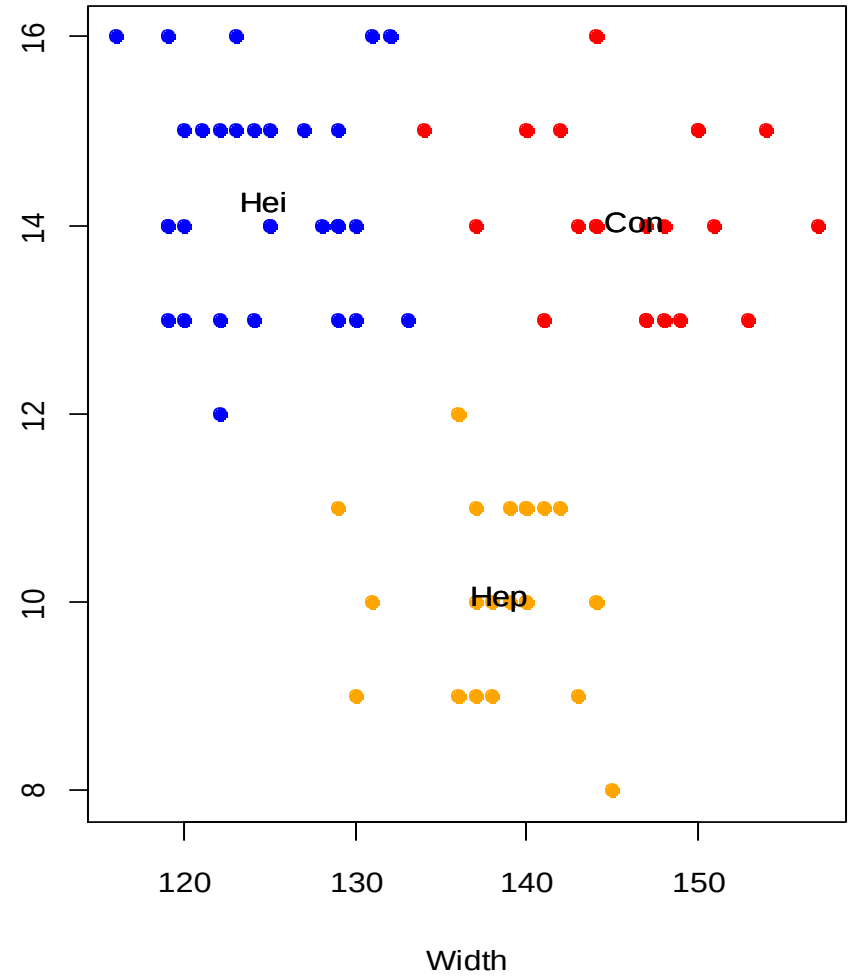
~ distance au centre de gravité qui prend en compte la variance

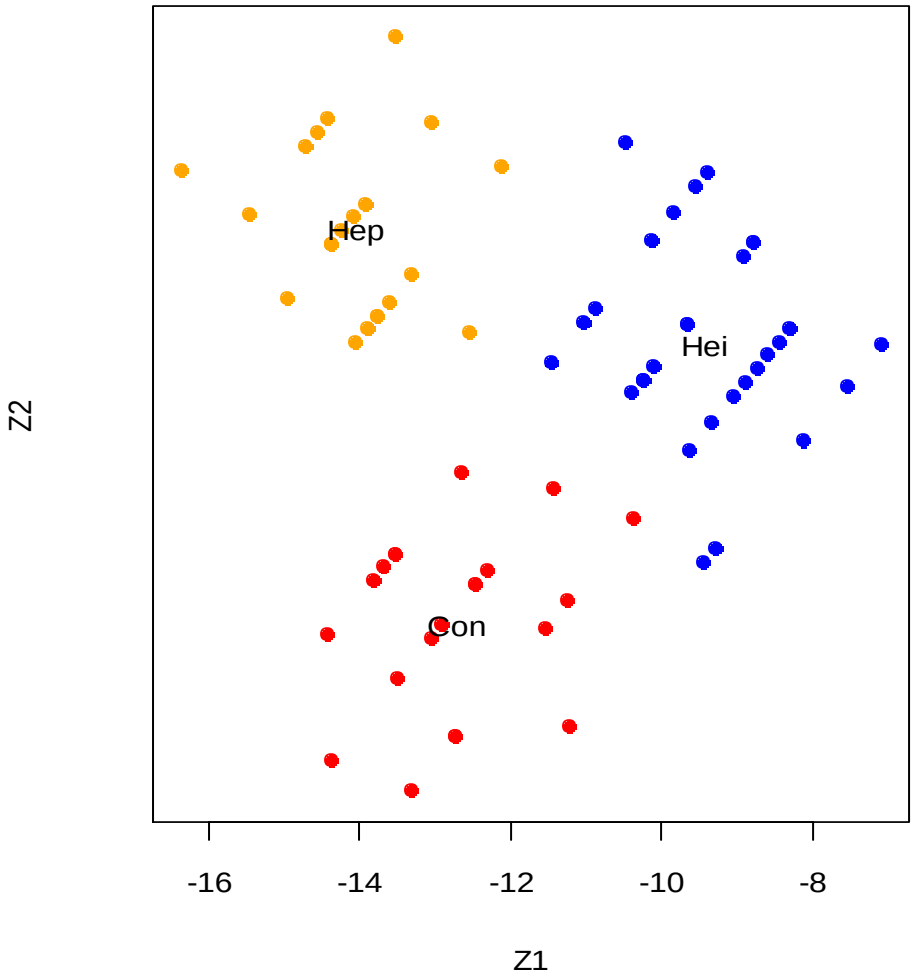
- Pouvoir discriminant d'un axe factoriel
  - variance portée par l'axe / valeur propre

