

TD 1 — Deux gaussiennes : erreur de Bayes, seuils et métriques

Séance 1

Christophe Ambroise

2026-09-29

Objectifs

Calculer une erreur de Bayes à la main sur un modèle où tout est connu, puis retrouver les mêmes quantités sur un échantillon simulé. Comparer les deux.

i Déroulé

Ex. 1	Probabilité a posteriori et seuil de décision	15 min
Ex. 2	Seuil optimal, risque de Bayes, coûts asymétriques	20 min
Ex. 3	Simuler l'échantillon et voir les densités	15 min
Ex. 4	Matrices de confusion aux trois seuils	15 min
Ex. 5	Courbe ROC et AUC	15 min
Ex. 6	Score F_1 et déséquilibre des classes	10 min

Tout le TD porte sur le même modèle, celui du cours :

$$X \mid Y = 0 \sim \mathcal{N}(\mu_0, \sigma^2), \quad X \mid Y = 1 \sim \mathcal{N}(\mu_1, \sigma^2), \quad P(Y = 1) = \pi_1,$$

avec $\mu_0 = -1$, $\mu_1 = 1,5$, $\sigma = 1$ et $\pi_1 = 0,35$. La classe 1 est celle des **positifs** — les malades d'un dépistage. Une seule variable, des variances égales : tout se calcule en forme close.

Partie théorique

Exercice 1 — Probabilité a posteriori et seuil de décision

1. Écrire $P(Y = 1 \mid X = x)$ à l'aide de π_0 , π_1 , f_0 et f_1 .
2. Montrer que cette probabilité s'écrit $\sigma(ax + b)$, où $\sigma(u) = 1/(1 + e^{-u})$, et donner a et b .
3. En déduire que décider « positif » lorsque $P(Y = 1 \mid X = x) > s$ revient à décider « positif » lorsque $x > t$, pour un seuil t que l'on exprimera en fonction de s . Que deviendrait cette équivalence si $\mu_1 < \mu_0$?

💡 Correction

1. Par la règle de Bayes, le dénominateur ne dépendant pas de la classe :

$$P(Y = 1 | X = x) = \frac{\pi_1 f_1(x)}{\pi_0 f_0(x) + \pi_1 f_1(x)} = \frac{1}{1 + \frac{\pi_0 f_0(x)}{\pi_1 f_1(x)}}.$$

2. Les variances étant égales, les termes en x^2 s'annulent dans le rapport :

$$\frac{\pi_1 f_1(x)}{\pi_0 f_0(x)} = \frac{\pi_1}{\pi_0} \exp\left(\frac{(\mu_1 - \mu_0)x}{\sigma^2} - \frac{\mu_1^2 - \mu_0^2}{2\sigma^2}\right),$$

d'où $P(Y = 1 | X = x) = \sigma(ax + b)$ avec

$$a = \frac{\mu_1 - \mu_0}{\sigma^2} = 2,5, \quad b = -\frac{\mu_1^2 - \mu_0^2}{2\sigma^2} + \ln \frac{\pi_1}{\pi_0} = -0,625 - 0,619 = -1,244.$$

C'est la forme d'une régression logistique, avec des coefficients déduits du modèle génératif et non estimés.

3. σ est strictement croissante, donc pour $a > 0$:

$$\sigma(ax + b) > s \iff ax + b > \ln \frac{s}{1-s} \iff x > t = \frac{1}{a} \left(\ln \frac{s}{1-s} - b \right).$$

Si $a < 0$ (c'est-à-dire $\mu_1 < \mu_0$), le sens de l'inégalité s'inverse : on déciderait « positif » pour les **petites** valeurs de x .

Un seuil sur le score et un seuil sur x sont donc interchangeables : la suite travaille directement sur x .

Exercice 2 — Seuil optimal, risque de Bayes, coûts asymétriques

1. Exprimer le risque sous coût 0-1 d'une règle « décider 1 si $x > t$ » :

$$R(t) = \pi_0 P(X > t | Y = 0) + \pi_1 P(X \leq t | Y = 1).$$

L'écrire avec la fonction de répartition Φ de la loi normale centrée réduite.

2. Minimiser $R(t)$ et retrouver le seuil de Bayes t^* . Le calculer numériquement.
3. Un faux négatif coûte c fois un faux positif. Reprendre le calcul et donner t_c . Dans quel sens la frontière se déplace-t-elle quand c croît ?
4. Calculer $R(t)$ pour trois seuils : t^* , le seuil $t = 0,25$ obtenu en ignorant les proportions a priori, et t_3 (faux négatif trois fois plus coûteux).

