Apprentissage supervisé
Cours 1 : Introduction, concepts fondamentaux et classification binaire

27 Septembre 2021

Présentation (succincte) du cours

- Jour 1 : Introduction, concepts fondamentaux et classification binaire
  - Contexte de l'apprentissage supervisé et problèmes standards
  - Vocabulaire, concepts fondamentaux
  - Classification binaire, lien avec la régression logistique, illustration
- Jour 2 : Support Vector Machines (and if time permits Neural Networks)
  - SVM pour la classification linéaire
  - Introduction aux noyaux et SVM non linéaire
- Jour 3 : Arbres de régression et classification, forêts aléatoires
  - Arbres de classification et de régression
  - Forêts aléatoires de Breiman
  - Fondement des forêts aléatoires, bootstrapping, bagging
- Jour 4 : Introduction à la théorie de l'apprentissage
  - Cadre d'apprentissage PAC, erreur de généralisation
  - Bornes sur l'erreur de généralisation
  - Minimisation du risque empirique
  - Sélection de modèle, validation croisée
- Jour 5 : Projet final
  - Exercices théoriques
  - Projet Kaggle

Organisation du cours

- **Cours :**
  - Présentation des méthodes d'apprentissage
  - Exemples pratiques avec Python
  - Exercices théoriques
  - Exercices d'entraînement à la maison
- **Notation :**
  - DM (en groupe de 2-3 étudiants) : partie théorique et analyse d'un jeu de données

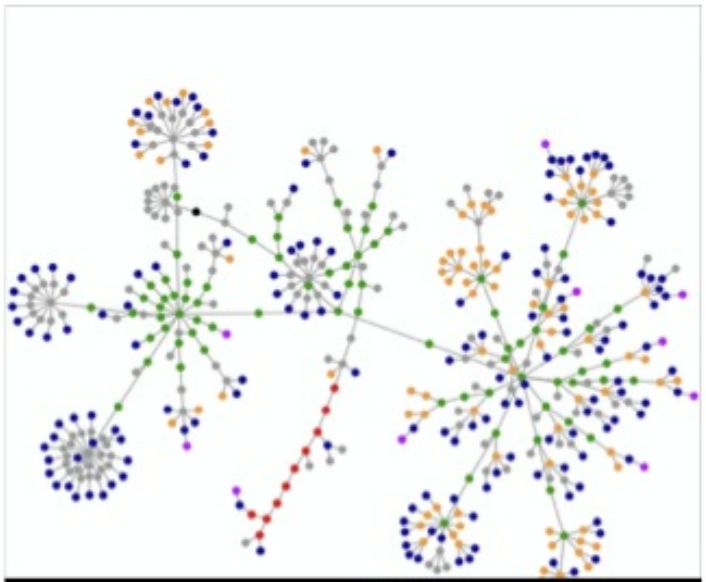
Une définition de l'apprentissage automatique

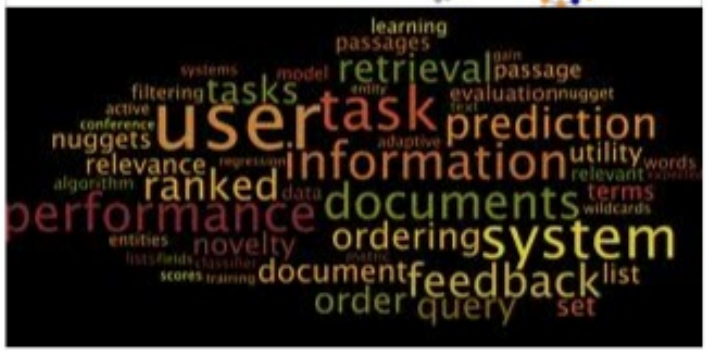
Une définition de Tom Mitchell (http://www.cs.cmu.edu/~tom/)

On dit d'un programme informatique qu'il apprend de l'expérience E par rapport à une classe de tâches T au sens d'une mesure de performance P, si sa performance aux tâches de T, telle que mesurée par P, s'améliore avec l'expérience E.

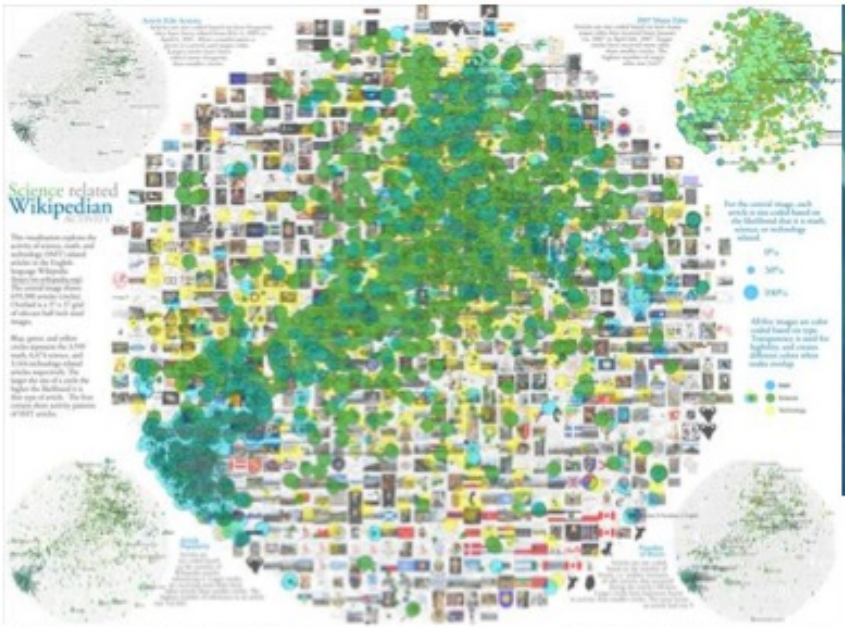
Machine learning

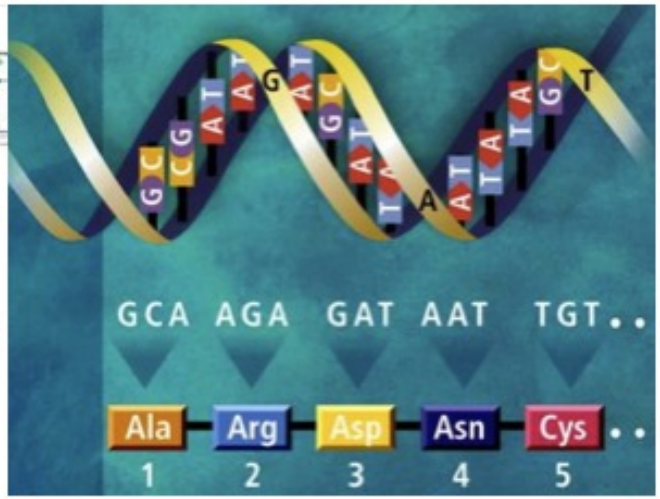
Motivation

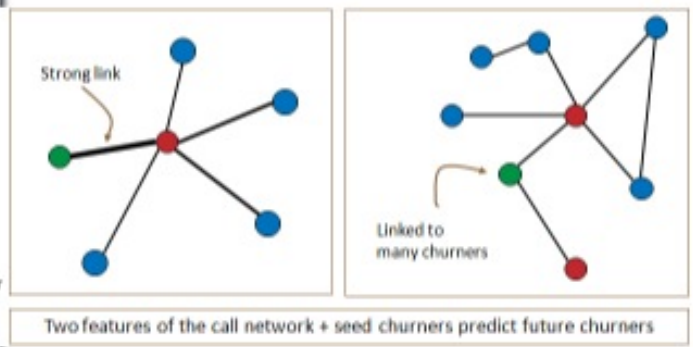




Machine Learning everywhere !







Two features of the call network + seed churners predict future churners

Machine Learning everywhere !

Utiliser les données pour prédire

- Moteur de recherche, text mining
- Diagnostique, Détection d'anomalie
- Business analytics
- Social networks
- Data-mining

Une définition de l'apprentissage automatique

Une définition de Tom Mitchell (http://www.cs.cmu.edu/~tom/)

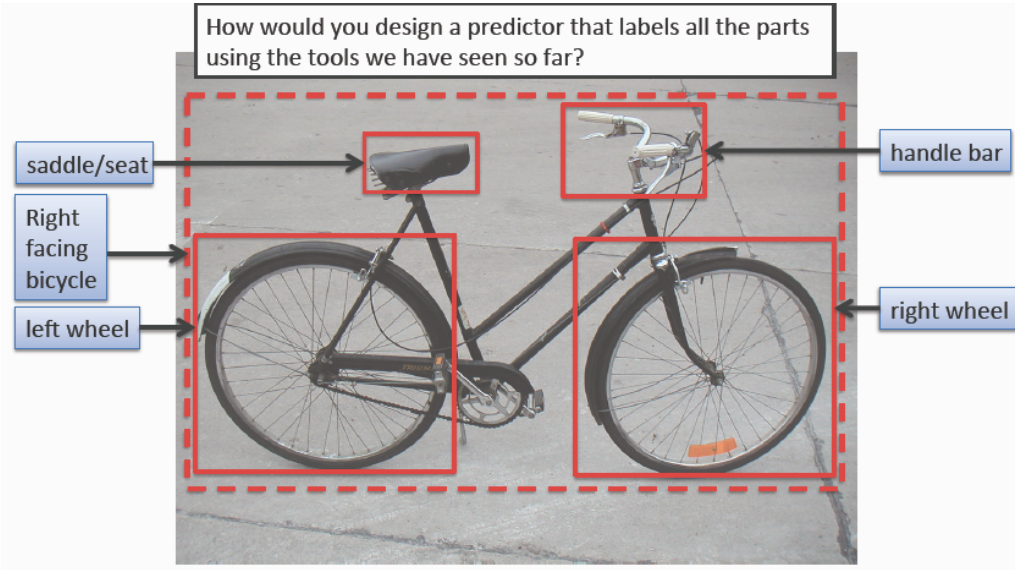
On dit d'un programme informatique qu'il apprend de l'expérience E par rapport à une classe de tâches T au sens d'une mesure de performance P, si sa performance aux tâches de T, telle que mesurée par P, s'améliore avec l'expérience E.