- Robustesse
  - peut fonctionner même si les hypothèses de départ de ne sont respectées

Analyse discriminante : Exemple à 3 classes

- Con : Concinna
- Hei : Heikertingeri
- Hep : Heptapotamica



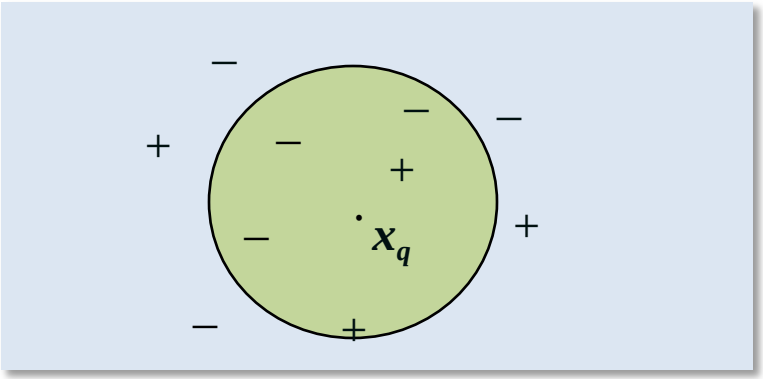


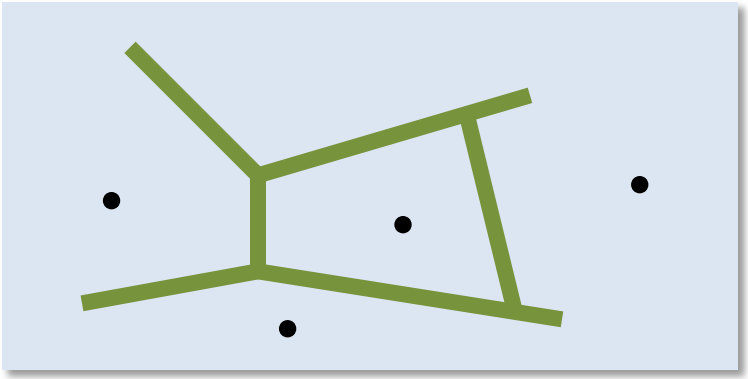
Méthodes basées sur des instances (Instance-Based Methods)

- Apprentissage basé sur des instances :
  - Stocke le jeu d'apprentissage et effectue le traitement quand une nouvelle instance doit être classée
- Approches typiques
  - *k* plus proches voisins (*k* nearest neighbors)
    - chaque objet représente un point dans l'espace
  - régression
    - loess, approximation locale

Algorithme des *k* plus proches voisins

- Chaque objet est représenté par un point dans un espace à *n* dimensions
- Les plus proches voisins sont définis en terme de distance (euclidienne, manhattan, …) ou dissimilarité
- La fonction de prédiction peut être discrète/nominale ou continue
  - discrète : valeur la plus fréquente
  - continue : moyenne
- Diagramme de Voronoï : surface de décision induite par le plus proche voisin





