KNN et Régression logistique

Jérémie Cabessa

Laboratoire DAVID, UVSQ

Classification

- Dans le cadre de l'**apprentissage supervisé**, on distingue deux types de méthodes:
- **Méthodes de régression**

  La variable d'output (réponse) est **quantitative**.
- **Méthodes de classification**

  La variable d'output (réponse) est **qualitative**.

Classification

- Dans le cadre de l'**apprentissage supervisé**, on distingue deux types de méthodes:
- **Méthodes de régression**

  La variable d'output (réponse) est **quantitative**.
- **Méthodes de classification**

  La variable d'output (réponse) est **qualitative**.

Classification

- Dans le cadre de l'**apprentissage supervisé**, on distingue deux types de méthodes:
- **Méthodes de régression**

  La variable d'output (réponse) est **quantitative**.
- **Méthodes de classification**

  La variable d'output (réponse) est **qualitative**.

K-Nearest Neighbors

- Soient $\boldsymbol{X} = (X_1, \ldots, X_p)$ des variables explicatives et Y une variable réponse qualitative (C classes différentes).
- On codes les réalisations de Y par $1, 2, \ldots, C$.
- Pour tout $\boldsymbol{x}$, on aimerait estimer les probabilités que $Y = c$, pour $c = 1, \ldots, C$, étant donnée la réalisation $\boldsymbol{X} = \boldsymbol{x}$:

$$\hat{\Pr}(Y = c \mid \boldsymbol{X} = \boldsymbol{x}) \text{ , pour } c = 1, \ldots, C.$$

- Ensuite, pour tout $\boldsymbol{x}$, on prédit la classe $\hat{c}$ associée à la probabilité maximale

$$\hat{c} = \underset{c \in \{1,\ldots,C\}}{\arg\max} \hat{\Pr}(Y = c \mid \boldsymbol{X} = \boldsymbol{x}).$$

K-Nearest Neighbors

- Soient $\boldsymbol{X} = (X_1, \ldots, X_p)$ des variables explicatives et Y une variable réponse qualitative (C classes différentes).
- On codes les réalisations de Y par $1, 2, \ldots, C$.
- Pour tout $\boldsymbol{x}$, on aimerait estimer les probabilités que $Y = c$, pour $c = 1, \ldots, C$, étant donnée la réalisation $\boldsymbol{X} = \boldsymbol{x}$:

$$\hat{\Pr}(Y = c \mid \boldsymbol{X} = \boldsymbol{x}) \text{ , pour } c = 1, \ldots, C.$$

- Ensuite, pour tout $\boldsymbol{x}$, on prédit la classe $\hat{c}$ associée à la probabilité maximale

$$\hat{c} = \underset{c \in \{1,\ldots,C\}}{\arg\max} \hat{\Pr}(Y = c \mid \boldsymbol{X} = \boldsymbol{x}).$$

K-Nearest Neighbors

- Soient $\boldsymbol{X} = (X_1, \ldots, X_p)$ des variables explicatives et Y une variable réponse qualitative (C classes différentes).
- On codes les réalisations de Y par $1, 2, \ldots, C$.
- Pour tout $\boldsymbol{x}$, on aimerait estimer les probabilités que $Y = c$, pour $c = 1, \ldots, C$, étant donnée la réalisation $\boldsymbol{X} = \boldsymbol{x}$:

$$\hat{\Pr}(Y = c \mid \boldsymbol{X} = \boldsymbol{x}) \text{ , pour } c = 1, \ldots, C.$$

- Ensuite, pour tout $\boldsymbol{x}$, on prédit la classe $\hat{c}$ associée à la probabilité maximale

$$\hat{c} = \underset{c \in \{1,\ldots,C\}}{\arg\max} \hat{\Pr}(Y = c \mid \boldsymbol{X} = \boldsymbol{x}).$$

K-Nearest Neighbors

- Soient $\boldsymbol{X} = (X_1, \ldots, X_p)$ des variables explicatives et Y une variable réponse qualitative (C classes différentes).
- On codes les réalisations de Y par $1, 2, \ldots, C$.
- Pour tout $\boldsymbol{x}$, on aimerait estimer les probabilités que $Y = c$, pour $c = 1, \ldots, C$, étant donnée la réalisation $\boldsymbol{X} = \boldsymbol{x}$:

$$\hat{\Pr}(Y = c \mid \boldsymbol{X} = \boldsymbol{x}) \text{ , pour } c = 1, \ldots, C.$$

- Ensuite, pour tout $\boldsymbol{x}$, on prédit la classe $\hat{c}$ associée à la probabilité maximale

$$\hat{c} = \underset{c \in \{1,\ldots,C\}}{\arg\max} \hat{\Pr}(Y = c \mid \boldsymbol{X} = \boldsymbol{x}).$$

K-Nearest Neighbors

- Soit un training set $S_{\text{train}} = \{(\boldsymbol{x_1}, y_1), \ldots, (\boldsymbol{x_N}, y_N)\}$.
- K-Nearest Neighbors (KNN): On prend les K plus proches voisins de $\boldsymbol{x}$ dans S_{train}, on compte combien de ces voisins font partie de la classe c, disons n_c, et on a

$$\hat{\Pr}(Y = c \mid \boldsymbol{X} = \boldsymbol{x}) = \frac{1}{K} \sum_{\boldsymbol{x_i} \in \mathcal{N}_{\boldsymbol{x}}^K} \mathrm{I}(y_i = c) = \frac{n_c}{K}$$

pour tout $c = 1, \ldots, C$, où $\mathcal{N}_{\boldsymbol{x}}^K$ désigne les K plus proches voisins de $\boldsymbol{x}$ et $\mathrm{I}(.)$ la fonction indicatrice.

- Pour tout $\boldsymbol{x}$, on prédit la classe $\hat{c}$ donnée par

$$\hat{c} = \underset{c \in \{1,\ldots,C\}}{\arg\max} \hat{\Pr}(Y = c \mid \boldsymbol{X} = \boldsymbol{x}).$$

K-Nearest Neighbors

- Soit un training set $S_{\text{train}} = \{(\boldsymbol{x_1}, y_1), \ldots, (\boldsymbol{x_N}, y_N)\}$.
- **K-Nearest Neighbors (KNN):** On prend les K plus proches voisins de $\boldsymbol{x}$ dans S_{train}, on compte combien de ces voisins font partie de la classe c, disons n_c, et on a

$$\hat{\Pr}(Y = c \mid \boldsymbol{X} = \boldsymbol{x}) = \frac{1}{K} \sum_{\boldsymbol{x_i} \in \mathcal{N}_{\boldsymbol{x}}^K} \mathrm{I}(y_i = c) = \frac{n_c}{K}$$

pour tout $c = 1, \ldots, C$, où $\mathcal{N}_{\boldsymbol{x}}^K$ désigne les K plus proches voisins de $\boldsymbol{x}$ et $\mathrm{I}(.)$ la fonction indicatrice.

- Pour tout $\boldsymbol{x}$, on prédit la classe $\hat{c}$ donnée par

$$\hat{c} = \underset{c \in \{1,\ldots,C\}}{\arg\max} \hat{\Pr}(Y = c \mid \boldsymbol{X} = \boldsymbol{x}).$$

K-Nearest Neighbors

- Soit un training set $S_{\text{train}} = \{(\boldsymbol{x_1}, y_1), \ldots, (\boldsymbol{x_N}, y_N)\}$.
- **K-Nearest Neighbors (KNN):** On prend les K plus proches voisins de $\boldsymbol{x}$ dans S_{train}, on compte combien de ces voisins font partie de la classe c, disons n_c, et on a

$$\hat{\Pr}(Y = c \mid \boldsymbol{X} = \boldsymbol{x}) = \frac{1}{K} \sum_{\boldsymbol{x_i} \in \mathcal{N}_{\boldsymbol{x}}^K} \mathrm{I}(y_i = c) = \frac{n_c}{K}$$

pour tout $c = 1, \ldots, C$, où $\mathcal{N}_{\boldsymbol{x}}^K$ désigne les K plus proches voisins de $\boldsymbol{x}$ et $\mathrm{I}(.)$ la fonction indicatrice.

- Pour tout $\boldsymbol{x}$, on prédit la classe $\hat{c}$ donnée par

$$\hat{c} = \underset{c \in \{1,\ldots,C\}}{\arg\max} \hat{\Pr}(Y = c \mid \boldsymbol{X} = \boldsymbol{x}).$$

K-Nearest Neighbors

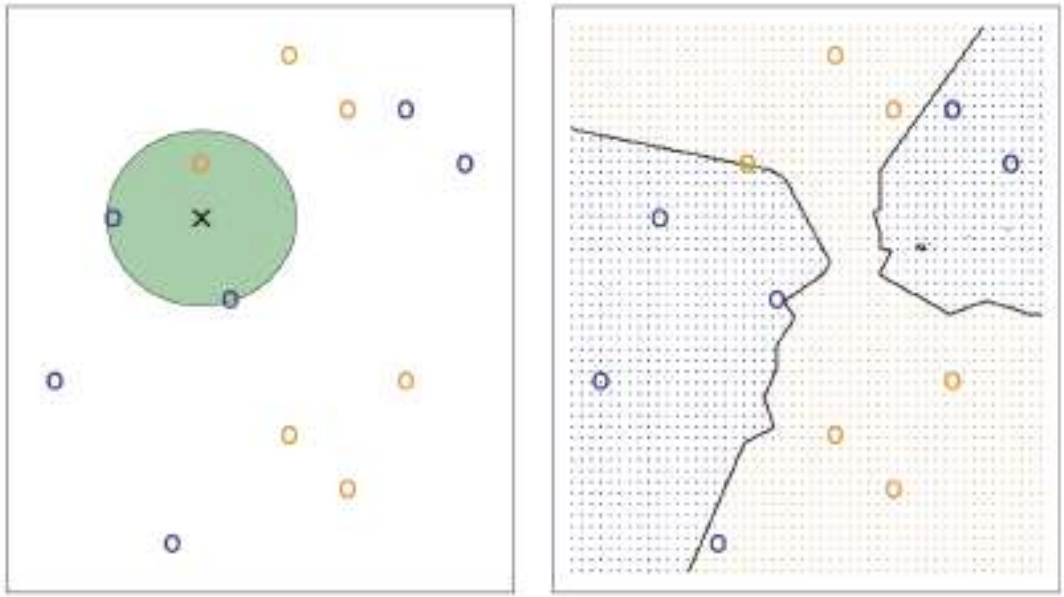


FIGURE 2.14. *The KNN approach, using $K = 3$, is illustrated in a simple situation with six blue observations and six orange observations.* Left: *a test observation at which a predicted class label is desired is shown as a black cross. The three closest points to the test observation are identified, and it is predicted that the test observation belongs to the most commonly-occurring class, in this case blue.* Right: *The KNN decision boundary for this example is shown in black. The blue grid indicates the region in which a test observation will be assigned to the blue class, and the orange grid indicates the region in which it will be assigned to the orange class.*