Apprentissage d'un robot

Un robot doté d'un ensemble de capteurs et d'un algorithme d'apprentissage en ligne :

- **Tâche** : jouer au football (américain)
- **Performance** : score
- **Expérience** :
  - environnement courant et résultats,
  - matches passés

Reconnaissance d'objet dans les image



- **Tâche** : dire si un objet est présent ou non dans l'image.
- **Performance** : nombre d'erreurs
- **Expérience** : ensemble d'images étiquetées vues précédemment

Deux types d'apprentissage 1/2

Online learning : *l'algorithme d'apprentissage continue d'interagir avec l'environnement.*

- robotiques
- social networks
- serveurs cloud
- publicité personnalisée
- voitures automatiques
- médecine personnalisée

Deux types d'apprentissage 2/2

Offline ou batch learning : *l'algorithme d'apprentissage reçoit un fichier de données et produit une fonction qui peut être utilisée à son tour pour de nouvelles données.*

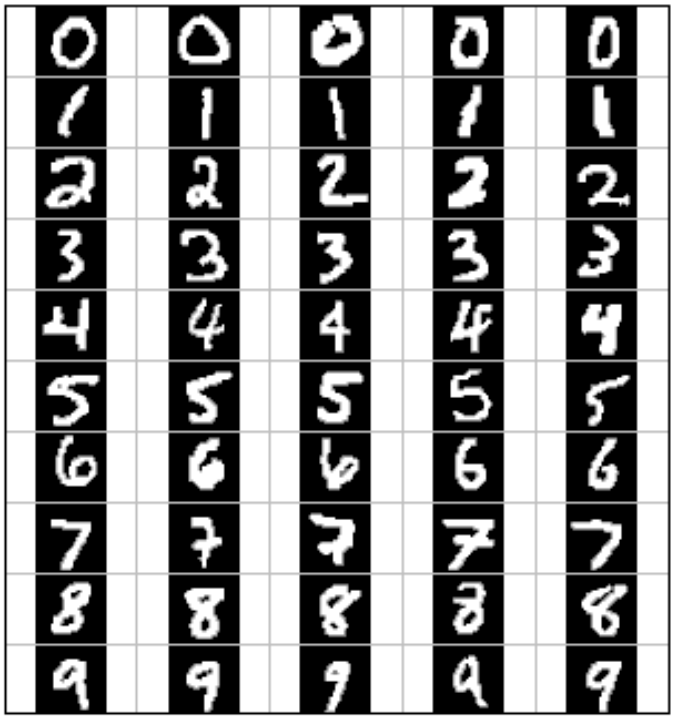
- Reconnaissance d'objet
- Diagnostic (santé, usines, ...)
- Prédiction de liens dans les résaux

Dans ce cours $\rightarrow$ offline learning

Supervisé et non-supervisé

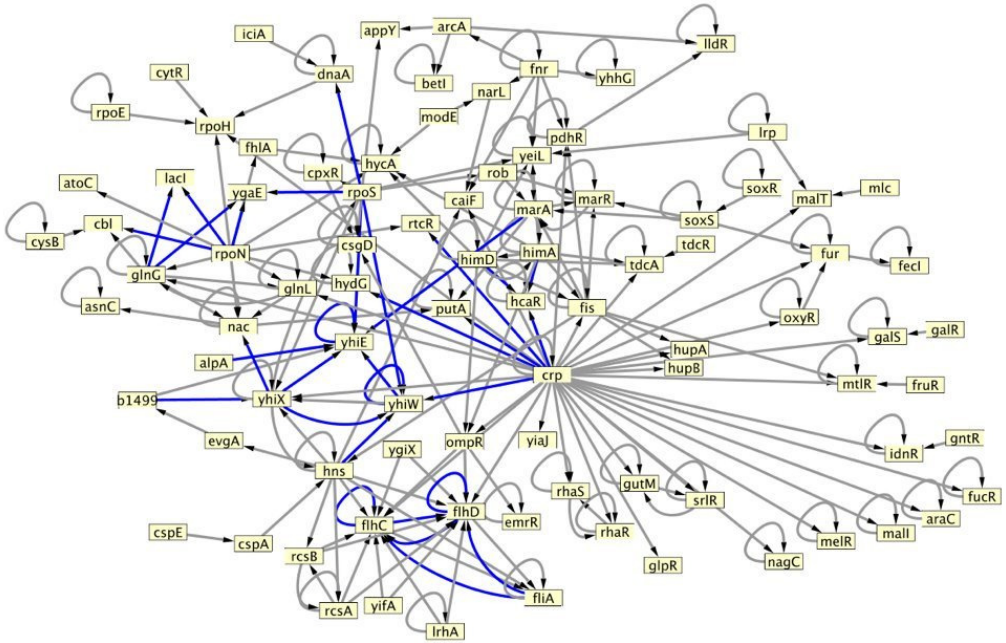
- Apprentissage supervisé/Supervised learning :
  - Objectif : Apprendre une fonction f pour prédire une variable Y pour un individu de features $\mathbf{X}$.
  - Data : Données d'apprentissage $(\mathbf{X}_i, Y_i)$
- Apprentissage non-supervisé/Unsupervised learning :
  - Objectif : découvrir une structure au sein d'un ensemble d'individus $(\mathbf{X}_i)$.
  - Data : Données d'apprentissage $(\mathbf{X}_i)$
- Le premier cas est mieux posé.
- Dans ce cours $\rightarrow$ supervised learning

Données MNIST



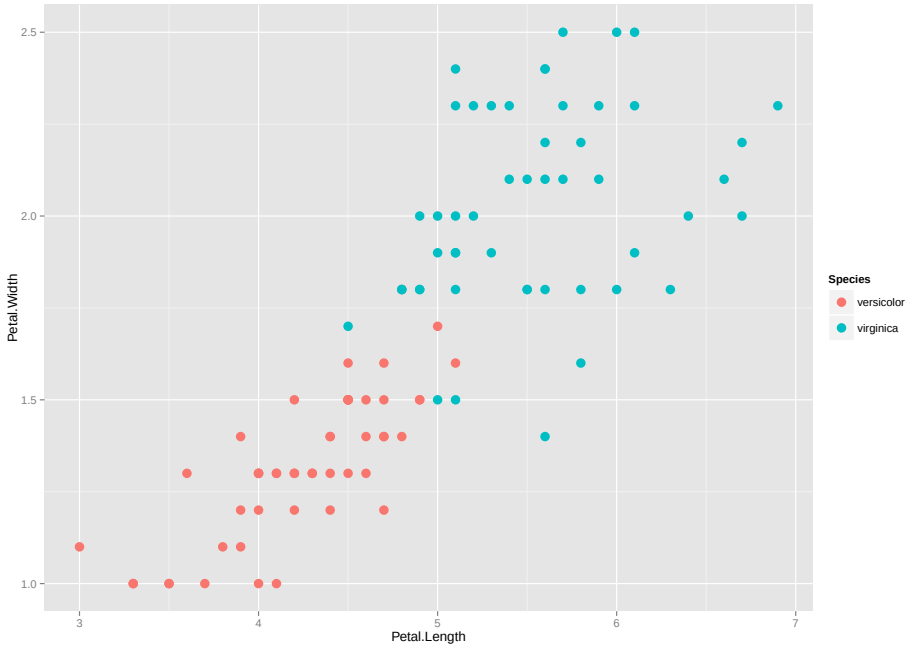
- Lire un code postal sur une enveloppe.
- Objectif : donner un nombre à partir d'une image.
- $\mathbf{X} =$ image.
- $Y =$ nombre correspondant.

Biologie



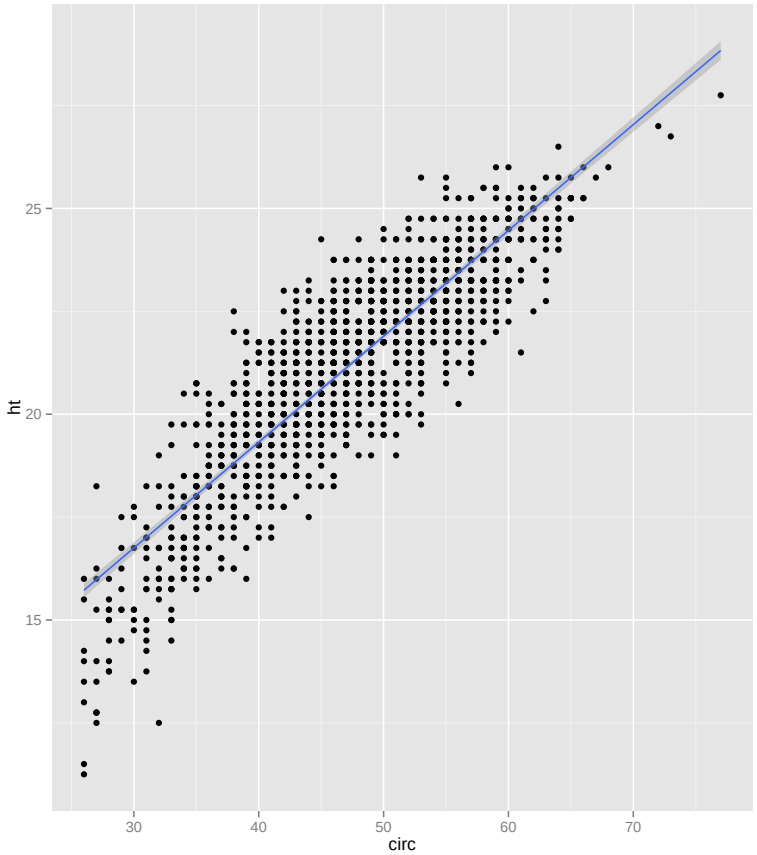
- Prédiction de réseaux d'interactions protéiques.
- Objectif : Prédire les interactions (inconnues) entre les protéines.
- $\mathbf{X}$ = paire de protéines.
- Y = existence ou non d'une interaction.
- Nombreuses questions similaires en bio(informatique) : génomique,...

Iris



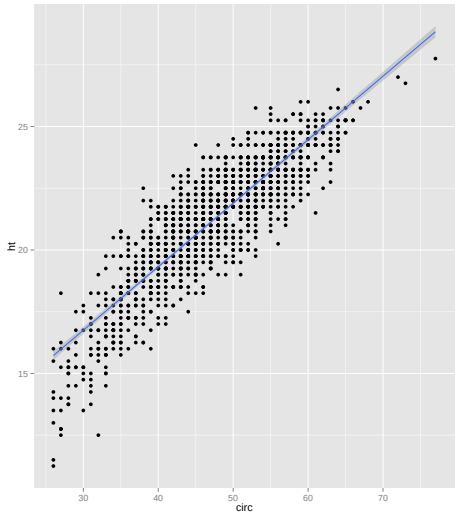
- Jeu de données classique.
- Objectif : prédire l'espèce à partir de la forme du pétale..
- $\mathbf{X}$ = hauteur/largeur du pétale.
- Y = espèce.

Eucalyptus



- Jeu de données simple et classique.
- Objectif : prédire la hauteur à partir de la circonférence
- $\mathbf{X} =$ **circ** $=$ circonférence.
- $Y = ht =$ hauteur.

