Projet Machine Learning

Wine Quality Dataset

Analyse et Modélisation avec Réduction de Dimension, Clustering et Classification

Brahim Semlali
Master ISI (Ingenierie des Systèmes d'Information)

GitHub Repository

14 février 2026

Lien du dépôt GitHub :

https://github.com/Brahim-semlali/ML-project

Table des matières

1 Introduction

Ce projet, réalisé dans le cadre du **Master ISI** (Informatique et Systèmes d'Information), consiste à analyser et modéliser le dataset *Wine Quality* portant sur les vins rouges. L'objectif principal est double :

1. **Prédire la qualité du vin** à partir de ses propriétés physico-chimiques
2. **Explorer la structure des données** à l'aide de méthodes de réduction de dimension, de clustering et de classification

1.1 Motivation du projet

Le jeu de données choisi présente plusieurs atouts qui justifient son utilisation dans un contexte pédagogique et pratique :

Avantages du dataset Wine Quality

- **Interprétabilité** : Variables correspondant à des mesures chimiques réelles (acidité, alcool, sulfates)
- **Taille adaptée** : 1599 échantillons et 12 colonnes, idéal pour l'apprentissage
- **Documentation** : Référence académique claire (Cortez et al., 2009)
- **Applications pratiques** : Contrôle qualité en œnologie et industrie viticole

1.2 Structure du rapport

Ce rapport détaille successivement :

1. La description du dataset et son exploration (EDA)
2. La réduction de dimension (PCA, t-SNE, NMF, LDA)
3. Le clustering (K-Means, Agglomerative, DBSCAN, GMM)
4. La comparaison de modèles de classification (Logistic Regression, Naive Bayes, KNN, Decision Tree, SVM, Random Forest, AdaBoost, Gradient Boosting, Neural Network)

2 Dataset

2.1 Présentation et source

Le dataset **Wine Quality (Red Wine)** provient d'une étude de Cortez et al. (2009), dont l'objectif était de modéliser les préférences de dégustation à partir de propriétés physico-chimiques mesurables en laboratoire.

Informations techniques

Nom	Wine Quality Dataset (Red Wine)
Référence	Cortez et al., *Decision Support Systems*, 2009
Échantillons	1599 vins rouges
Variables	12 (11 features + 1 cible)
Source Kaggle	https://www.kaggle.com/datasets/uciml/red-wine-quality-cortez-et-al-2009
Source UCI	https://archive.ics.uci.edu/ml/datasets/wine+quality

2.2 Variables du dataset

2.2.1 Variables d'entrée (Features)

Les **11 variables d'entrée** sont des mesures physico-chimiques :

Variable	Description	Unité
fixed acidity	Acidité fixe (acide tartrique)	g/dm^3
volatile acidity	Acidité volatile (acide acétique)	g/dm^3
citric acid	Acide citrique	g/dm^3
residual sugar	Sucre résiduel après fermentation	g/dm^3
chlorides	Chlorures (salinité)	g/dm^3
free sulfur dioxide	SO_2 libre (anti-oxydant)	mg/dm^3
total sulfur dioxide	SO_2 total (libre + lié)	mg/dm^3
density	Densité	g/cm^3
pH	Acidité/alcalinité	0–14
sulphates	Sulfates (additif)	g/dm^3
alcohol	Degré d'alcool	% vol.

Table 1 – *Variables physico-chimiques du dataset*

2.2.2 Variable cible

Variable cible : quality

La variable **quality** est une note entière de 0 à 10, dérivée de la médiane d'au moins trois évaluations par des dégustateurs experts. Dans ce dataset, les notes observées vont de **3 à 8**.

Deux approches :

— **Binaire** : qualité ≥ 6 = "bon", < 6 = "moyen/mauvais"

— **Multi-classes** : chaque note comme classe distincte

2.3 Exploration des données (EDA)

L'exploration des données permet de vérifier la qualité des données, visualiser les distributions et identifier les corrélations entre variables.