Figures taken from [James et al., 2013]

K-Nearest Neighbors

- Plus K est petit, plus la frontière de décision est non-linaire:
  $\rightarrow$ petit biais mais grande variance (bias-variance trade-off)
- Plus K est grand, plus la frontière de décision est linaire:
  $\rightarrow$ petite variance mais grand biais (bias-variance trade-off)

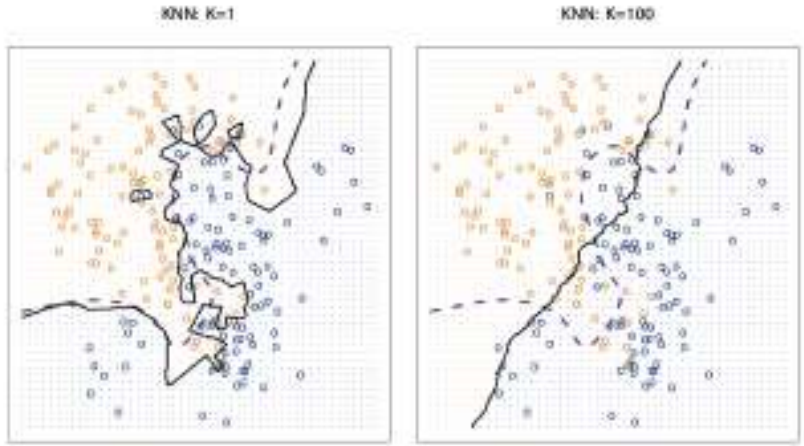


FIGURE 2.16. *A comparison of the KNN decision boundaries (solid black curves) obtained using $K = 1$ and $K = 100$ on the data from Figure 2.13. With $K = 1$, the decision boundary is overly flexible, while with $K = 100$ it is not sufficiently flexible. The Bayes decision boundary is shown as a purple dashed line.*

Figures taken from [James et al., 2013]

K-Nearest Neighbors

- Plus K est petit, plus la frontière de décision est non-linaire:
  $\rightarrow$ petit biais mais grande variance (bias-variance trade-off)
- Plus K est grand, plus la frontière de décision est linaire:
  $\rightarrow$ petite variance mais grand biais (bias-variance trade-off)

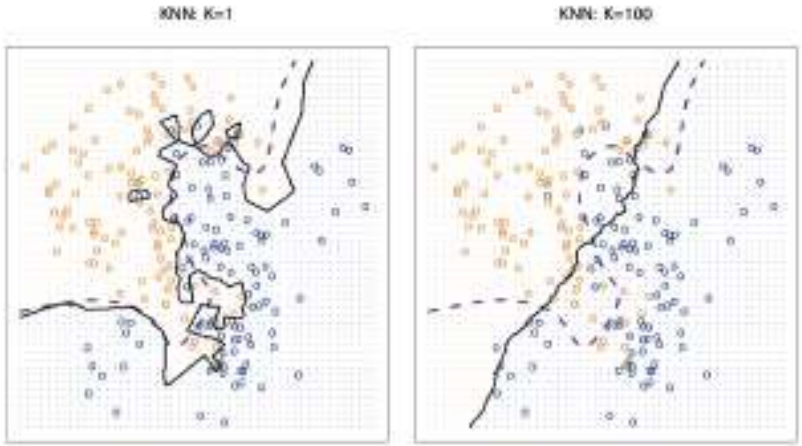


FIGURE 2.16. *A comparison of the KNN decision boundaries (solid black curves) obtained using $K = 1$ and $K = 100$ on the data from Figure 2.13. With $K = 1$, the decision boundary is overly flexible, while with $K = 100$ it is not sufficiently flexible. The Bayes decision boundary is shown as a purple dashed line.*

Figures taken from [James et al., 2013]

K-Nearest Neighbors

- Plus K est petit, plus la frontière de décision est non-linaire:
  $\rightarrow$ petit biais mais grande variance (bias-variance trade-off)
- Plus K est grand, plus la frontière de décision est linaire:
  $\rightarrow$ petite variance mais grand biais (bias-variance trade-off)

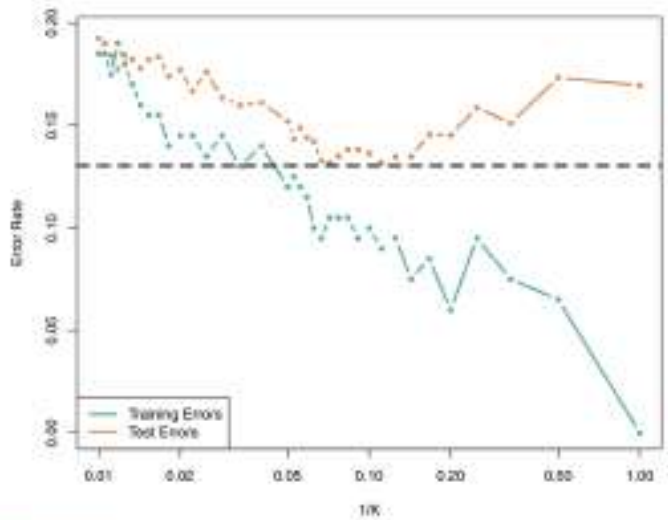


FIGURE 2.17. *The KNN training error rate (blue, 200 observations) and test error rate (orange, 5,000 observations) on the data from Figure 2.13, as the level of flexibility (assessed using $1/K$) increases, or equivalently as the number of neighbors K decreases. The black dashed line indicates the Bayes error rate. The jumpiness of the curves is due to the small size of the training data set.*

Figures taken from [James et al., 2013]

Régression logistique

- Soient $\boldsymbol{X} = (X_1, \ldots, X_p)$ des variables explicatives et Y une variable réponse qualitative *binaire*.
- On code les réalisations possibles de Y par 0 ou 1.
- Pour tout $\boldsymbol{x}$, on aimerait modéliser la probabilité que $Y = 1$ étant donnée la réalisation $\boldsymbol{X} = \boldsymbol{x}$

$$\hat{\Pr}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}).$$

- Si on sait modéliser $\hat{\Pr}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})$, alors on sait également modéliser $\hat{\Pr}(Y = 0 \mid \boldsymbol{X} = \boldsymbol{x}) = 1 - \Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})$.

Régression logistique

- Soient $\boldsymbol{X} = (X_1, \ldots, X_p)$ des variables explicatives et Y une variable réponse qualitative *binaire*.
- On code les réalisations possibles de Y par 0 ou 1.
- Pour tout $\boldsymbol{x}$, on aimerait modéliser la probabilité que $Y = 1$ étant donnée la réalisation $\boldsymbol{X} = \boldsymbol{x}$

$$\hat{\Pr}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}).$$

- Si on sait modéliser $\hat{\Pr}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})$, alors on sait également modéliser $\hat{\Pr}(Y = 0 \mid \boldsymbol{X} = \boldsymbol{x}) = 1 - \Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})$.

Régression logistique

- Soient $\boldsymbol{X} = (X_1, \ldots, X_p)$ des variables explicatives et Y une variable réponse qualitative *binaire*.
- On code les réalisations possibles de Y par 0 ou 1.
- Pour tout $\boldsymbol{x}$, on aimerait modéliser la probabilité que $Y = 1$ étant donnée la réalisation $\boldsymbol{X} = \boldsymbol{x}$

$$\hat{\Pr}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}).$$

- Si on sait modéliser $\hat{\Pr}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})$, alors on sait également modéliser $\hat{\Pr}(Y = 0 \mid \boldsymbol{X} = \boldsymbol{x}) = 1 - \Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})$.

Régression logistique

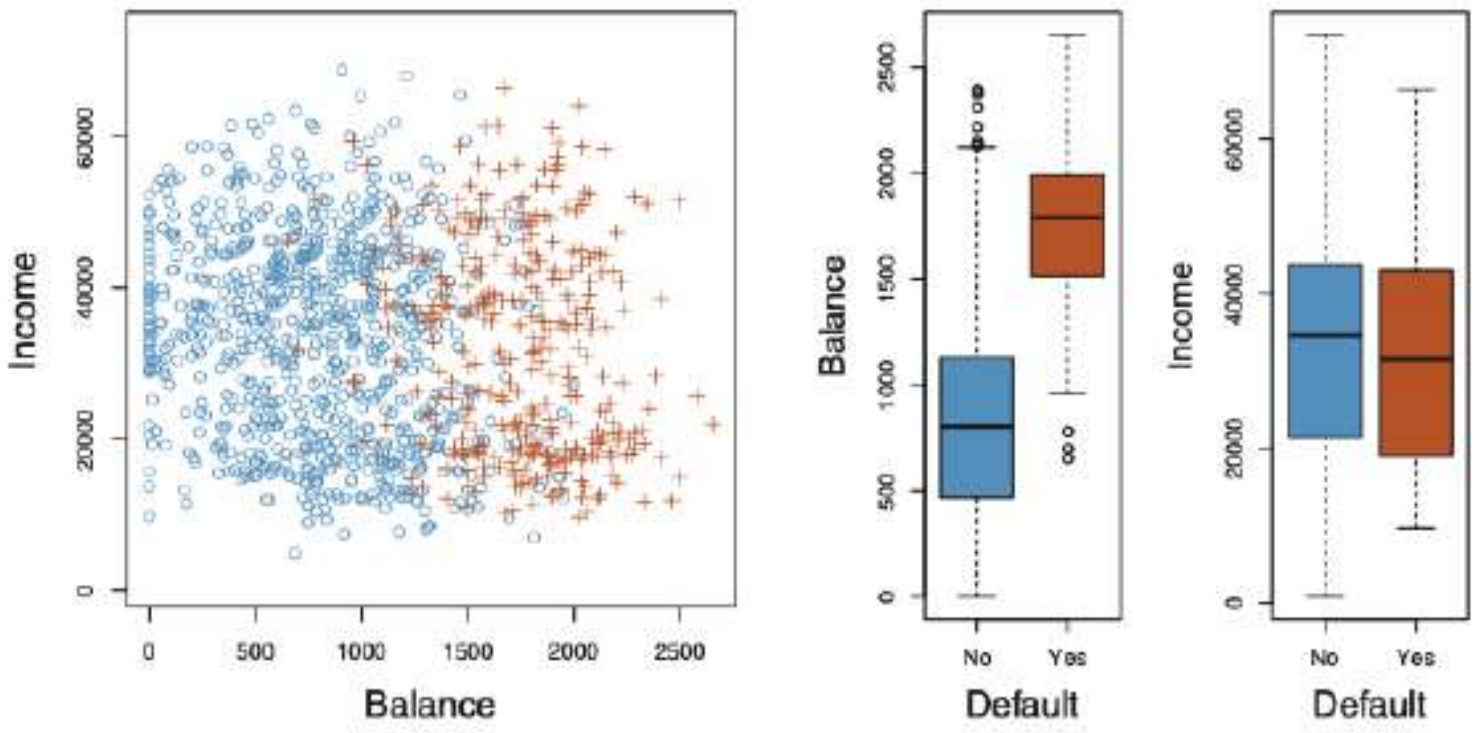
- Soient $\boldsymbol{X} = (X_1, \ldots, X_p)$ des variables explicatives et Y une variable réponse qualitative *binaire*.
- On code les réalisations possibles de Y par 0 ou 1.
- Pour tout $\boldsymbol{x}$, on aimerait modéliser la probabilité que $Y = 1$ étant donnée la réalisation $\boldsymbol{X} = \boldsymbol{x}$

