Cours 3 - Introduction à l'Intelligence Artificielle

Alice Cohen-Hadria

Maîtresse de conférences à Sorbonne Université
alice.cohenhadria@gmail.com

Classification

Intuition

On cherche à regrouper nos données en un certain nombre K de classes, connues à l'avance.

Si k=2, on parle de classification binaire.

Si k>2, on parle de classification multi-classes.

Dans ce cours, on va voir deux méthodes de classification : KNN et les réseaux de neurones.

Exemple

Données: emails

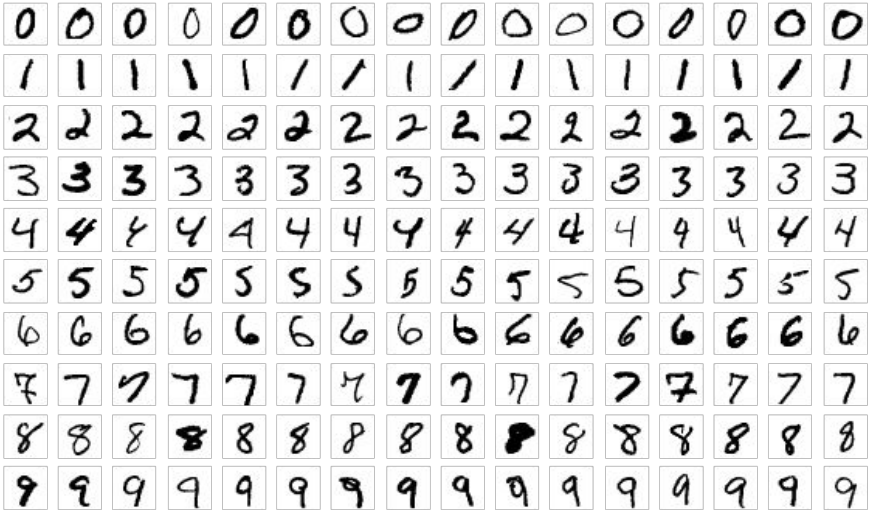
Classes : Spam ou Non spam

Exemple

Lire un code postal sur une enveloppe, un montant sur un chèque.

Données : Image

Classes: Les 10 chiffres.

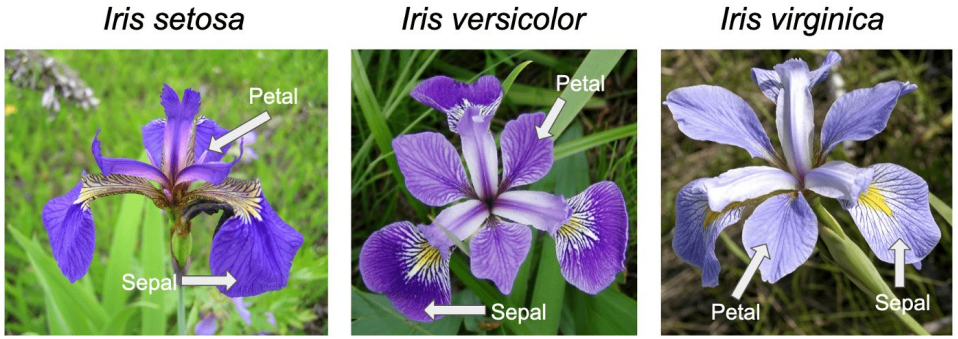


Exemple

Retrouver le type d’iris

Données : Deux indicateurs extraits des images

Classes : Le type d’iris.



Classification binaire

On a des données d'apprentissage pour des individus i = 1, …, n.

Pour chaque individu i, on a :

- Un vecteur de variables x_i qui décrivent les attributs de l'individu
- La valeur du label $y_i \in \{-1; 1\}$ qui correspond à la classe que l'on veut prédire.

Attention ! Dans une tâche de classification, ce que l'on veut prédire est obligatoirement un ensemble fini et discret. Quand on veut prédire une quantité on parle de **régression.**

Classification binaire - But

Avec l'aide des données d'apprentissage, on veut prédire la classe d'un nouvel individu x_{test} avec un algorithme choisi.

Notre algorithme peut être vu comme une fonction f qui pour tout x_i approxime :

$$f(x_i) \simeq y_i$$

Pour avoir la fonction f, on utilise les données d'apprentissage

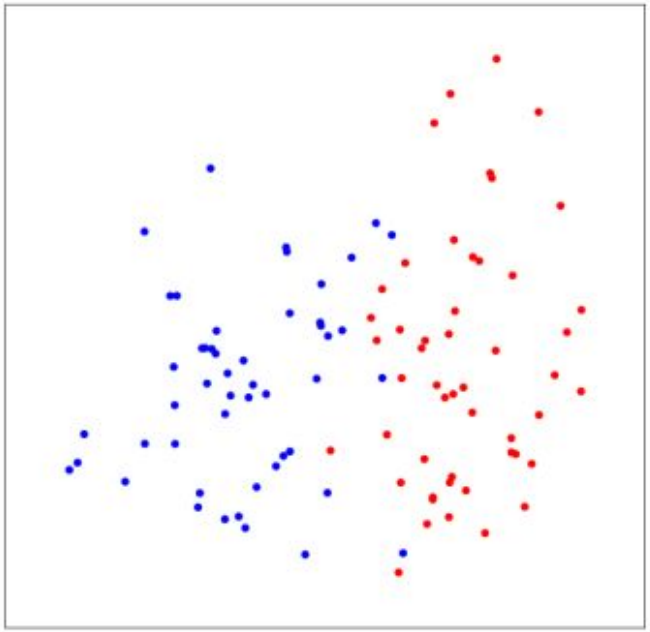
$$\mathcal{D}_n = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$$

Exemple simple

Les données sont de dimensions 2 (qu'on a représenté en x et y).

