

TP 1 — Vins : explorer sans étiquette, puis prédire

ACP, k -means, modèles gaussiens et k plus proches voisins

Christophe Ambroise

2026-10-19

Objectifs

Explorer un jeu de données sans regarder les étiquettes (ACP, k -means), puis apprendre à les prédire (LDA, QDA, k -PPV), et relier les deux démarches : la structure trouvée sans étiquette annonce-t-elle la difficulté du problème supervisé ? Fil rouge : **le rôle des unités**.

Données et consignes

`data/wine.csv` — jeu UCI *Wine* (Forina et al., 1988, diffusé par l'UCI Machine Learning Repository) : 178 vins d'une même région d'Italie, issus de trois cépages (*cultivars*), décrits par 13 mesures chimiques (alcool, acide malique, flavonoïdes, intensité de la couleur, proline...). La colonne `cultivar` (1, 2 ou 3) est l'étiquette.

- Travailler en Python **ou** en R ; les squelettes sont donnés dans les deux langages.
- Rendre un compte rendu (Quarto ou notebook) qui contient le code, les figures et une réponse **rédigée** à chaque question. Une figure sans commentaire ne répond à rien.
- Fixer les graines (`random_state=0`, `set.seed(0)`) pour que les résultats soient reproductibles.

```
from pathlib import Path

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

vins = pd.read_csv(Path("..") / "data" / "wine.csv")
y = vins["cultivar"].to_numpy()
X_df = vins.drop(columns="cultivar")      # l'étiquette sort avant tout calcul
X = X_df.to_numpy()
```

```
vins <- read.csv("../data/wine.csv")
y <- factor(vins$cultivar)
X <- as.matrix(vins[, setdiff(names(vins), "cultivar")]) # étiquette retirée
```

Partie 1 — Exploration

1. Donner n , p et les effectifs des trois cépages.

2. Calculer moyenne, écart-type, minimum et maximum de chaque variable. Quelle variable a le plus grand écart-type ? Dans quel rapport avec le plus petit ?
3. Tracer la matrice des corrélations (une carte de chaleur suffit). Repérer un groupe de variables fortement corrélées entre elles. Qu'en attendre pour l'ACP ?

Partie 2 — ACP : brute ou standardisée ?

L'étiquette reste **hors** de tout calcul dans cette partie et la suivante. On note $S = \frac{1}{n} X_c^T X_c$ (diviseur n , comme au CM 6).

```
def acp(A):
    """Valeurs propres décroissantes, vecteurs propres, scores du nuage centré."""
    Ac = A - A.mean(axis=0)
    lam, U = np.linalg.eigh(Ac.T @ Ac / len(Ac))
    ordre = lam.argsort()[::-1]
    lam, U = lam[ordre], U[:, ordre]
    return lam, U, Ac @ U
```

```
Z = (X - X.mean(axis=0)) / X.std(axis=0) # écart-type population
```

```
ecart_type_pop <- function(v) sqrt(mean((v - mean(v))^2))
Z <- scale(X, center = TRUE, scale = apply(X, 2, ecart_type_pop))

acp <- function(A) {
  Ac <- scale(A, center = TRUE, scale = FALSE)
  e <- eigen(crossprod(Ac) / nrow(Ac), symmetric = TRUE) # valeurs décroissantes
  list(lam = e$values, U = e$vectors, C = Ac %*% e$vectors)
}
```

1. Faire l'ACP des données **brutes**. Quelle part de l'inertie porte le premier axe ? Quelle variable domine le premier vecteur propre ? Relier la réponse à la question 1.2. Cette ACP décrit-elle les vins ou les unités de mesure ?
2. Faire l'ACP des données **standardisées**. Tracer l'ébouillis des valeurs propres et le taux d'inertie cumulé ρ_q . Combien d'axes retenir, et selon quel critère ?
3. Tracer le plan principal (c_1, c_2) sans couleur. Voit-on des groupes ?
4. Calculer les corrélations variable-composante $\sqrt{\lambda_k} u_{jk}$ et tracer le cercle des corrélations. Interpréter les deux premiers axes en termes chimiques.
5. Colorier *a posteriori* le plan principal par cépage. Commenter, en tenant compte de ρ_2 : faut-il une forte part d'inertie pour voir des groupes ?