💡 Correction

1. Avec Φ la fonction de répartition de $\mathcal{N}(0, 1)$:

$$R(t) = \pi_0 \left[1 - \Phi\left(\frac{t - \mu_0}{\sigma}\right) \right] + \pi_1 \Phi\left(\frac{t - \mu_1}{\sigma}\right).$$

Le premier terme compte les faux positifs, le second les faux négatifs.

2. En dérivant, $R'(t) = -\pi_0 f_0(t) + \pi_1 f_1(t)$, qui s'annule quand $\pi_0 f_0(t) = \pi_1 f_1(t)$ — c'est bien le point d'égalité de la figure du cours. En prenant le logarithme et en simplifiant les termes en t^2 :

$$t^* = \frac{\mu_0 + \mu_1}{2} + \frac{\sigma^2}{\mu_1 - \mu_0} \ln \frac{\pi_0}{\pi_1} = 0,25 + 0,4 \times \ln \frac{0,65}{0,35} = \mathbf{0,498}.$$

Les négatifs étant majoritaires, le seuil est décalé vers la classe rare.

3. Un faux négatif coûtant c , la décision « positif » devient optimale dès que $c \pi_1 f_1(x) > \pi_0 f_0(x)$, ce qui remplace π_1 par $c \pi_1$:

$$t_c = \frac{\mu_0 + \mu_1}{2} + \frac{\sigma^2}{\mu_1 - \mu_0} \ln \frac{\pi_0}{c \pi_1}.$$

Quand c croît, le seuil **descend**. Pour $c = 3$: $t_3 = 0,25 + 0,4 \times \ln(0,65/1,05) = \mathbf{0,058}$.

4. Calcul des trois seuils et de leur risque :

```
import numpy as np
from scipy.stats import norm

MU0, MU1, SIGMA, PI1 = -1.0, 1.5, 1.0, 0.35
PI0 = 1 - PI1

def seuil(c=1):
    return (MU0 + MU1) / 2 + SIGMA**2 / (MU1 - MU0) * np.log(PI0 / (c * PI1))

def risque(t):
    return PI0 * (1 - norm.cdf(t, MU0, SIGMA)) + PI1 * norm.cdf(t, MU1, SIGMA)

for t in [seuil(), 0.25, seuil(3)]:
    print(round(t, 3), round(risque(t), 4),
          1 - norm.cdf(t, MU1, SIGMA), 1 - norm.cdf(t, MU0, SIGMA))
```

```
mu0 <- -1.0; mu1 <- 1.5; sigma <- 1.0; pi1 <- 0.35; pi0 <- 1 - pi1

seuil <- function(c = 1) {
  (mu0 + mu1) / 2 + sigma^2 / (mu1 - mu0) * log(pi0 / (c * pi1))
}

risque <- function(t) {
  pi0 * (1 - pnorm(t, mu0, sigma)) + pi1 * pnorm(t, mu1, sigma)
}

for (t in c(seuil(), 0.25, seuil(3))) {
  cat(round(t, 3), round(risque(t), 4),
      1 - pnorm(t, mu1, sigma), 1 - pnorm(t, mu0, sigma), "\n")
}
```

Sous coût 0-1 :

Seuil	t	$R(t)$	TPR	FPR
t^* — optimal	0,498	0,0990	0,842	0,067
0,25 — a priori ignorés	0,250	0,1056	0,894	0,106
t_3 — faux négatif $\times 3$	0,058	0,1204	0,925	0,145

$R(t^*) = 9,9\%$ est le **risque de Bayes**. t_3 est sous-optimal pour le coût 0-1, mais optimal pour le coût qui l'a produit. Entre t^* et 0,25, l'écart n'est que de 0,7 point : R est plate près de son minimum.

Partie empirique

Exercice 3 — Simuler l'échantillon et voir les densités

Le code de simulation et d'affichage est fourni. L'exécuter, puis répondre.

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

MU0, MU1, SIGMA, PI1 = -1.0, 1.5, 1.0, 0.35
PI0 = 1 - PI1

rng = np.random.default_rng(0)
n = 2000
y = (rng.random(n) < PI1).astype(int)      # étiquettes
x = rng.normal(np.where(y == 1, MU1, MU0), SIGMA)  # observations

grille = np.linspace(-5, 6, 400)
plt.hist(x[y == 0], bins=40, density=True, alpha=0.25,
         color="#41597a", label="échantillon, classe 0")
plt.hist(x[y == 1], bins=40, density=True, alpha=0.25,
         color="#d96125", label="échantillon, classe 1")
plt.plot(grille, PI0 * norm.pdf(grille, MU0, SIGMA), color="#41597a", lw=2,
         label=r"$\pi_0 f_0$")
plt.plot(grille, PI1 * norm.pdf(grille, MU1, SIGMA), color="#d96125", lw=2,
         label=r"$\pi_1 f_1$")
plt.axvline(0.498, color="black", ls="--", label="seuil $t^*$")
plt.legend(); plt.xlabel("x"); plt.ylabel("densité")
plt.show()
```