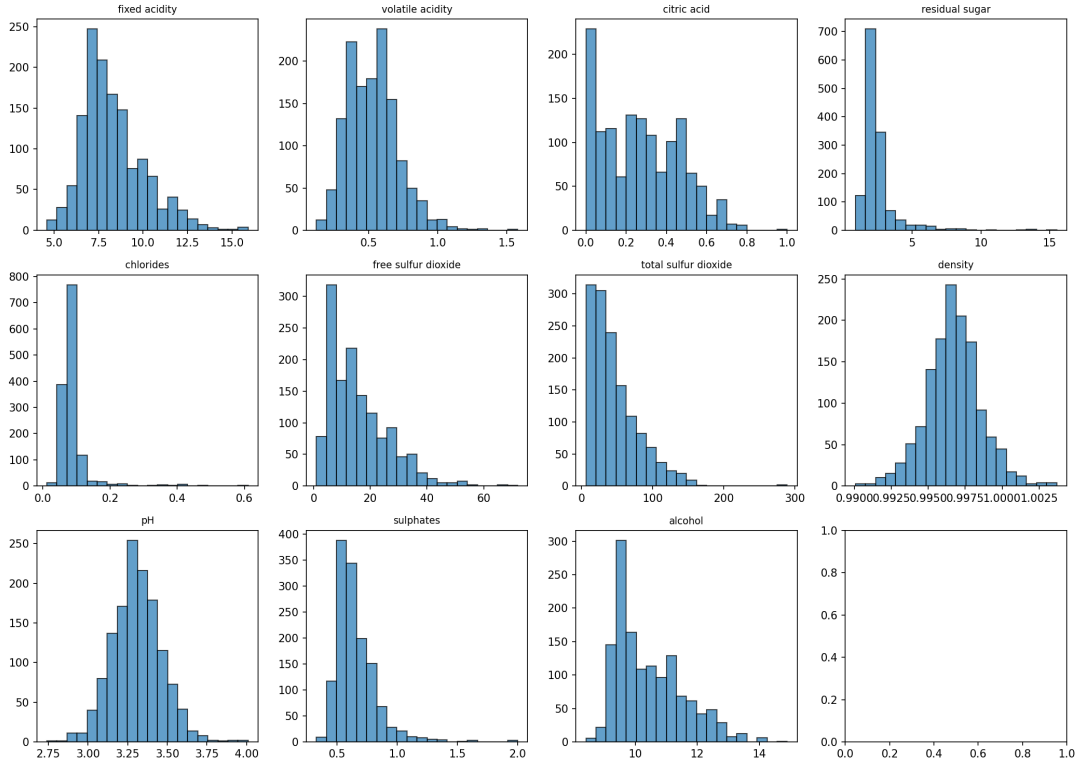


Figure 1 – *Distribution des features du dataset Wine Quality*

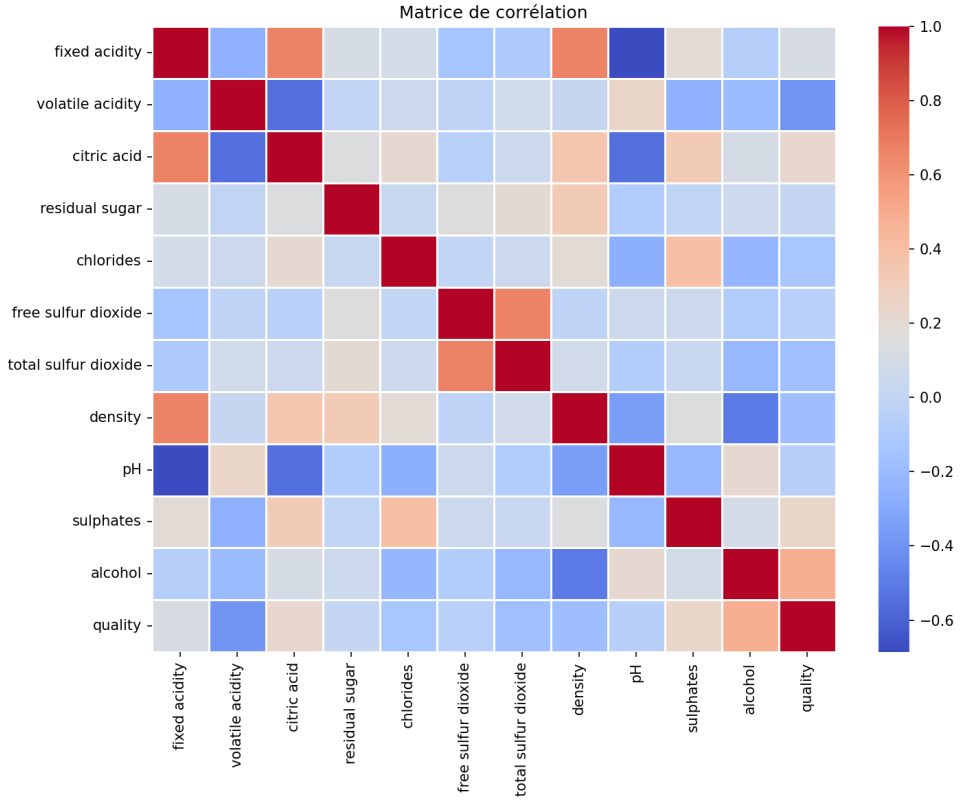


Figure 2 – *Matrice de corrélation entre les features*

Interprétation de l'EDA

Distributions : Les histogrammes révèlent des distributions souvent asymétriques (ex. **residual sugar**, **chlorides**), justifiant la standardisation.

Corrélations :

- **density** fortement corrélée à **fixed acidity**
- **alcohol** négativement corrélé à **density**
- Corrélations modérées avec **quality**

3 Réduction de dimension

La réduction de dimension permet de visualiser les données dans un espace de plus faible dimension. Nous utilisons quatre méthodes complémentaires (PCA, t-SNE, NMF et LDA).

3.1 PCA (Principal Component Analysis)

La PCA est une méthode **linéaire** qui maximise la variance expliquée par des composantes orthogonales.

3.1.1 Résultats PCA 2D

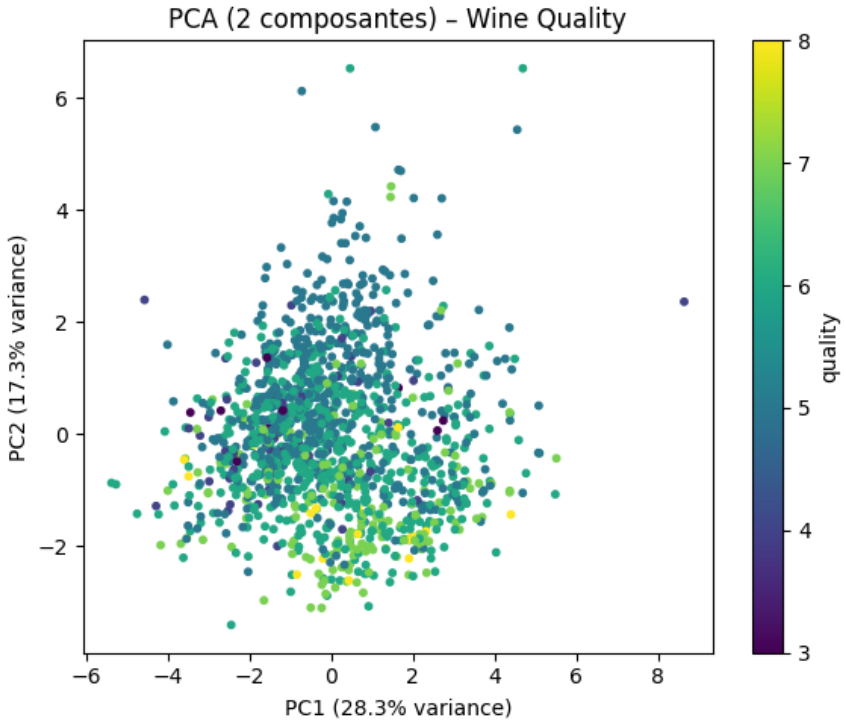


Figure 3 – *PCA avec 2 composantes principales - Visualisation colorée par qualité*

Résultats PCA 2D

Variance expliquée PC1	28,3 %
Variance expliquée PC2	17,3 %
Variance cumulée totale	45,6 %

Interprétation : La variance cumulée (45,6 %) indique qu'une part importante de l'information est perdue en 2D ; la visualisation reste utile pour observer un

gradient partiel selon la qualité. Le chevauchement des couleurs suggère que la frontière bon/mauvais vin n'est pas linéairement séparable, ce qui motive l'usage de modèles non linéaires en classification.