$$\hat{\Pr}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}).$$

- Si on sait modéliser $\hat{\Pr}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})$, alors on sait également modéliser $\hat{\Pr}(Y = 0 \mid \boldsymbol{X} = \boldsymbol{x}) = 1 - \Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})$.

Régression logistique

- Pour diverses raisons, les régressions de types linéaires ne sont pas appropriées...



Figures taken from [James et al., 2013]

Régression logistique

- Pour diverses raisons, les régressions de types linéaires ne sont pas appropriées...

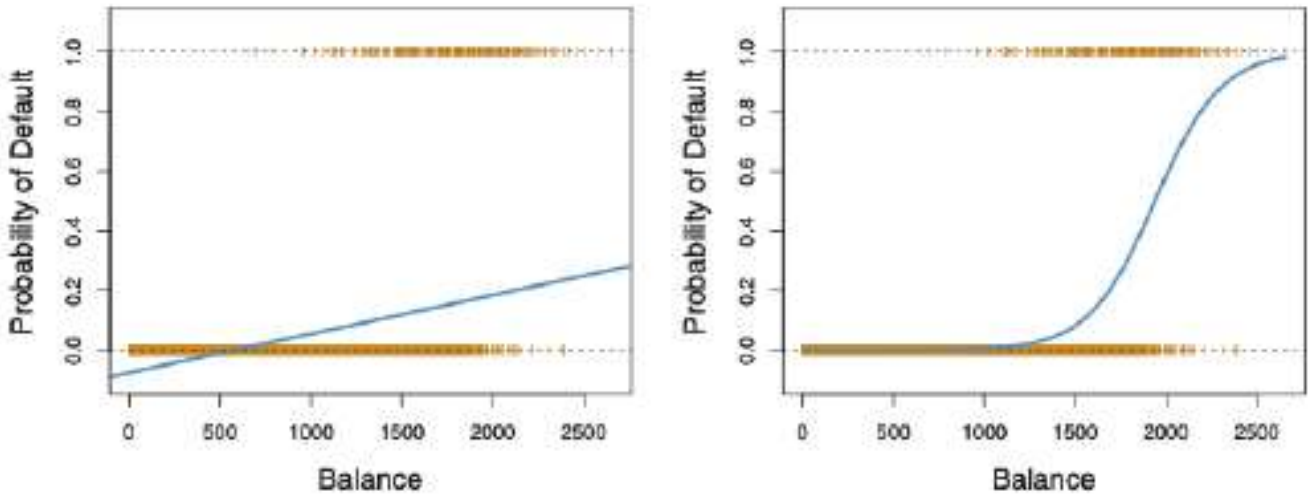


FIGURE 4.2. *Classification using the* `Default` *data.* Left: *Estimated probability of* `default` *using linear regression. Some estimated probabilities are negative! The orange ticks indicate the 0/1 values coded for* `default` *(*`No` *or* `Yes`*).* Right: *Predicted probabilities of* `default` *using logistic regression. All probabilities lie between* 0 *and* 1.

Figures taken from [James et al., 2013]

Régression logistique

- On veut absolument que $\hat{\Pr}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) \in [0, 1]$, puisque c'est une probabilité.
- On suppose alors que la "vraie" $\Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})$ est donnée par la **fonction logistique** suivante:

$$\Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) = \frac{e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p}}{1 + e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p}}$$

- Remarque:

$$\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p \to +\infty \Longrightarrow \Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) \to 1$$
$$\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p \to -\infty \Longrightarrow \Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) \to 0$$

- Si on connait $\Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})$ alors on peut immédiatement déduire $\Pr(Y = 0 \mid \boldsymbol{X} = \boldsymbol{x}) = 1 - \Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})$.

Régression logistique

- On veut absolument que $\hat{\Pr}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) \in [0, 1]$, puisque c'est une probabilité.
- On suppose alors que la "vraie" $\Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})$ est donnée par la **fonction logistique** suivante:

$$\Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) = \frac{e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p}}{1 + e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p}}$$

- Remarque:

$$\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p \to +\infty \Longrightarrow \Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) \to 1$$
$$\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p \to -\infty \Longrightarrow \Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) \to 0$$

- Si on connait $\Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})$ alors on peut immédiatement déduire $\Pr(Y = 0 \mid \boldsymbol{X} = \boldsymbol{x}) = 1 - \Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})$.

Régression logistique

- On veut absolument que $\hat{\Pr}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) \in [0, 1]$, puisque c'est une probabilité.
- On suppose alors que la "vraie" $\Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})$ est donnée par la **fonction logistique** suivante:

$$\Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) = \frac{e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p}}{1 + e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p}}$$

- Remarque:

$$\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p \to +\infty \Longrightarrow \Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) \to 1$$
$$\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p \to -\infty \Longrightarrow \Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) \to 0$$

- Si on connait $\Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})$ alors on peut immédiatement déduire $\Pr(Y = 0 \mid \boldsymbol{X} = \boldsymbol{x}) = 1 - \Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})$.

Régression logistique

- On veut absolument que $\hat{\Pr}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) \in [0, 1]$, puisque c'est une probabilité.
- On suppose alors que la "vraie" $\Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})$ est donnée par la **fonction logistique** suivante:

$$\Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) = \frac{e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p}}{1 + e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p}}$$

- Remarque:

$$\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p \to +\infty \Longrightarrow \Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) \to 1$$
$$\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p \to -\infty \Longrightarrow \Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) \to 0$$

- Si on connait $\Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})$ alors on peut immédiatement déduire $\Pr(Y = 0 \mid \boldsymbol{X} = \boldsymbol{x}) = 1 - \Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})$.

Régression logistique

- Note hypothèse sur la (vraie) forme de $\Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})$

$$\Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) = \frac{e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}}{1 + e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}}$$

implique que **la fonction logit** est de la forme linéaire suivante:

$$\log\left(\frac{\Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})}{1 - \Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

- On aimerait estimer les paramètres $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)$ de la fonction logit de manière pertinente...
- **Remarque:** La regression logistique correspond donc à une régression linéaire classique sur la fonction logit.

Régression logistique

- Note hypothèse sur la (vraie) forme de $\Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})$

$$\Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) = \frac{e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}}{1 + e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}}$$

implique que **la fonction logit** est de la forme linéaire suivante:

$$\log\left(\frac{\Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})}{1 - \Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

- On aimerait estimer les paramètres $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)$ de la fonction logit de manière pertinente...
- **Remarque:** La regression logistique correspond donc à une régression linéaire classique sur la fonction logit.

Régression logistique

- Note hypothèse sur la (vraie) forme de $\Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})$

$$\Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) = \frac{e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}}{1 + e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}}$$

  implique que **la fonction logit** est de la forme linéaire suivante:

$$\log\left(\frac{\Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})}{1 - \Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

- On aimerait estimer les paramètres $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)$ de la fonction logit de manière pertinente...
- **Remarque:** La regression logistique correspond donc à une régression linéaire classique sur la fonction logit.

Régression logistique

- Note hypothèse sur la (vraie) forme de $\Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})$

$$\Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) = \frac{e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}}{1 + e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}}$$

  implique que **la fonction logit** est de la forme linéaire suivante:

$$\log\left(\frac{\Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})}{1 - \Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x})}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

- On aimerait estimer les paramètres $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)$ de la fonction logit de manière pertinente...
- **Remarque:** La regression logistique correspond donc à une régression linéaire classique sur la fonction logit.