```
mu0 <- -1.0; mu1 <- 1.5; sigma <- 1.0; pi1 <- 0.35; pi0 <- 1 - pi1

set.seed(0)
n <- 2000
y <- rbinom(n, 1, pi1)      # étiquettes
x <- rnorm(n, ifelse(y == 1, mu1, mu0), sigma)  # observations

grille <- seq(-5, 6, length.out = 400)
hist(x[y == 0], breaks = 40, freq = FALSE, col = "#41597a40", border = NA,
     xlim = c(-5, 6), xlab = "x", ylab = "densité", main = "")
hist(x[y == 1], breaks = 40, freq = FALSE, col = "#d9612540", border = NA,
     add = TRUE)
lines(grille, pi0 * dnorm(grille, mu0, sigma), col = "#41597a", lwd = 2)
lines(grille, pi1 * dnorm(grille, mu1, sigma), col = "#d96125", lwd = 2)
abline(v = 0.498, lty = 2)
legend("topright", bty = "n", lwd = 2, col = c("#41597a", "#d96125"),
     legend = c(expression(pi[0] * f[0]), expression(pi[1] * f[1])))
```

1. Les histogrammes sont normalisés séparément par classe ; les courbes tracées sont $\pi_k f_k$. Pourquoi les deux ne se superposent-ils pas ? Comment corriger l'affichage pour qu'ils coïncident ?
2. Vérifier sur l'échantillon la proportion de positifs. À quelle distance de $\pi_1 = 0,35$ est-elle ? Cet écart est-il compatible avec $n = 2000$?

3. Repérer sur la figure la zone qui correspond au risque de Bayes.

💡 Correction

1. `density=True` (`freq = FALSE` en R) normalise **chaque** histogramme à une aire de 1 : il estime f_k , pas $\pi_k f_k$. Il faut que l'aire de l'histogramme de la classe k vaille $\hat{\pi}_k$, pas 1.

Attention, remettre simplement `weights` avec `density=True` ne suffit pas : NumPy renormalise à une aire de 1 même en présence de poids. Il faut `density=False` et des poids qui portent aussi la largeur des classes :

```
bords = np.linspace(-5, 6, 41)
largeur = bords[1] - bords[0]
for classe, couleur in ((0, "#41597a"), (1, "#d96125")):
    xs = x[y == classe]
    plt.hist(xs, bins=bords, alpha=0.25, color=couleur,
            weights=np.full(len(xs), 1 / (len(x) * largeur)))
```