2 classes : points rouges et points bleus.

Comment séparer les deux classes ?

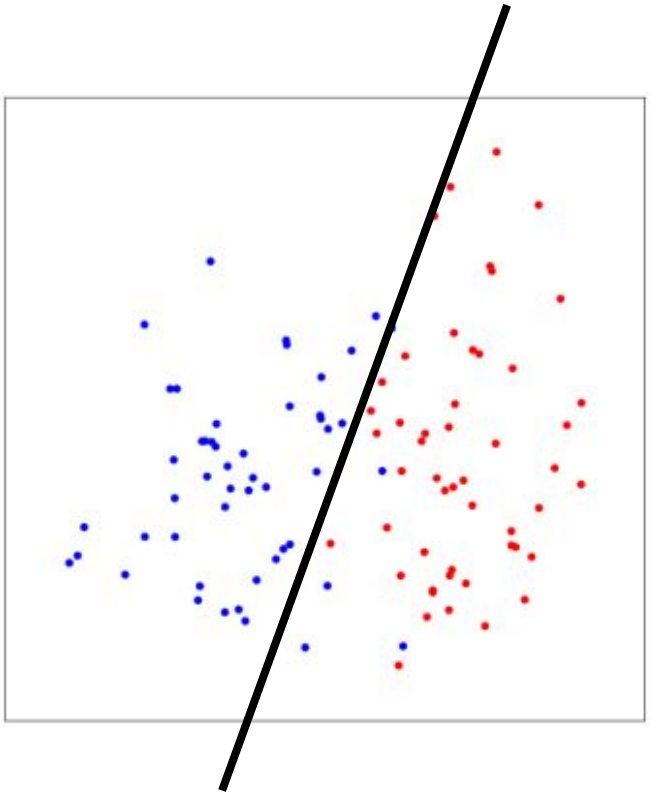


Exemple simple

Les données sont de dimensions 2 (qu'on a représenté en x et y).

2 classes : points rouges et points bleus.

Comment séparer les deux classes ?

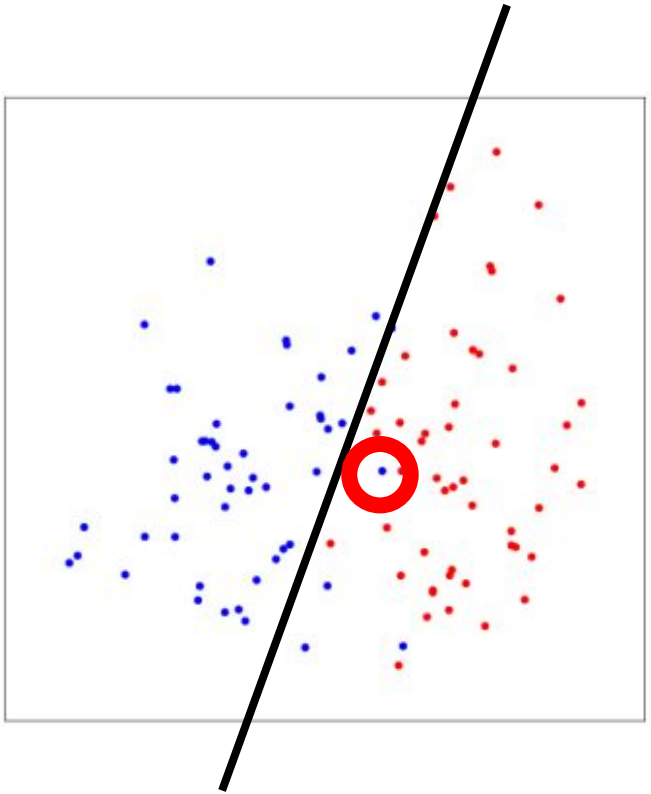


Exemple simple

Les données sont de dimensions 2 (qu'on a représenté en x et y).

2 classes : points rouges et points bleus.

Comment séparer les deux classes ?

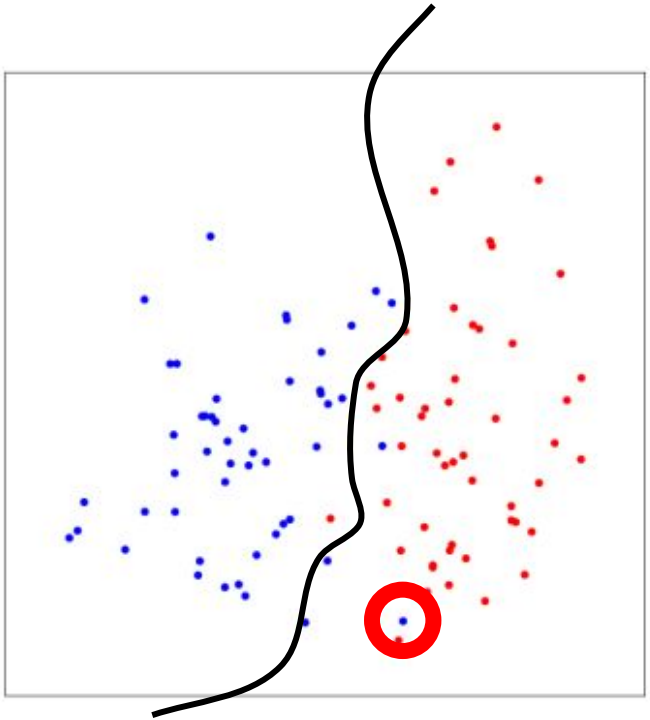


Exemple simple

Les données sont de dimensions 2 (qu'on a représenté en x et y).