Eucalyptus



Modèle linéaire

- Modèle paramétrique :

$$f_\beta(\textbf{circ}) = \beta_1 + \beta_2 \textbf{circ}$$

- Comment choisir $\beta = (\beta_1, \beta_2)$?

Moindres carrés

Méthodologie

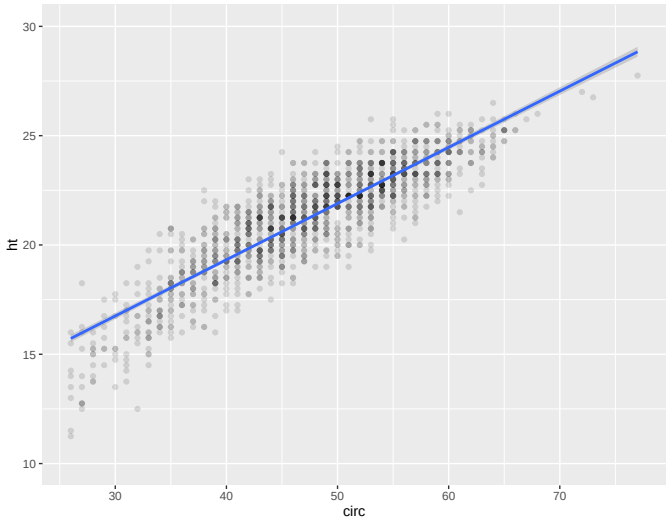
- Critère de qualité naturel :

$$\sum_{i=1}^{n} |Y_i - f_\beta(\mathbf{X}_i)|^2 = \sum_{i=1}^{n} |\mathsf{ht}_i - f_\beta(\mathbf{circ}_i)|^2$$
$$= \sum_{i=1}^{n} |\mathsf{ht}_i - (\beta_1 + \beta_2 \mathbf{circ}_i)|^2$$

- On choisit le β qui minimise ce critère

$$\widehat{\beta} = \arg\min_{\beta \in \mathbb{R}^2} \sum_{i=1}^{n} |h_i - (\beta_1 + \beta_2 \mathbf{circ}_i)|^2$$

Prédiction

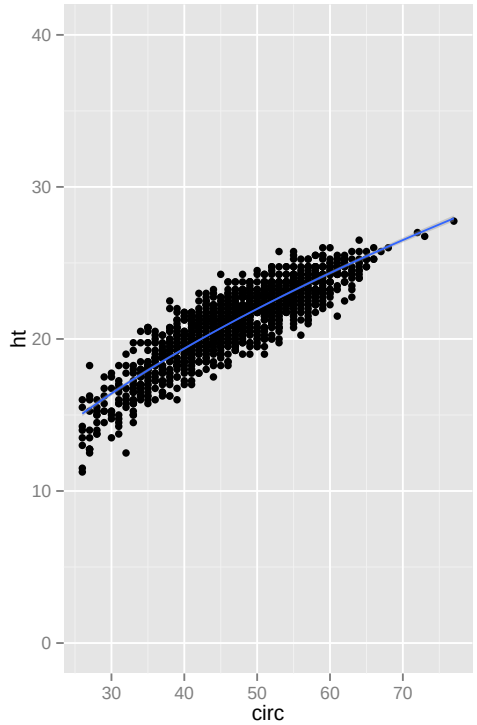


Prédiction

- Prédiction linéaire pour la hauteur :

$$\widehat{\text{ht}} = f_{\widehat{\beta}}(\textbf{circ}) = \widehat{\beta}_1 + \widehat{\beta}_2 \textbf{circ}$$

Régression polynomiale

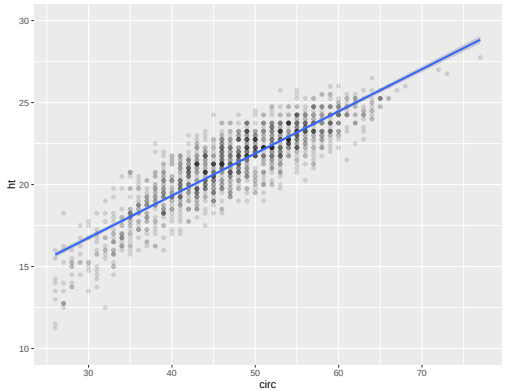


Modèle polynomial

- Modèle polynomial : $f_\beta(\mathbf{circ}) = \sum_{l=1}^{p} \beta_l \mathbf{circ}^{l-1}$

- Linéaire en β !
- Estimation facile des moindres carrés pour tout degré !

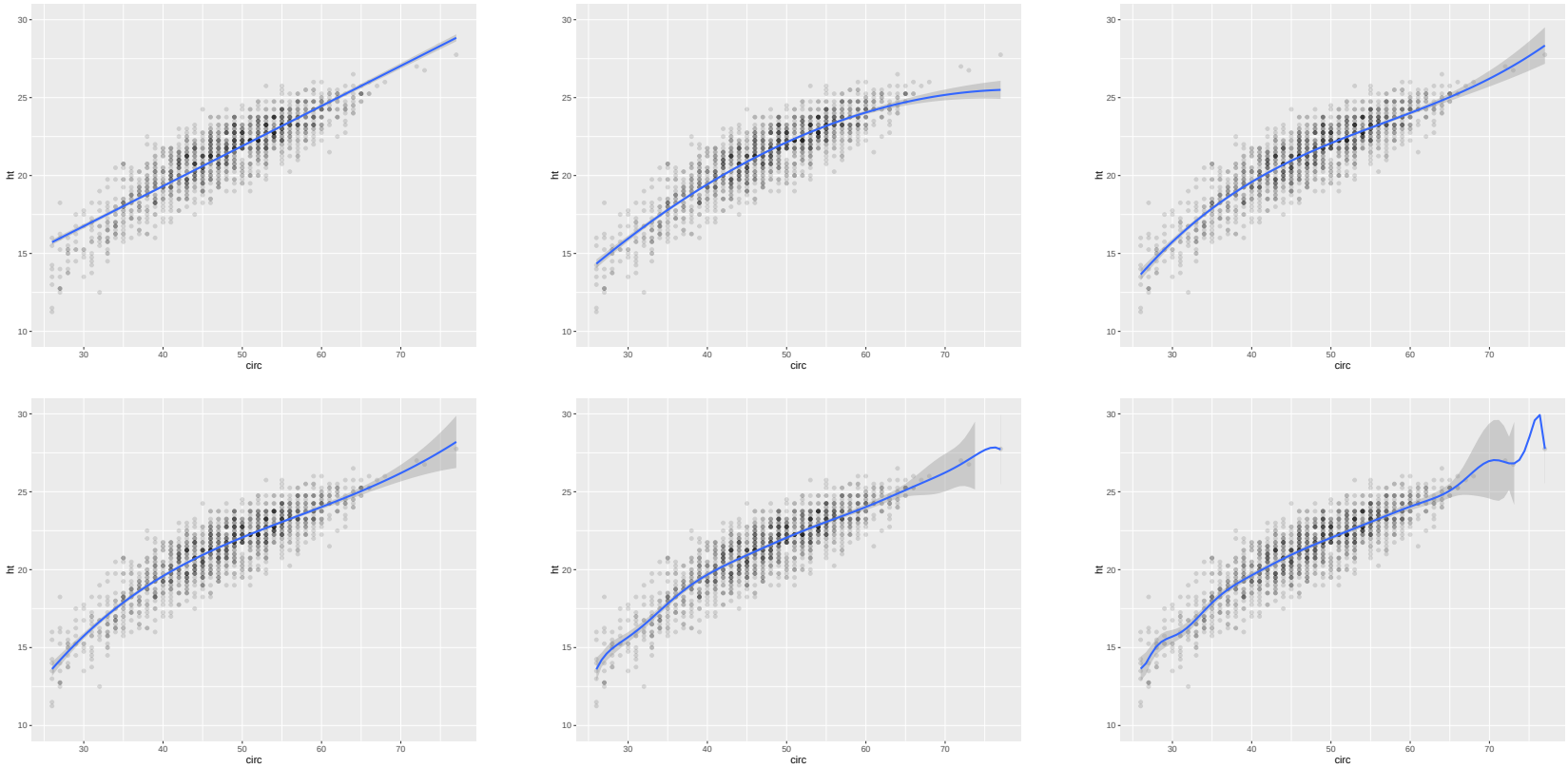
Comment choisir le degré ?



Modèles

- Augmentation du degré = complexité croissante et une meilleure adéquation sur les données

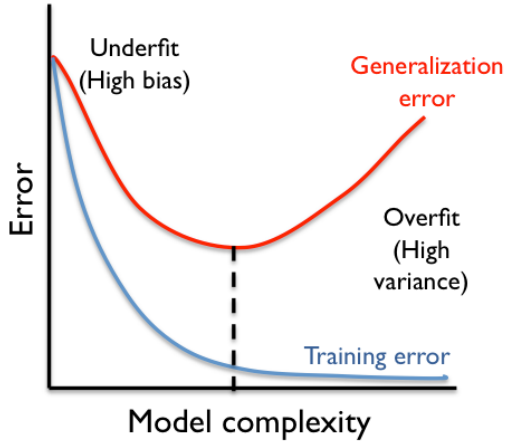
Comment choisir le degré ?



Meilleur degré ?

- Comment choisir parmi ces solutions ?

Over-fitting



Comportement de l'erreur

- Le risque empirique (erreur faite sur l'ensemble d'apprentissage) décroît lorsque la complexité du modèle augmente.
- Comportement tout à fait différent lorsque l'erreur est calculée sur de nouvelles observations (risque réel / erreur de généralisation).

- Overfit pour les modèles complexes : les paramètres appris sont trop spécifiques à l'ensemble d'apprentissage.

Validation croisée et pénalisation

Deux directions

- Comment estimer l'erreur de généralisation ?
- Trouver un moyen de corriger l'erreur empirique ?

Deux approches

- **Validation croisée :** Estimer l'erreur sur un autre ensemble données :
  - Très efficace (c'est la solution la plus usitée)
  - Nécessite plus de données pour calculer l'erreur de généralisation
- **Approche par pénalisation :** Corriger l'optimisme de l'erreur empirique :
  - Exige de trouver la correction (pénalité).