3.1.2 Résultats PCA 3D

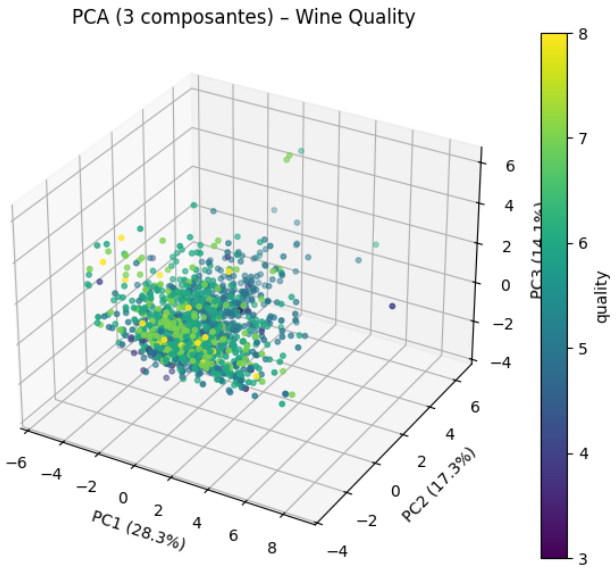


Figure 4 – *PCA avec 3 composantes principales - Visualisation 3D*

Interprétation PCA 3D

La vue 3D peut révéler des séparations entre qualités peu visibles en 2D. On observe souvent un continuum avec des zones plus denses pour certaines notes, sans séparation nette, cohérent avec la subjectivité des dégustations.

3.2 t-SNE (t-Distributed Stochastic Neighbor Embedding)

Le t-SNE est une méthode **non linéaire** qui préserve les voisinages locaux.

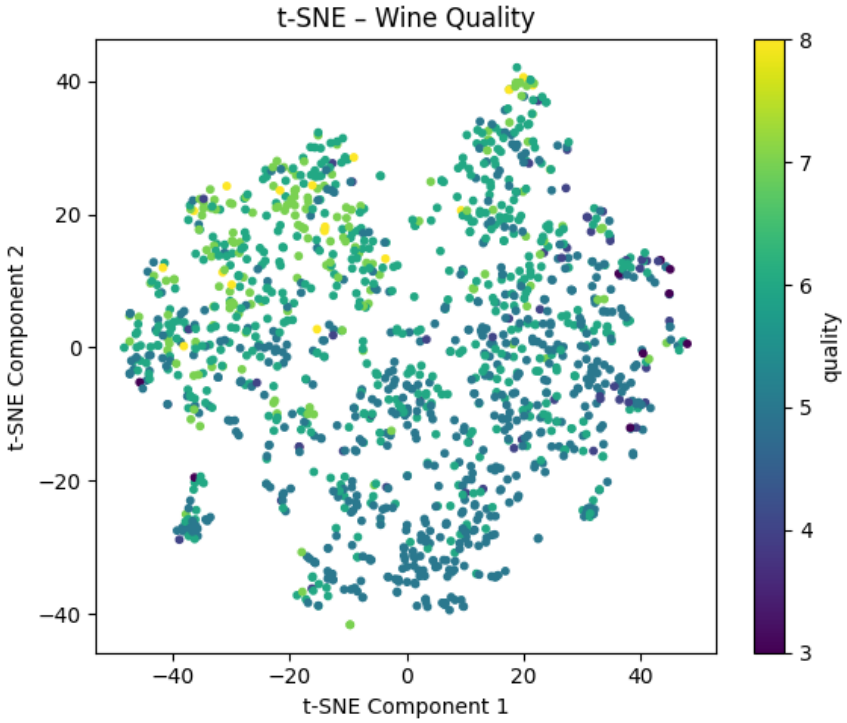


Figure 5 – *t-SNE 2D - Visualisation de la structure des données (perplexity=30)*

Paramètres t-SNE

Perplexity : 30 | **Random state** : 42

Interprétation t-SNE

Si des amas émergent partiellement alignés avec les couleurs de qualité, la structure est non linéaire : des groupes de vins chimiquement proches correspondent à des niveaux de qualité similaires. Un mélange homogène indiquerait que la qualité n'est pas clairement reflétée par les voisinages dans l'espace des 11 variables.

3.3 NMF (Non-negative Matrix Factorization)

La NMF factorise les données en composantes **non-négatives**, interprétables comme des profils chimiques.

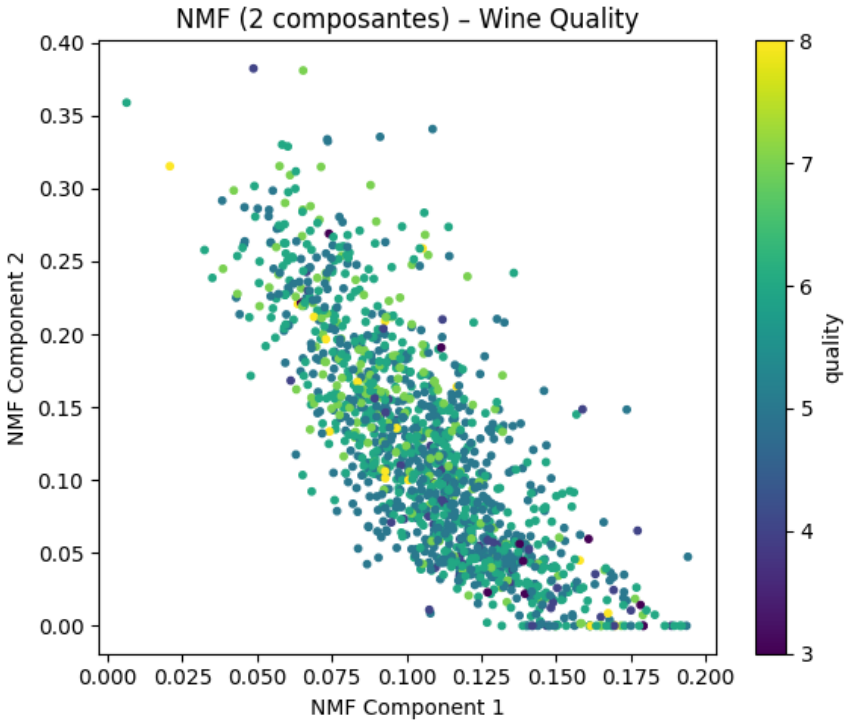


Figure 6 – *NMF avec 2 composantes - Identification des composants chimiques dominants*

Résultats NMF

Reconstruction error : 12,62

Interprétation : Les composantes identifient des patterns tels que “alcool/sulfates” vs “acidité/densité”. Chaque axe peut s’interpréter comme un profil chimique dominant ; les vins positionnés à un extrême ont un profil marqué par ce groupe de variables.