2 classes : points rouges et points bleus.

Comment séparer les deux classes ?



Classification multiclasses

On a des données d'apprentissage pour des individus i = 1, ..., n.

Pour chaque individu i, on a :

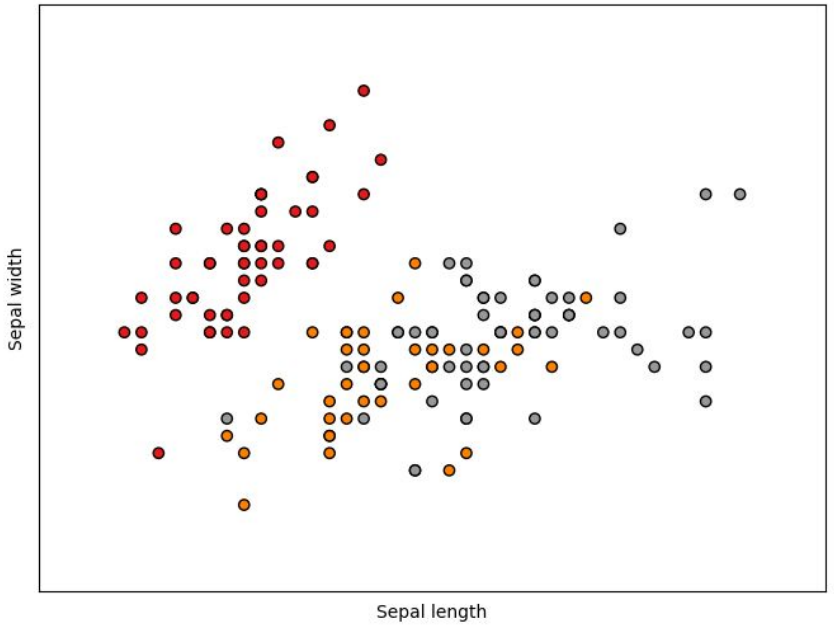
- Un vecteur de variables x_i qui décrivent les attributs de l'individu
- La valeur du label $\mathcal{C} = \{1, ..., K\}$ qui correspond à la classe que l'on veut prédire.

Exemple

Les données sont de dimensions 2 (qu'on a représenté en x et y).

Ici on a trois classes : Rouge, orange et gris

Comment séparer les deux classes ?

