

TD 2 — Erreur de Bayes gaussienne, Bayes naïf, LDA et QDA

Séance 2

Christophe Ambroise

2026-09-29

Objectifs

Calculer l'erreur de Bayes d'un modèle gaussien à deux classes dans le plan, où tout est explicite, puis estimer par validation croisée l'erreur de Bayes naïf, de LDA et de QDA sur Iris. Confronter ce que la théorie garantit à ce que l'échantillon permet d'atteindre.

i Déroulé

Ex. 1	Règle de Bayes et frontière linéaire	20 min
Ex. 2	Erreur de Bayes et distance de Mahalanobis	25 min
Ex. 3	Ce qu'apporte la seconde variable	10 min
Ex. 4	Bayes naïf, LDA et QDA sur Iris	20 min
Ex. 5	Variables de bruit : théorie contre pratique	15 min

Le modèle de la partie théorique :

$$X | Y = 0 \sim \mathcal{N}_2(\mu_0, I), \quad X | Y = 1 \sim \mathcal{N}_2(\mu_1, I), \quad \mu_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \mu_1 = \begin{pmatrix} 1 \\ \varepsilon \end{pmatrix}.$$

Les deux classes sont supposées **équiprobables** ($\pi_0 = \pi_1 = 1/2$), sauf mention contraire. La première variable sépare toujours les deux classes d'une unité ; la seconde n'apporte quelque chose que si $\varepsilon \neq 0$.

Partie théorique

Exercice 1 — Règle de Bayes et frontière linéaire

1. Écrire $\delta_k(x) = \log \pi_k + \log f_k(x)$ pour ce modèle, et simplifier.
2. Montrer que la règle de Bayes décide « classe 1 » lorsque $w^\top x > c$, et donner w et c explicitement en fonction de $\mu_0, \mu_1, \pi_0, \pi_1$.
3. Quelle est la direction de w ici ? Tracer la frontière pour $\varepsilon = 0$ puis $\varepsilon = 1$, sur le carré $[-3, 4]^2$.
4. Pourquoi la frontière est-elle une **droite** et non une conique ?

Correction

1. Avec $\Sigma = I$, $\log f_k(x) = -\frac{1}{2}\|x - \mu_k\|^2 - \log(2\pi)$. Le terme $-\log(2\pi)$ ne dépend pas de k et disparaît :

$$\delta_k(x) = \log \pi_k - \frac{1}{2}\|x\|^2 + x^\top \mu_k - \frac{1}{2}\|\mu_k\|^2.$$

2. Le terme $-\frac{1}{2}\|x\|^2$ est **commun aux deux classes** : il s'élimine de la différence. Il reste

$$\delta_1(x) - \delta_0(x) = x^\top \underbrace{(\mu_1 - \mu_0)}_w - \underbrace{\left[\frac{1}{2}(\|\mu_1\|^2 - \|\mu_0\|^2) + \log \frac{\pi_0}{\pi_1}\right]}_c,$$

soit, avec $\mu_0 = 0$ et $\pi_0 = \pi_1$:

$$w = \begin{pmatrix} 1 \\ \varepsilon \end{pmatrix}, \quad c = \frac{1}{2}\|\mu_1\|^2 = \frac{1+\varepsilon^2}{2}.$$

En général $w = \Sigma^{-1}(\mu_1 - \mu_0)$; ici $\Sigma = I$ ne corrige rien, donc w est exactement l'écart des moyennes.

3. w pointe de μ_0 vers μ_1 . La frontière est la droite perpendiculaire à w passant par le milieu $\frac{1}{2}(\mu_0 + \mu_1)$ — c'est la médiatrice du segment des deux centres, puisque la distance de Mahalanobis se réduit ici à la distance euclidienne.