Partie 3 — k -means

1. Sur les données standardisées, tracer l'inertie intra-classe I_W en fonction de $K \in \{1, \dots, 8\}$ (`n_init=50` ou `nstart = 50`). Le coude désigne-t-il clairement un K ?
2. Avec $K = 3$, croiser les groupes avec les cépages (table de contingence) et calculer la **pureté** : la part des vins qui appartiennent au cépage majoritaire de leur groupe.
3. Recommencer sur les données **brutes**, puis sur les deux premières composantes principales seulement. Classer les trois versions et expliquer l'écart par la Partie 2.
4. Les numéros de groupes de k -means sont arbitraires : pourquoi la pureté ne dépend-elle pas de cette numérotation, et pourquoi le taux d'« erreur » brut $\frac{1}{n} \sum_i \mathbb{1}\{g_i \neq y_i\}$ en dépendrait-il ?

```

from sklearn.cluster import KMeans

def purete(groupees, y):
    table = pd.crosstab(groupees, y)
    return table.max(axis=1).sum() / len(y)

groupees = KMeans(3, n_init=50, random_state=0).fit_predict(Z)

```

```

purete <- function(groupees, y) {
  table_c <- table(groupees, y)
  sum(apply(table_c, 1, max)) / length(y)
}
set.seed(0)
groupees <- kmeans(Z, centers = 3, nstart = 50)$cluster

```

Partie 4 — Classification supervisée

Les étiquettes entrent en jeu. Toutes les erreurs sont estimées par validation croisée stratifiée à 10 plis. **Toute** opération apprise sur les données (centrage, réduction, ACP) s'apprend sur le pli d'apprentissage seul.

```

from sklearn.discriminant_analysis import (LinearDiscriminantAnalysis,
                                           QuadraticDiscriminantAnalysis)
from sklearn.model_selection import StratifiedKFold, cross_val_score
from sklearn.neighbors import KNeighborsClassifier
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler

plis = StratifiedKFold(n_splits=10, shuffle=True, random_state=0)

def erreur_vc(modele, A=X):
    return 1 - cross_val_score(modele, A, y, cv=plis).mean()

knn_std = make_pipeline(StandardScaler(), KNeighborsClassifier(5))

```

```

library(class)
library(MASS)

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
}
pli <- plis_stratifies(y)

# `ajuste_predit(app, y_app, val)` renvoie les classes prédites sur `val`.
erreur_vc <- function(ajuste_predit, A = X) {

```

```

mean(sapply(1:10, function(k) {
  pred <- ajuste_predit(A[pli != k, , drop = FALSE], y[pli != k],
                        A[pli == k, , drop = FALSE])
  mean(pred != y[pli == k])
}))
}

lda_ap <- function(app, y_app, val) predict(lda(app, y_app), val)$class
knn_ap <- function(k, standardise) function(app, y_app, val) {
  if (standardise) { # centre et échelle appris sur le seul pli d'apprentissage
    centre <- colMeans(app); echelle <- apply(app, 2, sd)
    app <- scale(app, centre, echelle); val <- scale(val, centre, echelle)
  }
  knn(app, val, y_app, k = k)
}

```

1. Estimer l'erreur de la LDA et de la QDA (CM 2). Combien de paramètres chacune estime-t-elle ici ? Le rapport $n/\text{paramètres}$ de la QDA est-il rassurant ?
2. Refaire la LDA sur les données standardisées. Comparer et **démontrer** le résultat : que devient la règle de la LDA quand on remplace x par $Dx + b$, avec D diagonale inversible ?
3. Tracer l'erreur des k -PPV pour $k \in \{1, 3, \dots, 31\}$, sur données brutes et sur données standardisées, sur la même figure. Commenter l'écart, et le comparer à ce que donnait la standardisation sur Iris au TD 3.
4. Pourquoi la standardisation doit-elle être apprise dans chaque pli plutôt que sur les 178 vins une fois pour toutes ? Ici, l'écart serait-il visible ?
5. Relancer la question 3 avec quatre autres graines de découpage. Le k optimal est-il stable ? Quel k retenir, et sur quel argument ?

Partie 5 — Réduire avant de classer

1. Pour $q \in \{1, \dots, 13\}$, tracer l'erreur des 5-PPV sur les q premières composantes des données standardisées, ACP apprise **dans chaque pli** (`make_pipeline(StandardScaler(), PCA(q), ...)` ; en R, adapter `knn_ap`).
2. L'ACP a-t-elle été construite pour séparer les cépages ? Pourquoi peut-elle néanmoins aider les k -PPV ? Pourquoi n'est-ce pas garanti ?
3. Synthèse : réunir dans un tableau les meilleures erreurs de la LDA, de la QDA, des k -PPV et des k -PPV après ACP. Lequel recommanderiez-vous, en tenant compte de l'erreur, de sa variabilité d'une graine à l'autre et de l'interprétabilité ?