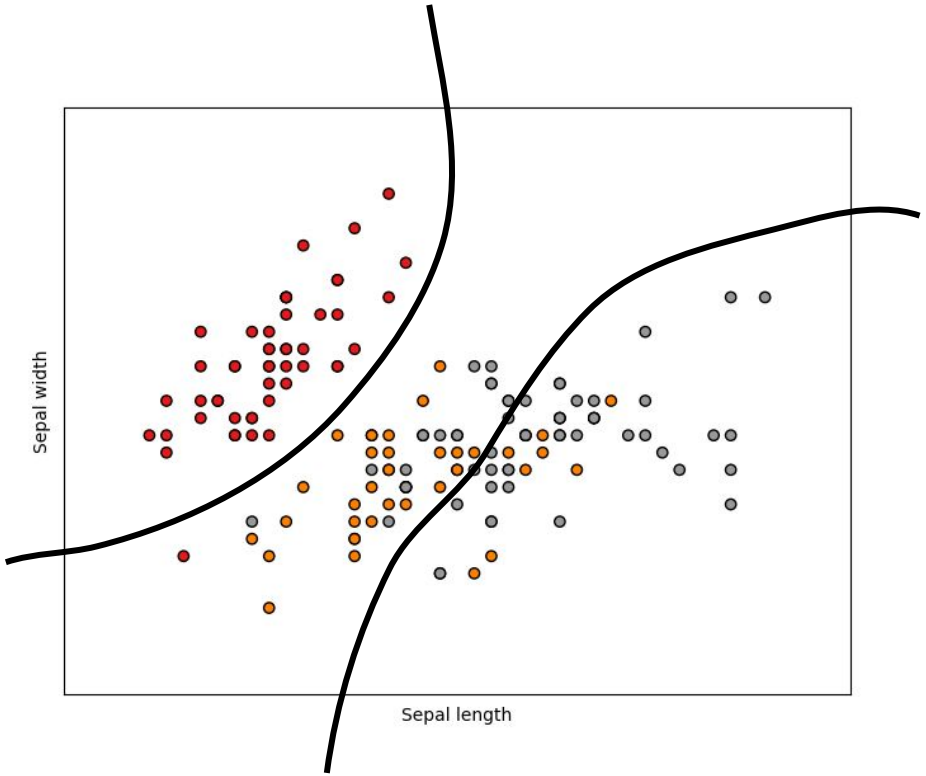


Exemple

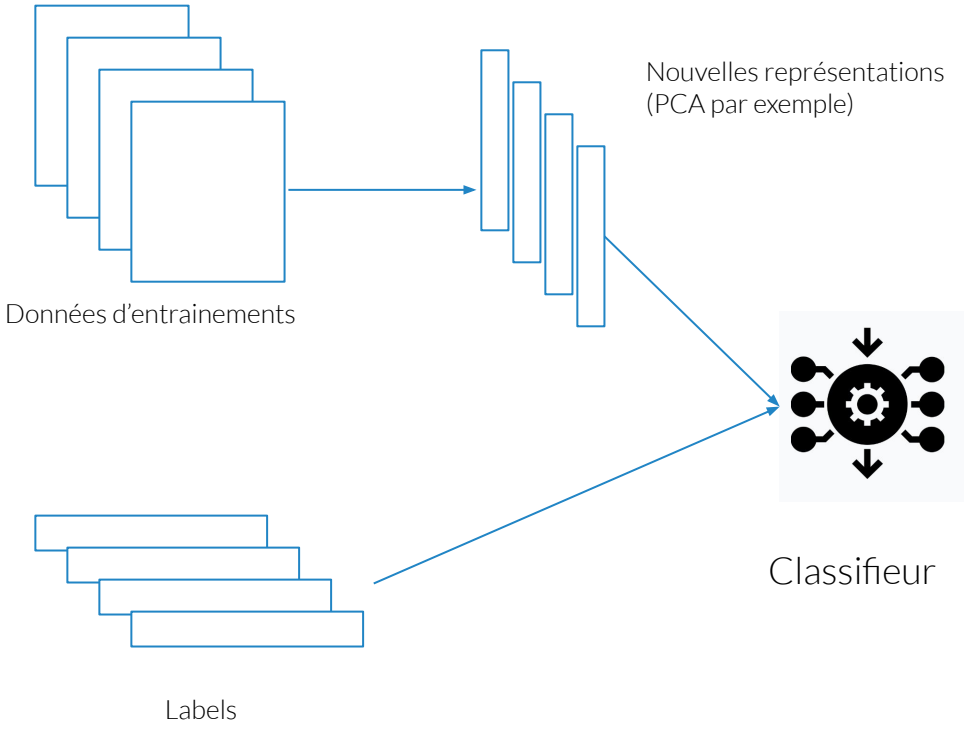
Les données sont de dimensions 2 (qu'on a représenté en x et y).

Ici on a trois classes : Rouge, orange et gris

Comment séparer les deux classes ?



Résumé - Construction du classifieur



Résumé - Nouvelle prédiction

Nouvelle donnée

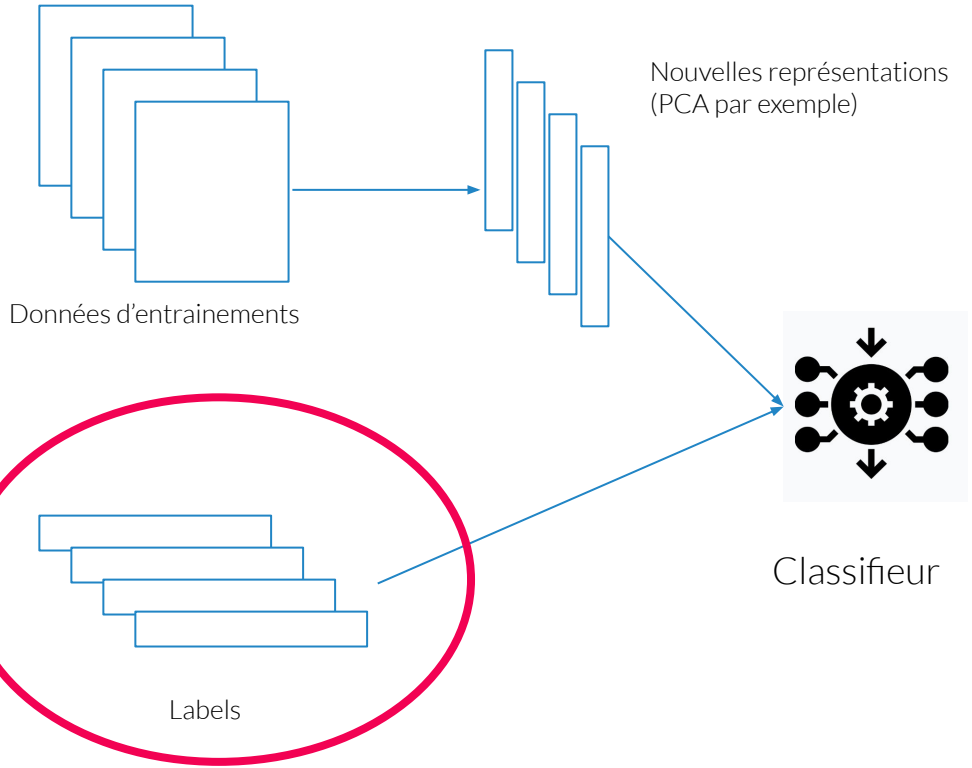
Nouvelles représentations
(PCA par exemple)

Classifieur

Label prédit

On utilise pas le label de la nouvelle donnée pour faire la prédiction (des fois, on ne l'a même pas à disposition).