Discussion sur l'algorithme des *k* plus proches voisins

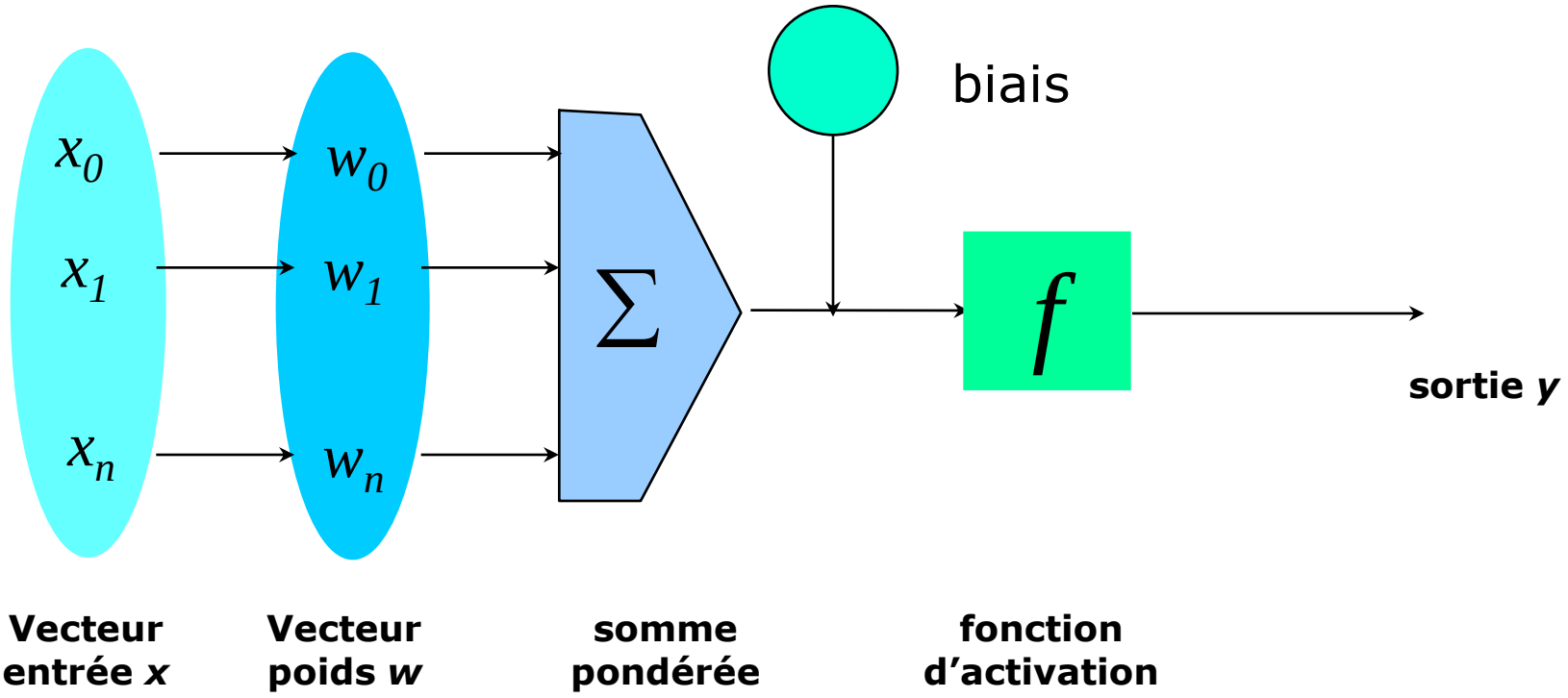
- Pondération en fonction de la distance
  - pondère la contribution de chacun des *k* voisins en fonction de sa distance à l'objet à classer
  - plus on est proche, plus on a de poids

$$w \equiv \frac{1}{d(x_q, x_i)^2}$$

- Robuste dans le cas de données bruitées
- Problème de dimensionnalité : la distance peut être dominée par des attributs non pertinents
  - solution : normalisation des dimensions ou élimination des attributs non pertinents

- Avantages
  - précision souvent élevée
  - robuste, marche lorsque le jeu d'apprentissage contient des erreurs
  - sortie peut être valeur discrète, continue, ou un vecteur de plusieurs valeurs discrètes ou continues
- Critiques
  - apprentissage long
  - difficile de comprendre le modèle
  - difficile d'incorporer des connaissances