Apprentissage supervisé

Cadre de l'apprentissage statistique supervisé

- Données d'input (features) $\mathbf{X} = (X^{(1)}, X^{(2)}, \ldots, X^{(d)}) \in \mathcal{X}$
- Données d'output (réponse/label) $Y \in \mathcal{Y}$.
- $(\mathbf{X}, Y) \sim \mathbf{P}$ avec $\mathbf{P}$ inconnu.
- Données d'apprentissage/Training data : $\mathcal{D}_n = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$ (i.i.d. $\sim \mathbf{P}$)
- Souvent
  - $\mathbf{X} \in \mathbb{R}^d$ et $Y \in \{-1, 1\}$ (classification)
  - ou $\mathbf{X} \in \mathbb{R}^d$ et $Y \in \mathbb{R}$ (régression).

- Un prédicteur est une fonction de $\mathcal{F} = \{f : \mathcal{X} \to \mathcal{Y} \text{ mesurable}\}$

Objectif

- Construire un bon predicteur $\widehat{f}$ à partir des données d'apprentissage.

- Il faut donc définir la notion de bon prédicteur.
- Formellement, la classification et la régression sont le même problème.

Fonction de coût et cadre probabiliste

Fonction de coût/Loss function

- Loss function : $\ell(Y, f(\mathbf{X}))$ mesure la qualité du prédicteur $f(\mathbf{X})$ pour *prédire* Y.
- Exemples :
  - Perte de prédiction : $\ell(Y, f(\mathbf{X})) = \mathbf{1}_{Y \neq f(\mathbf{X})}$
  - Perte quadratique : $\ell(Y, \mathbf{X}) = |Y - f(\mathbf{X})|^2$

Risque d'un classificateur générique

- Risque mesuré comme la perte moyenne pour un nouveau couple :

$$\mathcal{R}(f) = \mathbb{E}\left[\ell(Y, f(\mathbf{X}))\right] = \mathbb{E}_X\left[\mathbb{E}_{Y|\mathbf{X}}\left[\ell(Y, f(\mathbf{X}))\right]\right]$$

- Exemples :
  - Perte de prédiction : $\mathbb{E}\left[\ell(Y, f(\mathbf{X}))\right] = \mathbb{P}\left[Y \neq f(\mathbf{X})\right]$
  - Perte quadratique : $\mathbb{E}\left[\ell(Y, f(\mathbf{X}))\right] = \mathbb{E}\left[|Y - f(\mathbf{X})|^2\right]$

- **Attention :** Comme $\widehat{f}$ depend de $\mathcal{D}_n$, $\mathcal{R}(\widehat{f})$ est une variable aléatoire !

Apprentissage supervisé

Expérience, Tâche et mesure de performance

- Données d'apprentissage : $\mathcal{D} = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$ (i.i.d. $\sim \mathbf{P}$)
- Prédicteur : $f : \mathcal{X} \to \mathcal{Y}$ mesurable
- Coût/Loss function : $\ell(Y, f(\mathbf{X}))$ mesure la qualité de la prédiction de $f(\mathbf{X})$ par rapport Y
- Risque :

$$\mathcal{R}(f) = \mathbb{E}\left[\ell(Y, f(\mathbf{X}))\right] = \mathbb{E}_X\left[\mathbb{E}_{Y|\mathbf{X}}\left[\ell(Y, f(\mathbf{X}))\right]\right]$$

- Souvent $\ell(Y, f(\mathbf{X})) = |f(\mathbf{X}) - Y|^2$ ou $\ell(Y, f(\mathbf{X})) = \mathbf{1}_{Y \neq f(\mathbf{X})}$

Objectif

- Apprendre une règle pour construire un prédicteur $\widehat{f} \in \mathcal{F}$ à partir des données d'apprentissage $\mathcal{D}_n$ s.t. le risque $\mathcal{R}(\widehat{f})$ soit petit en moyenne ou avec une forte probabilité par rapport à $\mathcal{D}_n$.

Meilleure Solution

- La meilleure solution f^* (qui est indépendante de $\mathcal{D}_n$) est

$$f^* = \arg\min_{f \in \mathcal{F}} R(f) = \arg\min_{f \in \mathcal{F}} \mathbb{E}\left[\ell(Y, f(\mathbf{X}))\right] = \arg\min_{f \in \mathcal{F}} \mathbb{E}_{\mathbf{X}}\left[\mathbb{E}_{Y|\mathbf{X}}\left[\ell(Y, f(\mathbf{x}))\right]\right]$$

Prédicteur de Bayes (solution explicite)

- En classification binaire avec perte $0-1$:

$$f^*(\mathbf{X}) = \begin{cases} +1 & \text{si } \mathbb{P}\left[Y=+1|\mathbf{X}\right] \geq \mathbb{P}\left[Y=-1|\mathbf{X}\right] \\ & \Leftrightarrow \mathbb{P}\left[Y=+1|\mathbf{X}\right] \geq 1/2 \\ -1 & \text{sinon} \end{cases}$$

- Dans la régression avec la perte quadratique

$$f^*(\mathbf{X}) = \mathbb{E}\left[Y|\mathbf{X}\right]$$

Problème : La solution explicite nécessite de savoir $EspY|\mathbf{X}$ pour toutes les valeurs de $\mathbf{X}$!

Objectif

Machine Learning

- Apprendre une règle pour construire un classifieur $\widehat{f} \in \mathcal{F}$ à partir des données d'apprentissage $\mathcal{D}_n$ s.t. le risque $\mathcal{R}(\widehat{f})$ est petit en moyenne ou avec grande probabilité par rapport à $\mathcal{D}_n$.

Exemple canonique : Minimiseur du risque empirique

- On restreint f à un sous-ensemble de fonctions $\mathcal{S} = \{f_\theta, \theta \in \Theta\}$
- On remplace la minimisation de la perte moyenne par la minimisation de la perte empirique

$$\widehat{f} = f_{\widehat{\theta}} = \underset{f_\theta, \theta \in \Theta}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, f_\theta(\mathbf{X}_i))$$

- Exemples :
  - Régression linéaire
  - Discrimination linéaire avec

$$\mathcal{S} = \{\mathbf{x} \mapsto \mathtt{sign}\{\beta^T \mathbf{x} + \beta_0\} / \beta \in \mathbb{R}^d, \beta_0 \in \mathbb{R}\}$$

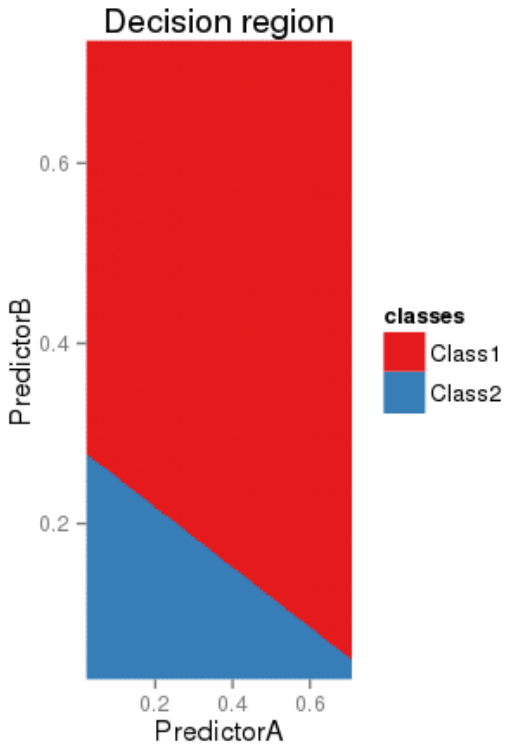
Exemple : Jeu de données à deux classes

Données synthétiques

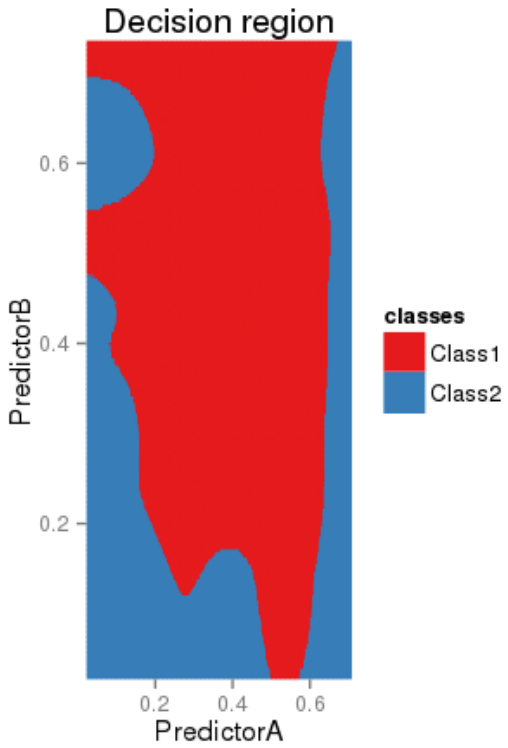
- Deux features/covariables.
- Deux classes.

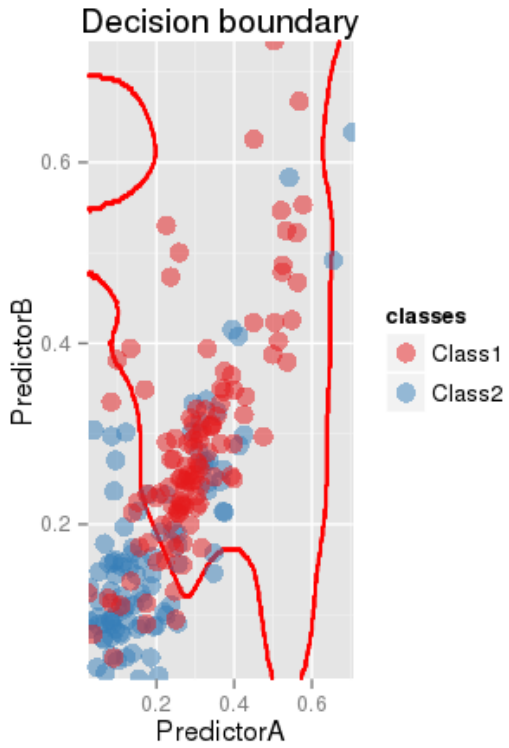
- Dataset pris dans *Applied Predictive Modeling*, M. Kuhn and K. Johnson, Springer
- Expériences numériques avec **R** et le package **caret**.