Apprentissage supervisé



On se sert des labels pour construire notre classifieur

Apprentissage supervisé

On peut apprendre notre classifieur de deux manières :

1. On peut utiliser les labels (parce qu'on les connaît). On parle alors **d'apprentissage supervisé.**
    - Inconvénient : On doit connaître les labels à l'avance. Annoter des bases de données est coûteux.
2. On n'utilise pas les labels ou on ne les connaît pas à l'avance. On cherche à regrouper nos données avec des ressemblances significatives.

Dans ce cours, on ne voit que l'apprentissage supervisé.

Apprentissage : la vraie vie vs les expérimentations

Dans la vraie vie

On a des données qui existent. Ensuite on utilise notre modèle pour prédire des choses dans la vraie vie avec des nouvelles données qui arrivent tous les jours.

On n'a rien appris sur ces données.

Par exemple, lecture de montant de chèques.

Nouveaux clients tous les jours.

Expérimentations

Quand on fait des expérimentations, on n'a pas de nouvelles données qui arrivent en permanence.

On simule ça en splitant nos données en plusieurs sous ensemble:

- Ensemble d'entraînement (train set). Sert à apprendre le classifieur
- Ensemble de test (test set). Sert à évaluer notre classifieur. Simule les données de la vraie vie.

Expérimentations

	Area Abbreviation	Area Code	Area	Item Code	Item	Element Code	Element	Unit	latitude	longitude	...	Y2004	Y2005	Y2006	Y2007	Y2008	Y2009
0	AF	2	Afghanistan	2511	Wheat and products	5142	Food	1000 tonnes	33.94	67.71	...	3249.0	3486.0	3704.0	4164.0	4252.0	4538.0
1	AF	2	Afghanistan	2805	Rice (Milled Equivalent)	5142	Food	1000 tonnes	33.94	67.71	...	419.0	445.0	546.0	455.0	490.0	415.0
2	AF	2	Afghanistan	2513	Barley and products	5521	Feed	1000 tonnes	33.94	67.71	...	58.0	236.0	262.0	263.0	230.0	379.0
3	AF	2	Afghanistan	2513	Barley and products	5142	Food	1000 tonnes	33.94	67.71	...	185.0	43.0	44.0	48.0	62.0	55.0
4	AF	2	Afghanistan	2514	Maize and products	5521	Feed	1000 tonnes	33.94	67.71	...	120.0	208.0	233.0	249.0	247.0	195.0
5	AF	2	Afghanistan	2514	Maize and products	5142	Food	1000 tonnes	33.94	67.71	...	231.0	67.0	82.0	67.0	69.0	71.0
6	AF	2	Afghanistan	2517	Millet and products	5142	Food	1000 tonnes	33.94	67.71	...	15.0	21.0	11.0	19.0	21.0	18.0
7	AF	2	Afghanistan	2520	Cereals, Other	5142	Food	1000 tonnes	33.94	67.71	...	2.0	1.0	1.0	0.0	0.0	0.0
8	AF	2	Afghanistan	2531	Potatoes and products	5142	Food	1000 tonnes	33.94	67.71	...	276.0	294.0	294.0	260.0	242.0	250.0
9	AF	2	Afghanistan	2536	Sugar cane	5521	Feed	1000 tonnes	33.94	67.71	...	50.0	29.0	61.0	65.0	54.0	114.0
10	AF	2	Afghanistan	2537	Sugar beet	5521	Feed	1000 tonnes	33.94	67.71	...	0.0	0.0	0.0	0.0	0.0	0.0

Train/test split

La répartition est soit :

- Donnée par la base de données. En anglais on parle de fold (train ou test), qui est indiqué dans les données
- Soit faite par le programmeur. Dans ce cas, on choisit un pourcentage arbitraire. Le train est majoritaire.
  - On peut choisir par exemple 80% des données pour le train et 20% pour le test.

Evaluer le classifieur

Comment évaluer le classifieur ?

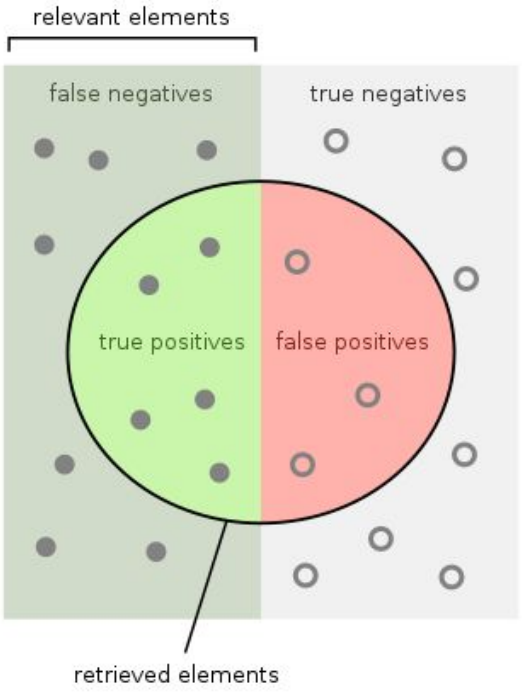
1. Accuracy
2. Précision
3. Recall/ Rappel (sensibilité)
4. F measure

Comment qualifier les prédictions du classifieur ?