3.4 LDA (Linear Discriminant Analysis)

La LDA est une méthode de réduction de dimension **supervisée** qui maximise la séparation entre classes. Contrairement à PCA (non supervisée), LDA utilise les étiquettes **quality** pour projeter les données dans un sous-espace où les classes sont le mieux séparées.

PCA vs LDA

PCA : maximise la variance totale, indépendamment des classes.
LDA : maximise le ratio variance inter-classes / variance intra-classe.

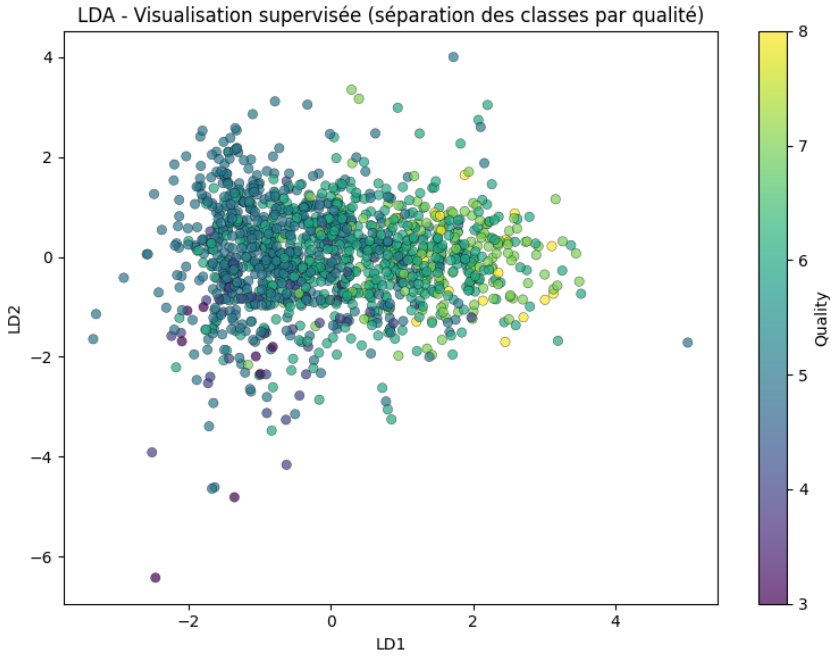


Figure 7 – *LDA 2D - Visualisation supervisée (séparation des classes par qualité)*

Résultats LDA

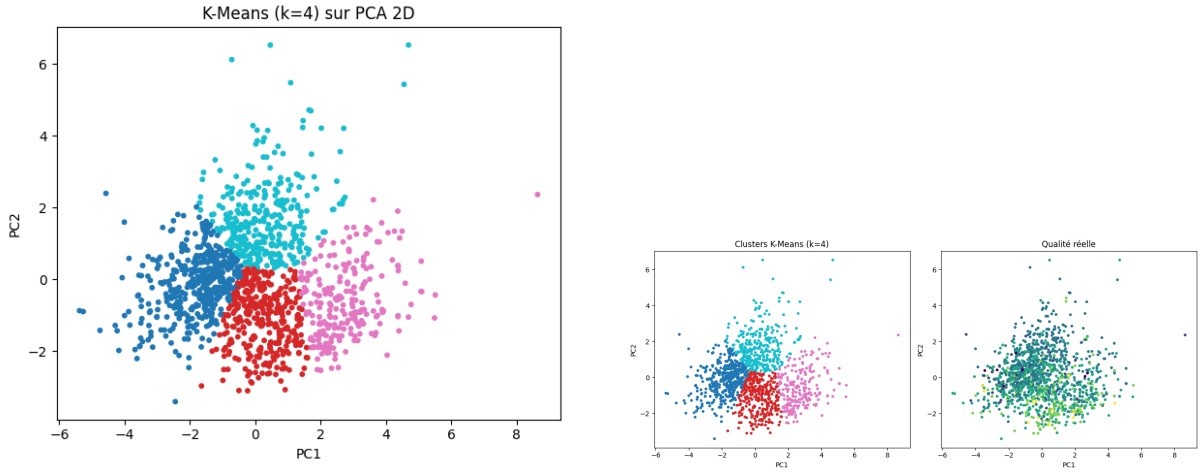
Interprétation : La LDA projette sur au plus $\min(n_{\text{features}}, n_{\text{classes}} - 1)$ composantes. Pour le dataset Wine Quality (6 classes), on obtient jusqu'à 5 composantes discriminantes. La visualisation 2D révèle une meilleure séparation des notes que PCA lorsqu'on exploite explicitement l'information de qualité.

4 Clustering

Le clustering groupe les échantillons sans utiliser l'étiquette **quality**.

4.1 K-Means

K-Means partitionne en k clusters en minimisant l'inertie.