Régression logistique

- Soit un training set $S_{\text{train}} = \{(\boldsymbol{x_1}, y_1), \ldots, (\boldsymbol{x_N}, y_N)\}$.
- Soit

$$\hat{\Pr}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p}}$$

une estimation de

$$\Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) = \frac{e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p}}{1 + e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p}}$$

- Idéalement, on aimerait obtenir des estimateurs $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \ldots, \hat{\beta}_p)$ de $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_p)$ tels que, pour tout $(\boldsymbol{x_i}, y_i) \in S_{\text{train}}$ on ait:

$$y_i = 1 \Rightarrow \hat{\Pr}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x_i}) > 0.5$$
$$y_i = 0 \Rightarrow \hat{\Pr}(Y = 0 \mid \boldsymbol{X} = \boldsymbol{x_i}) = 1 - \hat{\Pr}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x_i}) \leq 0.5.$$

Régression logistique

- Soit un training set $S_{\text{train}} = \{(\boldsymbol{x_1}, y_1), \ldots, (\boldsymbol{x_N}, y_N)\}$.
- Soit

$$\hat{\Pr}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p}}$$

une estimation de

$$\Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) = \frac{e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p}}{1 + e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p}}$$

- Idéalement, on aimerait obtenir des estimateurs $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \ldots, \hat{\beta}_p)$ de $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_p)$ tels que, pour tout $(\boldsymbol{x_i}, y_i) \in S_{\text{train}}$ on ait:

$$y_i = 1 \Rightarrow \hat{\Pr}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x_i}) > 0.5$$

$$y_i = 0 \Rightarrow \hat{\Pr}(Y = 0 \mid \boldsymbol{X} = \boldsymbol{x_i}) = 1 - \hat{\Pr}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x_i}) \leq 0.5.$$

Régression logistique

- Soit un training set $S_{\text{train}} = \{(\boldsymbol{x_1}, y_1), \ldots, (\boldsymbol{x_N}, y_N)\}$.
- Soit

$$\hat{\Pr}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p}}$$

une estimation de

$$\Pr(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) = \frac{e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p}}{1 + e^{\beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p}}$$

- Idéalement, on aimerait obtenir des estimateurs $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \ldots, \hat{\beta}_p)$ de $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_p)$ tels que, pour tout $(\boldsymbol{x_i}, y_i) \in S_{\text{train}}$ on ait:

$$y_i = 1 \;\Rightarrow\; \hat{\Pr}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x_i}) > 0.5$$
$$y_i = 0 \;\Rightarrow\; \hat{\Pr}(Y = 0 \mid \boldsymbol{X} = \boldsymbol{x_i}) = 1 - \hat{\Pr}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x_i}) \leq 0.5.$$

Régression logistique

- Pour cela, on choisit les paramètres $\hat{\beta}$ qui maximisent la **fonction de vraisemblance (likelihood)** suivante:

$$\mathcal{L}(\boldsymbol{\beta}) := \prod_{\{i:y_i=1\}} \Pr(Y=1 \mid \boldsymbol{X}=\boldsymbol{x_i}) \prod_{\{i':y_{i'}=0\}} \left(\Pr(Y=0 \mid \boldsymbol{X}=\boldsymbol{x_{i'}})\right) =$$

$$\prod_{\{i:y_i=1\}} \frac{e^{\beta_0+\sum_{k=1}^{p}\beta_k x_{ik}}}{1+e^{\beta_0+\sum_{k=1}^{p}\beta_k x_{ik}}} \prod_{\{i':y_{i'}=0\}} \left(1-\frac{e^{\beta_0+\sum_{k=1}^{p}\beta_k x_{i'k}}}{1+e^{\beta_0+\sum_{k=1}^{p}\beta_k x_{i'k}}}\right)$$

- On a donc

$$\hat{\beta} = \arg\max_{\beta} \mathcal{L}(\beta)$$

- En pratique, les paramètres $\hat{\beta}$ sont calculés par une méthode de gradient itérative (pas de solution exacte).

Régression logistique

- Pour cela, on choisit les paramètres $\hat{\boldsymbol{\beta}}$ qui maximisent la **fonction de vraisemblance (likelihood)** suivante:

$$\mathcal{L}(\boldsymbol{\beta}) := \prod_{\{i:y_i=1\}} \Pr(Y=1 \mid \boldsymbol{X}=\boldsymbol{x_i}) \prod_{\{i':y_{i'}=0\}} \left(\Pr(Y=0 \mid \boldsymbol{X}=\boldsymbol{x_{i'}})\right) =$$

$$\prod_{\{i:y_i=1\}} \frac{e^{\beta_0+\sum_{k=1}^{p}\beta_k x_{ik}}}{1+e^{\beta_0+\sum_{k=1}^{p}\beta_k x_{ik}}} \prod_{\{i':y_{i'}=0\}} \left(1-\frac{e^{\beta_0+\sum_{k=1}^{p}\beta_k x_{i'k}}}{1+e^{\beta_0+\sum_{k=1}^{p}\beta_k x_{i'k}}}\right)$$

- On a donc

$$\hat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}} \mathcal{L}(\boldsymbol{\beta})$$

- En pratique, les paramètres $\hat{\beta}$ sont calculés par une méthode de gradient itérative (pas de solution exacte).

Régression logistique

- Pour cela, on choisit les paramètres $\hat{\boldsymbol{\beta}}$ qui maximisent la **fonction de vraisemblance (likelihood)** suivante:

$$\mathcal{L}(\boldsymbol{\beta}) := \prod_{\{i:y_i=1\}} \Pr(Y=1 \mid \boldsymbol{X}=\boldsymbol{x_i}) \prod_{\{i':y_{i'}=0\}} \left(\Pr(Y=0 \mid \boldsymbol{X}=\boldsymbol{x_{i'}})\right) =$$

$$\prod_{\{i:y_i=1\}} \frac{e^{\beta_0+\sum_{k=1}^{p}\beta_k x_{ik}}}{1+e^{\beta_0+\sum_{k=1}^{p}\beta_k x_{ik}}} \prod_{\{i':y_{i'}=0\}} \left(1-\frac{e^{\beta_0+\sum_{k=1}^{p}\beta_k x_{i'k}}}{1+e^{\beta_0+\sum_{k=1}^{p}\beta_k x_{i'k}}}\right)$$

- On a donc

$$\hat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}} \mathcal{L}(\boldsymbol{\beta})$$

- En pratique, les paramètres $\hat{\boldsymbol{\beta}}$ sont calculés par une méthode de gradient itérative (pas de solution exacte).

Régression logistique

- Une fois les paramètres $\hat{\beta}$ calculés, on peut faire des prédictions de manière très simple.
- Soit $\boldsymbol{x} = (x_1, \ldots, x_p)$ un point. La prédiction $\hat{y}$ associée à x est donnée par:

$$\hat{y} := \begin{cases} 1, \text{ si } \hat{\Pr}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) = \dfrac{e^{\hat{\beta}_0 + \sum_{k=1}^{p} \hat{\beta}_k x_k}}{1 + e^{\hat{\beta}_0 + \sum_{k=1}^{p} \hat{\beta}_k x_k}} > 0.5 \\ 0, \text{ si } \hat{\Pr}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) = \dfrac{e^{\hat{\beta}_0 + \sum_{k=1}^{p} \hat{\beta}_k x_k}}{1 + e^{\hat{\beta}_0 + \sum_{k=1}^{p} \hat{\beta}_k x_k}} \leq 0.5 \end{cases}$$

Régression logistique

- Une fois les paramètres $\hat{\boldsymbol{\beta}}$ calculés, on peut faire des prédictions de manière très simple.
- Soit $\boldsymbol{x} = (x_1, \ldots, x_p)$ un point. La prédiction $\hat{y}$ associée à x est donnée par:

$$\hat{y} := \begin{cases} 1, \text{ si } \hat{\Pr}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) = \dfrac{e^{\hat{\beta}_0 + \sum_{k=1}^{p} \hat{\beta}_k x_k}}{1 + e^{\hat{\beta}_0 + \sum_{k=1}^{p} \hat{\beta}_k x_k}} > 0.5 \\ 0, \text{ si } \hat{\Pr}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) = \dfrac{e^{\hat{\beta}_0 + \sum_{k=1}^{p} \hat{\beta}_k x_k}}{1 + e^{\hat{\beta}_0 + \sum_{k=1}^{p} \hat{\beta}_k x_k}} \leq 0.5 \end{cases}$$

Régression logistique

- **Généralisation à un contexte multi-classes:** Supposons que Y possède $k \geq 2$ valeurs possibles $c_1, \ldots, c_k$.
- On code Y par une variable $\boldsymbol{Y} = (Y_1, \ldots, Y_k)$ telle que $Y_i = 1$ ssi $Y = c_i$ (1-hot encoding).
- On estime les paramètres $\hat{\boldsymbol{\beta}}$ en maximisant une fonction de likelihood similaire $\mathcal{L}(\boldsymbol{\beta})$, i.e., $\hat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}} \mathcal{L}(\boldsymbol{\beta})$.
- Grâce aix $\hat{\boldsymbol{\beta}}$, on a alors une estimation $\hat{\Pr}(\boldsymbol{Y} = \boldsymbol{y} \mid \boldsymbol{X} = \boldsymbol{x})$ pour toute réalisation $(\boldsymbol{x}, \boldsymbol{y})$ de $(\boldsymbol{X}, \boldsymbol{Y})$.
- Pour tout point $\boldsymbol{x}$, la prédiction $\hat{y}$ est donnée par:

$$\hat{y} := c_i \text{ si et seulement si}$$

$$\hat{\Pr}(\boldsymbol{Y} = \mathbf{1}_i \mid X = \boldsymbol{x}) > \hat{\Pr}(\boldsymbol{Y} = \mathbf{1}_j \mid X = \boldsymbol{x}) \text{ pour tout } j \neq i.$$

Régression logistique

- **Généralisation à un contexte multi-classes:** Supposons que Y possède $k \geq 2$ valeurs possibles $c_1, \ldots, c_k$.
- On code Y par une variable $\boldsymbol{Y} = (Y_1, \ldots, Y_k)$ telle que $Y_i = 1$ ssi $Y = c_i$ (1-hot encoding).
- On estime les paramètres $\hat{\boldsymbol{\beta}}$ en maximisant une fonction de likelihood similaire $\mathcal{L}(\boldsymbol{\beta})$, i.e., $\hat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}} \mathcal{L}(\boldsymbol{\beta})$.
- Grâce aix $\hat{\boldsymbol{\beta}}$, on a alors une estimation $\hat{\Pr}(\boldsymbol{Y} = \boldsymbol{y} \mid \boldsymbol{X} = \boldsymbol{x})$ pour toute réalisation $(\boldsymbol{x}, \boldsymbol{y})$ de $(\boldsymbol{X}, \boldsymbol{Y})$.
- Pour tout point $\boldsymbol{x}$, la prédiction $\hat{y}$ est donnée par:

$$\hat{y} := c_i \text{ si et seulement si}$$

$$\hat{\Pr}(\boldsymbol{Y} = \mathbf{1}_i \mid X = \boldsymbol{x}) > \hat{\Pr}(\boldsymbol{Y} = \mathbf{1}_j \mid X = \boldsymbol{x}) \text{ pour tout } j \neq i.$$

