Algorithmique des données
Régression

Charlotte Pelletier
MCF Univ. Bretagne Sud – IRISA Vannes
Basé sur le cours de Chloé Friguet (MCF UBS/IRISA).

19 février 2020

Plan

Rappel

- Modélisation de la relation entre la variable à expliquer y et les variables explicatives $X^1, X^2, \cdots, X^d$
  - y est une variable à expliquer **qualitative** (binaire)
  - X^i : variables explicatives quantitatives ou qualitatives

➜ Le **modèle linéaire** n'est **pas adapté**

Rappel

- Modélisation de la relation entre la variable à expliquer y et les variables explicatives $X^1, X^2, \cdots, X^d$
  - y est une variable à expliquer **qualitative** (binaire)
  - X^i : variables explicatives quantitatives ou qualitatives

➜ Le **modèle linéaire** n'est **pas adapté**car y n'est pas quantitative !

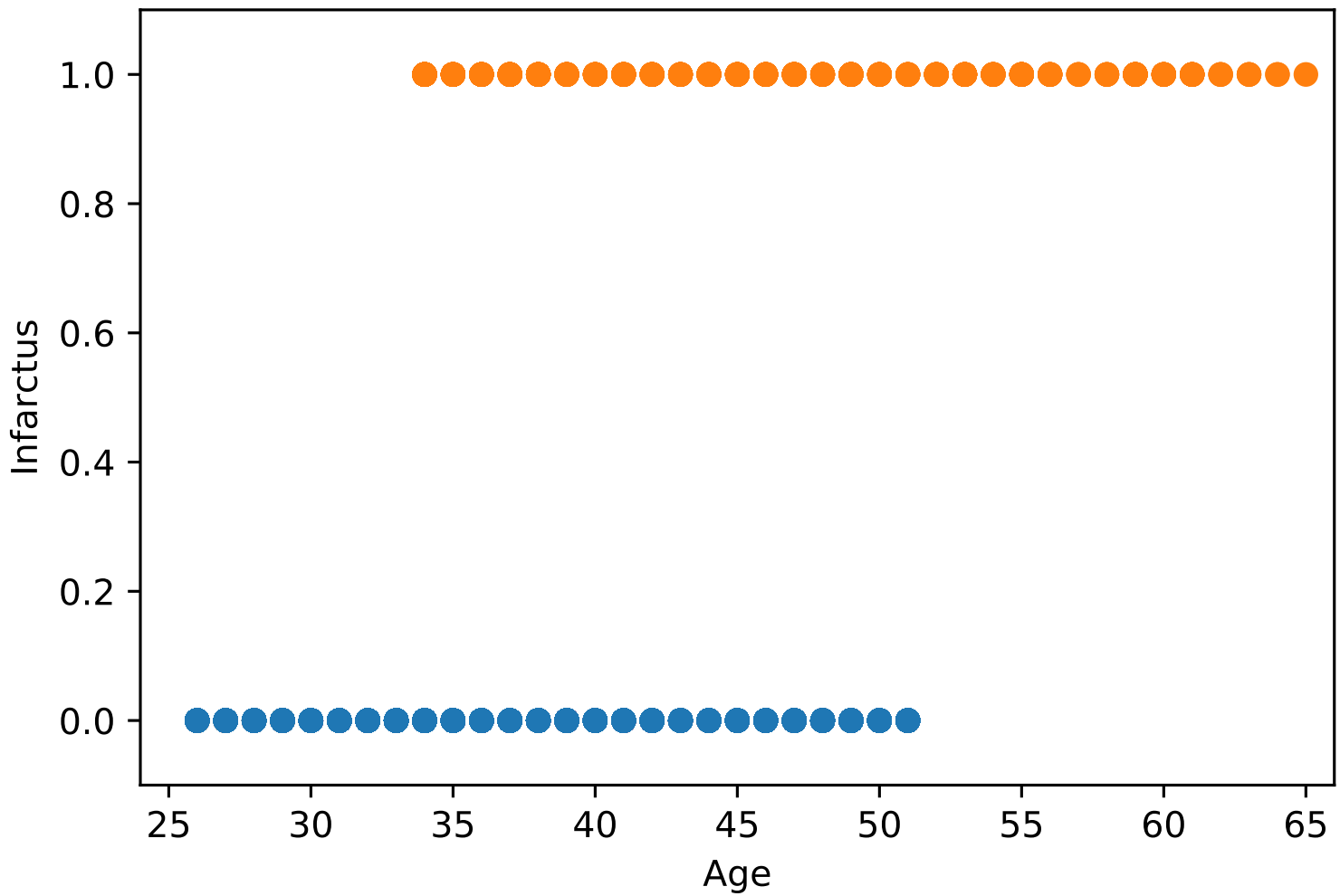
Exemple B

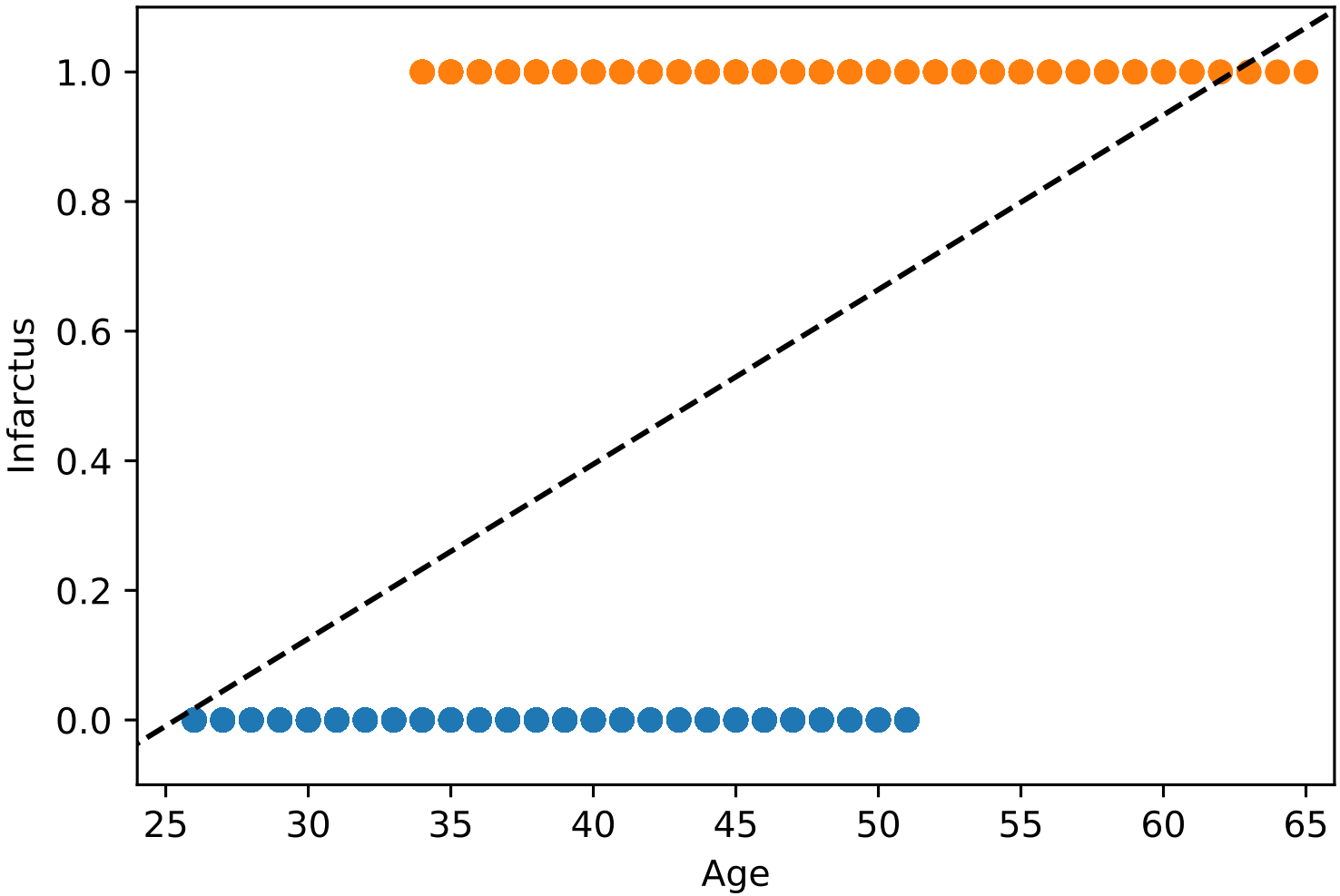
On souhaite évaluer l'existence d'un risque plus élevé de survenue d'un infarctus du myocarde chez les femmes à partir d'une étude cas-témoins. Les facteurs d'exposition recueillis sont : la prise de contraceptif, l'âge, le poids, la taille, la consommation de tabac, l'hypertension artérielle, les antécédents familiaux de maladies cardio-vasculaires, etc.