Positif/Negatif

Pour un exemple labélisé comme positif :

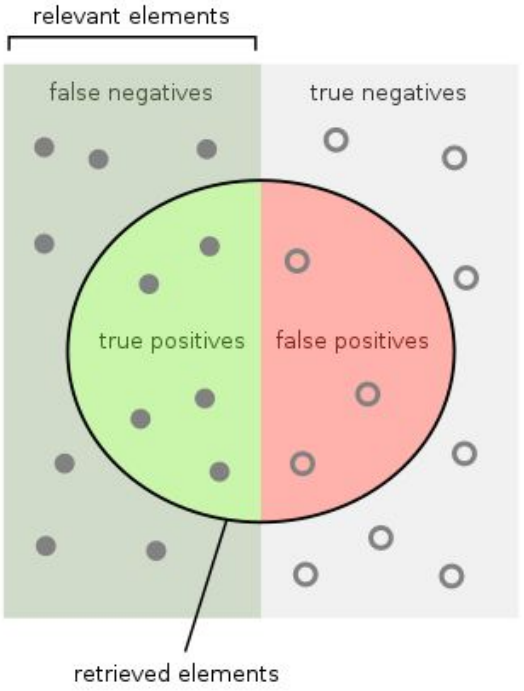
- Si le modèle prédit qu'il est positif alors c'est un vrai positif (TP)
- Si le modèle prédit qu'il est négatif, alors c'est un faux négatif (FN)



Positif/Negatif

Pour un exemple labélisé comme négatif :

- Si le modèle prédit qu'il est négatif, alors c'est un vrai négatif (TN)
- Si le modèle prédit qu'il est positif alors c'est un faux positif (FP)



Faux/Vrai Positif/Négatif

Réalité (labels dans la base de données)

Prédiction du classifieur

	Negatif = 0	Positif = 1
Negatif = 0	Vrai négatif (TN)	Faux négatif (FN)
Positif = 1	Faux positif (FP)	Vrai positif (TP)

Accuracy

Accuracy = % de bonnes classification

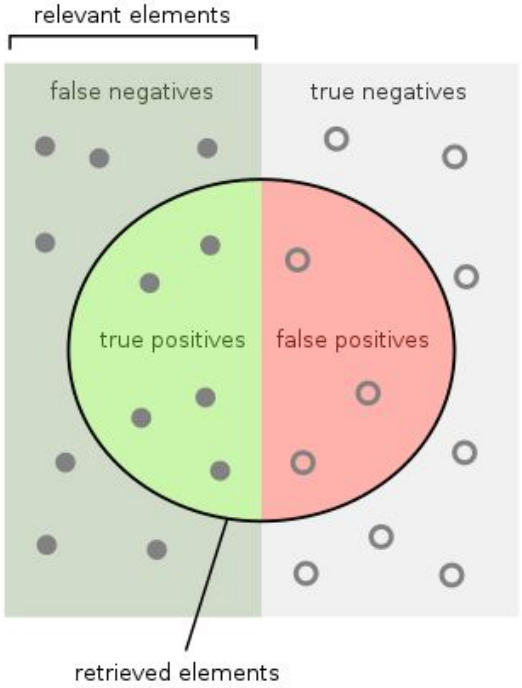
$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{TP + TN}{|\mathcal{D}|}$$

Calcule, en pourcentage, le nombre de fois où le modèle a prédit la bonne classe, positive ou négative.

Précision

Elle permet de connaître le nombre de **prédictions positives** bien effectuées.

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



A chaque fois que le modèle a prédit positif, combien de fois il a eu raison .

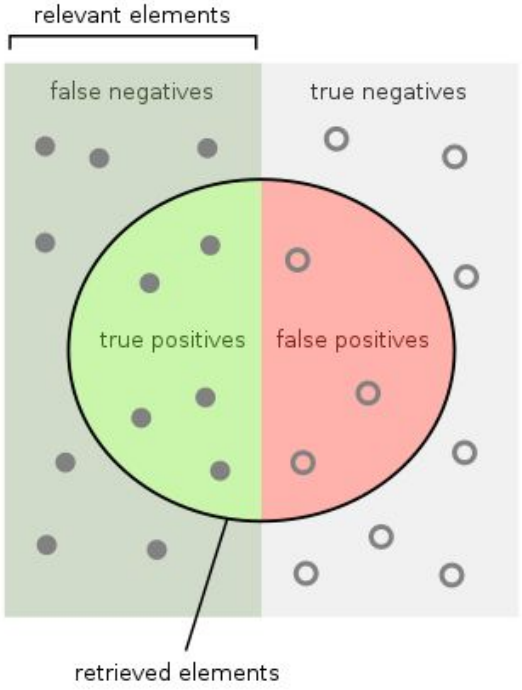
Plus la précision est élevée, moins il y a de faux positifs.

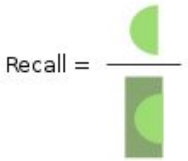
Recall / Rappel

Permet de savoir le pourcentage de positifs bien prédit par notre modèle.

$$\text{Recall} = \frac{TP}{TP + FN}$$

Plus le recall est élevée, plus on maximise le nombre de vrai positif





Classification multi classes - matrice de confusion

Visualisation pour se rendre compte des performances d'un modèle de classification multi classes.

Matrice de taille KxK (nombre de classes par nombre de classes).

Prenons l'exemple de classification en 3 classes d'images : Chat, Chien et Vache.

Notre matrice aura 3 lignes et 3 colonnes. Les éléments seront les prédictions de notre modèle.