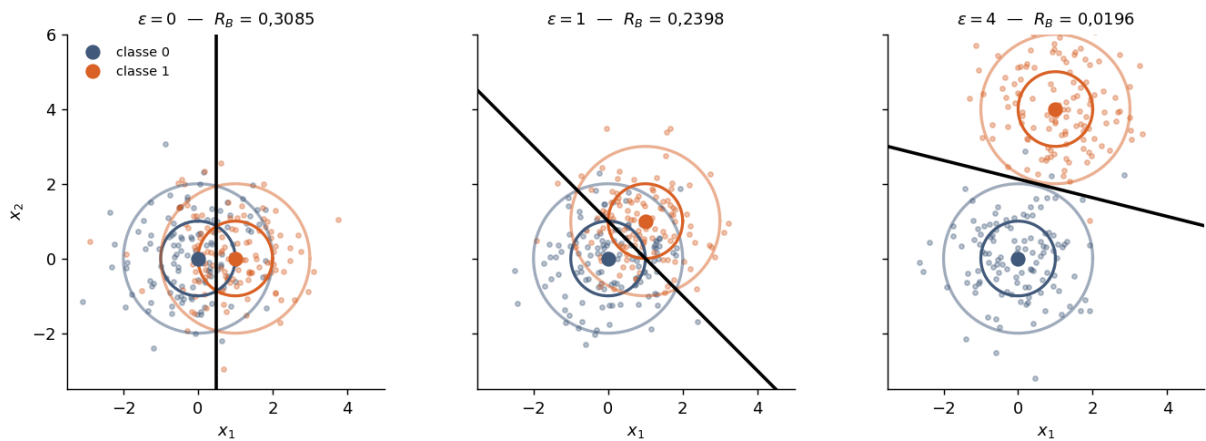


FIGURE 1 — Trois valeurs de ε . Ellipses d'isodensité à un et deux écarts-types, frontière de Bayes en noir. Elle pivote avec μ_1 mais reste une droite.

4. Parce que les deux covariances sont **égales**. Le terme quadratique $-\frac{1}{2}x^\top \Sigma_k^{-1}x$ est alors le même pour les deux classes et disparaît de la différence. C'est exactement l'hypothèse d'homoscédasticité de LDA.

Exercice 2 — Erreur de Bayes et distance de Mahalanobis

1. Poser $U = w^\top X$. Donner la loi de U sous chaque classe. Que vaut $\text{Var}(U)$?
2. En déduire les deux erreurs conditionnelles, puis montrer que, sous $\pi_0 = \pi_1 = 1/2$:

$$R_B = \Phi\left(-\frac{\Delta}{2}\right), \quad \Delta^2 = (\mu_1 - \mu_0)^\top \Sigma^{-1}(\mu_1 - \mu_0).$$

3. Que vaut Δ ici ? Calculer R_B pour $\varepsilon \in \{0; 0,5; 1; 2; 4\}$.
4. Reprendre le calcul avec des proportions quelconques $\pi_0 \neq \pi_1$. Vérifier la formule obtenue sur le TD 1 : une seule dimension, $\mu_0 = -1$, $\mu_1 = 1,5$, $\sigma = 1$, $\pi_1 = 0,35$.

Correction

1. $U = w^\top X$ est une combinaison linéaire d'un vecteur gaussien, donc gaussienne. Sous la classe k : $U \sim \mathcal{N}(w^\top \mu_k, w^\top \Sigma w)$. Or $w = \Sigma^{-1}(\mu_1 - \mu_0)$, donc

$$w^\top \Sigma w = (\mu_1 - \mu_0)^\top \Sigma^{-1}(\mu_1 - \mu_0) = \Delta^2, \quad w^\top \mu_1 - w^\top \mu_0 = \Delta^2.$$

Les deux moyennes sont séparées de Δ^2 , avec un écart-type Δ : en unités d'écart-type, la séparation vaut donc Δ . Le problème se ramène à la dimension 1 du TD 1.