Extrait des données :

Infarctus	Contraceptif oral	Age	Poids	$\cdots$
non	non	47	48	$\cdots$
non	non	35	53	$\cdots$
non	oui	62	41	$\cdots$
oui	oui	47	45	$\cdots$
oui	oui	63	$\cdots$	
oui	non	45	69	$\cdots$
oui	non	90	84	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Données Institut de Santé Publique, d'Epidémiologie et de Développement (ISPED), Bordeaux
Source : `http://www.biostatisticien.eu/springeR/jeuxDonnees5.html`


Infarctus
1.0
0.8
0.6
0.4
0.2
0.0
25
30
35
40
45
50
55
60
65
Age


1.0
0.8
0.6
0.4
0.2
0.0
Infarctus
25
30
35
40
45
50
55
60
65
Age

Plan

Avec des probabilités

On s'adapte

- on ne va plus modéliser Y par une relation linéaire
- on s'intéresse aux probabilités : $\mathbf{P}(Y = 0|X = x)$ et $\mathbf{P}(Y = 1|X = x)$
- la connaissance de $\mathbf{P}(Y = 1|X = x)$ implique la connaissance de $\mathbf{P}(Y = 0|X = x)$ car

Avec des probabilités

On s'adapte

- on ne va plus modéliser Y par une relation linéaire
- on s'intéresse aux probabilités : $\mathbf{P}(Y = 0|X = x)$ et $\mathbf{P}(Y = 1|X = x)$
- la connaissance de $\mathbf{P}(Y = 1|X = x)$ implique la connaissance de $\mathbf{P}(Y = 0|X = x)$ car $\mathbf{P}(Y = 0|X = x) = 1 - \mathbf{P}(Y = 1|X = x)$

Loi de Bernoulli

Posons

$$\mathbf{P}(Y = 1|X = x) = \pi(x)$$

et

$$\mathbf{P}(Y = 0|X = x) = 1 - \pi(x),$$

avec $\pi(x) \in [0, 1]$.

On peut donc modéliser Y par

$$Y|X = x \sim \mathcal{B}(\pi(x))$$

(loi de Bernoulli)

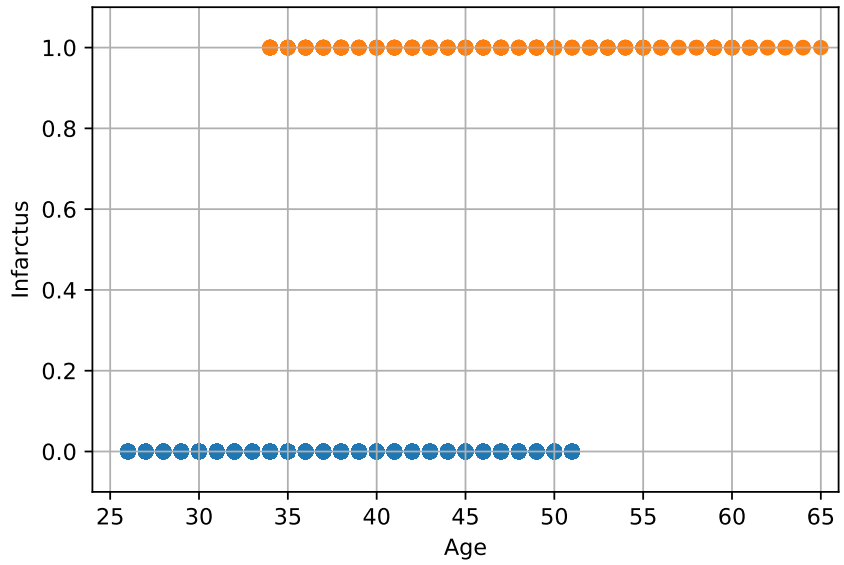
Propriétés (loi de Bernoulli)

$$\mathbf{P}(Y = y|X = x) = \pi(x)^y \big(1 - \pi(x)\big)^{1-y}$$

$$\mathbb{E}(Y|X = x) = \pi(x) \qquad Var(Y|X = x) = \pi(x)\big(1 - \pi(x)\big)$$

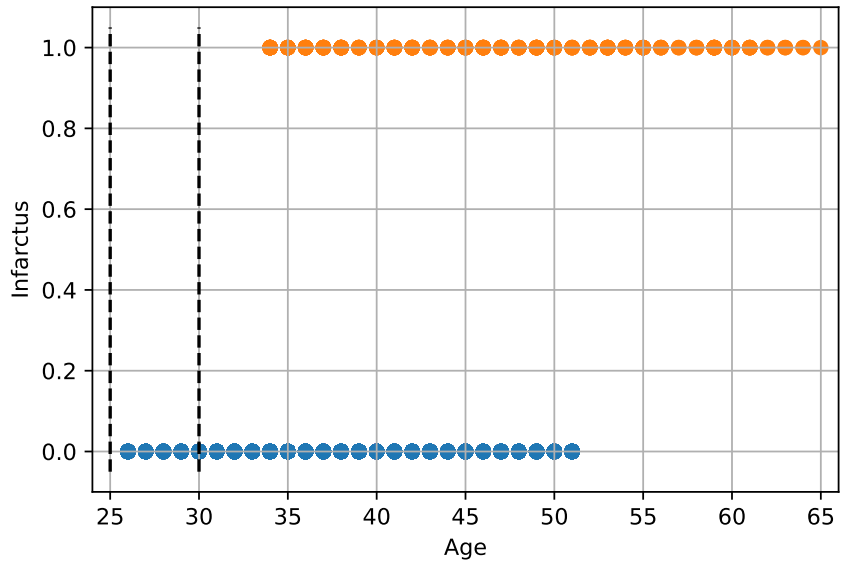
- Estimation de $\pi(x)$: $\hat{\pi}(x)$ = proportion de $Y = 1$ pour X mis en classes

classes	n_c	#0	#1	$\hat{\pi}(x)$



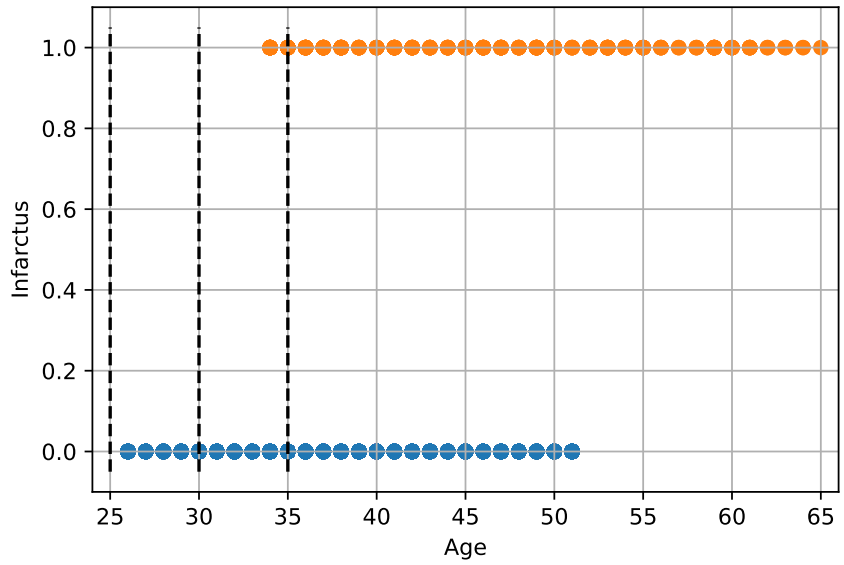
- Estimation de $\pi(x)$: $\hat{\pi}(x)$ = proportion de $Y = 1$ pour X mis en classes