(a) *Clusters K-Means (k=4)* **(b)** *Comparaison avec qualité réelle*

Figure 8 – *Analyse K-Means clustering*

Résultats K-Means

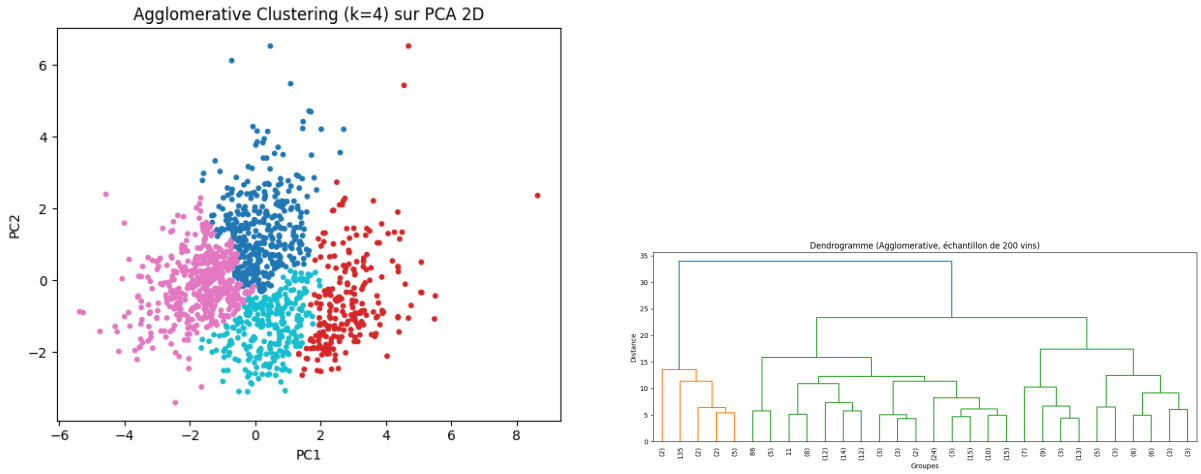
Clusters testés	k = 3, 4, 5
Silhouette score (k=4)	0,367
Observations	Clusters partiellement alignés avec les catégories de qualité ; les segments reflètent des "types" chimiques (ex. alcool/sulfates vs acidité/densité) plutôt qu'une séparation stricte par note.

Interprétation K-Means

Un score de silhouette d'environ 0,37 indique des clusters discernables mais avec chevauchement. La figure de comparaison montre si chaque cluster correspond à une ou plusieurs notes ; un accord partiel rappelle que la qualité perçue dépend aussi de facteurs non mesurés.

4.2 Agglomerative Clustering

Clustering hiérarchique avec méthode de Ward.



(a) *Clusters agglomératifs* **(b)** *Dendrogramme*

Figure 9 – *Agglomerative Clustering (k=4)*

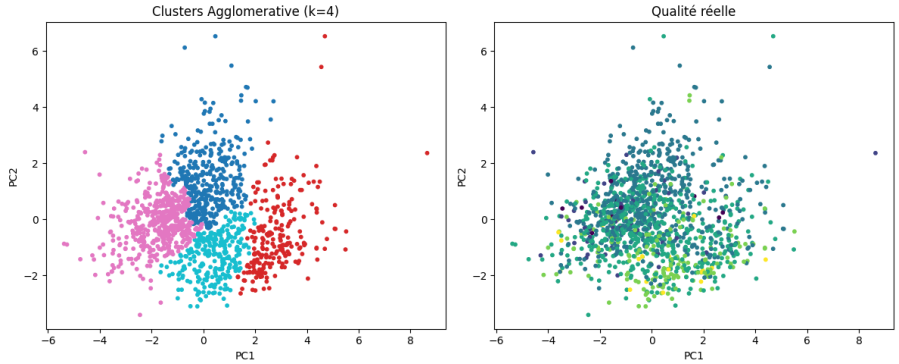


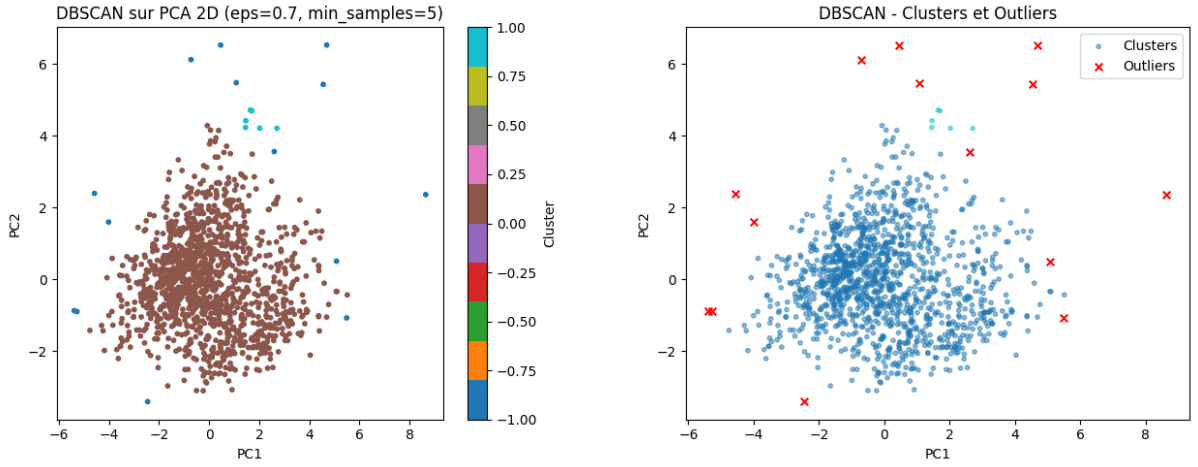
Figure 10 – *Comparaison clusters agglomératifs vs qualité réelle*

Résultats Agglomerative

Méthode de linkage	Ward
Silhouette score (k=4)	0,343
Interprétation	Le dendrogramme révèle la structure hiérarchique ; des sauts de distance indiquent des coupures naturelles. La figure de comparaison montre l'alignement entre clusters et catégories de qualité (comme pour K-Means) ; un accord partiel indique que la segmentation chimique ne correspond pas exactement aux notes de dégustation.

4.3 DBSCAN

DBSCAN identifie des régions denses et détecte les outliers.