Exemple : Discrimination linéaire

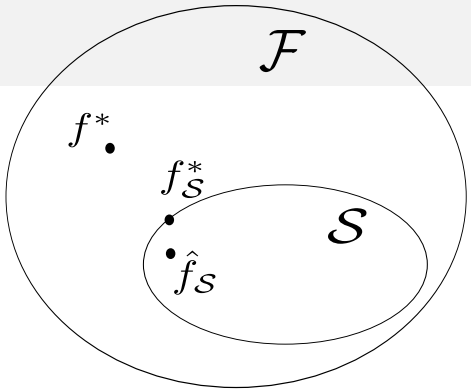


Exemple : modèle plus complexe





Dilemme biais-variance I



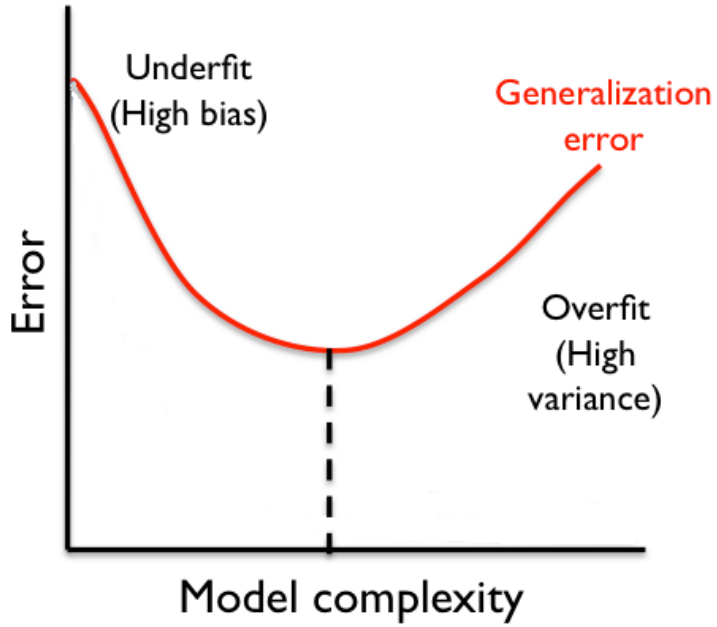
- Cadre général :
  - $\mathcal{F} = \{\text{fonctions mesurables } \mathcal{X} \to \mathcal{Y}\}$
  - Meilleure solution : $f^* = \arg\min_{f \in \mathcal{F}} \mathcal{R}(f)$
  - Classe de fonctions $\mathcal{S} \subset \mathcal{F}$
  - Cible idéale dans $\mathcal{S}$: $f^*_{\mathcal{S}} = \arg\min_{f \in \mathcal{S}} \mathcal{R}(f)$
  - Estimateur dans$\mathcal{S}$: $\widehat{f}_{\mathcal{S}}$ obtenu avec un algorithme quelconque

Erreur d'approximation et erreur d'estimation (Bias/Variance)

$$\mathcal{R}(\widehat{f}_{\mathcal{S}}) - \mathcal{R}(f^*) = \underbrace{\mathcal{R}(f^*_{\mathcal{S}}) - \mathcal{R}(f^*)}_{\text{Approximation error}} + \underbrace{\mathcal{R}(\widehat{f}_{\mathcal{S}}) - \mathcal{R}(f^*_{\mathcal{S}})}_{\text{Estimation error}}$$

- L'erreur d'approximation peut être grande si le modèle $\mathcal{S}$ n'est pas adapté.
- L'erreur d'estimation peut être importante si le modèle est complexe.

Sous-apprentissage / Sur-apprentissage



- Comportement différent selon la complexité du modèle
- Les modèles simples sont faciles à apprendre mais leur erreur d'approximation ("biais") peut être importante (sous-apprentissage).
- Les modèles complexes peuvent contenir une bonne cible idéale, mais l'erreur d'estimation ("variance") peut être importante (Sur-apprentissage)

Trade-off Biais-variance $\Leftrightarrow$ éviter le Sur-apprentissage et le sous-apprentissage

But du cours

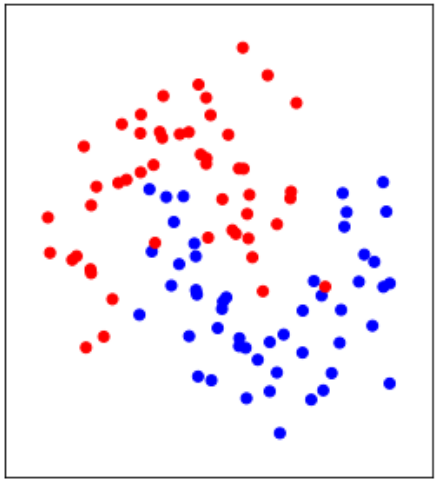
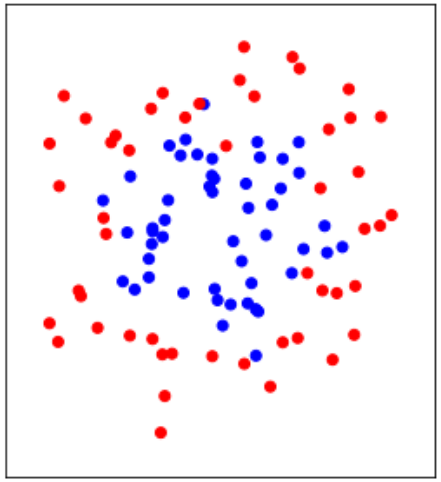
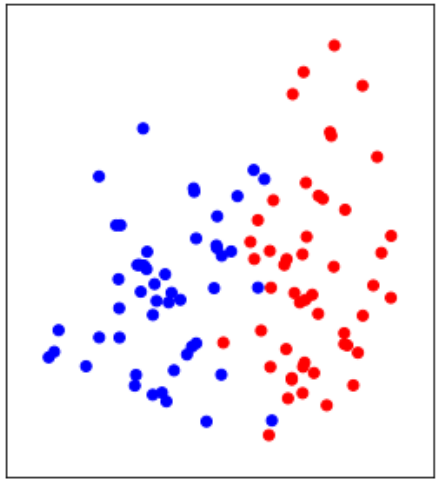
- Le but de ce cours est à la fois d'étudier de nouveaux algorithmes de classification/régression mais également d'obtenir des résultats mathématiques sur les algorithmes décrits

- On parlera beaucoup de classification mais, dans la plupart des cas, cela s'étend aux problèmes d'apprentissage supervisé quand les labels sont continus (voir remarques, exercices)

Spam detection

- Données : emails
- Input : email
- Output : Spam or No Spam

Classification binaire : toy datasets

- But : retrouver la classe
- Input : 2 predicteurs
- Output : classe

Classification multi-classes

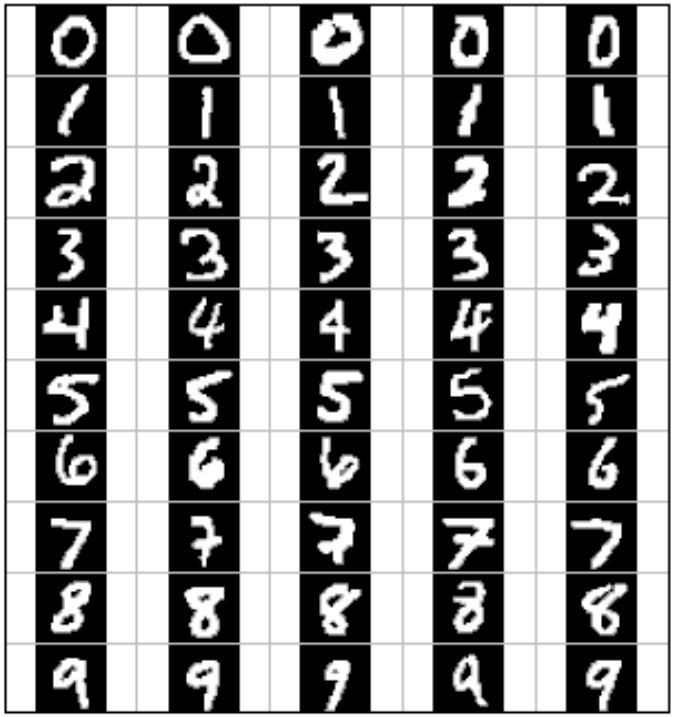
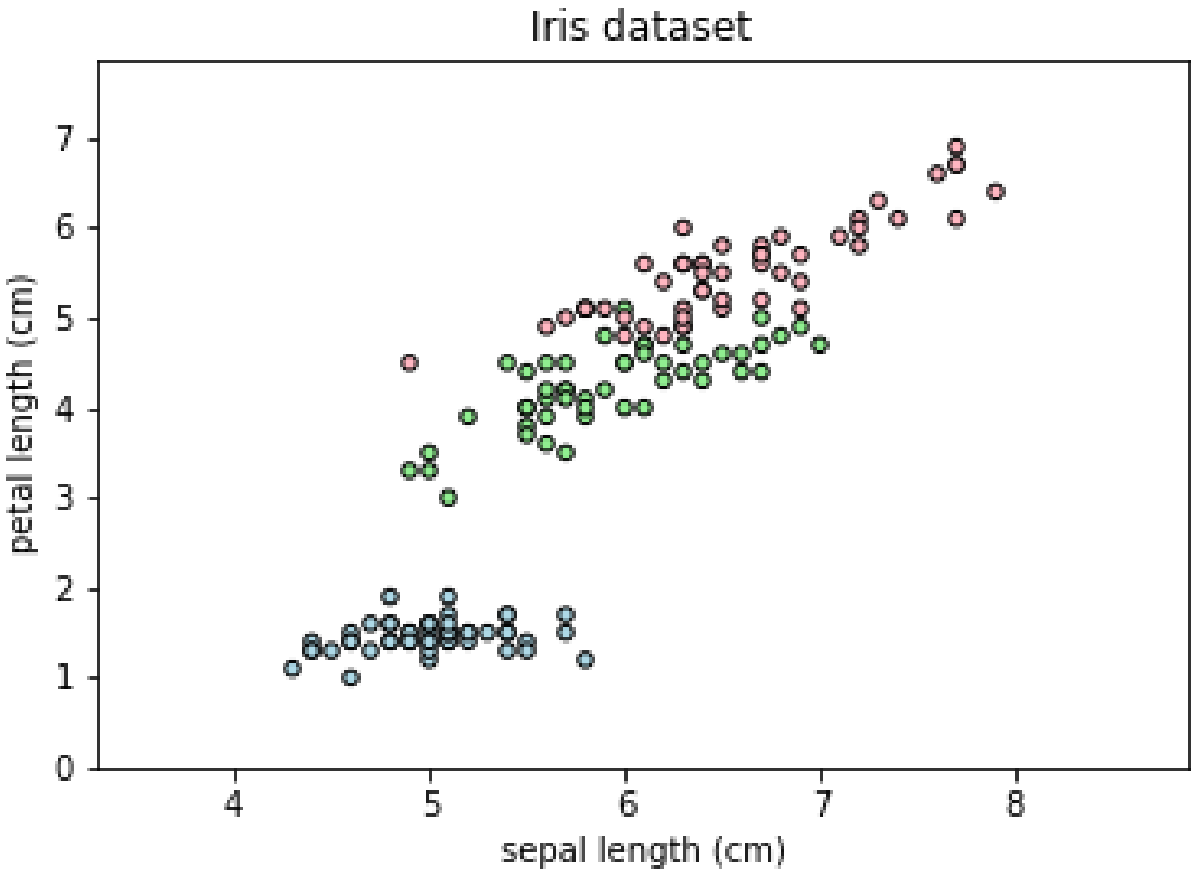


FIGURE – Jeu de données MNIST

- Lire un code postal sur une enveloppe.
- But : assigner un chiffre à une image.
- Input : image.
- Output : chiffre correspondant.