classes	n_c	#0	#1	$\hat{\pi}(x)$
[25,30]	454	454	0	0.00



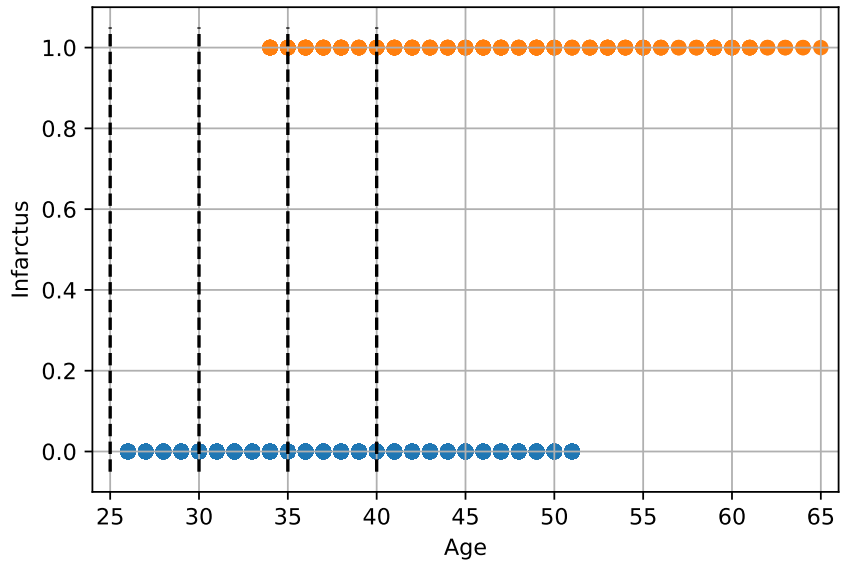
- Estimation de $\pi(x)$: $\hat{\pi}(x)$ = proportion de $Y = 1$ pour X mis en classes

classes	n_c	#0	#1	$\hat{\pi}(x)$
[25,30]	454	454	0	0.00
(30,35]	958	913	45	0.04



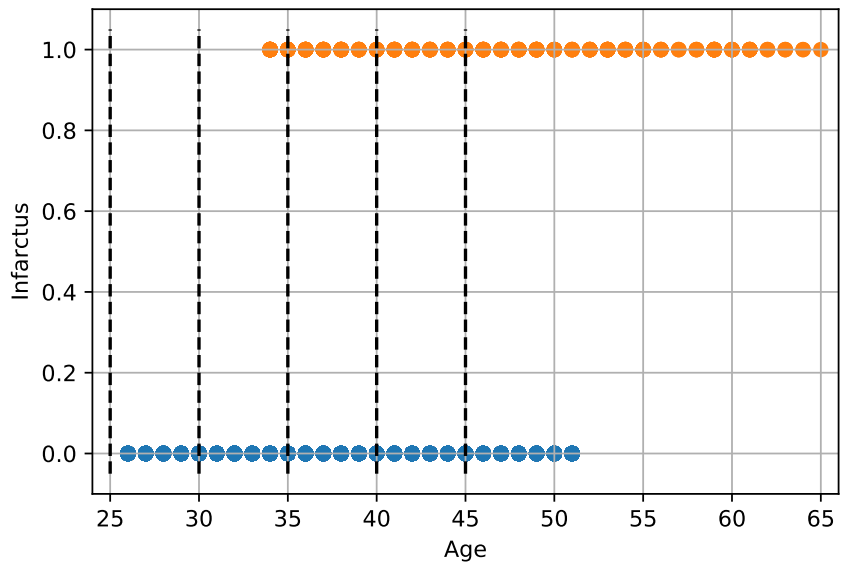
- Estimation de $\pi(x)$: $\hat{\pi}(x)$ = proportion de $Y = 1$ pour X mis en classes

classes	n_c	#0	#1	$\hat{\pi}(x)$
[25,30]	454	454	0	0.00
(30,35]	958	913	45	0.04
(35,40]	780	709	71	0.10



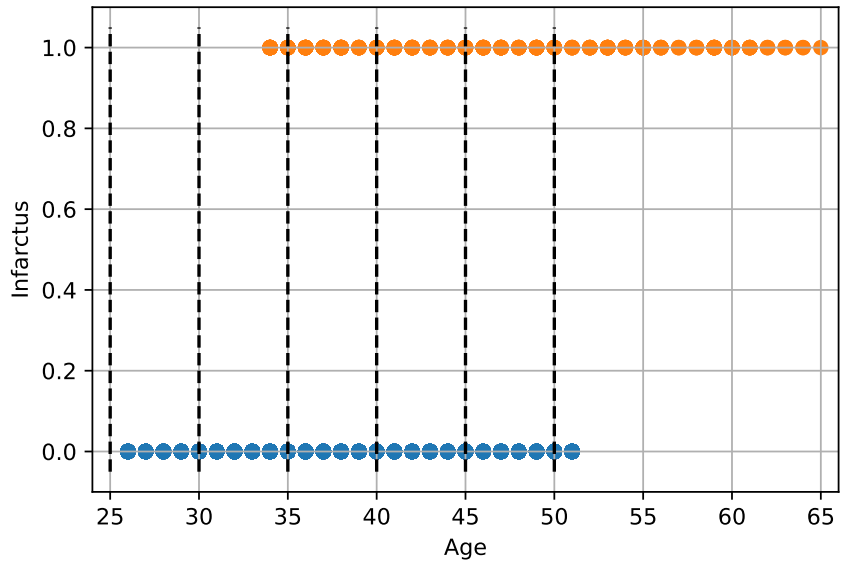
- Estimation de $\pi(x)$: $\hat{\pi}(x)$ = proportion de $Y = 1$ pour X mis en classes

classes	n_c	#0	#1	$\hat{\pi}(x)$
[25,30]	454	454	0	0.00
(30,35]	958	913	45	0.04
(35,40]	780	709	71	0.10
(40,45]	608	505	103	0.17



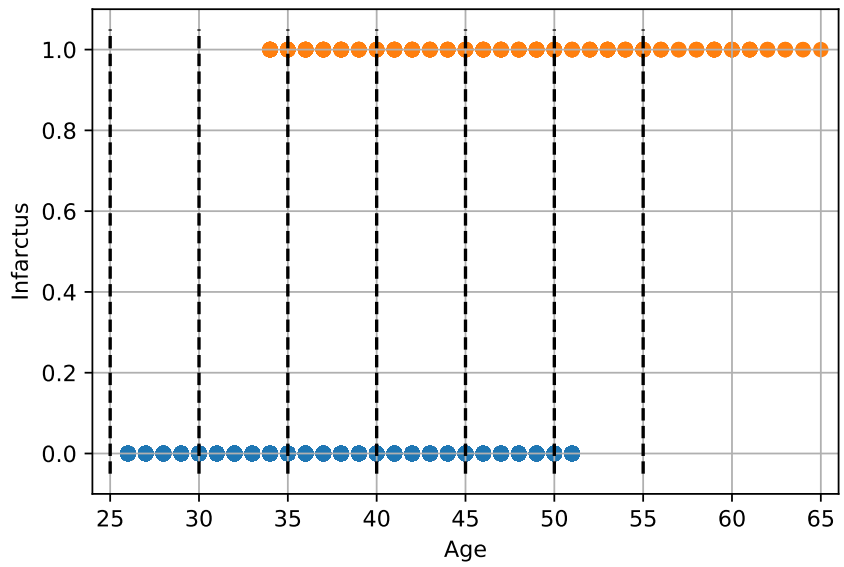
- Estimation de $\pi(x)$: $\hat{\pi}(x)$ = proportion de $Y = 1$ pour X mis en classes

classes	n_c	#0	#1	$\hat{\pi}(x)$
[25,30]	454	454	0	0.00
(30,35]	958	913	45	0.04
(35,40]	780	709	71	0.10
(40,45]	608	505	103	0.17
(45,50]	554	355	199	0.36