(a) *Clusters DBSCAN* **(b)** *Outliers détectés*

Figure 11 – *DBSCAN - Détection de groupes et outliers*

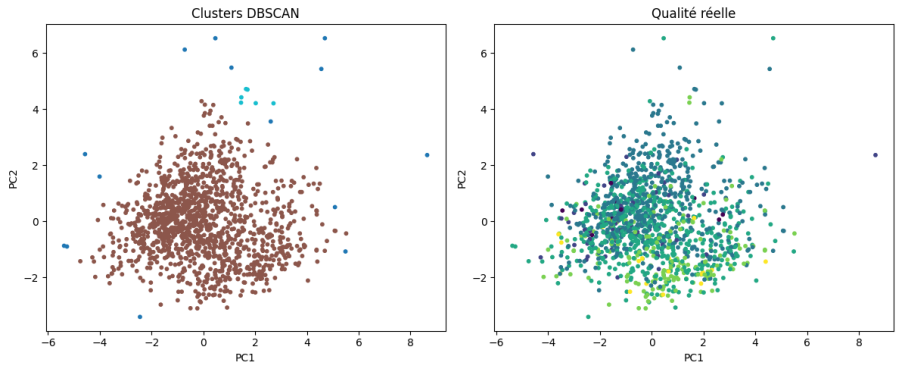


Figure 12 – *Comparaison DBSCAN vs qualité réelle*

Résultats DBSCAN

Paramètres	eps = 0,7, min_samples = 5
Clusters détectés	2
Outliers détectés	14
Silhouette score	0,441 (hors bruit)

Interprétation : DBSCAN produit 2 groupes denses et 14 outliers (vins au profil chimique atypique). La figure de comparaison montre si les clusters correspondent aux catégories de qualité ; les outliers ne sont pas nécessairement de mauvaise qualité et méritent une analyse ciblée pour détecter d'éventuelles valeurs aberrantes ou vins.

4.4 GMM (Gaussian Mixture Models)

Le GMM est un clustering **probabiliste** qui modélise les données comme un mélange de distributions gaussiennes. Contrairement à K-Means (assignment dur), le GMM réalise un *soft assignment* : chaque point appartient à chaque cluster avec une probabilité. Le critère BIC (Bayesian Information Criterion) permet de sélectionner automatiquement le nombre optimal de composantes.

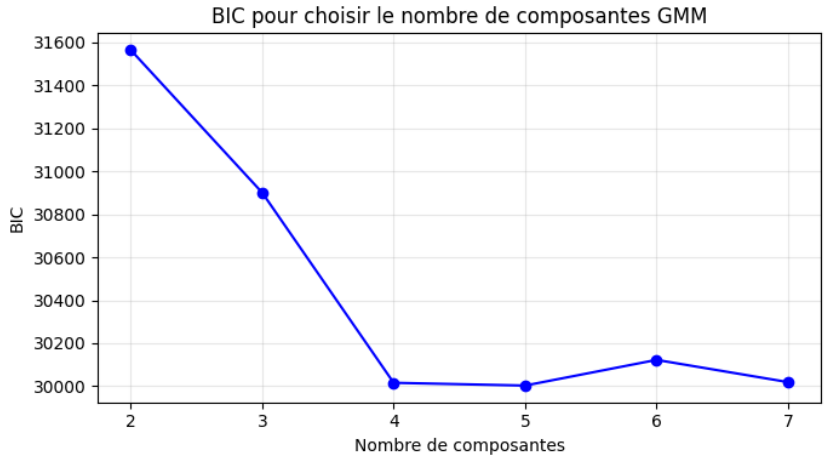


Figure 13 – *BIC en fonction du nombre de composantes GMM – le minimum indique le k optimal*

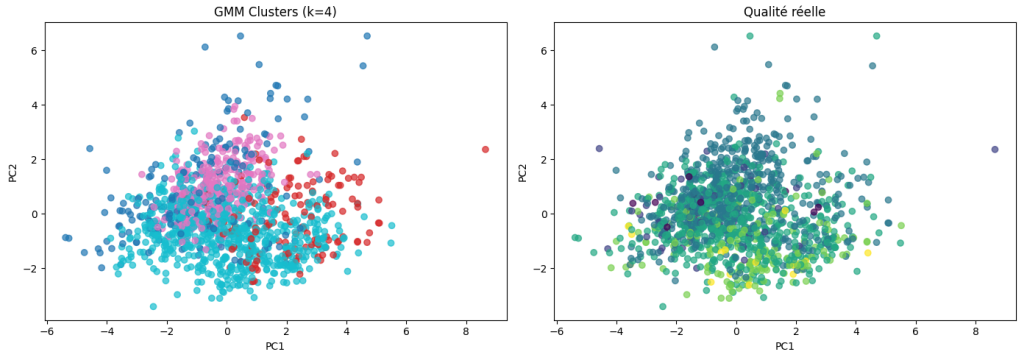


Figure 14 – *GMM - Clusters probabilistes (gauche) vs qualité réelle (droite)*

Résultats GMM

Composantes	k = 4 (sélection BIC)
Silhouette score	0,35–0,40
Avantage	Clusters de forme elliptique ; sélection automatique de k via BIC

Interprétation : La courbe BIC permet de choisir objectivement le nombre de composantes : on sélectionne le k correspondant au BIC minimal. Le GMM offre une modélisation plus flexible que K-Means pour des clusters non sphériques.

4.5 Comparaison des méthodes de clustering

Algorithme	Caractéristiques	Silhouette	k/Clusters
K-Means	Centroïdes, k fixé	0,367	4
Agglomerative	Hiérarchique Ward	0,343	4
DBSCAN	Densité, outliers	0,441	Auto (2 + 14 bruits)
GMM	Probabiliste, soft assignment	0,35–0,40	4 (BIC)

Table 2 – *Comparaison des performances - Clustering*

5 Classification

L'objectif est de prédire la qualité du vin en classification binaire et multi-classes. Nous utilisons neuf modèles : Logistic Regression, Naive Bayes, KNN, Decision Tree, SVM, Random Forest, AdaBoost, Gradient Boosting et Neural Network (MLP).

5.1 Logistic Regression

Modèle linéaire de référence (baseline).