2. D'où vient la règle « décider 1 si $U > c$ ». La règle de Bayes choisit la classe de plus grand score : elle décide 1 lorsque $\delta_1(x) > \delta_0(x)$. L'exercice 1 a montré que cette différence s'écrit $w^\top x - c$; la règle revient donc à **projeter** l'observation sur la direction w , puis à comparer la projection $U = w^\top x$ à un **seuil** c . Avec une covariance Σ quelconque (commune aux deux classes), le même calcul donne

$$w = \Sigma^{-1}(\mu_1 - \mu_0), \quad c = \frac{1}{2}(\mu_1 + \mu_0)^\top \Sigma^{-1}(\mu_1 - \mu_0) + \log \frac{\pi_0}{\pi_1} = \underbrace{\frac{1}{2}(w^\top \mu_0 + w^\top \mu_1)}_{\text{milieu des deux moyennes de } U} + \log \frac{\pi_0}{\pi_1}.$$

Le seuil est donc le **milieu** des deux moyennes projetées, décalé vers la classe la moins probable : si $\pi_1 < \pi_0$, le logarithme est positif, c monte, et il faut une projection plus grande pour oser dire « 1 ». Pour $\varepsilon = 0$: $w = (1, 0)^\top$, $U = x_1$, $c = 1/2$ — on décide 1 dès que x_1 dépasse le milieu entre 0 et 1.

Position du seuil par rapport à chaque moyenne. En soustrayant $w^\top \mu_0$, puis $w^\top \mu_1$, et en utilisant $w^\top \mu_1 - w^\top \mu_0 = \Delta^2$ (question 1) :

$$c - w^\top \mu_0 = \frac{\Delta^2}{2} + \log \frac{\pi_0}{\pi_1}, \quad c - w^\top \mu_1 = -\frac{\Delta^2}{2} + \log \frac{\pi_0}{\pi_1}.$$

Les deux erreurs. Sous la classe k , $Z = (U - w^\top \mu_k)/\Delta$ suit une $\mathcal{N}(0, 1)$. On divise donc chaque écart par l'écart-type Δ , et on utilise $P(Z > a) = \Phi(-a)$:

$$\begin{aligned} \text{FPR} &= P(U > c \mid Y = 0) = P\left(Z > \frac{c - w^\top \mu_0}{\Delta}\right) = \Phi\left(-\frac{\Delta}{2} - \frac{1}{\Delta} \log \frac{\pi_0}{\pi_1}\right), \\ \text{FNR} &= P(U \leq c \mid Y = 1) = P\left(Z \leq \frac{c - w^\top \mu_1}{\Delta}\right) = \Phi\left(-\frac{\Delta}{2} + \frac{1}{\Delta} \log \frac{\pi_0}{\pi_1}\right). \end{aligned}$$

L'erreur de Bayes est la moyenne des deux, pondérée par les proportions : $R_B = \pi_0 \text{FPR} + \pi_1 \text{FNR}$. Sous $\pi_0 = \pi_1$ le logarithme s'annule, le seuil tombe exactement au milieu, à $\Delta/2$ écart-type de chaque moyenne : les deux erreurs sont égales, et $R_B = \frac{1}{2}\Phi(-\Delta/2) + \frac{1}{2}\Phi(-\Delta/2) = \Phi(-\Delta/2)$.

Note

L'erreur de Bayes ne dépend des paramètres **qu'à travers** Δ . Deux problèmes gaussiens homoscedastiques de même distance de Mahalanobis entre centres sont exactement aussi difficiles l'un que l'autre, quelles que soient la dimension et l'orientation.

3. Ici $\Sigma = I$, donc $\Delta = \|\mu_1 - \mu_0\| = \sqrt{1 + \varepsilon^2}$.

ε	Δ	$R_B = \Phi(-\Delta/2)$
0	1,000	0,3085
0,5	1,118	0,2881
1	1,414	0,2398
2	2,236	0,1318
4	4,123	0,0196