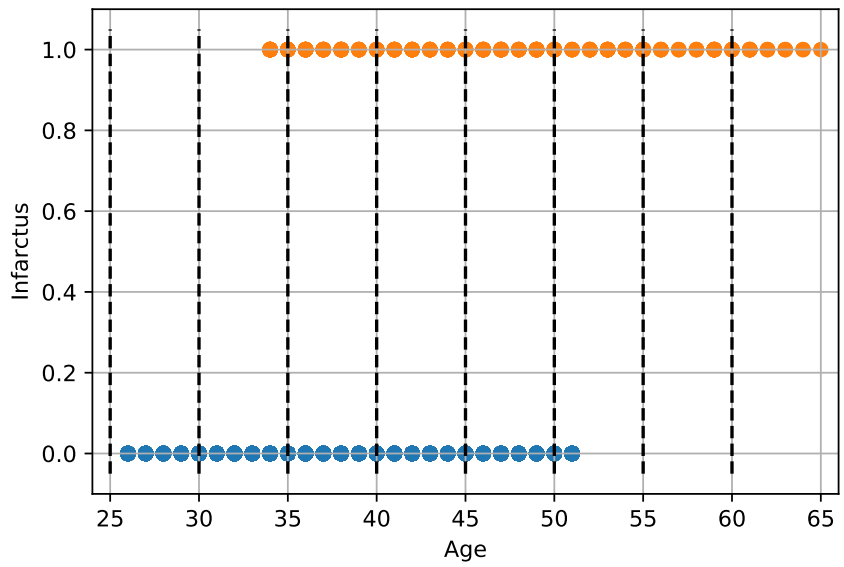
- Estimation de $\pi(x)$: $\hat{\pi}(x)$ = proportion de $Y = 1$ pour X mis en classes

classes	n_c	#0	#1	$\hat{\pi}(x)$
[25,30]	454	454	0	0.00
(30,35]	958	913	45	0.04
(35,40]	780	709	71	0.10
(40,45]	608	505	103	0.17
(45,50]	554	355	199	0.36
(50,55]	134	30	104	0.78



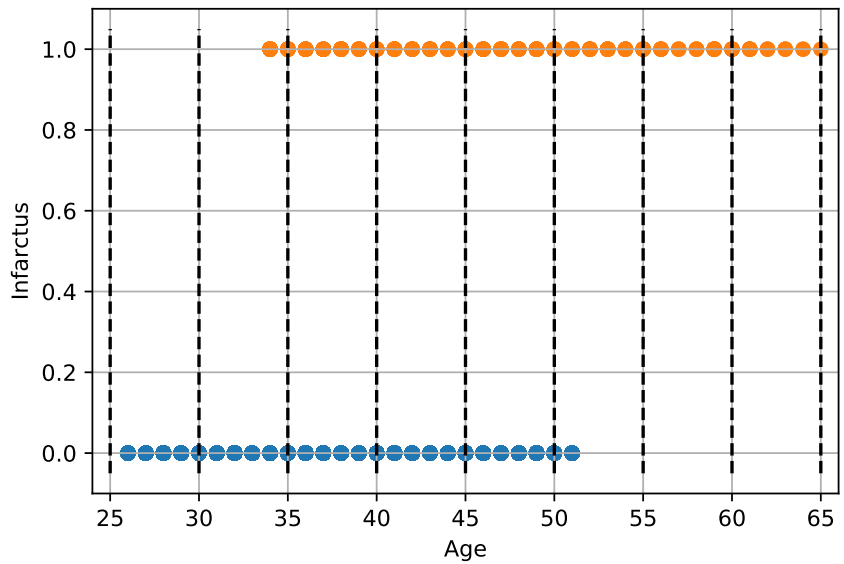
- Estimation de $\pi(x)$: $\hat{\pi}(x)$ = proportion de $Y = 1$ pour X mis en classes

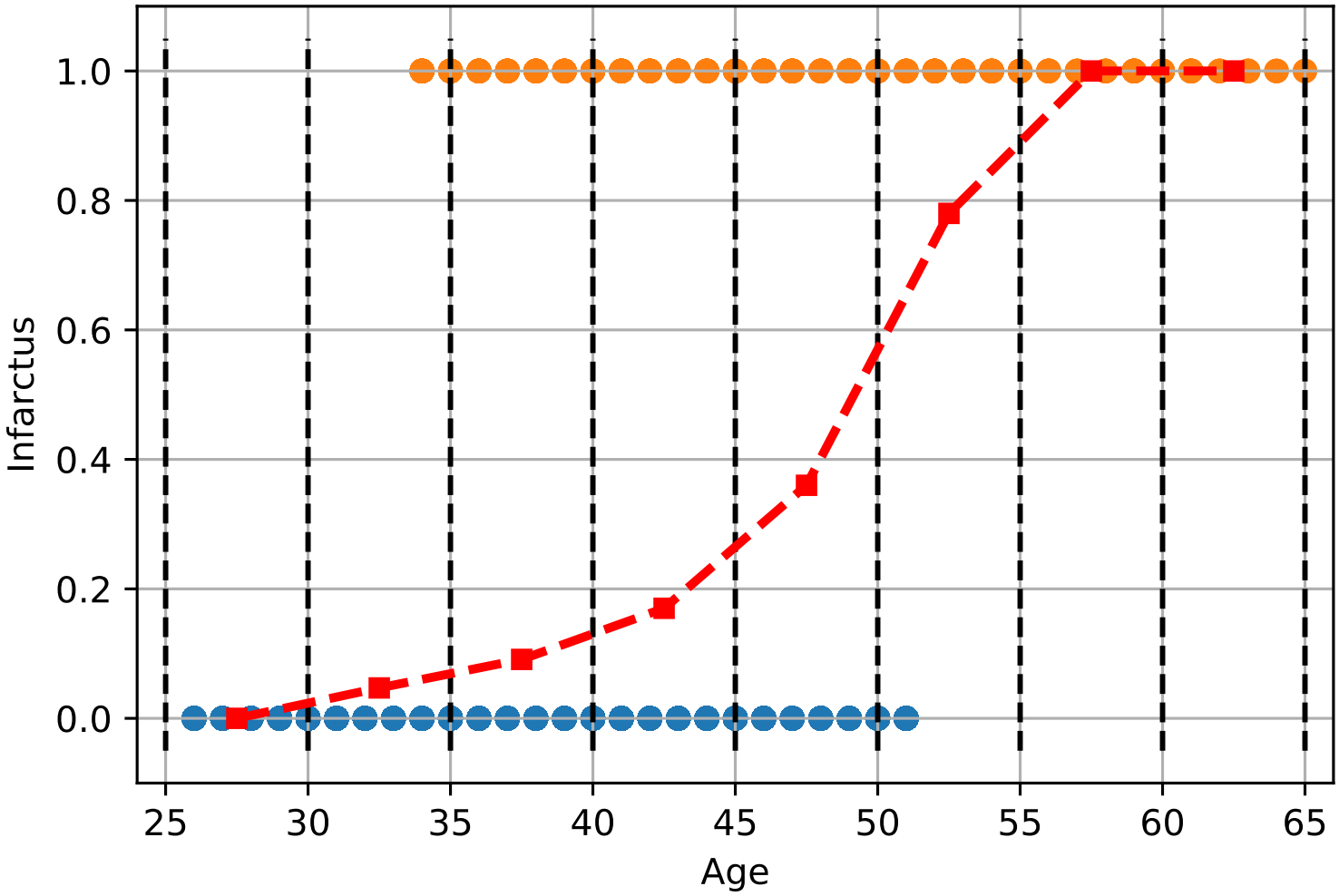
classes	n_c	#0	#1	$\hat{\pi}(x)$
[25,30]	454	454	0	0.00
(30,35]	958	913	45	0.04
(35,40]	780	709	71	0.10
(40,45]	608	505	103	0.17
(45,50]	554	355	199	0.36
(50,55]	134	30	104	0.78
(55,60]	33	0	33	1.00