```
bords <- seq(-5, 6, length.out = 41)
plot(NULL, xlim = c(-5, 6), ylim = c(0, 0.3),
     xlab = "x", ylab = "densité pondérée")
for (k in 0:1) {
  h <- hist(x[y == k], breaks = bords, plot = FALSE)
  rect(head(bords, -1), 0, tail(bords, -1), h$density * mean(y == k),
      col = c("#41597a40", "#d9612540")[k + 1], border = NA)
}
```

Vérification : l'aire vaut alors 0,648 pour la classe 0 et 0,352 pour la classe 1, et les sommets des histogrammes (0,267 et 0,153) rejoignent ceux des courbes $\pi_k f_k$ (0,259 et 0,140).

2. `y.mean()` en Python, `mean(y)` en R. Python : 704 positifs sur 2000, $\hat{\pi}_1 = 0,352$; R : 699, $\hat{\pi}_1 = 0,3495$. L'écart-type de la proportion vaut $\sqrt{0,35 \times 0,65/2000} = 0,011$: les deux écarts sont inférieurs à un cinquième d'écart-type.

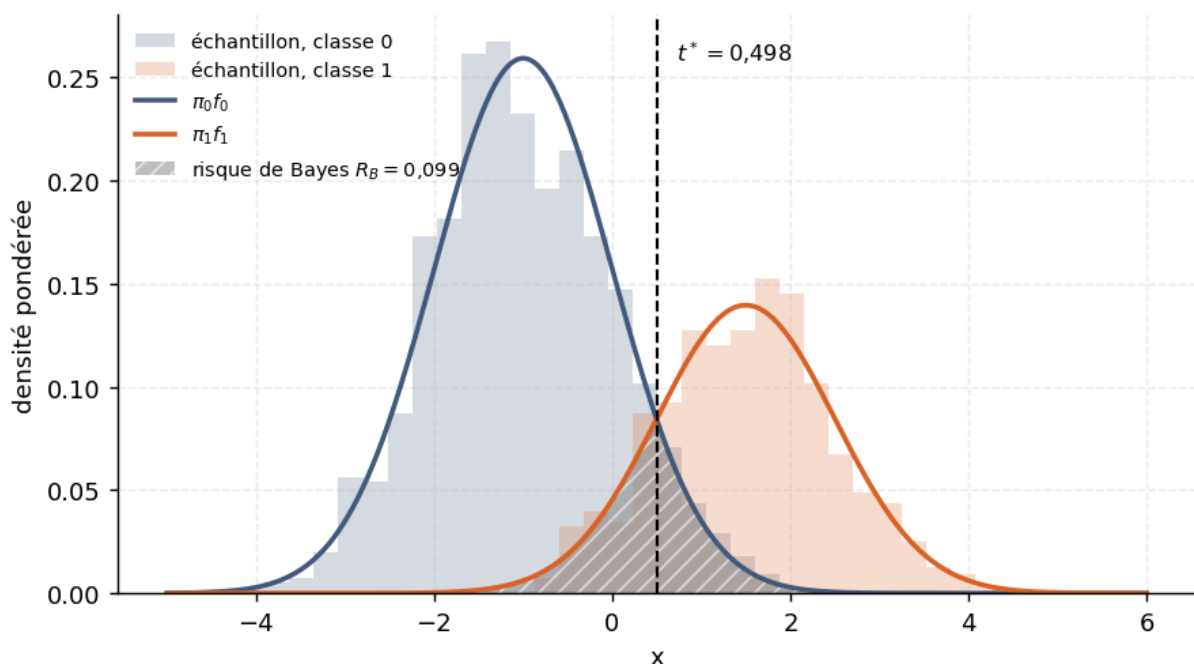


FIGURE 1 – Histogrammes pondérés par $\hat{\pi}_k$ et densités $\pi_k f_k$. La zone hachurée est le risque de Bayes.

3. La zone où la courbe la plus basse passe sous l'autre, c'est-à-dire l'aire sous $\min\{\pi_0 f_0(x), \pi_1 f_1(x)\}$. Elle vaut $R_B = 0,099$ et se situe de part et d'autre du seuil t^* : à droite de t^* , la queue de $\pi_0 f_0$ (faux positifs) ; à gauche, celle de $\pi_1 f_1$ (faux négatifs).

Exercice 4 — Matrices de confusion aux trois seuils

Pour chacun des trois seuils de l'exercice 2, construire la matrice de confusion sur l'échantillon simulé, puis calculer l'erreur, la précision et le rappel.

1. Écrire le code (une décision est simplement $x > t$).
2. Comparer l'erreur empirique au $R(t)$ théorique, seuil par seuil.
3. Que devient la matrice de confusion du classifieur constant « toujours négatif » ? Quelle est son exactitude ? Son rappel ?

💡 Correction

```
from sklearn.metrics import confusion_matrix, precision_score, recall_score

for nom, t in [("t*", 0.498), ("0,25", 0.250), ("t_3", 0.058)]:
    pred = (x > t).astype(int)
    vn, fp, fn, vp = confusion_matrix(y, pred).ravel()
    print(nom, vp, fp, fn, vn, np.mean(pred != y),
          precision_score(y, pred), recall_score(y, pred))
```

```
for (t in c(0.498, 0.250, 0.058)) {
  pred <- as.integer(x > t)
  vp <- sum(pred == 1 & y == 1); fp <- sum(pred == 1 & y == 0)
  fn <- sum(pred == 0 & y == 1); vn <- sum(pred == 0 & y == 0)
  cat(t, vp, fp, fn, vn, mean(pred != y),
      vp / (vp + fp), vp / (vp + fn), "\n") # erreur, précision, rappel
}
```

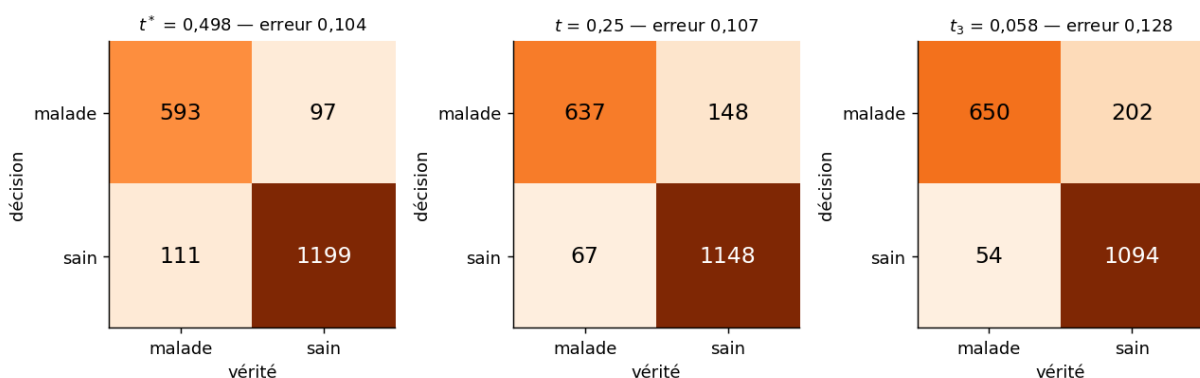


FIGURE 2 — Matrices de confusion aux trois seuils. Baisser le seuil déplace les effectifs de la diagonale FN vers la colonne des alertes.

2. Résultats (Python, graine 0, $n = 2000$) :

Seuil	VP	FP	FN	VN	erreur	$R(t)$	précision	rappel
t^*	593	97	111	1199	0,1040	0,0990	0,859	0,842

0,25	637	148	67	1148	0,1075	0,1056	0,811	0,905
t_3	650	202	54	1094	0,1280	0,1204	0,763	0,923

Écarts de quelques millièmes, de l'ordre de l'incertitude de Monte-Carlo $\sqrt{R(1-R)/n} \approx 0,007$. De t^* à t_3 , le rappel passe de 0,842 à 0,923 (57 malades détectés en plus) pour 105 fausses alertes supplémentaires.

R tire un autre échantillon : effectifs voisins, non identiques (à t^* : VP = 595, FP = 90, FN = 104, VN = 1211, erreur 0,0970).

3.