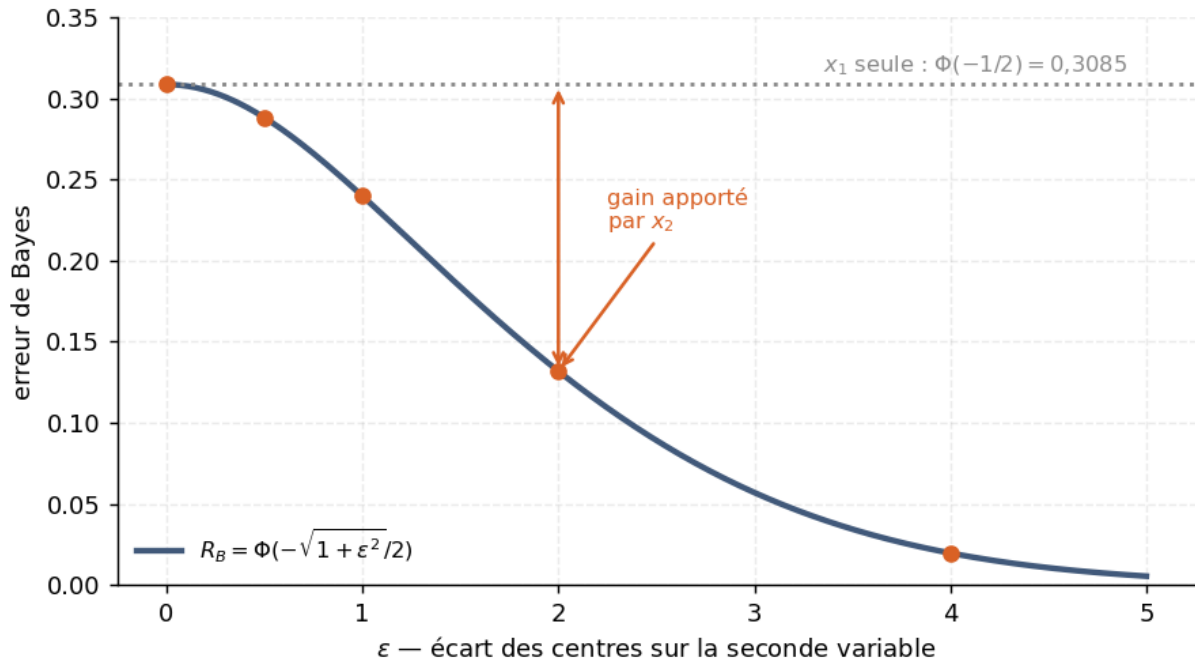


FIGURE 2 – L’erreur de Bayes décroît avec ε . En pointillé : l’erreur avec x_1 seule.

4. Sans hypothèse sur les proportions :

$$R_B = \pi_0 \Phi\left(-\frac{\Delta}{2} - \frac{\ell}{\Delta}\right) + \pi_1 \Phi\left(-\frac{\Delta}{2} + \frac{\ell}{\Delta}\right), \quad \ell = \log \frac{\pi_0}{\pi_1}.$$

Vérification sur le TD 1. En dimension 1, $\Delta = |\mu_1 - \mu_0|/\sigma = 2,5$ et $\ell = \log(0,65/0,35) = 0,6190$:

$$R_B = 0,65 \Phi(-1,4976) + 0,35 \Phi(-1,0024) = 0,0436 + 0,0553 = \mathbf{0,0990}.$$

C’est le risque de Bayes obtenu au TD 1 par intégration numérique.

Exercice 3 — Ce qu’apporte la seconde variable

1. Un observateur ne dispose que de x_1 . Quel est Δ ? Quelle erreur de Bayes ?
2. Comparer à la question 3 de l’exercice 2 et remplir la colonne « gain ».
3. Démontrer que, dans ce modèle, ajouter une variable ne peut **jamais** augmenter l’erreur de Bayes. Le résultat vaut-il encore si la variable ajoutée est du bruit pur ($\varepsilon = 0$) ?
4. En pratique, ajouter une variable inutile dégrade pourtant les modèles. Où est la contradiction ?