- Estimation de $\pi(x)$: $\hat{\pi}(x)$ = proportion de $Y = 1$ pour X mis en classes

classes	n_c	#0	#1	$\hat{\pi}(x)$
[25,30]	454	454	0	0.00
(30,35]	958	913	45	0.04
(35,40]	780	709	71	0.10
(40,45]	608	505	103	0.17
(45,50]	554	355	199	0.36
(50,55]	134	30	104	0.78
(55,60]	33	0	33	1.00
(60,65]	17	0	17	1.00




Infarctus
1.0
0.8
0.6
0.4
0.2
0.0
25
30
35
40
45
50
55
60
65
Age

Fonction logistique

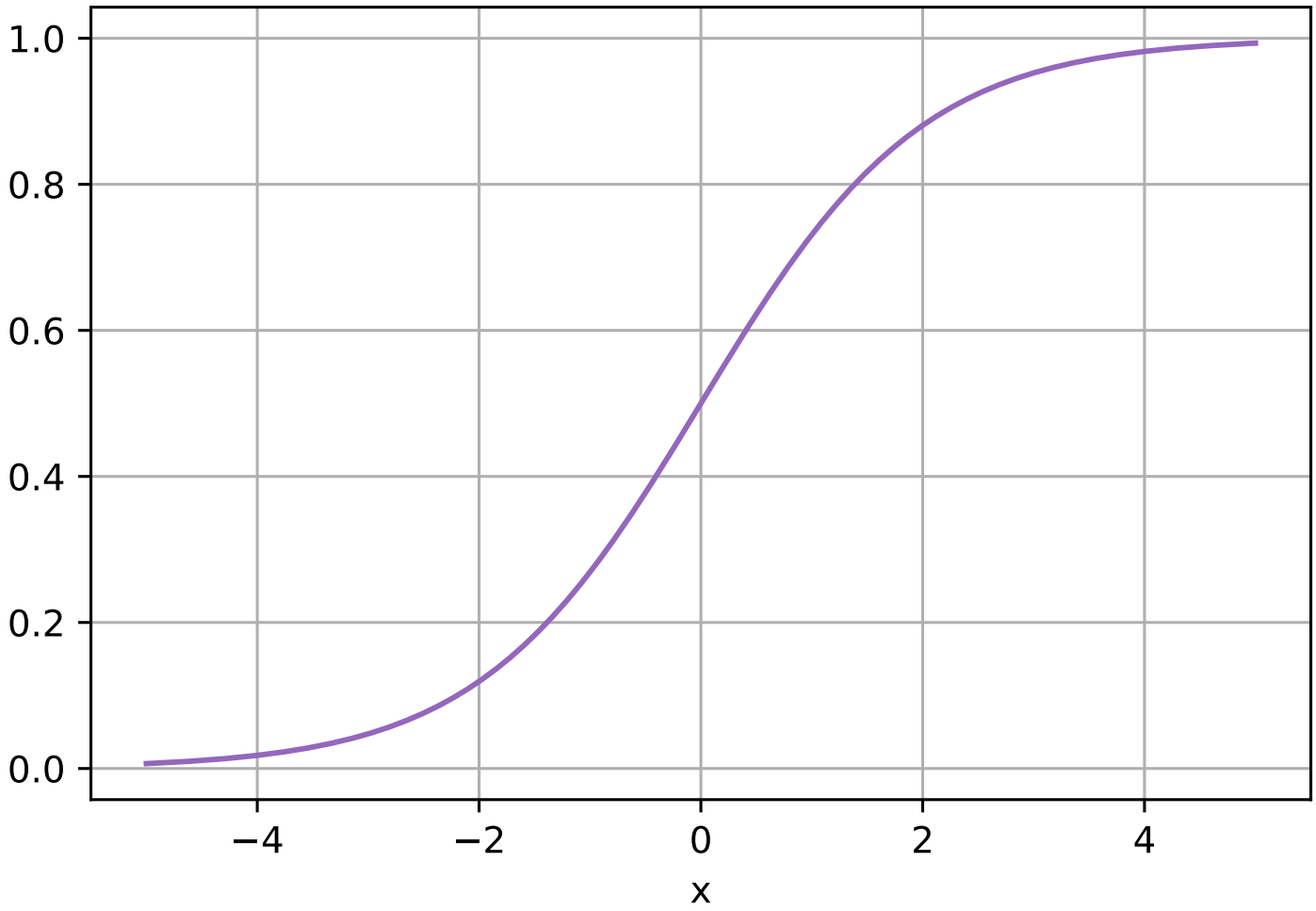
- $\pi(x)$ n'est pas linéaire en x !
- On a plutôt envie de modéliser $\pi(x)$ par une fonction "en $\mathcal{S}$" $\Rightarrow$ une possibilité est la fonction logistique

Fonction logistique

$$\begin{aligned} s \quad : \quad & \mathbb{R} \to]0,1[\\ & x \to s(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{1+e^x} \end{aligned}$$

Cas particulier de la fonction sigmoïde

Fonction logistique



Fonction logistique

Allure de la courbe de la fonction (plus générale)