Classification multi-classes : Iris dataset



- But : retrouver la classe
- Input : 2 predicteurs
- Output : classe

Le problème de classification binaire

On a des données d'apprentissage (learning data) pour des individus $i = 1, \ldots, n$. Pour chaque individu i :

- on a un vecteur de covariables (features) $X_i \in \mathcal{X} \subset \mathbb{R}^d$
- la valeur de son label $Y_i \in \{-1, 1\}$.
- on suppose que les couples (X_i, Y_i) sont des copies i.i.d. de (X, Y) de loi inconnue et que l'on observe leurs réalisations (x_i, y_i) $(i = 1, \ldots, n)$.

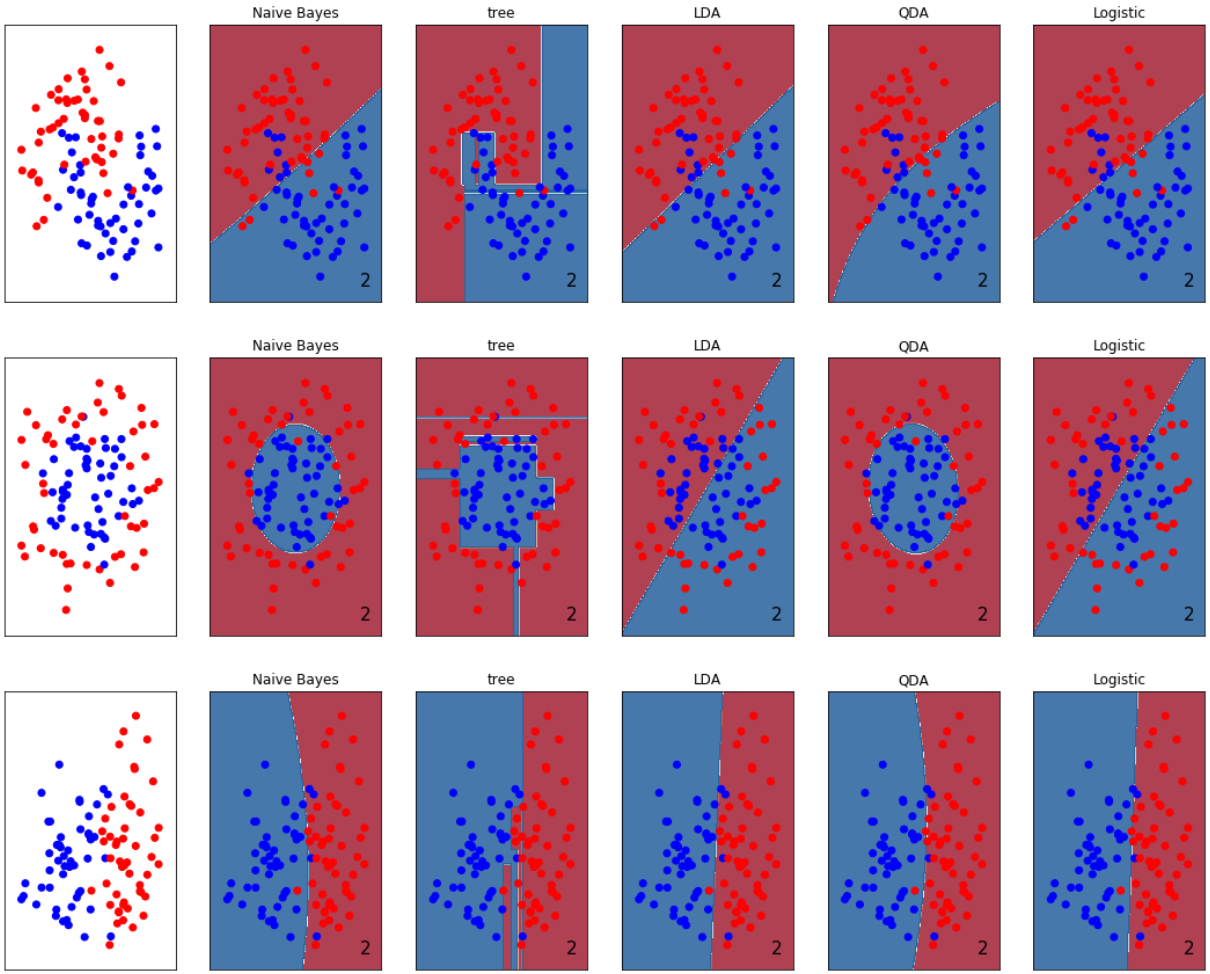
But

- On veut, pour un nouveau vecteur X_+ de features, prédire la valeur du label Y_+ par $\hat{Y}_+ \in \{-1, 1\}$
- Pour cela, on utilise les données d'apprentissage $\mathcal{D}_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ pour construire un **classifieur** $\hat{c}$ de telle sorte que

$$\hat{Y}_+ = \hat{c}(X_+).$$

et $\hat{Y}$ est proche de Y_+ (dans un sens à préciser).

Classification binaire : toy datasets



Le problème de classification multi-classes

On a des données d'apprentissage (learning data) pour des individus $i = 1, \ldots, n$. Pour chaque individu i :

- on a un vecteur de covariables (features) $X_i \in \mathbb{R}^d$
- la valeur de son label $Y_i \in \mathcal{C} = \{1, \ldots, K\}$.
- on suppose que les couples (X_i, Y_i) sont des copies i.i.d. de (X, Y) de loi inconnue et que l'on observe leurs réalisations (x_i, y_i) $(i = 1, \ldots, n)$.

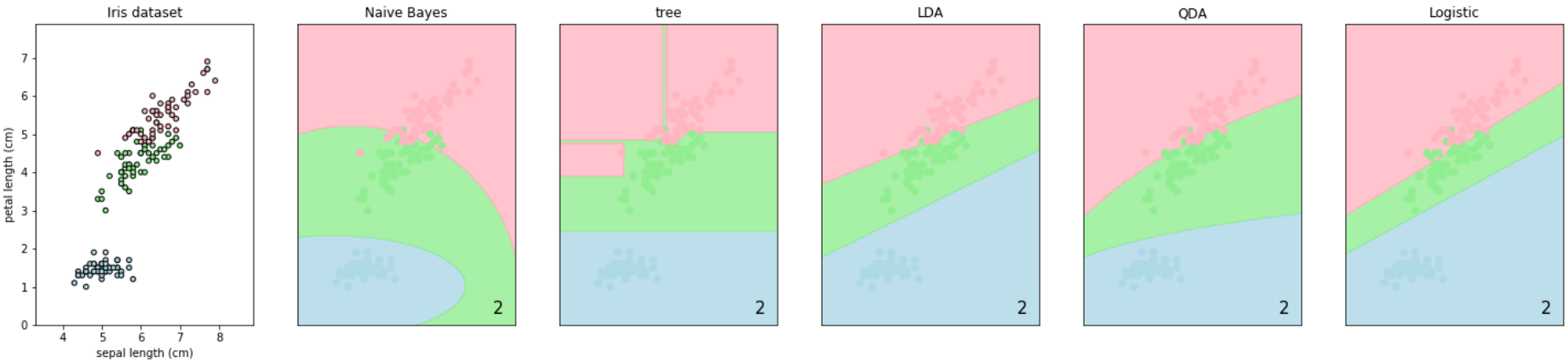
But

- On veut pour un nouveau vecteur X_+ de features prédire la valeur du label Y_+ par $\hat{Y}_+ \in \mathcal{C} = \{1, \ldots, K\}$
- Pour cela, on utilise les données d'apprentissage $\mathcal{D}_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ pour construire un **classifieur** $\hat{c}$ de telle sorte que

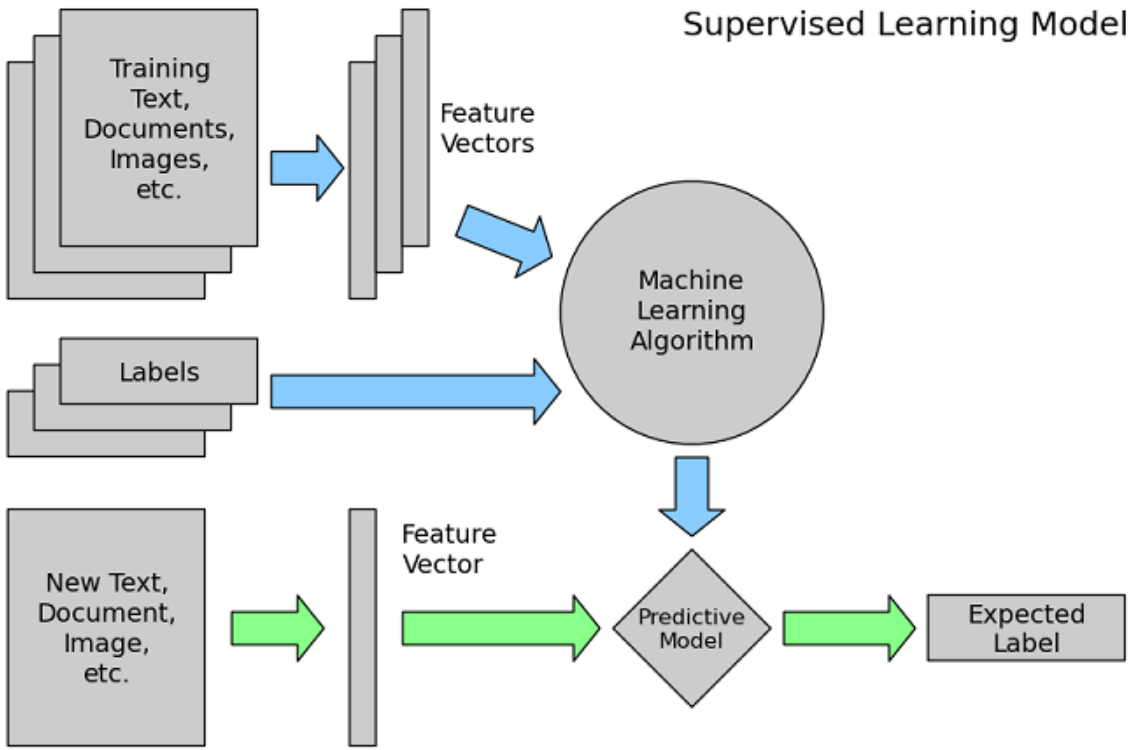
$$\hat{Y}_+ = \hat{c}(X_+)$$

et $\hat{Y}_+$ est proche de Y_+ (dans un sens à préciser).