```
pred = np.zeros_like(y)
print(np.mean(pred == y), recall_score(y, pred, zero_division=0))
```

```
pred <- rep(0L, length(y))
c(mean(pred == y), sum(pred == 1 & y == 1) / sum(y == 1))
```

VP = 0, FP = 0, FN = 704, VN = 1296 : exactitude **64,8 %**, rappel **0** (R : 65,05 %). C'est la référence minimale à battre.

Exercice 5 — Courbe ROC et AUC

1. Donner les expressions théoriques de $\text{TPR}(t)$ et $\text{FPR}(t)$, et tracer la courbe ROC théorique en faisant varier t .
2. Construire la courbe ROC empirique à partir de l'échantillon, et la superposer à la théorique.
3. Montrer que l'AUC théorique vaut $\Phi\left(\frac{\mu_1 - \mu_0}{\sigma\sqrt{2}}\right)$, et la comparer à l'AUC empirique.
4. Placer les trois seuils de l'exercice 2 sur la courbe. Où se trouve le seuil optimal ? Est-ce le point le plus proche du coin supérieur gauche ?

💡 Correction

1. $\text{TPR}(t) = P(X > t \mid Y = 1) = 1 - \Phi\left(\frac{t - \mu_1}{\sigma}\right)$ et $\text{FPR}(t) = 1 - \Phi\left(\frac{t - \mu_0}{\sigma}\right)$. Faire varier t de $+\infty$ à $-\infty$ parcourt la courbe de (0,0) à (1,1).

```
from scipy.stats import norm
from sklearn.metrics import roc_curve, roc_auc_score

t = np.linspace(6, -5, 400)
plt.plot(1 - norm.cdf(t, MU0, SIGMA), 1 - norm.cdf(t, MU1, SIGMA),
         lw=2, label="ROC théorique")
fpr, tpr, _ = roc_curve(y, x)
plt.step(fpr, tpr, where="post", label="ROC empirique")
plt.plot([0, 1], [0, 1], ls=":", color="grey")
plt.xlabel("FPR"); plt.ylabel("TPR"); plt.legend(); plt.show()
print(roc_auc_score(y, x))
```