Matrice de confusion

Sachant la ééalité (labels dans la base de données)

Prédiction du classifieur

	Chien	Chat	Vache
Chien	50	10	5
Chat	25	40	0
Vache	10	0	55

Matrice de confusion

Sachant la ééalité (labels dans la base de données)

Prédiction du classifieur

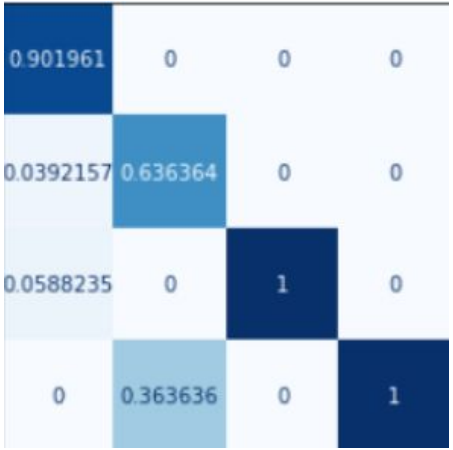
	Chien	Chat	Vache
Chien	50	10	5
Chat	25	40	0
Vache	10	0	55

Signifie que le modèle a prédit chat pour 25 images qui étaient en fait labélisées chien dans la base de données

Matrice de confusion

Idéalement, on voudrait une matrice diagonale (nulle partout sauf sur la diagonale).

Permet de savoir quelles classes sont confondues par notre modèle.



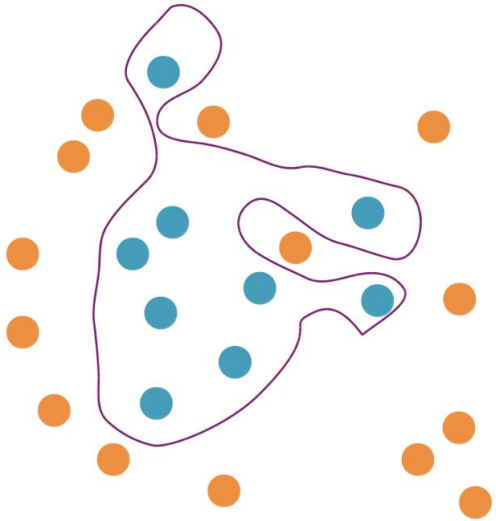
Sur apprentissage, Généralisation

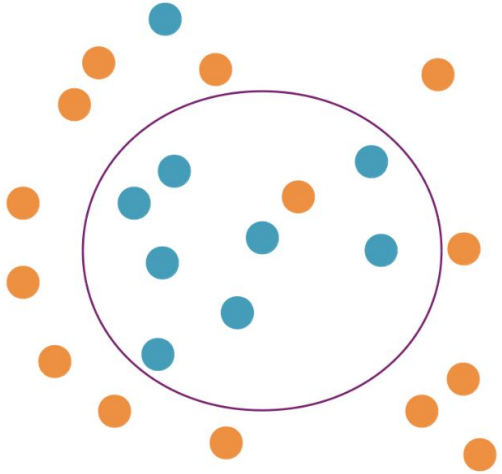
Généralisation : capacité d'un modèle à faire des prédictions correctes sur de nouvelles données

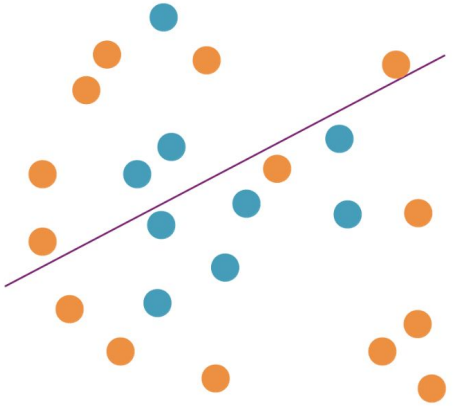
C'est pour cela qu'on parle d'apprentissage.

Si un modèle est très bon sur les données d'apprentissage, mais très mauvais sur de nouvelles données, on parle alors de **surapprentissage.**

Surapprentissage







Quel modèle préfère-t-on ?