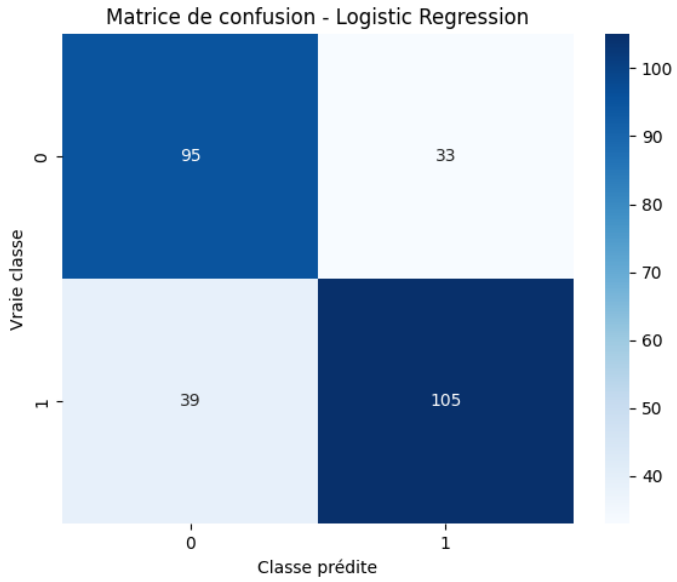


Figure 15 – *Matrice de confusion - Logistic Regression*

Résultats Logistic Regression

Métrique	Valeur
Accuracy	0,735
F1-score	0,745
Precision	0,761
Recall	0,729

Interprétation

Bonne baseline ; la régression logistique suppose une frontière linéaire. Les performances indiquent que la relation qualité/variables est partiellement linéaire.

5.2 Naive Bayes

Naive Bayes est un classificateur probabiliste basé sur le théorème de Bayes et l'hypothèse d'indépendance conditionnelle des features. **GaussianNB** est adapté aux variables continues et constitue un excellent baseline rapide.

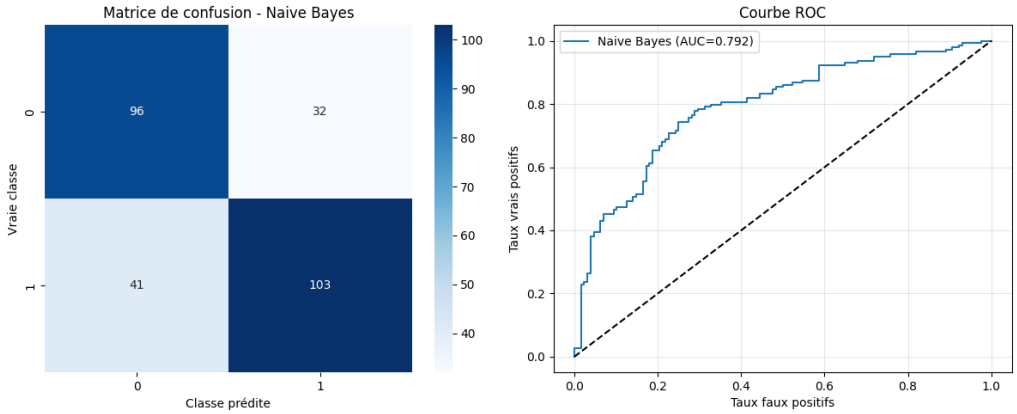


Figure 16 – *Naive Bayes - Matrice de confusion et courbe ROC*

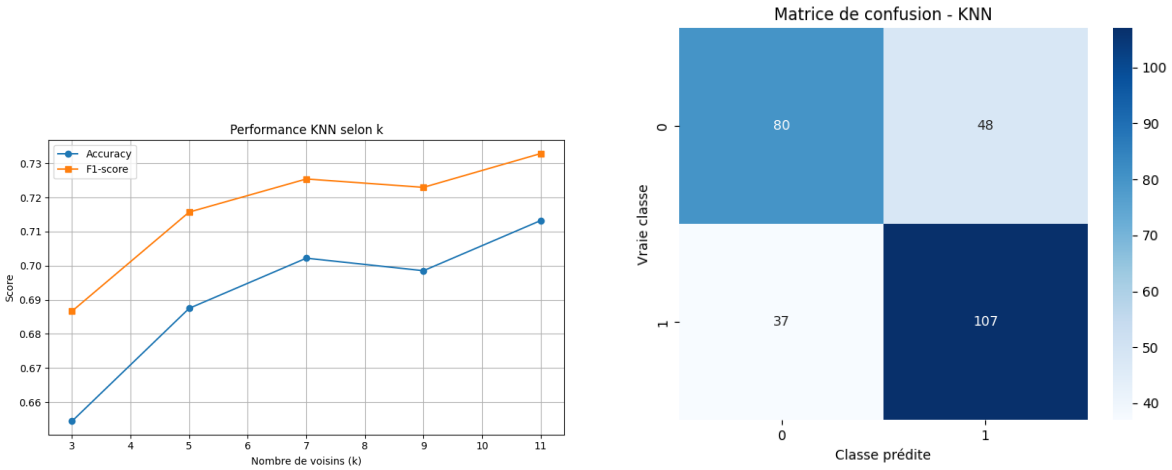
Résultats Naive Bayes

Accuracy	0,68–0,72
F1-score	0,70–0,74
ROC-AUC	Métrique complémentaire pour classificateurs probabilistes

Interprétation : Naive Bayes est très rapide et offre une baseline solide. La courbe ROC et l'AUC permettent d'évaluer la capacité discriminative indépendamment du seuil de décision. L'hypothèse d'indépendance n'est pas respectée (corrélations entre features) mais le modèle reste utile en pratique.

5.3 K-Nearest Neighbors (KNN)

Classification basée sur les voisins proches.



(a) *Performance vs k* **(b)** *Matrice de confusion*

Figure 17 – *Analyse KNN*

Résultats KNN

k optimal : 5 | **Accuracy** : 0,688 | **F1-score** : 0,716

Interprétation

KNN est sensible à la normalisation (appliquée). Un $k = 5$ limite le bruit tout en gardant des frontières locales ; les performances restent en deçà des modèles à base d'arbres.

5.4 Decision Tree

Arbre de décision avec importance des features. L'arbre fournit des règles de décision explicites (seuils sur les variables) et une visualisation directe du processus de classification.