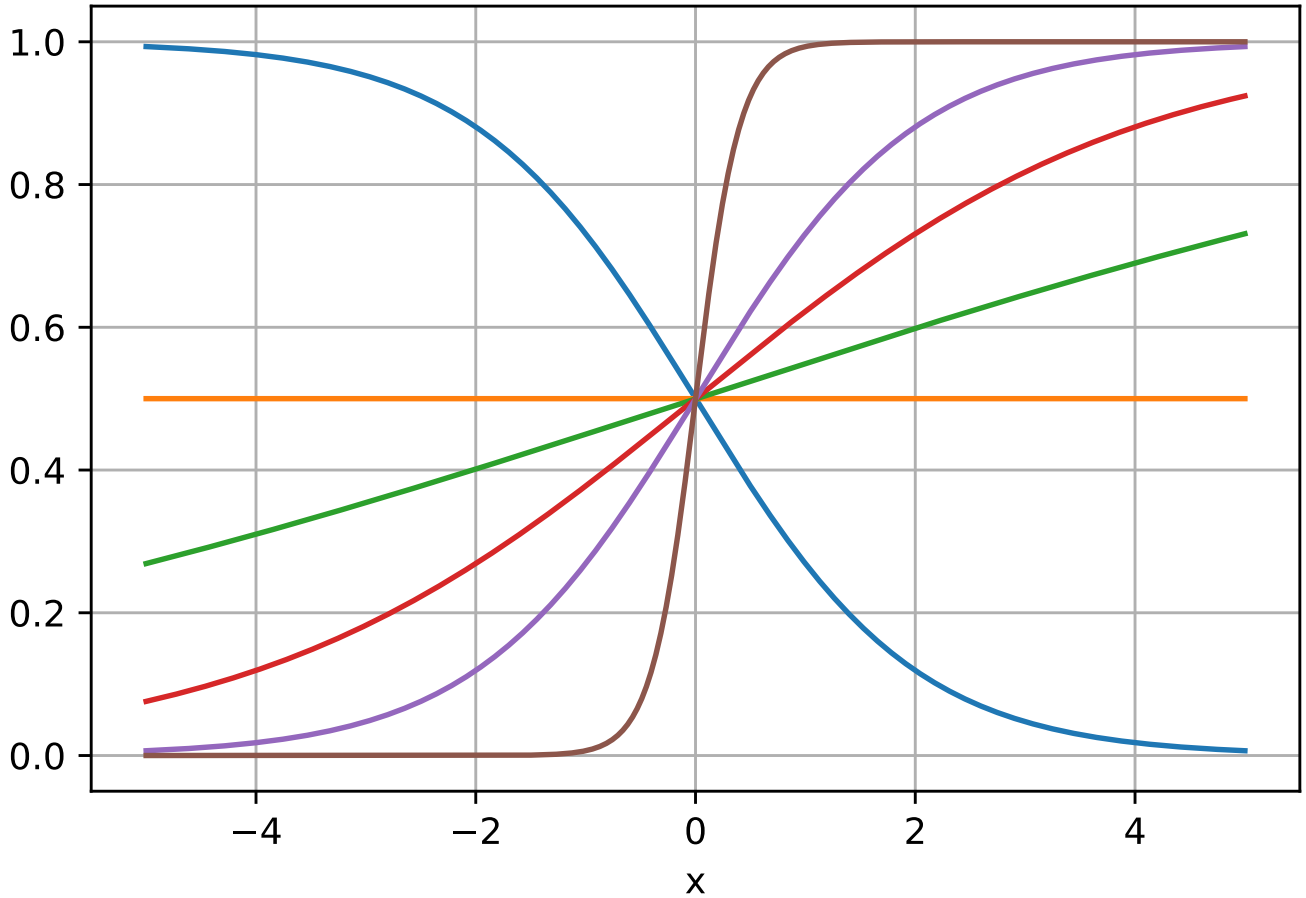
$$\begin{aligned} s \quad : \quad & \mathbb{R} \to]0,1[\\ & x \to s(x) = \frac{1}{1+e^{-x\beta}} = \frac{e^x}{1+e^{x\beta}} \end{aligned}$$

pour différentes valeurs de β

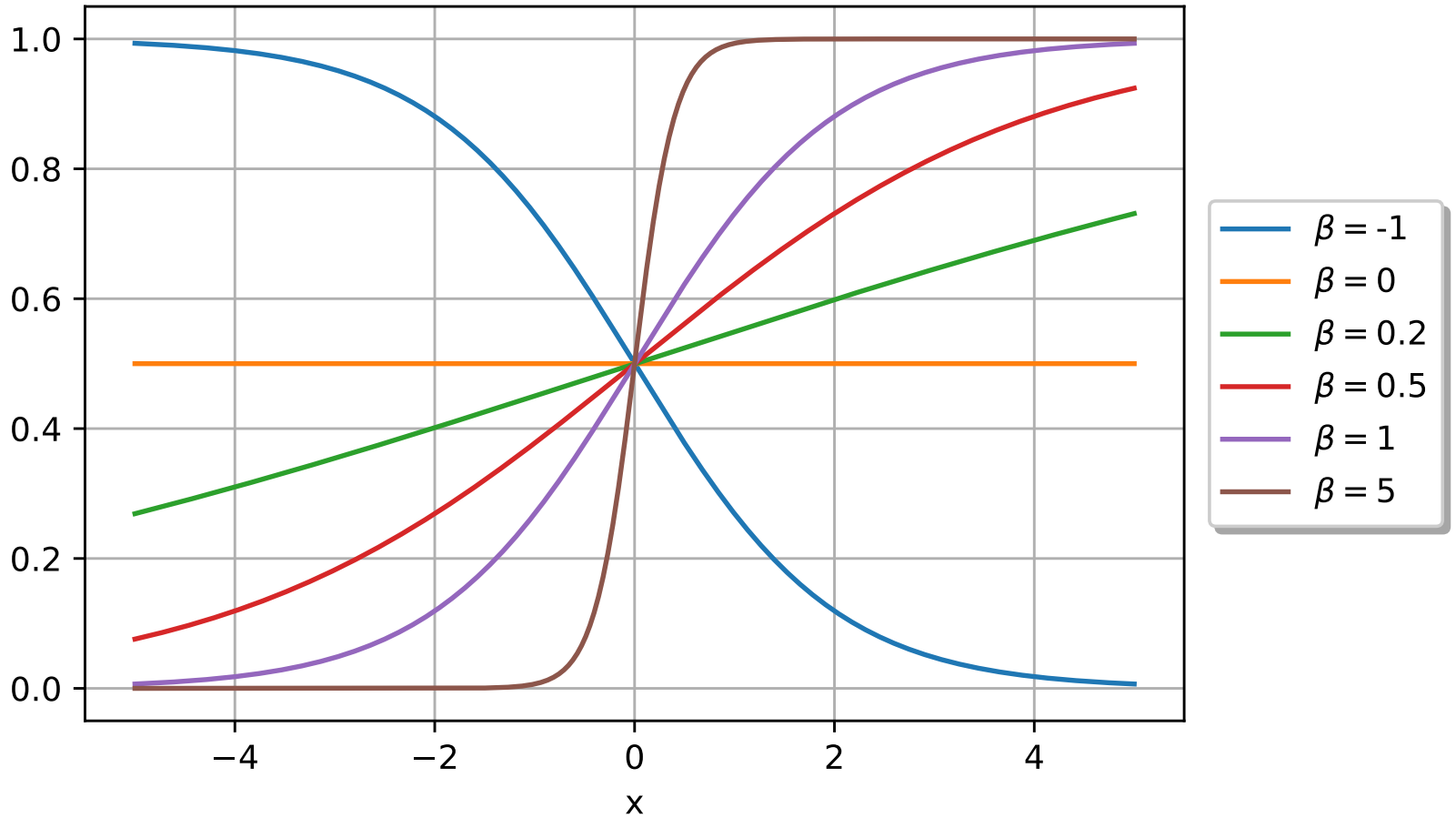
- $\beta = 0 \Rightarrow$ fonction consante
- β "petit" $\Rightarrow$ large plage de valeurs de x pour lesquelles la fonction se situe aux alentours de 0.5 $\Rightarrow$ discrimination difficile
- β "grand" $\Rightarrow$ petite plage de valeurs de x pour lesquelles la fonction se situe aux alentours de 0.5 $\Rightarrow$ discrimination plus facile

⚠ Dépend de l'échelle de x !