Surapprentissage et généralisation

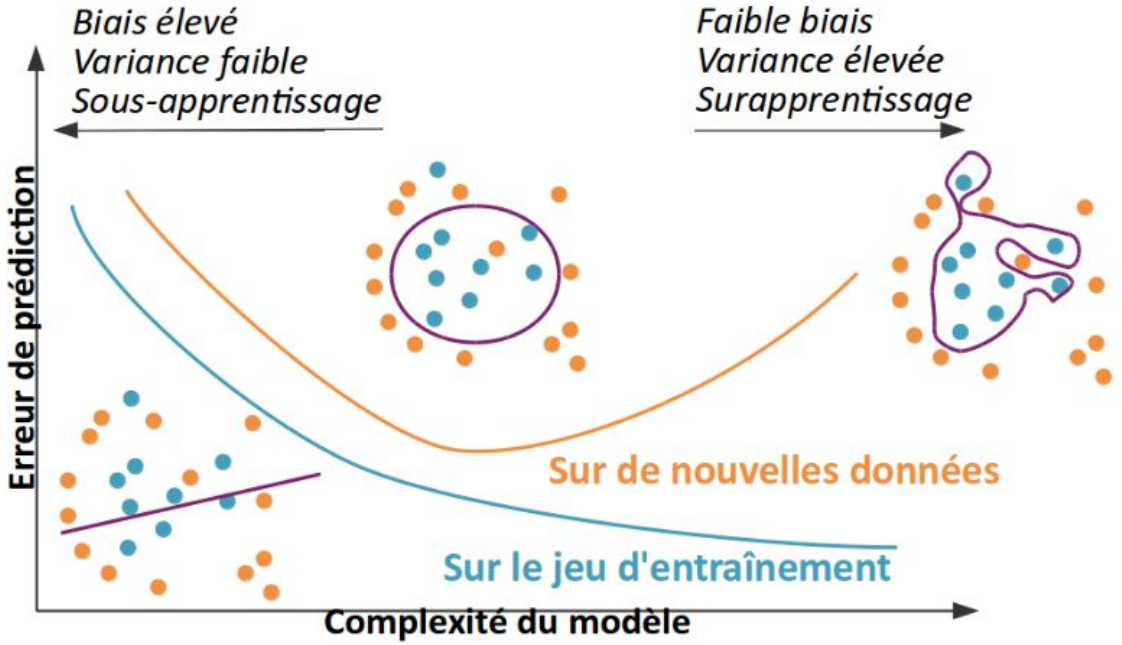
Peu importe si on fait quelques erreurs sur les données d'entrainement, l'important est que notre modèle capture le phénomène qui nous intéresse.

Un modèle simple (variance faible) risque le sous apprentissage (biais élevé sur les données d'entrainement).

Un modèle complexe (variance élevée) risque le sur apprentissage (biais faible sur les données d'entrainement, élevé sur des nouvelles).

On souhaite trouver un modèle intermédiaire.

Compromis biais / variance



KNN: K plus proche voisin

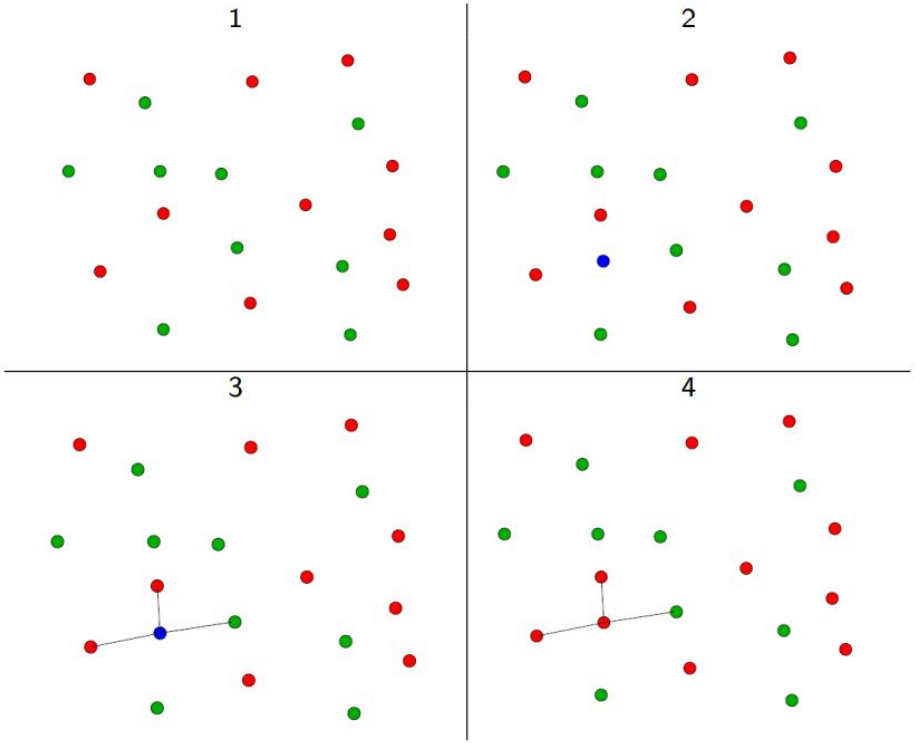
Principe

Les exemples d'entrainements sont des vecteurs de dimension p. On stocke ces vecteurs et leur label associé.

On cherche à classifier un nouvel exemple.

1. On va calculer sa **distance** à tous les autres exemples de la base d'entrainement
2. On sélectionne les K plus proche voisins
3. On choisit la classe du nouvel exemple en fonction de la classe de ses K plus proches voisins.

Exemple avec K=3



A choisir

On doit choisir à l'avance :

1. La distance que l'on souhaite utiliser
2. Le nombre de voisin que l'on considère, K
    a. En général, si K est grand, la classification est meilleure

Distance

On peut utiliser plusieurs distances pour deux vecteurs X et Y de taille n:

- Distance euclidienne $d(X,Y) = \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2}$
- Distance de Manhattan $d(X,Y) = \sum_{i=1}^{n}|y_i - x_i|$

Et il en existe plein d'autres …

Implémentation

- Implémenter une fonction de distance (en fonction de la distance choisie)
- Calculer la distance pour chaque exemple du test avec tous les points de la base de données
- Stocker ces distances dans une liste
- Trier la liste (`np.sort`)
- Choisir les k plus proches voisins
- Implémenter une fonction de décision de la classe
- Calculer les métriques de classification

Représentations

On a vu que l'on faisait la distance sur les exemples de la base de données.

On peut aussi choisir une autre représentation que celle brute des données.

Par exemple, utiliser la sortie d'une PCA pour faire un KNN dessus !