Régression logistique

- **Généralisation à un contexte multi-classes:** Supposons que Y possède $k \geq 2$ valeurs possibles $c_1, \ldots, c_k$.
- On code Y par une variable $\boldsymbol{Y} = (Y_1, \ldots, Y_k)$ telle que $Y_i = 1$ ssi $Y = c_i$ (1-hot encoding).
- On estime les paramètres $\hat{\boldsymbol{\beta}}$ en maximisant une fonction de likelihood similaire $\mathcal{L}(\boldsymbol{\beta})$, i.e., $\hat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}} \mathcal{L}(\boldsymbol{\beta})$.
- Grâce aix $\hat{\beta}$, on a alors une estimation $\hat{\Pr}(\boldsymbol{Y} = \boldsymbol{y} \mid \boldsymbol{X} = \boldsymbol{x})$ pour toute réalisation $(\boldsymbol{x}, \boldsymbol{y})$ de $(\boldsymbol{X}, \boldsymbol{Y})$.
- Pour tout point $\boldsymbol{x}$, la prédiction $\hat{y}$ est donnée par:

$$\hat{y} := c_i \text{ si et seulement si}$$

$$\hat{\Pr}(\boldsymbol{Y} = \mathbf{1}_i \mid X = \boldsymbol{x}) > \hat{\Pr}(\boldsymbol{Y} = \mathbf{1}_j \mid X = \boldsymbol{x}) \text{ pour tout } j \neq i.$$

Régression logistique

- **Généralisation à un contexte multi-classes:** Supposons que Y possède $k \geq 2$ valeurs possibles $c_1, \ldots, c_k$.
- On code Y par une variable $\boldsymbol{Y} = (Y_1, \ldots, Y_k)$ telle que $Y_i = 1$ ssi $Y = c_i$ (1-hot encoding).
- On estime les paramètres $\hat{\boldsymbol{\beta}}$ en maximisant une fonction de likelihood similaire $\mathcal{L}(\boldsymbol{\beta})$, i.e., $\hat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}} \mathcal{L}(\boldsymbol{\beta})$.
- Grâce aix $\hat{\boldsymbol{\beta}}$, on a alors une estimation $\hat{\Pr}(\boldsymbol{Y} = \boldsymbol{y} \mid \boldsymbol{X} = \boldsymbol{x})$ pour toute réalisation $(\boldsymbol{x}, \boldsymbol{y})$ de $(\boldsymbol{X}, \boldsymbol{Y})$.
- Pour tout point x, la prédiction $\hat{y}$ est donnée par:

$$\hat{y} := c_i \text{ si et seulement si}$$

$$\hat{\Pr}(\boldsymbol{Y} = \mathbf{1}_i \mid X = \boldsymbol{x}) > \hat{\Pr}(\boldsymbol{Y} = \mathbf{1}_j \mid X = \boldsymbol{x}) \text{ pour tout } j \neq i.$$

Régression logistique

- **Généralisation à un contexte multi-classes:** Supposons que Y possède $k \geq 2$ valeurs possibles $c_1, \ldots, c_k$.
- On code Y par une variable $\boldsymbol{Y} = (Y_1, \ldots, Y_k)$ telle que $Y_i = 1$ ssi $Y = c_i$ (1-hot encoding).
- On estime les paramètres $\hat{\boldsymbol{\beta}}$ en maximisant une fonction de likelihood similaire $\mathcal{L}(\boldsymbol{\beta})$, i.e., $\hat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}} \mathcal{L}(\boldsymbol{\beta})$.
- Grâce aix $\hat{\boldsymbol{\beta}}$, on a alors une estimation $\hat{\Pr}(\boldsymbol{Y} = \boldsymbol{y} \mid \boldsymbol{X} = \boldsymbol{x})$ pour toute réalisation $(\boldsymbol{x}, \boldsymbol{y})$ de $(\boldsymbol{X}, \boldsymbol{Y})$.
- Pour tout point $\boldsymbol{x}$, la prédiction $\hat{y}$ est donnée par:

$$\hat{y} := c_i \text{ si et seulement si}$$

$$\hat{\Pr}(\boldsymbol{Y} = \mathbf{1}_i \mid X = \boldsymbol{x}) > \hat{\Pr}(\boldsymbol{Y} = \mathbf{1}_j \mid X = \boldsymbol{x}) \text{ pour tout } j \neq i.$$

Matrice de confusion

- Comment évaluer la performance d'un classifieur binaire?
- On utilise les **matrices de confusion** et les métriques qui en découlent.
- **Exemple:** Soit un groupe de 1000 personnes dans lequel se trouve 10 individus dangereux.
- Supposons qu'on ait entraîné qui modèle (classifieur binaire) qui prédise si un individu est dangereux ou non.
- Mais ce modèle est très défaillant: parmi 10 individus dangereux, seul 1 est détecté comme dangereux; sinon, parmi les 990 personnes inoffensives, toutes sont détectées comme inoffensives.

Matrice de confusion

- Comment évaluer la performance d'un classifieur binaire?
- On utilise les **matrices de confusion** et les métriques qui en découlent.
- **Exemple:** Soit un groupe de 1000 personnes dans lequel se trouve 10 individus dangereux.
- Supposons qu'on ait entraîné qui modèle (classifieur binaire) qui prédise si un individu est dangereux ou non.
- Mais ce modèle est très défaillant: parmi 10 individus dangereux, seul 1 est détecté comme dangereux; sinon, parmi les 990 personnes inoffensives, toutes sont détectées comme inoffensives.

Matrice de confusion

- Comment évaluer la performance d'un classifieur binaire?
- On utilise les **matrices de confusion** et les métriques qui en découlent.
- **Exemple:** Soit un groupe de 1000 personnes dans lequel se trouve 10 individus dangereux.
- Supposons qu'on ait entraîné qui modèle (classifieur binaire) qui prédise si un individu est dangereux ou non.
- Mais ce modèle est très défaillant: parmi 10 individus dangereux, seul 1 est détecté comme dangereux; sinon, parmi les 990 personnes inoffensives, toutes sont détectées comme inoffensives.

Matrice de confusion

- Comment évaluer la performance d'un classifieur binaire?
- On utilise les **matrices de confusion** et les métriques qui en découlent.
- **Exemple:** Soit un groupe de 1000 personnes dans lequel se trouve 10 individus dangereux.
- Supposons qu'on ait entraîné qui modèle (classifieur binaire) qui prédise si un individu est dangereux ou non.
- Mais ce modèle est très défaillant: parmi 10 individus dangereux, seul 1 est détecté comme dangereux; sinon, parmi les 990 personnes inoffensives, toutes sont détectées comme inoffensives.

Matrice de confusion

- Comment évaluer la performance d'un classifieur binaire?
- On utilise les **matrices de confusion** et les métriques qui en découlent.
- **Exemple:** Soit un groupe de 1000 personnes dans lequel se trouve 10 individus dangereux.
- Supposons qu'on ait entraîné qui modèle (classifieur binaire) qui prédise si un individu est dangereux ou non.
- Mais ce modèle est très défaillant: parmi 10 individus dangereux, seul 1 est détecté comme dangereux; sinon, parmi les 990 personnes inoffensives, toutes sont détectées comme inoffensives.

Matrice de confusion

- Les résultats du classifieur sont les suivants:

		prédiction	
		inoffensif	dangereux
réalité	inoffensif	990	0
	dangereux	9	1

- L'erreur totale du modèle n'est que de $\frac{9+0}{1000} = 0.9\%$.
- Autrement dit, l'accuracy du modèle est de $\frac{990+1}{1000} = 99.1\%$.
- Pourtant, le classifieur est mauvais: il laisse passer presque tous les dangereux.
- C'est le paradoxe de l'accuracy.

Matrice de confusion

- Les résultats du classifieur sont les suivants:

		prédiction	
		inoffensif	dangereux
réalité	inoffensif	990	0
	dangereux	9	1

- L'erreur totale du modèle n'est que de $\frac{9+0}{1000} = 0.9\%$.
- Autrement dit, l'accuracy du modèle est de $\frac{990+1}{1000} = 99.1\%$.
- Pourtant, le classifieur est mauvais: il laisse passer presque tous les dangereux.
- C'est le paradoxe de l'accuracy.

Matrice de confusion

- Les résultats du classifieur sont les suivants:

		prédiction	
		inoffensif	dangereux
réalité	inoffensif	990	0
	dangereux	9	1

- L'erreur totale du modèle n'est que de $\frac{9+0}{1000} = 0.9\%$.
- Autrement dit, l'accuracy du modèle est de $\frac{990+1}{1000} = 99.1\%$.
- Pourtant, le classifieur est mauvais: il laisse passer presque tous les dangereux.
- C'est le **paradoxe de l'accuracy**.

Matrice de confusion

- Les résultats du classifieur sont les suivants:

		prédiction	
		inoffensif	dangereux
réalité	inoffensif	990	0
	dangereux	9	1

- L'erreur totale du modèle n'est que de $\frac{9+0}{1000} = 0.9\%$.
- Autrement dit, l'accuracy du modèle est de $\frac{990+1}{1000} = 99.1\%$.
- Pourtant, le classifieur est mauvais: il laisse passer presque tous les dangereux.
- C'est le paradoxe de l'accuracy.