Classification multi-classes : Iris dataset



Apprentissage statistique supervisé



- Input : covariables, variables explicatives, features $X = (X^1, \ldots, X^d)$
- Ouput : variable à expliquer, variable dépendante, réponse, label Y

Approche probabiliste / statistique en classification binaire

- Pour construire le classifieur $\hat{c}$, on construit des estimateurs $\hat{p}_1(x)$ et $\hat{p}_{-1}(x)$ de

$$p_1(x) = \mathbb{P}(Y = 1|X = x) \quad \text{et} \quad p_{-1}(x) = 1 - p_1(x)$$

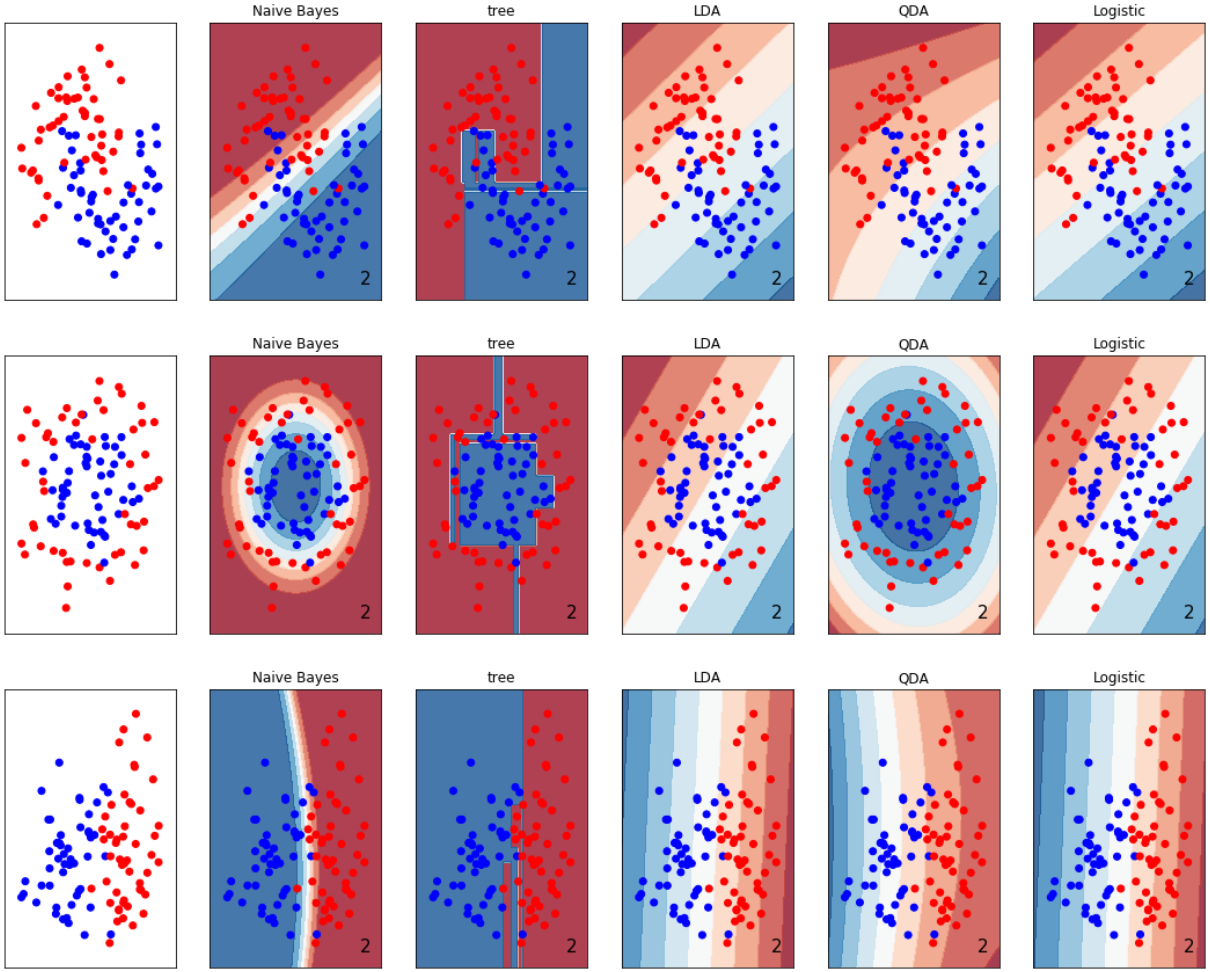
- en modélisant la loi de $Y|X$.
- Puis, conditionnellement à $X_+ = x$, on classifie en utilisant la règle

$$\hat{Y}_+ = \hat{c}(x) = \begin{cases} 1 & \text{si } \hat{p}_1(x) \geq s \\ -1 & \text{sinon} \end{cases}$$

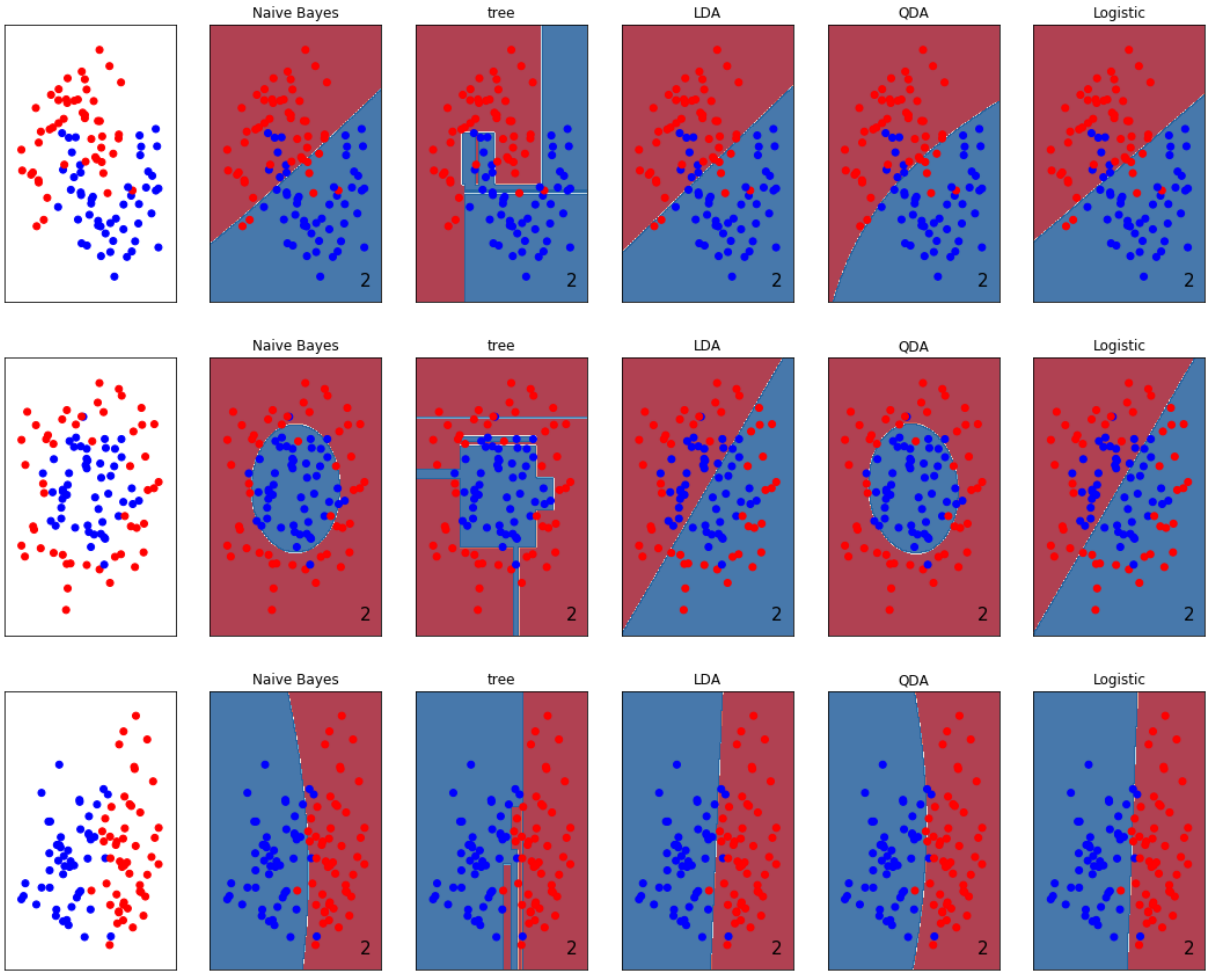
pour un seuil $s \in (0, 1)$.

- Si on choisit $s = 1/2$, cela revient à classifier suivant la plus grande valeur entre $\hat{p}_1(x)$ et $\hat{p}_{-1}(x)$ (on retient cette règle dans la suite).