```

t <- seq(6, -5, length.out = 400)
plot(1 - pnorm(t, mu0, sigma), 1 - pnorm(t, mu1, sigma), type = "l", lwd = 2,
     xlab = "FPR", ylab = "TPR")
seuils <- c(Inf, sort(unique(x), decreasing = TRUE), -Inf)
tpr <- sapply(seuils, function(s) sum(x > s & y == 1) / sum(y == 1))
fpr <- sapply(seuils, function(s) sum(x > s & y == 0) / sum(y == 0))
lines(fpr, tpr, col = "#d96125", type = "s")
abline(0, 1, lty = 3)
sum(diff(fpr) * (head(tpr, -1) + tail(tpr, -1)) / 2) # AUC, trapèzes

```

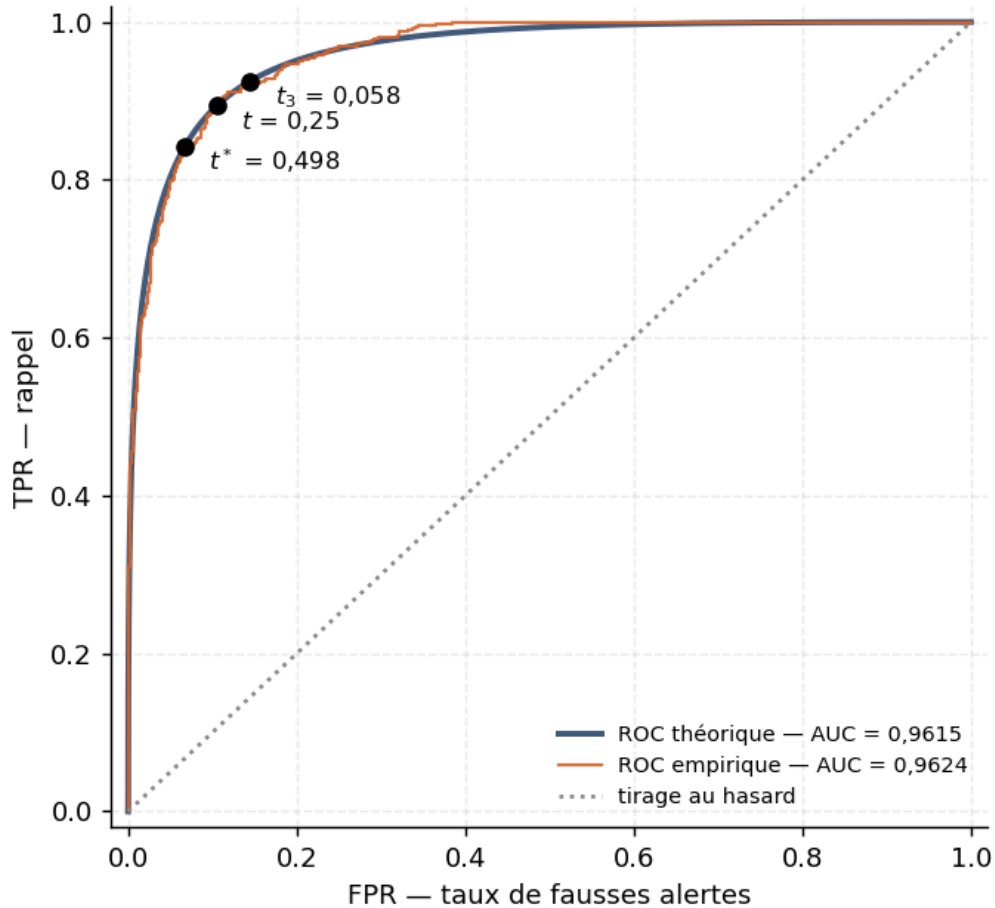


FIGURE 3 – ROC théorique et ROC empirique. Les trois seuils du sujet sont marqués; t^* n'est pas le point le plus proche du coin supérieur gauche.

2. La courbe empirique est en escalier, un palier par observation, et suit la théorique à $O(1/\sqrt{n})$ près.

3. Sous $Y = 1$, $X_1 \sim \mathcal{N}(\mu_1, \sigma^2)$; sous $Y = 0$, $X_0 \sim \mathcal{N}(\mu_0, \sigma^2)$, indépendantes. L'AUC est la probabilité qu'un positif reçoive un score supérieur à celui d'un négatif :

$$\text{AUC} = P(X_1 > X_0) = P(X_1 - X_0 > 0), \quad X_1 - X_0 \sim \mathcal{N}(\mu_1 - \mu_0, 2\sigma^2),$$

d'où $\text{AUC} = \Phi\left(\frac{\mu_1 - \mu_0}{\sigma\sqrt{2}}\right) = \Phi(1,768) = \mathbf{0,9615}$. L'échantillon donne **0,9624** (Python) et 0,9649 (R, autre tirage).

4. Les trois points, du haut vers le bas de la courbe : $t_3 \mapsto (0,145; 0,925)$, $0,25 \mapsto (0,106; 0,894)$, $t^* \mapsto (0,067; 0,842)$.

Non : la distance au coin supérieur gauche suppose des erreurs de même coût et des classes

équilibrées. Sous coût 0-1, t^* est le point où la pente de la ROC vaut $\pi_0/(c\pi_1) = 0,65/0,35 = 1,86$. La ROC seule ne désigne aucun seuil.

Exercice 6 — Score F_1 et déséquilibre des classes

1. Écrire la précision au niveau de la population : $\text{précision}(t) = \frac{\pi_1 \text{TPR}(t)}{\pi_1 \text{TPR}(t) + \pi_0 \text{FPR}(t)}$, puis $F_1(t)$.
2. Tracer $F_1(t)$ et l'erreur $R(t)$ sur le même axe des seuils. Les deux sont-elles optimales au même endroit ?
3. Refaire le calcul avec $\pi_1 = 0,02$, les moyennes inchangées. Que deviennent l'AUC, la précision au seuil t^* , et le F_1 ?
4. Conclure : quand publier une ROC, quand publier une précision-rappel ?

💡 Correction

1. La précision mélange les deux classes à son dénominateur, donc dépend de π_1 — contrairement à TPR et FPR. Avec $F_1 = 2 \text{précision} \times \text{rappel} / (\text{précision} + \text{rappel})$ et $\text{rappel} = \text{TPR}$:

```
from sklearn.metrics import f1_score

def f1(t, pi1):
    tpr = 1 - norm.cdf(t, MU1, SIGMA); fpr = 1 - norm.cdf(t, MU0, SIGMA)
    prec = pi1 * tpr / (pi1 * tpr + (1 - pi1) * fpr)
    return prec, 2 * prec * tpr / (prec + tpr)

for t in [0.498, 0.25, 0.058]:
    print(t, f1(t, PI1)[1], f1_score(y, x > t))
```

```
f1 <- function(t, pi1) {
  tpr <- 1 - pnorm(t, mu1, sigma); fpr <- 1 - pnorm(t, mu0, sigma)
  prec <- pi1 * tpr / (pi1 * tpr + (1 - pi1) * fpr)
  c(precision = prec, F1 = 2 * prec * tpr / (prec + tpr))
}
for (t in c(0.498, 0.250, 0.058)) {
  pred <- as.integer(x > t); vp <- sum(pred == 1 & y == 1)
  cat(t, f1(t, pi1)["F1"], 2 * vp / (sum(pred) + sum(y)), "\n")
}
```

F_1 empirique en Python ; R donne 0,860, 0,857 et 0,844.

Seuil	t	$R(t)$	F_1 théorique	F_1 empirique
t^*	0,498	0,0990	0,856	0,851
0,25	0,250	0,1056	0,856	0,856
t_3	0,058	0,1204	0,843	0,835

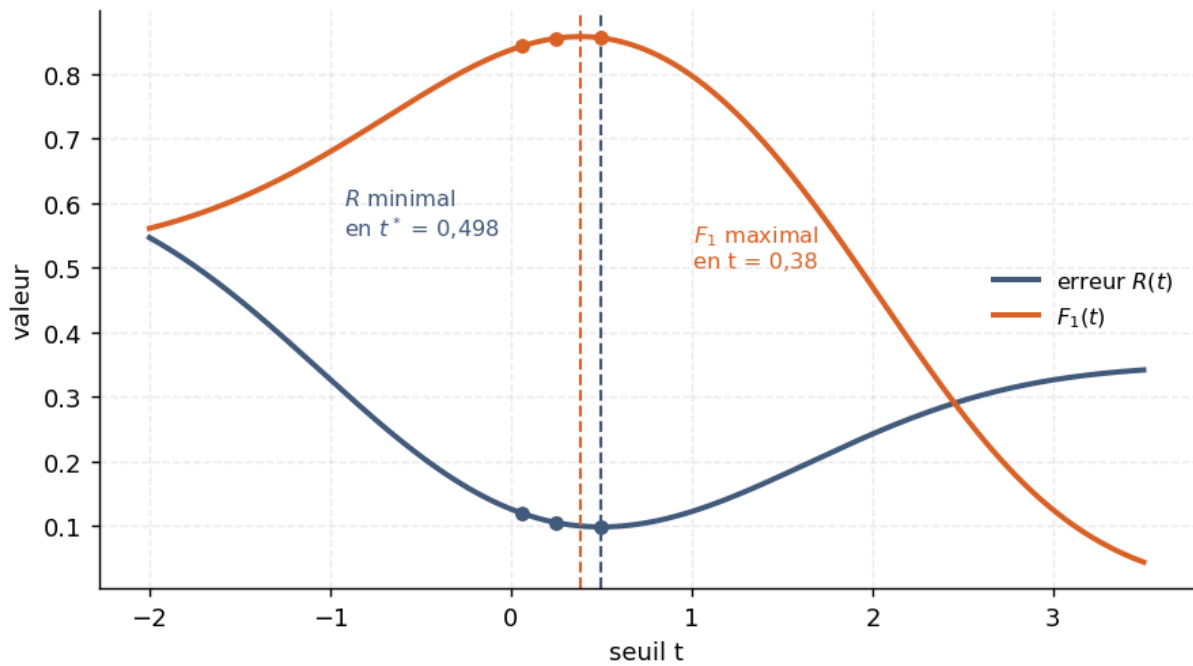


FIGURE 4 – Erreur $R(t)$ et score $F_1(t)$ en fonction du seuil. Les deux optima ne coïncident pas : F_1 ignore les vrais négatifs, R les compte.

2. Non. R est minimale en $t^* = 0,498$, F_1 maximal en $t \approx 0,38$ ($F_1 = 0,8582$) : F_1 ignore les vrais négatifs, R les compte.