💡 Correction

1. En n'observant que $x_1 : \Delta_1 = |1 - 0|/1 = 1$, donc $R_B^{(1)} = \Phi(-1/2) = 0,3085$, **quelle que soit la valeur de ε** .
2. Le gain est $R_B^{(1)} - R_B$:

ε	R_B (deux variables)	gain sur x_1 seule
0	0,3085	+0,0000
0,5	0,2881	+0,0205
1	0,2398	+0,0688
2	0,1318	+0,1768
4	0,0196	+0,2889

3. $\Delta^2 = 1 + \varepsilon^2 \geq 1 = \Delta_1^2$, et $\Phi(-\Delta/2)$ est décroissante en Δ : l'erreur ne peut que baisser. Pour $\varepsilon = 0$, $\Delta = \Delta_1$ et le gain est nul.

En général, la règle de Bayes sur (x_1, x_2) peut ignorer x_2 : elle fait donc au moins aussi bien que la règle sur x_1 seule.

4. Les deux affirmations ne portent pas sur la même quantité.

L'erreur de **Bayes** ne remonte jamais : elle suppose $p(z)$ connue.

L'erreur **estimée** remonte, parce qu'il faut estimer les paramètres de la variable ajoutée sur un échantillon fini.

Chaque variable de plus coûte des paramètres — K de plus pour Bayes naïf, $O(d)$ de plus pour LDA, $O(Kd)$ pour QDA. Le biais diminue (ou reste égal), la variance augmente (exercice 5).

Partie empirique

Exercice 4 — Bayes naïf, LDA et QDA sur Iris

Le squelette du code est fourni. Le compléter, puis répondre.

```
import pandas as pd
from sklearn.naive_bayes import GaussianNB
from sklearn.discriminant_analysis import (LinearDiscriminantAnalysis,
                                           QuadraticDiscriminantAnalysis)
from sklearn.model_selection import StratifiedKFold, cross_val_score

iris = pd.read_csv("../data/iris.csv")
X, y = iris.drop(columns="species"), iris["species"]
petale = ["petal length (cm)", "petal width (cm)"]

modeles = {"Bayes naïf": GaussianNB(),
           "LDA": LinearDiscriminantAnalysis(),
           "QDA": QuadraticDiscriminantAnalysis()}

plis = StratifiedKFold(n_splits=10, shuffle=True, random_state=0)
for nom, modele in modeles.items():
    scores = cross_val_score(modele, X[petale], y, cv=plis)
    print(nom, 1 - scores.mean(), scores.std())
```

```

library(e1071)
library(MASS)

iris_df <- read.csv("../data/iris.csv")
iris_df$species <- factor(iris_df$species)
petale <- grep("petal", names(iris_df), value = TRUE)

# Plis stratifiés : chaque classe est répartie également entre les K plis.
plis_stratifies <- function(y, K = 10, graine = 0) {
  set.seed(graine)
  pli <- integer(length(y))
  for (cls in levels(y)) {
    idx <- sample(which(y == cls))
    pli[idx] <- rep_len(seq_len(K), length(idx))
  }
  pli
}

modeles <- list(
  "Bayes naïf" = list(ajuste = function(d) naiveBayes(species ~ ., data = d),
    predict = function(m, d) predict(m, d)),
  "LDA" = list(ajuste = function(d) lda(species ~ ., data = d),
    predict = function(m, d) predict(m, d)$class),
  "QDA" = list(ajuste = function(d) qda(species ~ ., data = d),
    predict = function(m, d) predict(m, d)$class))

erreur_vc <- function(donnees, modele, K = 10, graine = 0) {
  pli <- plis_stratifies(donnees$species, K, graine)
  mean(sapply(seq_len(K), function(k) {
    ajuste <- modele$ajuste(donnees[pli != k, ])
    mean(modele$predit(ajuste, donnees[pli == k, ]) != donnees$species[pli == k])
  })))
}

donnees <- iris_df[, c(petale, "species")]
sapply(modeles, function(m) erreur_vc(donnees, m))

```

1. Estimer l'erreur des trois modèles par validation croisée stratifiée à dix plis, d'abord sur les **deux** variables de pétale, puis sur les **quatre** variables.
2. Pourquoi stratifier les plis ? Que se passerait-il sans mélange préalable, le fichier étant trié par espèce ?
3. Le classement des trois modèles est-il le même dans les deux cas ? Proposer une explication géométrique.
4. Relancer avec `random_state=1`, puis 2. De combien l'estimation bouge-t-elle ? Peut-on départager deux modèles séparés de 0,7 point ?