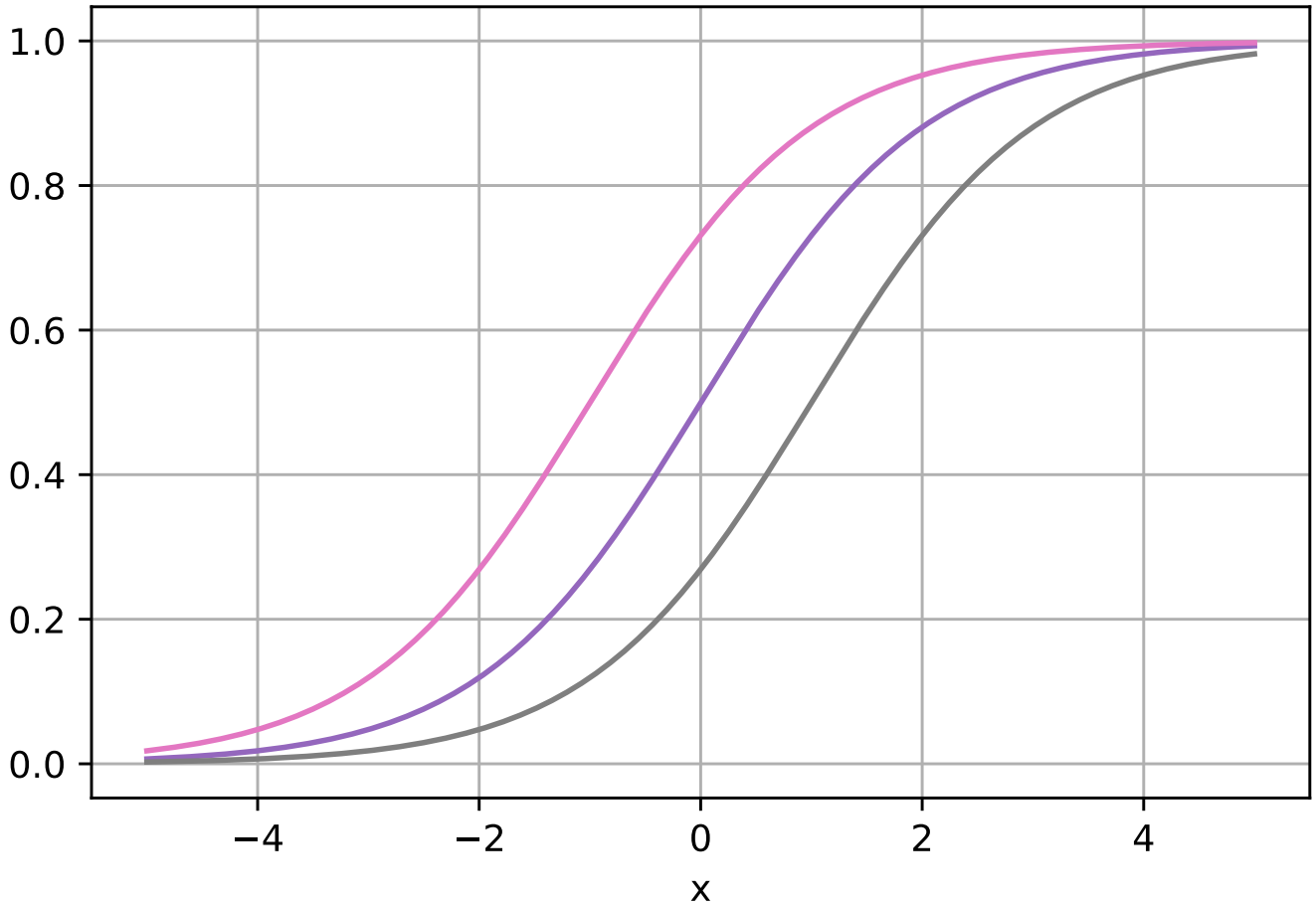
Fonction logistique



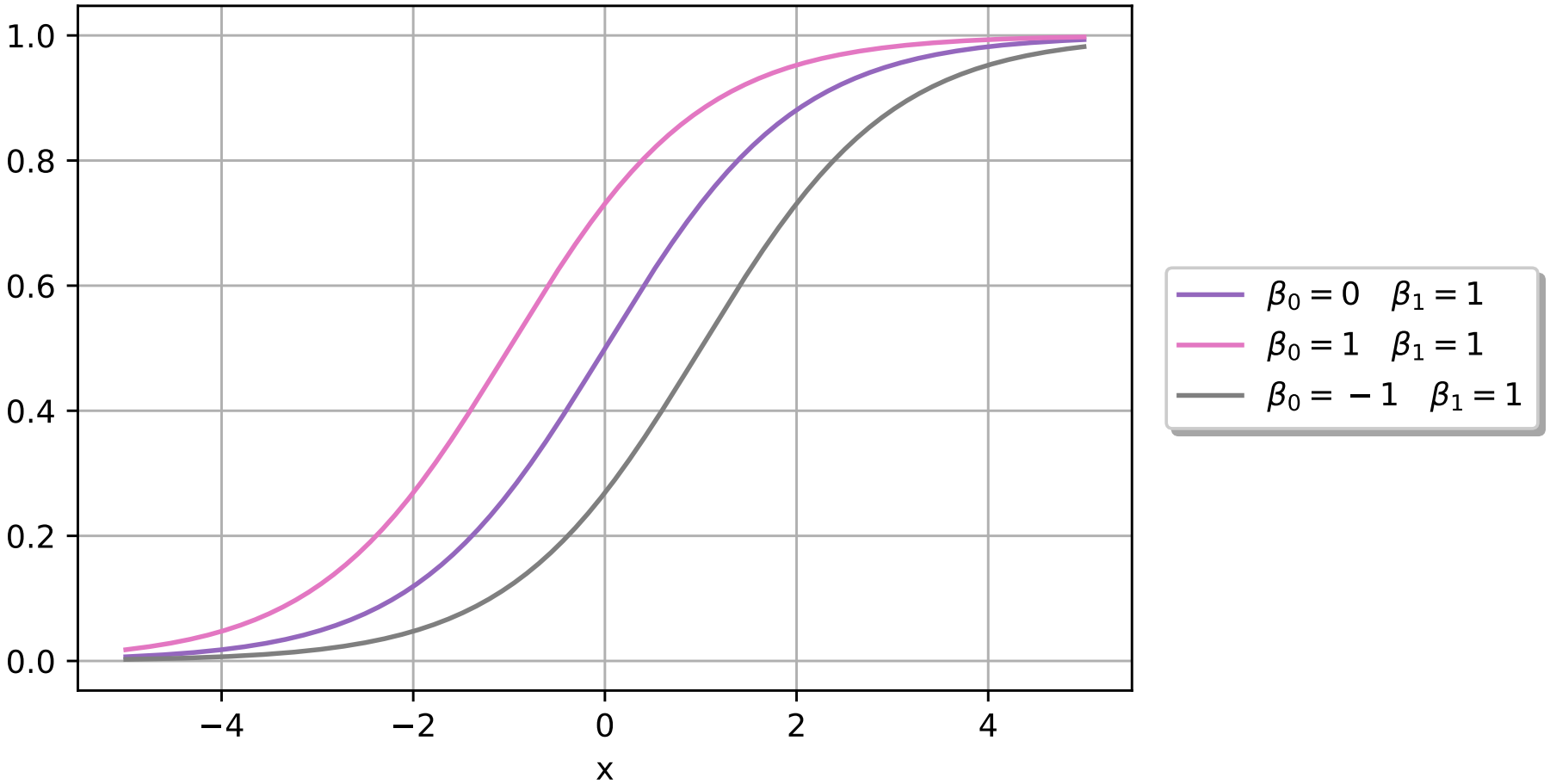
Fonction logistique



Fonction logistique



Fonction logistique



Modèle logistique

$$\pi(\mathbf{x}) = \frac{e^{\tilde{\mathbf{x}}\boldsymbol{\beta}}}{1 + e^{\tilde{\mathbf{x}}\boldsymbol{\beta}}} = \frac{e^{\beta_0 + \beta_1 x^1 + \ldots + \beta_d x^d}}{1 + e^{\beta_0 + \beta_1 x^1 + \ldots + \beta_d x^d}}$$

$$\Updownarrow$$

$$logit\Big(\pi(\mathbf{x})\Big) = log\Big(\frac{\pi(\mathbf{x})}{1 - \pi(\mathbf{x})}\Big) = \tilde{\mathbf{x}}\boldsymbol{\beta} = \beta_0 + \beta_1 x^1 + \ldots + \beta_d x^d$$

- Transformation g qui permet d'avoir un lien linéaire entre $g\Big(\pi(\mathbf{x})\Big)$ et $\mathbf{x}$: **fonction de lien**

Fonction logit

$$\begin{array}{rcl} logit & : &]0,1[\to \mathbb{R} \\ & & p \quad \to logit(p) = log\Big(\frac{p}{1-p}\Big) \end{array}$$