Matrice de confusion

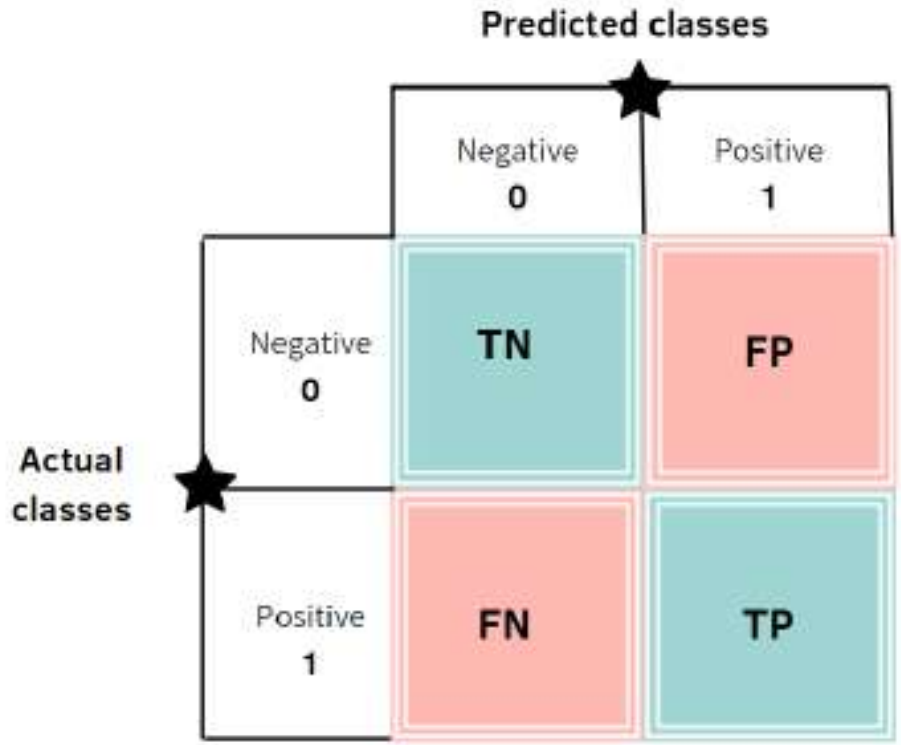
- Les résultats du classifieur sont les suivants:

		prédiction	
		inoffensif	dangereux
réalité	inoffensif	990	0
	dangereux	9	1

- L'erreur totale du modèle n'est que de $\frac{9+0}{1000} = 0.9\%$.
- Autrement dit, l'accuracy du modèle est de $\frac{990+1}{1000} = 99.1\%$.
- Pourtant, le classifieur est mauvais: il laisse passer presque tous les dangereux.
- C'est le **paradoxe de l'accuracy**.

Matrice de confusion

- De manière générale, la **matrice de confusion** d'un classifieur est la suivante:



Matrice de confusion

- **True positive:** prédit comme positif (positive), et c'est juste dans la réalité (True).
- **True negative:** prédit comme négatif (negative), et c'est juste dans la réalité (True).
- **False positive:** prédit comme positif (positive), et c'est faux dans la réalité (False).
- **False negative:** prédit comme négatif (negative), et c'est faux dans la réalité (False).

Matrice de confusion

- **True positive:** prédit comme positif (positive), et c'est juste dans la réalité (True).
- **True negative:** prédit comme négatif (negative), et c'est juste dans la réalité (True).
- **False positive:** prédit comme positif (positive), et c'est faux dans la réalité (False).
- **False negative:** prédit comme négatif (negative), et c'est faux dans la réalité (False).

Matrice de confusion

- **True positive:** prédit comme positif (positive), et c'est juste dans la réalité (True).
- **True negative:** prédit comme négatif (negative), et c'est juste dans la réalité (True).
- **False positive:** prédit comme positif (positive), et c'est faux dans la réalité (False).
- **False negative:** prédit comme négatif (negative), et c'est faux dans la réalité (False).

Matrice de confusion

- **True positive:** prédit comme positif (positive), et c'est juste dans la réalité (True).
- **True negative:** prédit comme négatif (negative), et c'est juste dans la réalité (True).
- **False positive:** prédit comme positif (positive), et c'est faux dans la réalité (False).
- **False negative:** prédit comme négatif (negative), et c'est faux dans la réalité (False).

Matrice de confusion

- **Accuracy**: $Accuracy = \frac{TP+TN}{N+P}$
- False positive rate: $FPR = \frac{FP}{N}$
- True positive rate / Recall: $TPR = Recall = \frac{TP}{P}$
- Precision: $Precision = \frac{TP}{P^*}$
- Negative prediction value : $NPV = \frac{TN}{N^*}$

		Predicted class		
		− or Null	+ or Non-null	Total
True class	− or Null	True Neg. (TN)	False Pos. (FP)	N
	+ or Non-null	False Neg. (FN)	True Pos. (TP)	P
	Total	N*	P*	

TABLE 4.6. *Possible results when applying a classifier or diagnostic test to a population.*

Matrice de confusion

- **Accuracy**: $Accuracy = \frac{TP+TN}{N+P}$
- **False positive rate**: $FPR = \frac{FP}{N}$
- **True positive rate / Recall**: $TPR = Recall = \frac{TP}{P}$
- **Precision**: $Precision = \frac{TP}{P^*}$
- **Negative prediction value** : $NPV = \frac{TN}{N^*}$

		Predicted class		
		− or Null	+ or Non-null	Total
True class	− or Null	True Neg. (TN)	False Pos. (FP)	N
	+ or Non-null	False Neg. (FN)	True Pos. (TP)	P
	Total	N*	P*	

TABLE 4.6. *Possible results when applying a classifier or diagnostic test to a population.*

Matrice de confusion

- **Accuracy**: $Accuracy = \frac{TP+TN}{N+P}$
- **False positive rate**: $FPR = \frac{FP}{N}$
- **True positive rate / Recall**: $TPR = Recall = \frac{TP}{P}$
- **Precision**: $Precision = \frac{TP}{P^*}$
- **Negative prediction value** : $NPV = \frac{TN}{N^*}$

		Predicted class		
		− or Null	+ or Non-null	Total
True class	− or Null	True Neg. (TN)	False Pos. (FP)	N
	+ or Non-null	False Neg. (FN)	True Pos. (TP)	P
	Total	N*	P*	

TABLE 4.6. *Possible results when applying a classifier or diagnostic test to a population.*

Matrice de confusion

- **Accuracy**: $Accuracy = \frac{TP+TN}{N+P}$
- **False positive rate**: $FPR = \frac{FP}{N}$
- **True positive rate / Recall**: $TPR = Recall = \frac{TP}{P}$
- **Precision**: $Precision = \frac{TP}{P^*}$
- Negative prediction value : $NPV = \frac{TN}{N^*}$

		Predicted class		
		− or Null	+ or Non-null	Total
True class	− or Null	True Neg. (TN)	False Pos. (FP)	N
	+ or Non-null	False Neg. (FN)	True Pos. (TP)	P
	Total	N^*	P^*	

TABLE 4.6. *Possible results when applying a classifier or diagnostic test to a population.*

Matrice de confusion

- **Accuracy**: $Accuracy = \frac{TP+TN}{N+P}$
- **False positive rate**: $FPR = \frac{FP}{N}$
- **True positive rate / Recall**: $TPR = Recall = \frac{TP}{P}$
- **Precision**: $Precision = \frac{TP}{P^*}$
- **Negative prediction value** : $NPV = \frac{TN}{N^*}$

		Predicted class		
		− or Null	+ or Non-null	Total
True class	− or Null	True Neg. (TN)	False Pos. (FP)	N
	+ or Non-null	False Neg. (FN)	True Pos. (TP)	P
	Total	N^*	P^*	

TABLE 4.6. *Possible results when applying a classifier or diagnostic test to a population.*

Matrice de confusion

Dans notre example, on a:

- $Accuracy = \frac{990+1}{1000} = 99.1$ (bon)
- $FPR = \frac{0}{990+0} = 0\%$ (bon, low=good)
- $Recall = \frac{TP}{P} = \frac{1}{9+1} = 10\%$: (mauvais!)
- $Precision = \frac{TP}{P^*} = \frac{1}{0+1} = 100\%$ (bon)
- $\frac{TN}{N^*} = \frac{990}{990+9} = 99.099\%$ (bon)
- Dans ce cas, les bonnes performances viennent du fait que les 2 classes sont fortement "imbalanced". Toutefois, le *recall* nous indique que notre modèle est mauvais en certains aspects.

		prédiction	
		inoffensif	dangereux
réalité	inoffensif	990	0
	dangereux	9	1

Matrice de confusion

Dans notre example, on a:

- $Accuracy = \frac{990+1}{1000} = 99.1$ (bon)
- $FPR = \frac{0}{990+0} = 0\%$ (bon, low=good)
- $Recall = \frac{TP}{P} = \frac{1}{9+1} = 10\%$: (mauvais!)
- $Precision = \frac{TP}{P^*} = \frac{1}{0+1} = 100\%$ (bon)
- $\frac{TN}{N^*} = \frac{990}{990+9} = 99.099\%$ (bon)
- Dans ce cas, les bonnes performances viennent du fait que les 2 classes sont fortement "imbalanced". Toutefois, le *recall* nous indique que notre modèle est mauvais en certains aspects.

		prédiction	
		inoffensif	dangereux
réalité	inoffensif	990	0
	dangereux	9	1

Matrice de confusion

Dans notre example, on a:

- $Accuracy = \frac{990+1}{1000} = 99.1$ (bon)
- $FPR = \frac{0}{990+0} = 0\%$ (bon, low=good)
- $Recall = \frac{TP}{P} = \frac{1}{9+1} = 10\%$: (mauvais!)
- $Precision = \frac{TP}{P^*} = \frac{1}{0+1} = 100\%$ (bon)
- $\frac{TN}{N^*} = \frac{990}{990+9} = 99.099\%$ (bon)
- Dans ce cas, les bonnes performances viennent du fait que les 2 classes sont fortement "imbalanced". Toutefois, le *recall* nous indique que notre modèle est mauvais en certains aspects.

		prédiction	
		inoffensif	dangereux
réalité	inoffensif	990	0
	dangereux	9	1

Matrice de confusion

Dans notre example, on a:

- $Accuracy = \frac{990+1}{1000} = 99.1$ (bon)
- $FPR = \frac{0}{990+0} = 0\%$ (bon, low=good)
- $Recall = \frac{TP}{P} = \frac{1}{9+1} = 10\%$: (mauvais!)
- $Precision = \frac{TP}{P^*} = \frac{1}{0+1} = 100\%$ (bon)
- $\frac{TN}{N^*} = \frac{990}{990+9} = 99.099\%$ (bon)
- Dans ce cas, les bonnes performances viennent du fait que les 2 classes sont fortement "imbalanced". Toutefois, le *recall* nous indique que notre modèle est mauvais en certains aspects.

		prédiction	
		inoffensif	dangereux
réalité	inoffensif	990	0
	dangereux	9	1

Matrice de confusion

Dans notre example, on a:

- $Accuracy = \frac{990+1}{1000} = 99.1$ (bon)
- $FPR = \frac{0}{990+0} = 0\%$ (bon, low=good)
- $Recall = \frac{TP}{P} = \frac{1}{9+1} = 10\%$: (mauvais!)
- $Precision = \frac{TP}{P^*} = \frac{1}{0+1} = 100\%$ (bon)
- $\frac{TN}{N^*} = \frac{990}{990+9} = 99.099\%$ (bon)
- Dans ce cas, les bonnes performances viennent du fait que les 2 classes sont fortement "imbalanced". Toutefois, le *recall* nous indique que notre modèle est mauvais en certains aspects.

		prédiction	
		inoffensif	dangereux
réalité	inoffensif	990	0
	dangereux	9	1

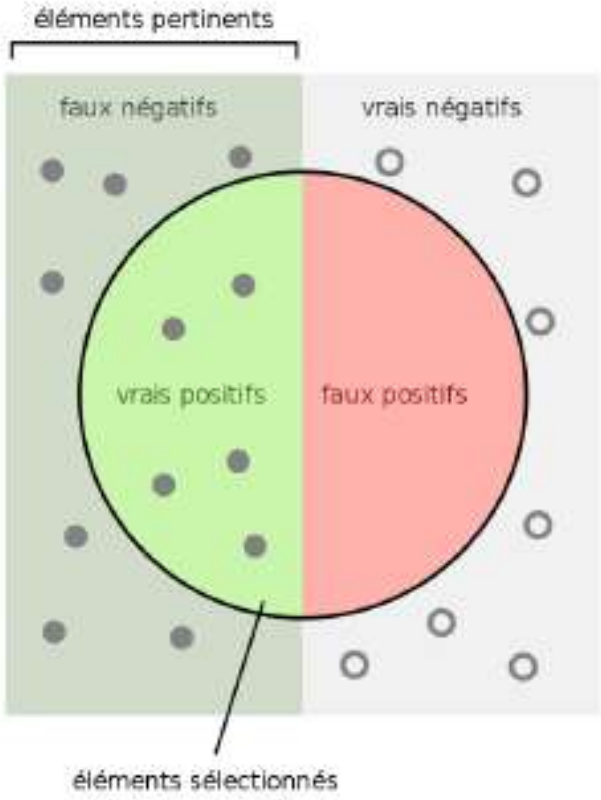
Matrice de confusion

Dans notre example, on a:

- $Accuracy = \frac{990+1}{1000} = 99.1$ (bon)
- $FPR = \frac{0}{990+0} = 0\%$ (bon, low=good)
- $Recall = \frac{TP}{P} = \frac{1}{9+1} = 10\%$: (mauvais!)
- $Precision = \frac{TP}{P^*} = \frac{1}{0+1} = 100\%$ (bon)
- $\frac{TN}{N^*} = \frac{990}{990+9} = 99.099\%$ (bon)
- Dans ce cas, les bonnes performances viennent du fait que les 2 classes sont fortement "imbalanced". Toutefois, le *recall* nous indique que notre modèle est mauvais en certains aspects.

		prédiction	
		inoffensif	dangereux
réalité	inoffensif	990	0
	dangereux	9	1

Matrice de confusion



Combien de candidats sélectionnés sont pertinents ?

$$\text{Précision} = \frac{\text{vrais positifs}}{\text{éléments sélectionnés}}$$

Combien d'éléments pertinents sont sélectionnés ?

$$\text{Rappel} = \frac{\text{vrais positifs}}{\text{éléments pertinents}}$$

Matrice de confusion

- On rappel que notre classifieur retourne une probabilité

$$\hat{p} = \hat{\Pr}(Y = 1 \mid \boldsymbol{X} = x)$$

et qu'on classifie x dans la classe 1 si $\hat{p} \geq 0.5$ et dans la classe 0 sinon.

- Le choix du seuil (threshold) de 0.5 est arbitraire...
- On peut créer un classifieur plus ou moins sévères en augmentant ou diminuant ce seuil.

Matrice de confusion

- On rappel que notre classifieur retourne une probabilité

$$\hat{p} = \hat{\Pr}(Y = 1 \mid \boldsymbol{X} = x)$$

  et qu'on classifie x dans la classe 1 si $\hat{p} \geq 0.5$ et dans la classe 0 sinon.
- Le choix du seuil (threshold) de 0.5 est arbitraire...
- On peut créer un classifieur plus ou moins sévères en augmentant ou diminuant ce seuil.

Matrice de confusion

- On rappel que notre classifieur retourne une probabilité

$$\hat{p} = \hat{\Pr}(Y = 1 \mid \boldsymbol{X} = x)$$

  et qu'on classifie x dans la classe 1 si $\hat{p} \geq 0.5$ et dans la classe 0 sinon.
- Le choix du seuil (threshold) de 0.5 est arbitraire...
- On peut créer un classifieur plus ou moins sévères en augmentant ou diminuant ce seuil.

Matrice de confusion

- Voici deux matrices de confusions pour un calssifieur qui utilise les seuils de 0.5 et 0.2, respectivement.
- On voit que le nombre de prédictions positives augmente de 104 à 430 (puisque le seuil est plus bas).

		True default status		
		No	Yes	Total
Predicted default status	No	9,644	252	9,896
	Yes	23	81	104
	Total	9,667	333	10,000

		True default status		
		No	Yes	Total
Predicted default status	No	9,432	138	9,570
	Yes	235	195	430
	Total	9,667	333	10,000

Matrice de confusion

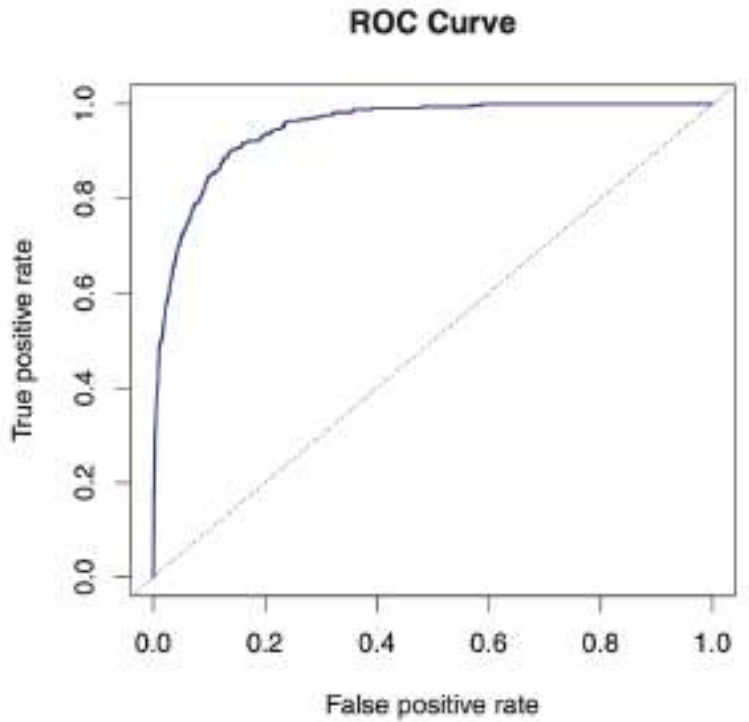
- Voici deux matrices de confusions pour un calssifieur qui utilise les seuils de 0.5 et 0.2, respectivement.
- On voit que le nombre de prédictions positives augmente de 104 à 430 (puisque le seuil est plus bas).

		True default status		
		No	Yes	Total
Predicted default status	No	9,644	252	9,896
	Yes	23	81	104
	Total	9,667	333	10,000

		True default status		
		No	Yes	Total
Predicted default status	No	9,432	138	9,570
	Yes	235	195	430
	Total	9,667	333	10,000

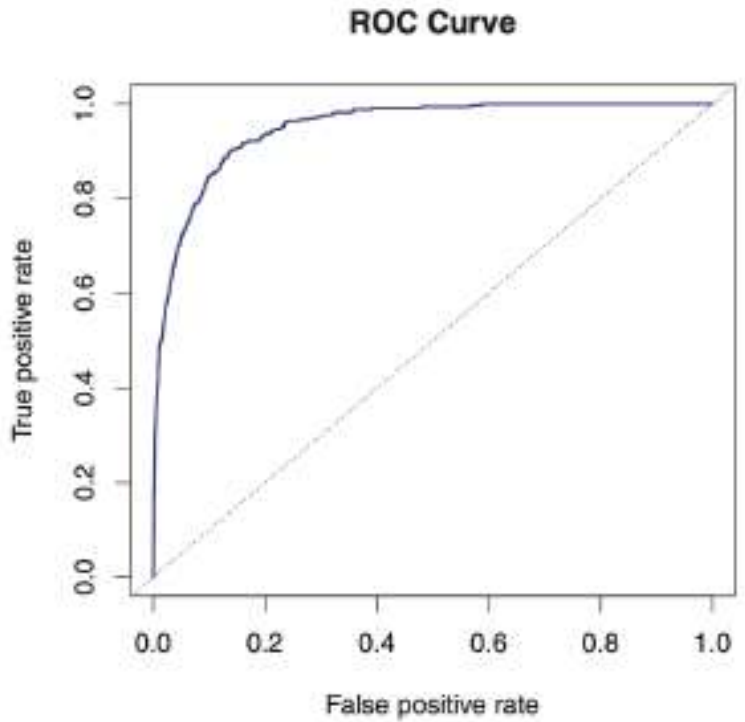
Matrice de confusion

- Un moyen de juger la performance d'un classifier est de faire sa **courbe ROC (ROC curve)**.
- C'est la courbe du "true positive rate / recall" en fonction du "false positive rate" paramétrée par le seuil θ (θ varie de 1 à 0).