Correction

1. On boucle sur les deux jeux de variables :

```

for titre, Xc in [("pétale", X[petale]), ("quatre", X)]:
    for nom, modele in modeles.items():
        scores = cross_val_score(modele, Xc, y, cv=plis)
        print(titre, nom, round(1 - scores.mean(), 4))

for (donnees in list(iris_df[, c(petale, "species")], iris_df)) {
    print(sapply(modeles, function(m) erreur_vc(donnees, m)))
}

```

Erreurs estimées, graine 0 (Python et R donnent ici les mêmes valeurs) :

Modèle	Deux variables de pétale	Les quatre variables
Bayes naïf	4,00 %	4,67 %
LDA	4,00 %	2,00 %
QDA	3,33 %	2,67 %

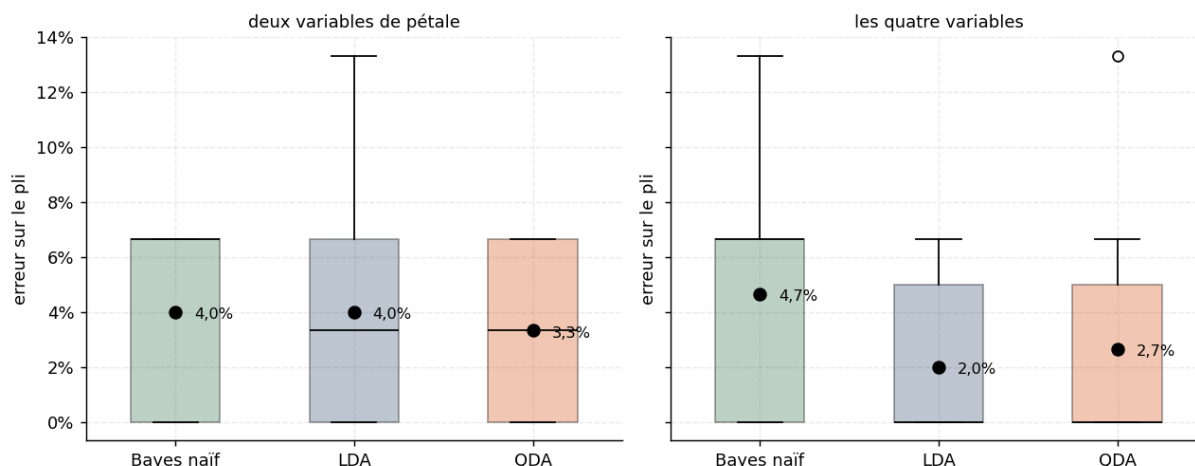


FIGURE 3 – Erreur par pli, sur les deux variables de pétale puis sur les quatre. Point noir : moyenne des plis.

2. Iris compte exactement cinquante fleurs par espèce et le fichier est trié. Sans mélange, dix blocs contigus donnent des plis d’une seule espèce, sous-représentée dans l’apprentissage. La stratification garde dans chaque pli les proportions du jeu complet.

3. Non, le classement s’inverse.

- **Deux** variables : QDA gagne. Les dispersions des espèces diffèrent (*setosa* serrée, *virginica* étalée) et QDA n’a que neuf paramètres de covariance à estimer.
- **Quatre** variables : LDA passe devant. QDA estime trente paramètres de covariance, dix par classe ; la variance d’estimation l’emporte. Bayes naïf traite comme indépendantes des variables corrélées.

4. Vingt graines :

```
for nom, modele in modeles.items():
    err = [1 - cross_val_score(modele, X, y, cv=StratifiedKFold(
        10, shuffle=True, random_state=g)).mean() for g in range(20)]
    print(nom, min(err), max(err))
```

```
sapply(modeles, function(m)
  range(sapply(0:19, function(g) erreur_vc(iris_df, m, graine = g))))
```

Minimum et maximum (Python) :

Modèle	Deux variables	Quatre variables
Bayes naïf	4,00 % (constant)	4,00 % – 5,33 %
LDA	4,00 % – 4,67 %	2,00 % (constant)
QDA	2,67 % – 3,33 %	2,00 % – 3,33 %

R donne les mêmes plages, sauf QDA sur deux variables : 2,00 % – 3,33 %.