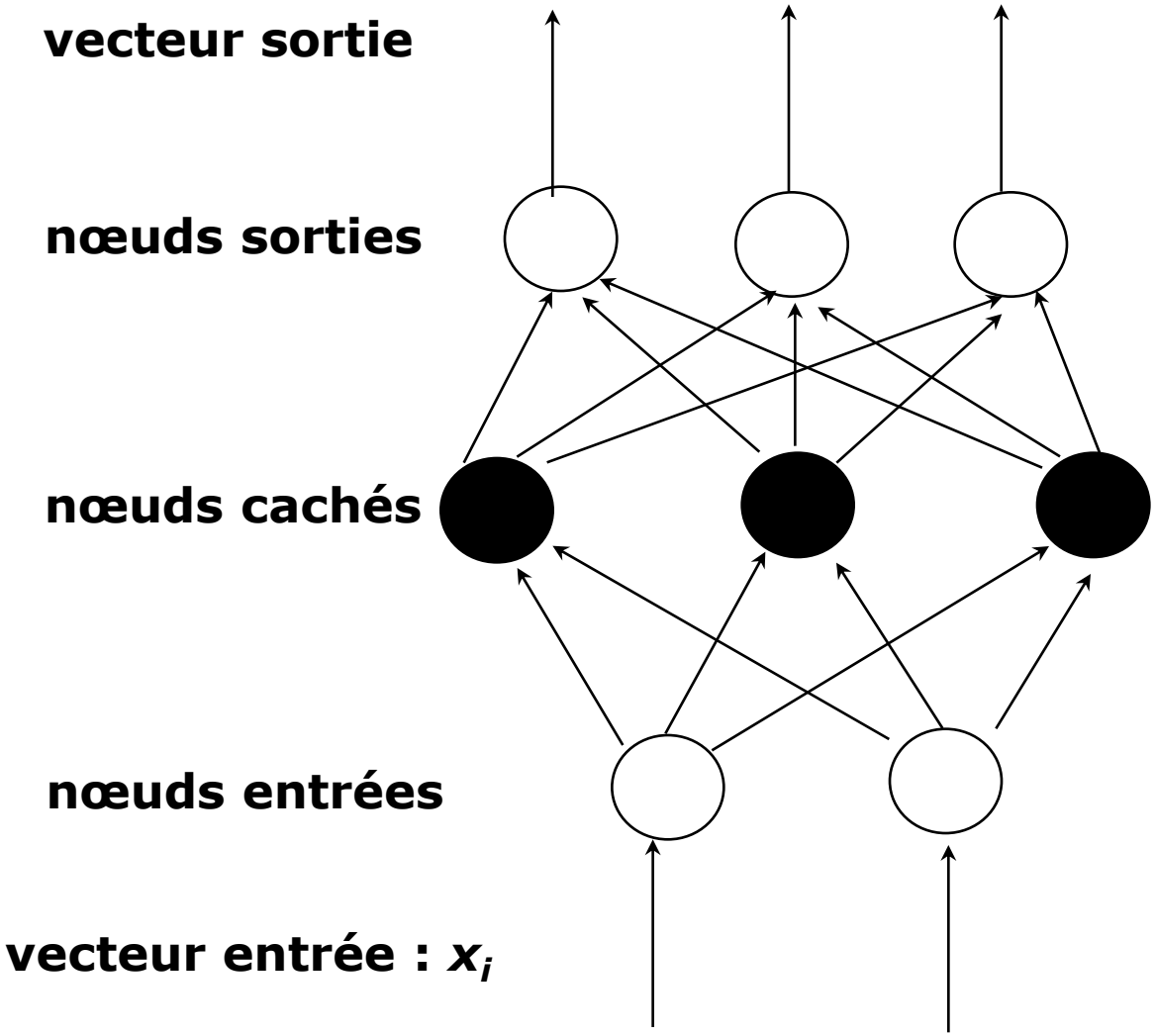
- vecteur *x* n-dimensionnel est intégré en *y* par le produit scalaire ($x_i.w_i$), le biais et la fonction d'activation



Entraînement

- Objectif
  - obtenir un ensemble de poids permettant de classer correctement (presque tous) les objets du jeu d'apprentissage
- Étapes
  - Initialiser les poids avec des valeurs aléatoires
  - Passer les objets au réseau un par un
  - pour chaque unité (neurone)
    - calculer l'entrée (combinaison linéaire de toutes les entrées)
    - calculer la sortie en utilisation la fonction d'activation
    - calculer l'erreur
    - mettre à jour le biais et les poids

Perceptron (multicouche) (eng: MLP pour multilayer perceptron)



1. calcul de l'erreur en sortie pour chaque neurone
2. augmentation ou diminution du biais et du poids pour obtenir une faible erreur locale
3. attribution d'une erreur au neurones précédents
4. retour à l'étape 2 pour la couche précédente

Estimation du taux d'erreurs

- partitionnement :
  - utilisation de jeux indépendants : apprentissage (2/3), test (1/3)
- validation croisée :
  - diviser les données en k partitions
  - utiliser k-1 partitions pour l'apprentissage et la dernière pour le test
  - précision = nombre d'objets bien classés lors des k itérations / nombre d'objets
  - leave-one-out (validation croisée avec k = s)
- bootstrapping
  - tirage aléatoire avec remise des objets constituant le jeu d'apprentissage

Performances du classificateur

- Positif (P=●+●), Négatif (N=●+●),
  Prédit Positif (PP=●+●), Prédit Négatif (PN=●+●),
  Vrai Positif (VP=●), Faux Positif (FP=●),
  Vrai Négatif(VN=●), Faux Négatif (FN=●)
- Sensibilité=VP/P (eng: sensitivity)
- Spécificité=VN/N
- Précision=VP/(VP+FP) = VP/PP
- Exactitude (accuracy) =sensibilité.P/(P+N) + spécificité.N/(P+N)
  = **(VP+VN)/(P+N)**

réalité → prédiction ↓	P	N
PP	VP	FP
PN	FN	VN

valeur prédictive positive

val. préd. négative

sensibilité

spécificité