Classification binaire : toy datasets



Classification binaire : toy datasets



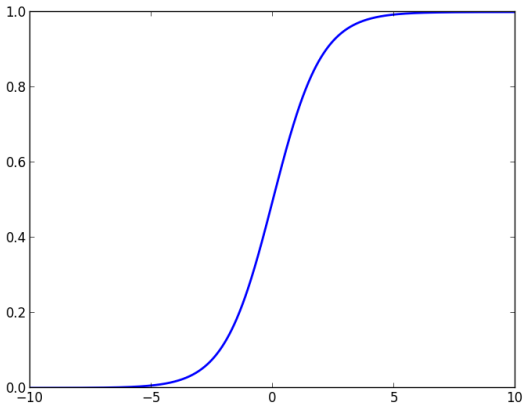
Régression logistique

- C'est le plus utilisé des algorithmes de classification
- On modélise la loi de $Y|X$ par

$$\mathbb{P}(Y = 1|X = x) = \sigma(\langle x, w \rangle + b)$$

où $w \in \mathbb{R}^d$ est un vecteur de régression ou de poids, $b \in \mathbb{R}$ est the **intercept**, et σ est la fonction **sigmoïde**

$$\sigma(z) = \frac{e^z}{1 + e^z} = \frac{1}{1 + e^{-z}}$$



Autres régressions

- Le choix de la fonction sigmoïde est lié à la loi de Bernoulli.
- On peut considérer d'autres fonctions de $\mathbb{R} \to [0,1]$ (car on veut modéliser une proba). Par exemple, toutes les fonctions de répartition

$$\mathbb{P}(Y = 1|X = x) = F(\langle x, w\rangle + b).$$

- Parmi celles-ci, la f.d.r. gaussienne est souvent utilisée

$$F(z) = \Phi(z) = \mathbb{P}(N(0,1) \leq z),$$

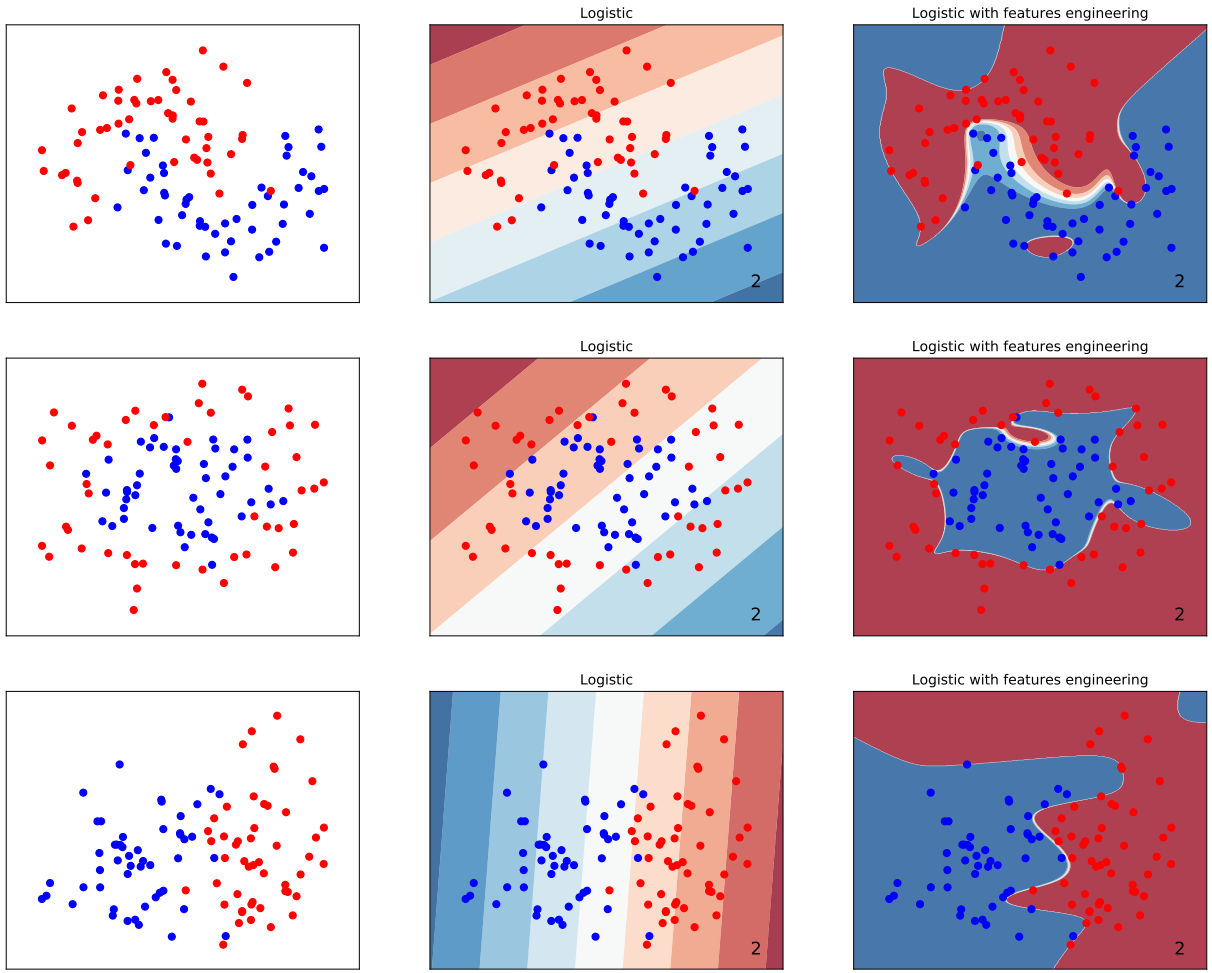
et on parle alors de régression **probit**.

- En classification multi-classes, on utilise la fonction **softmax** donnée par

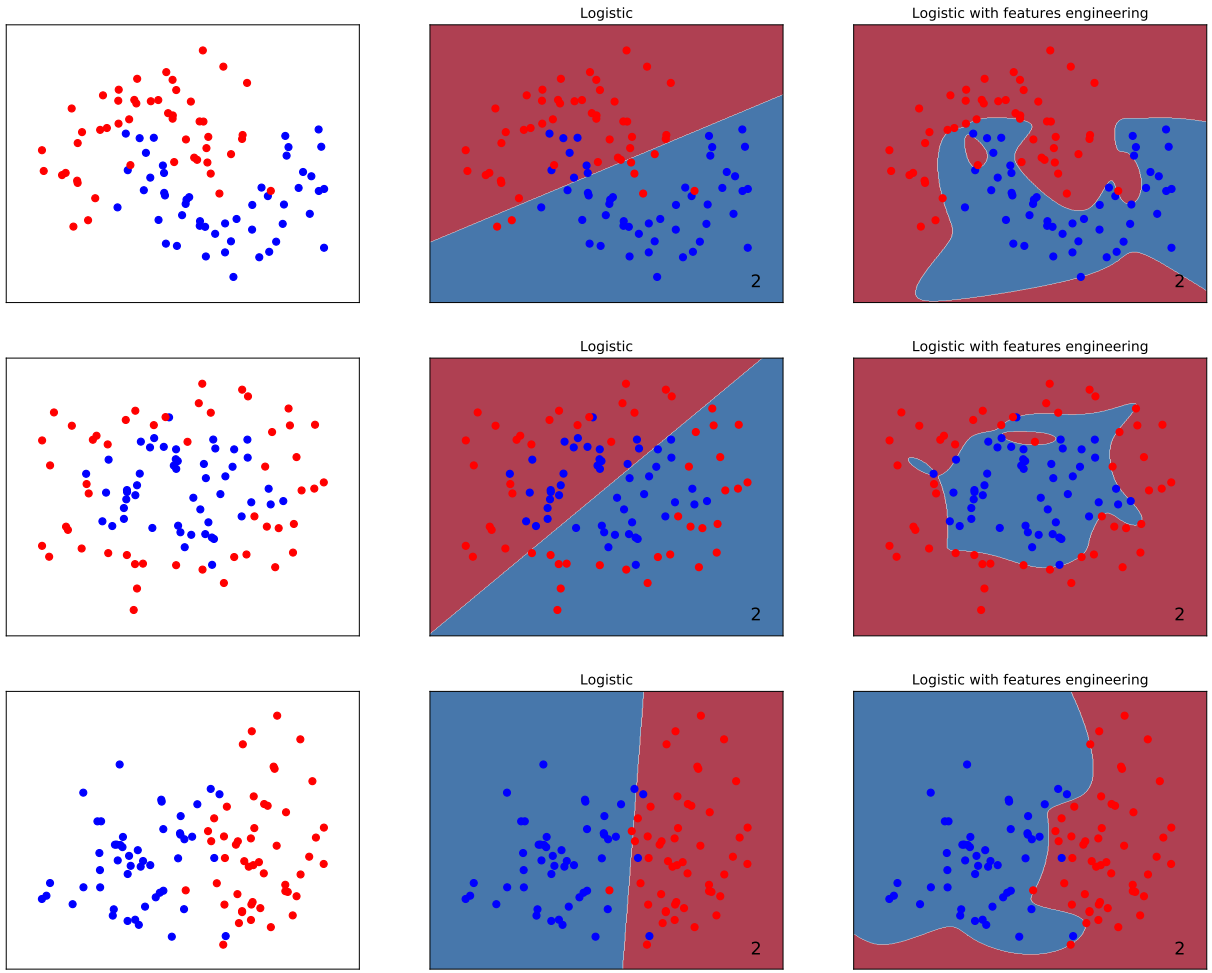
$$\mathbb{P}(Y = k|X = x) = \frac{\exp(\langle x, w_k\rangle + b_k)}{\sum_{k=1}^{K} \exp(\langle x, w_k\rangle + b_k)}$$

pour tout $k \in \mathcal{C} = \{1, \ldots, K\}$.

Illustration



Illustration



Règle de classification linéaire

- Remarquons que

$$\mathbb{P}(Y = 1|X = x) \geq \mathbb{P}(Y = -1|X = x)$$

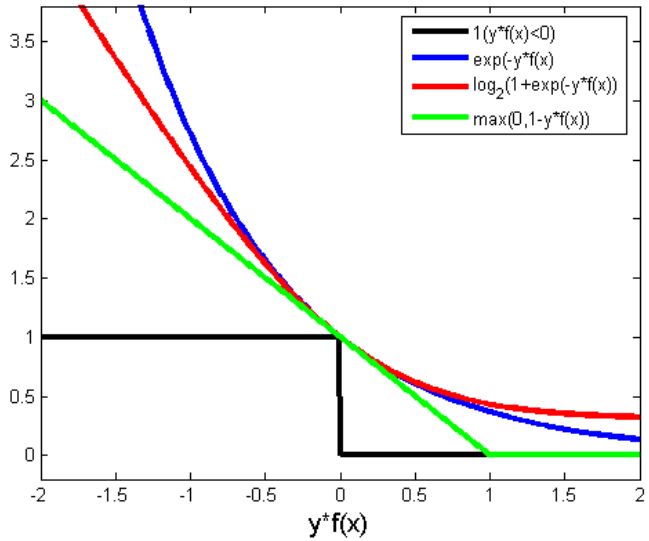
si et seulement si

$$\langle x, w \rangle + b \geq 0.$$

- On obtient une règle de classification linéaire, c'est-à-dire linéaire par rapport aux features
- Mais, on peut faire du **features engineering** (considérer comme covariables x^1,x^2, leur produit et leurs carrés, etc)

Autres pertes classiques pour la classification binaire

- Hinge loss (SVM), $\ell(y, y') = (1 - yy')_+$
- Quadratic hinge loss (SVM), $\ell(y, y') = \frac{1}{2}(1 - yy')_+^2$
- Huber loss $\ell(y, y') = -4yy' \mathbb{1}_{yy' < -1} + (1 - yy')_+^2 \mathbb{1}_{yy' \geq -1}$



- Toutes ces pertes peuvent être vus comme des approximations convexes de la perte $0/1$ $\ell(y, y') = \mathbb{1}_{yy' \leq 0}$