L’estimation bouge d’environ un point. Avec $n = 150$, 0,7 point est **une seule fleur** : l’écart n’est pas interprétable. Il faudrait répéter la validation croisée et comparer des distributions.

Exercice 5 — Variables de bruit : théorie contre pratique

L'exercice 3 a établi que l'erreur de Bayes ne remonte jamais quand on ajoute une variable. Vérifions ce qu'il en est de l'erreur *estimée*.

```
import numpy as np

def ajoute_bruit(X, n_bruit, graine=0):
    """Ajoute n_bruit variables gaussiennes sans aucun lien avec la classe."""
    rng = np.random.default_rng(graine)
    return np.hstack([np.asarray(X), rng.normal(size=(len(X), n_bruit))])
```

```
ajoute_bruit <- function(donnees, n_bruit, graine = 0) {
  set.seed(graine)
  bruit <- matrix(rnorm(nrow(donnees) * n_bruit), nrow = nrow(donnees))
  colnames(bruit) <- paste0("bruit", seq_len(n_bruit))
  cbind(donnees, bruit)
}
```

1. Reprendre les quatre variables d'Iris et leur ajouter 5, puis 20, puis 50 variables de bruit. Réestimer l'erreur des trois modèles à chaque étape.
2. Quel modèle se dégrade le plus vite ? Le rapprocher du nombre de paramètres de covariance de chacun.
3. À partir de combien de variables de bruit QDA devient-il inutilisable ? Que dit scikit-learn au passage ?
4. Conclure sur le rapport entre l'exercice 3 et celui-ci.

💡 Correction

1. Une réserve de cent colonnes de bruit, tirée une fois ; le jeu à m variables en prend les m premières (jeux **emboîtés**).

```
rng = np.random.default_rng(0)
reserve = rng.normal(size=(len(X), 100))
for m in [0, 5, 20, 50, 100]:
    Xm = np.hstack([X.to_numpy(), reserve[:, :m]])
    for nom, modele in modeles.items():
        scores = cross_val_score(modele, Xm, y, cv=plis)
        print(m, nom, round(1 - scores.mean(), 4))
```

```
set.seed(0)
reserve <- matrix(rnorm(nrow(iris_df) * 100), nrow = nrow(iris_df))
colnames(reserve) <- paste0("bruit", 1:100)
for (m in c(0, 5, 20, 50, 100)) {
  d <- cbind(iris_df, reserve[, seq_len(m), drop = FALSE])
  print(sapply(modeles, function(mod)
    tryCatch(erreur_vc(d, mod), error = function(e) NA)))
}
```

Erreur estimée, quatre variables d'Iris plus m variables de bruit :

m	Bayes naïf	LDA	QDA	Bayes naïf (R)	LDA (R)	QDA (R)
0	4,67 %	2,00 %	2,67 %	4,67 %	2,00 %	2,67 %

5	4,00 %	2,67 %	4,00 %	4,67 %	4,00 %	5,33 %
20	6,00 %	4,00 %	12,00 %	6,00 %	4,67 %	10,00 %
50	7,33 %	7,33 %	48,00 %	4,67 %	8,00 %	—
100	8,67 %	17,33 %	42,67 %	6,67 %	18,67 %	—

Les trois premières colonnes sont celles de Python ; le bruit tiré par R diffère.