```
grille = np.linspace(-2, 3.5, 5001)
print(grille[np.argmax(f1(grille, PI1)[1])])
```

```
grille <- seq(-2, 3.5, length.out = 5001)
grille[which.max(sapply(grille, function(t) f1(t, pi1)["F1"]))]
```

3. À seuil fixé, seule la prévalence change :

```
for pi1 in [0.35, 0.02]:
    print(pi1, f1(0.498, pi1)) # précision, F1
t02 = (MU0 + MU1) / 2 + SIGMA**2 / (MU1 - MU0) * np.log(0.98 / 0.02)
print(t02, 1 - norm.cdf(t02, MU1, SIGMA), f1(t02, 0.02))
```

```
sapply(c(0.35, 0.02), function(p) f1(0.498, p))
t02 <- (mu0 + mu1) / 2 + sigma^2 / (mu1 - mu0) * log(0.98 / 0.02)
c(t02, 1 - pnorm(t02, mu1, sigma), f1(t02, 0.02))
```

	$\pi_1 = 0,35$	$\pi_1 = 0,02$
AUC	0,9615	0,9615
au seuil $t = 0,498$: TPR / FPR	0,842 / 0,067	0,842 / 0,067
au seuil $t = 0,498$: précision	0,871	0,204
au seuil $t = 0,498$: F_1	0,856	0,328

AUC, TPR et FPR ne bougent pas : leurs dénominateurs viennent d'une seule classe. La précision mélange les deux classes et tombe à 0,204 : quatre alertes sur cinq sont fausses. Seuil optimal pour $\pi_1 = 0,02$: $t^* = 1,81$, erreur 1,5 %, rappel 0,380, $F_1 = 0,505$.

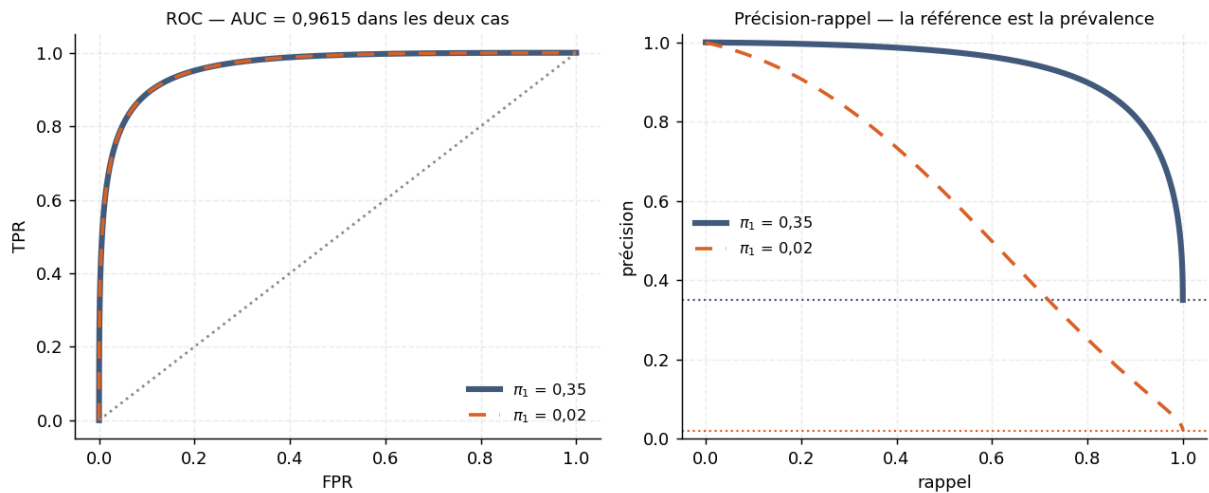


FIGURE 5 – Même modèle, deux prévalences. L'AUC est identique ; la courbe précision-rappel, elle, s'effondre quand les positifs se raréfient.

4. Classes équilibrées ou comparaison de modèles : la ROC, indépendante de la prévalence. Positifs rares : la courbe précision-rappel. Sur 1 000 personnes dont 20 malades, une FPR de 5 % donne 49 fausses alertes pour au plus 20 vrais positifs.

Remarque finale

! À retenir

Sur ce modèle, seuil optimal, risque de Bayes, ROC et AUC se calculent. L'échantillon les retrouve à $1/\sqrt{n}$ près. Sur données réelles, seule la colonne « empirique » existe.

Les deux scripts exécutables correspondants sont `python/td01_deux_gaussiennes.py` et `r/td01_deux_gaussiennes.R`.