Modèle logistique

- On pose donc :

$$f_{\boldsymbol{\beta}}(\mathbf{x}_i) = \frac{e^{\mathbf{x}_i' \boldsymbol{\beta}}}{1 + e^{\mathbf{x}_i' \boldsymbol{\beta}}} = \mathbb{P}(Y = 1 | X = \mathbf{x}_i)$$

- On cherche $\boldsymbol{\beta}$ tel que $f_{\boldsymbol{\beta}}(\mathbf{x}_i)$ est proche de y_i pour toutes les données d'apprentissage $\{\mathbf{x}_i, y_i\}_{i=1}^m$ avec $\mathbf{x}_i \in \mathbb{R}^d$

Modèle logistique

- Vraisemblance du modèle :

$$L(\boldsymbol{\beta}, y) \quad = \quad \prod_{i=1}^{m} \mathbf{P}(Y = y_i | X = \mathbf{x}_i) = \prod_{i=1}^{m} f_{\boldsymbol{\beta}}(\mathbf{x}_i)^{y_i} \times (1 - f_{\boldsymbol{\beta}}(\mathbf{x}_i))^{1-y_i}$$

Modèle logistique

- Vraisemblance du modèle :

$$L(\boldsymbol{\beta}, y) \quad = \quad \prod_{i=1}^{m} \mathbf{P}(Y = y_i | X = \mathbf{x}_i) = \prod_{i=1}^{m} f_{\boldsymbol{\beta}}(\mathbf{x}_i)^{y_i} \times (1 - f_{\boldsymbol{\beta}}(\mathbf{x}_i))^{1-y_i}$$

- Log-vraisemblance

$$\mathcal{L}(\boldsymbol{\beta}, y) \quad = \quad \sum_{i=1}^{m} \Big[y_i log\Big(f_{\boldsymbol{\beta}}(\mathbf{x}_i)\Big) + (1 - y_i) log\Big(1 - f_{\boldsymbol{\beta}}(\mathbf{x}_i)\Big) \Big]$$

Modèle logistique

- Vraisemblance du modèle :

$$L(\boldsymbol{\beta}, y) \quad = \quad \prod_{i=1}^{m} \mathbf{P}(Y = y_i | X = \mathbf{x}_i) = \prod_{i=1}^{m} f_{\boldsymbol{\beta}}(\mathbf{x}_i)^{y_i} \times (1 - f_{\boldsymbol{\beta}}(\mathbf{x}_i))^{1-y_i}$$

- Log-vraisemblance

$$\mathcal{L}(\boldsymbol{\beta}, y) \quad = \quad \sum_{i=1}^{m} \Big[y_i log\Big(f_{\boldsymbol{\beta}}(\mathbf{x}_i)\Big) + (1 - y_i) log\Big(1 - f_{\boldsymbol{\beta}}(\mathbf{x}_i)\Big) \Big]$$

- Maximisation de la (log-)vraisemblance
  $\Rightarrow$ annuler les dérivées de $L(\boldsymbol{\beta}, y)$
  - **pas de solution analytique explicite**
  - besoin d'algorithmes d'optimisation itératifs

Fonction-coût

- **Objectif** : trouver le meilleur $\boldsymbol{\beta}$ pour minimiser le coût (quadratique) **global** des erreurs :

$$\underset{\beta}{\operatorname{argmin}}\Big(\mathbf{J}(\beta)\Big)$$

avec

$$\mathbf{J}(\boldsymbol{\beta}) = -\frac{1}{m}\sum_{i=1}^{m}\Big[y_i log\Big(f_{\boldsymbol{\beta}}(\mathbf{x}_i)\Big) + (1-y_i) log\Big(1-f_{\boldsymbol{\beta}}(\mathbf{x}_i)\Big)\Big]$$

Plan

Principe

- Objectif : trouver le minimum d'une fonction-coût
- Principe : algorithme itératif
  1. initialisation : $\beta^{(0)}$
  2. à chaque étape k, modifier $\boldsymbol{\beta}^{(k-1)}$ pour faire diminuer $\mathbf{J}(\boldsymbol{\beta}^{(k)})$
  3. arrêt lorsque le minimum est atteint

Itération k de l'algorithme de descente du gradient

Pour le paramètre β_j

$$\beta_j^{(k)} \quad := \quad \beta_j^{(k-1)} - \alpha \frac{\partial}{\partial \beta_j} \mathbf{J}(\boldsymbol{\beta}^{(k-1)})$$

avec :

- $\frac{\partial}{\partial \beta_j}$:
- α :

Principe

- Objectif : trouver le minimum d'une fonction-coût
- Principe : algorithme itératif
  1. initialisation : $\beta^{(0)}$
  2. à chaque étape k, modifier $\boldsymbol{\beta}^{(k-1)}$ pour faire diminuer $\mathbf{J}(\boldsymbol{\beta}^{(k)})$
  3. arrêt lorsque le minimum est atteint

Itération k de l'algorithme de descente du gradient

Pour le paramètre β_j

$$\beta_j^{(k)} \quad := \quad \beta_j^{(k-1)} - \alpha \frac{\partial}{\partial \beta_j} \mathbf{J}(\boldsymbol{\beta}^{(k-1)})$$

avec :

- $\frac{\partial}{\partial \beta_j}$: dérivée partielle
- α : pas d'apprentissage (à fixer par l'utilisateur)

Pour la régression logistique

- Itération k de l'algorithme de descente du gradient ?

Itération k de l'algorithme de descente du gradient - régression logistique

$$\beta_j^{(k)} \quad := \quad \beta_j^{(k-1)} - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(f_{\boldsymbol{\beta}^{(k-1)}}(\mathbf{x}_i) - y_i \right) x_i^j$$

Remarque : $\forall i, x_i^0 = 1$

Pour la régression logistique

- Itération k de l'algorithme de descente du gradient ?
- Même formule que pour la régression linéaire...

Itération k de l'algorithme de descente du gradient - régression logistique

$$\beta_j^{(k)} \quad := \quad \beta_j^{(k-1)} - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(f_{\boldsymbol{\beta}^{(k-1)}}(\mathbf{x}_i) - y_i \right) x_i^j$$

Remarque : $\forall i, x_i^0 = 1$

Il existe des versions plus avancées de l'algorithme de descente du gradient :

- gradient conjugué, BFGS [1], L-BFGS (à mémoire limitée)
  - pas besoin de fixer α (fait automatiquement dans l'algorithme)
  - plus rapide (moins d'itérations) que l'algorithme de descente de gradient
  - mais plus complexe à implémenter (ces variantes sont disponibles dans les principaux logiciels/langages)

1. Broyden-Fletcher-Goldfarb-Shanno

Plan

Application

Les données portent sur une étude clinique sur le cancer du sein menée à l'Université du Winconsin. Il s'agit de prévoir le statut de la tumeur (maligne-1 ou bénine-0) à partir de caractéristiques de cellules prélevées chez les patientes

- radius (mean of distances from center to points on the perimeter)
- texture (standard deviation of gray-scale values)
- perimeter
- area
- . . .

Objectif : étudier le lien entre la probabilité d'avoir une tumeur maligne en fonction de certaines caractéristiques

The data was originally published by W.N. Street, W.H. Wolberg and O.L. Mangasarian. Nuclear feature extraction for breast tumor diagnosis. IS&T/SPIE 1993 International Symposium on Electronic Imaging : Science and Technology, volume 1905, pages 861-870, San Jose, CA, 1993.

TP / CR à rendre

- Importer les données
- Construire les fonctions nécessaires : f, coût, gradient puis implémenter l'algorithme de descente de gradient
  - Cas de la régression logistique simple
  - Etendre à la régression logistique multiple
  - Retourner les coefficients du modèle et les valeurs de la fonction-coût pour toutes les itérations
- **Faire varier les paramètres de l'algorithme (initialisation, pas) et commenter.**
- **Comparer avec les solutions d'optimisation natives de Python (pas de solution exacte ici)**