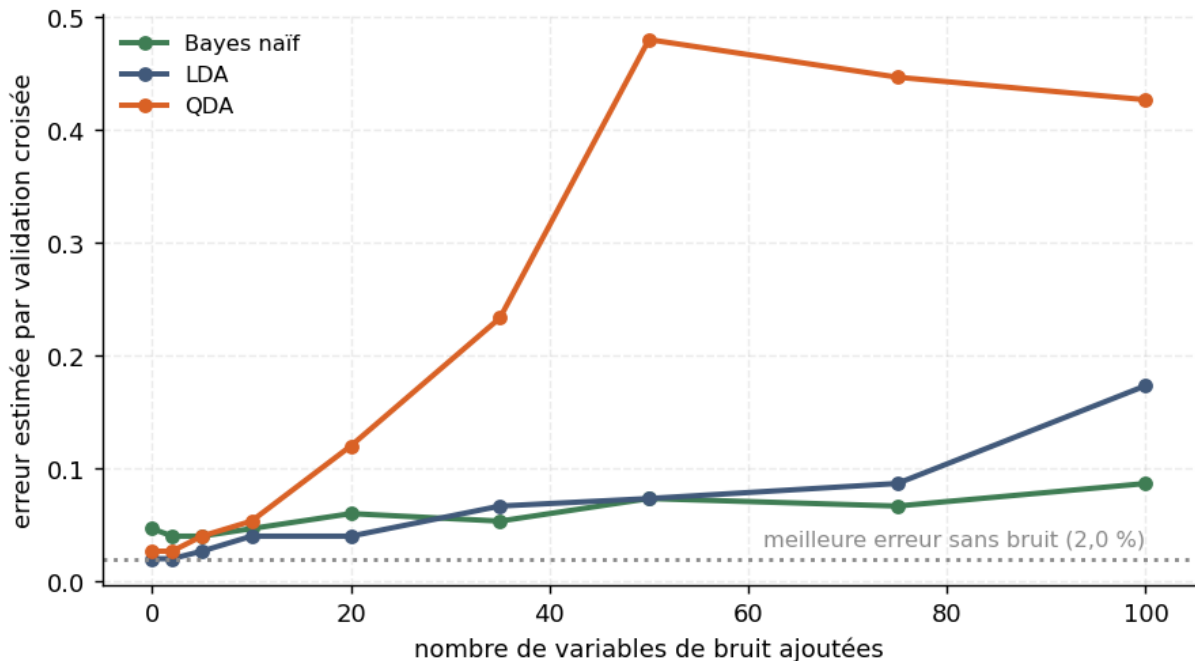


FIGURE 4 – Erreur estimée quand on ajoute des variables de bruit pur. L'erreur de Bayes, elle, reste constante.

2. QDA, de très loin. Avec d variables et $K = 3$ classes, il estime $K d(d + 1)/2$ paramètres de covariance : 30 pour $d = 4$, mais **2 106** pour $d = 24$ — sur 150 observations. LDA n'en estime que $d(d + 1)/2$, soit trois fois moins, et Bayes naïf Kd . L'ordre de robustesse est l'ordre inverse du nombre de paramètres.

3. QDA décroche dès vingt variables de bruit ; à cinquante, son erreur atteint 48 %. Chaque classe compte 45 fleurs d'apprentissage : au-delà de 45 variables, sa covariance empirique est singulière. Scikit-learn émet un `LinAlgWarning` (*covariance matrix of class k is not full rank*) et prédit quand même ; `qda` de R refuse d'ajuster (*some group is too small for 'qda'*). Remède : `reg_param`, la régularisation de QDA vers LDA vue en cours.

4. Les deux exercices mesurent deux quantités différentes.

L'exercice 3 porte sur l'erreur de **Bayes** : $p(z)$ est connue, ajouter une variable ne peut pas nuire. L'exercice 5 porte sur l'erreur **estimée** : $p(z)$ est inconnue, chaque variable ajoutée doit être payée en paramètres, donc en variance.

Remarque finale

! À retenir

Bayes naïf, LDA et QDA sont trois contraintes sur la même matrice Σ . La théorie dit que la plus riche ne peut pas être moins bonne ; l'échantillon dit souvent le contraire. Entre les deux, il n'y a que le nombre de paramètres à estimer et la taille de n .

Les deux scripts exécutables correspondants sont `python/td02_classification.py` et `r/td02_classification.R`.