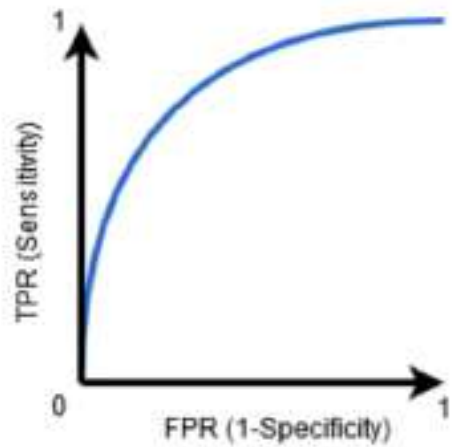
Matrice de confusion

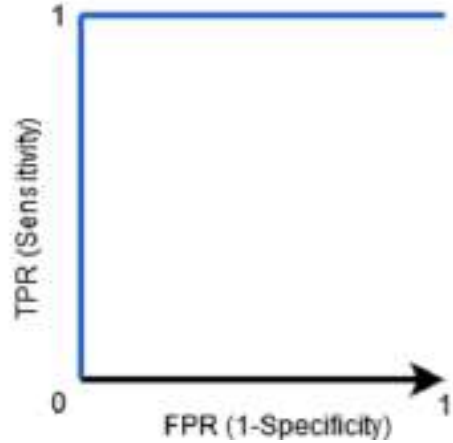
- Un moyen de juger la performance d'un classifier est de faire sa **courbe ROC (ROC curve)**.
- C'est la courbe du "true positive rate / recall" en fonction du "false positive rate" paramétrée par le seuil θ (θ varie de 1 à 0).



Matrice de confusion

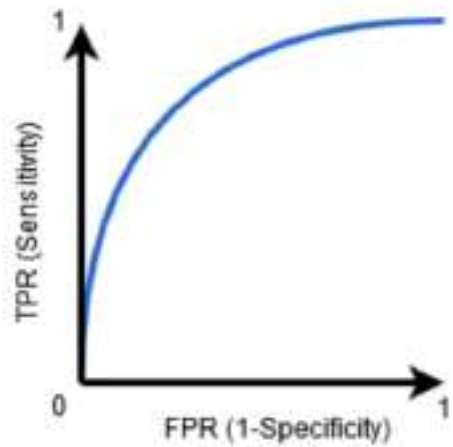
- Pour $\theta = 1$: Tout est classifié "négatif". Ainsi, $FPR = \frac{FP}{N} = 0$ et $TPR = \frac{TP}{P} = 0$. C'est le point $(0, 0)$.
- Pour $\theta = 0$: Tout est classifié "positif". Ainsi, $FPR = \frac{FP}{N} = \frac{N}{N} = 1$ et $TPR = \frac{TP}{P} = \frac{P}{P} = 1$. C'est le point $(1, 1)$.
- Le point $(0, 1)$ représente le classifieur parfait: $FPR = \frac{FP}{N} = 0 \Rightarrow FP = 0$ et $TPR = \frac{TP}{P} = 1 \Rightarrow TP = P$.

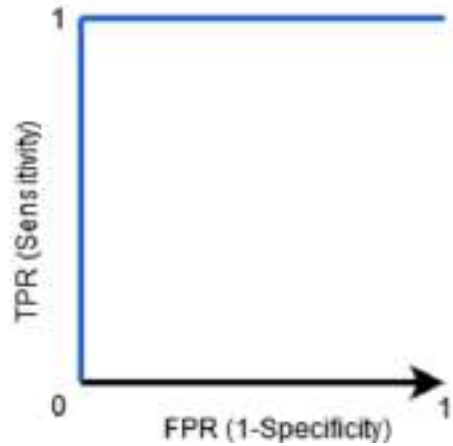




Matrice de confusion

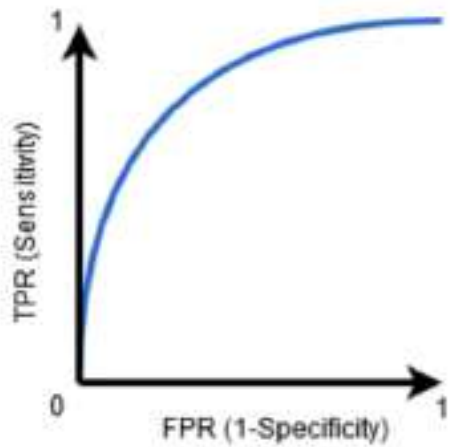
- Pour $\theta = 1$: Tout est classifié "négatif". Ainsi, $FPR = \frac{FP}{N} = 0$ et $TPR = \frac{TP}{P} = 0$. C'est le point $(0, 0)$.
- Pour $\theta = 0$: Tout est classifié "positif". Ainsi, $FPR = \frac{FP}{N} = \frac{N}{N} = 1$ et $TPR = \frac{TP}{P} = \frac{P}{P} = 1$. C'est le point $(1, 1)$.
- Le point $(0, 1)$ représente le classifieur parfait: $FPR = \frac{FP}{N} = 0 \Rightarrow FP = 0$ et $TPR = \frac{TP}{P} = 1 \Rightarrow TP = P$.

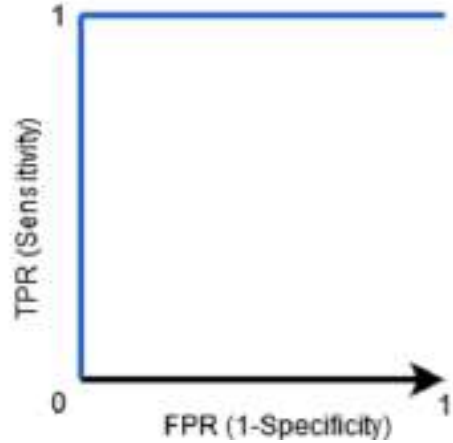




Matrice de confusion

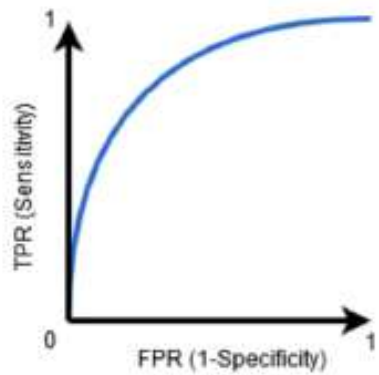
- Pour $\theta = 1$: Tout est classifié "négatif". Ainsi, $FPR = \frac{FP}{N} = 0$ et $TPR = \frac{TP}{P} = 0$. C'est le point $(0, 0)$.
- Pour $\theta = 0$: Tout est classifié "positif". Ainsi, $FPR = \frac{FP}{N} = \frac{N}{N} = 1$ et $TPR = \frac{TP}{P} = \frac{P}{P} = 1$. C'est le point $(1, 1)$.
- Le point $(0, 1)$ représente le classifieur parfait: $FPR = \frac{FP}{N} = 0 \Rightarrow FP = 0$ et $TPR = \frac{TP}{P} = 1 \Rightarrow TP = P$.

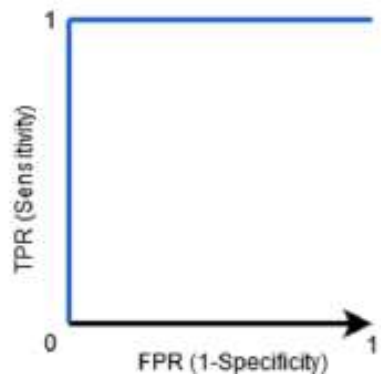




Matrice de confusion

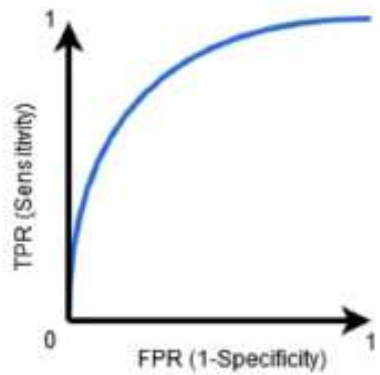
- Ainsi, on peut évaluer la performance d'un classifier en mesurant de combien il s'écarte du classifieur parfait.
- Pour cela, on mesure l'aire sous la courbe ROC, appelée **area under curve (AUC)**.
- Si $AUC = 1$, on est dans le cas du classifieur parfait. Si $AUC = 0.5$, c'est un classifieur random. Si $AUC = 0$, on a un classifieur qui classifie tout à l'envers (donc "anti-parfait").

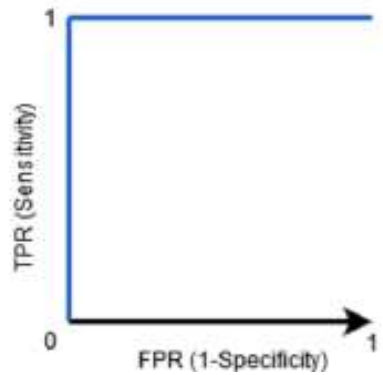




Matrice de confusion

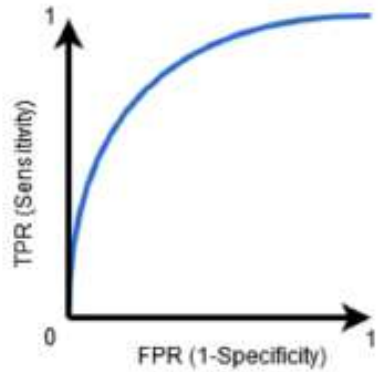
- Ainsi, on peut évaluer la performance d'un classifier en mesurant de combien il s'écarte du classifieur parfait.
- Pour cela, on mesure l'aire sous la courbe ROC, appelée **area under curve (AUC).**
- Si $AUC = 1$, on est dans le cas du classifieur parfait. Si $AUC = 0.5$, c'est un classifieur random. Si $AUC = 0$, on a un classifieur qui classifie tout à l'envers (donc "anti-parfait").

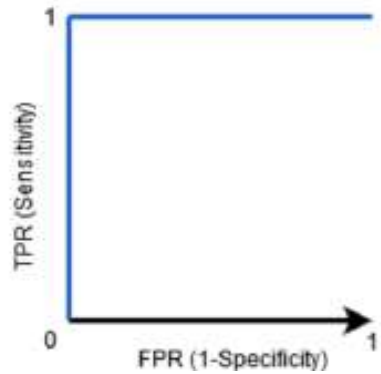




Matrice de confusion

- Ainsi, on peut évaluer la performance d'un classifier en mesurant de combien il s'écarte du classifieur parfait.
- Pour cela, on mesure l'aire sous la courbe ROC, appelée **area under curve (AUC).**
- Si $AUC = 1$, on est dans le cas du classifieur parfait. Si $AUC = 0.5$, c'est un classifieur random. Si $AUC = 0$, on a un classifieur qui classifie tout à l'envers (donc "anti-parfait").





Bibliographie

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013).
An Introduction to Statistical Learning: with Applications in R, volume 103 of *Springer Texts in Statistics*.
Springer, New York.