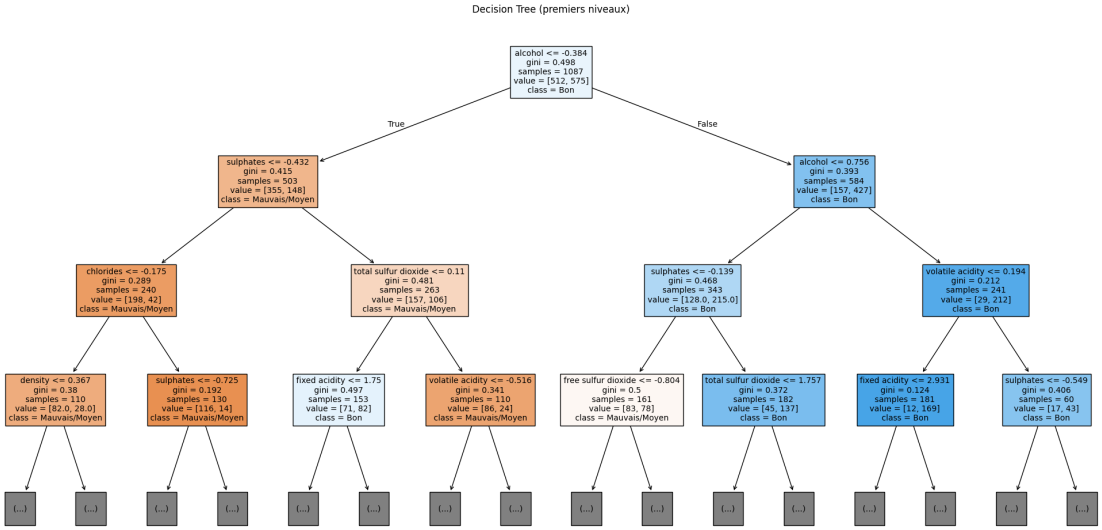
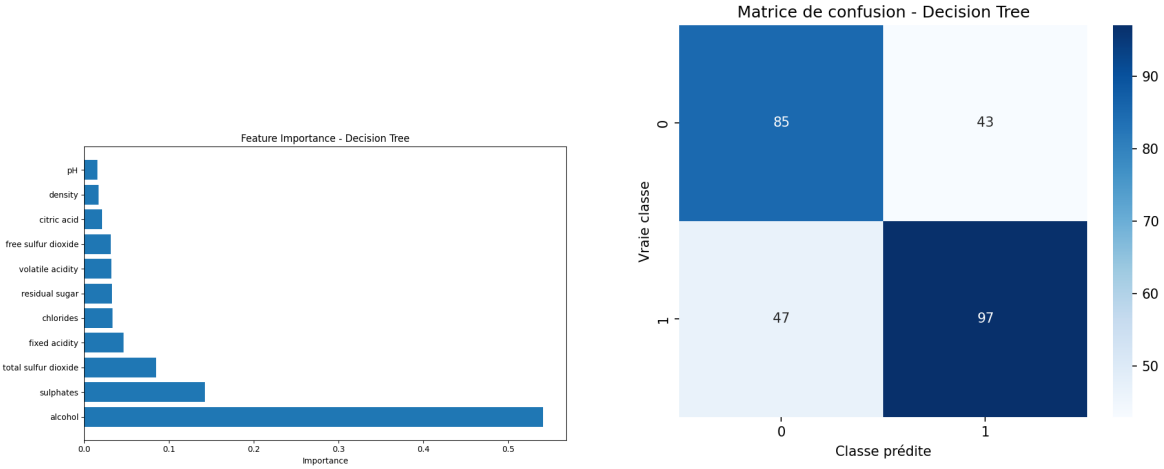


Figure 18 – *Structure de l'arbre de décision – nœuds et seuils interprétables (ex. alcohol < 10,2)*



(a) *Feature Importance* **(b)** *Matrice de confusion*

Figure 19 – *Analyse Decision Tree*

Résultats Decision Tree

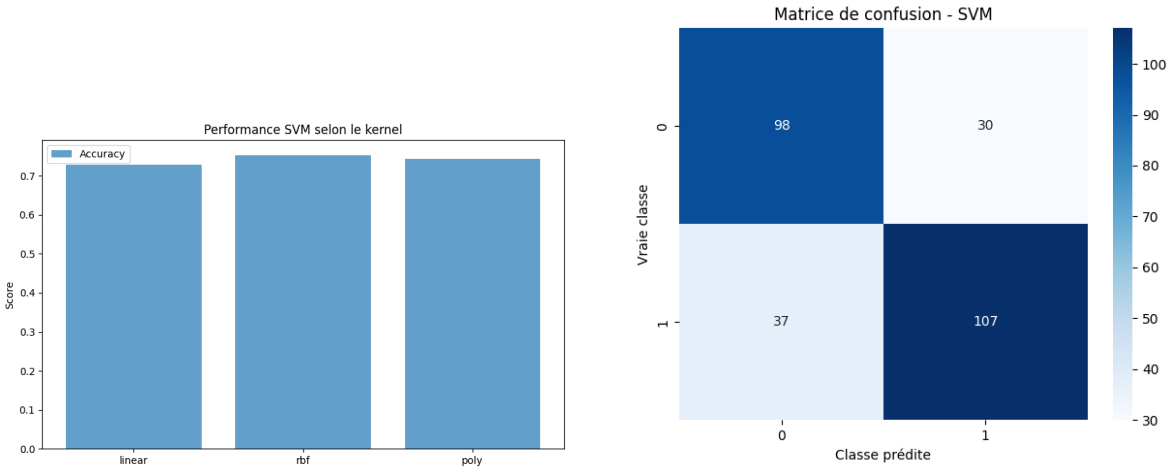
Accuracy	0,669
F1-score	0,683
Top features	alcohol, sulphates, volatile acidity

Interprétation

La figure de l'arbre montre les règles de décision : chaque nœud indique une variable et un seuil (ex. alcohol $\leq$ 10,2 $\rightarrow$ classe 0). Les features les plus utilisées en racine sont les plus discriminantes. Une profondeur limitée (max_depth=5) évite le sur-ajustement ; les importance confirment le rôle de l'alcool, des sulfates et de l'acidité volatile.

5.5 Support Vector Machine (SVM)

SVM avec différents kernels testés.



(a) *Comparaison kernels* **(b)** *Matrice de confusion (RBF)*

Figure 20 – *Analyse SVM*

Résultats SVM

Kernel optimal : RBF | **Accuracy** : 0,754 | **F1-score** : 0,762

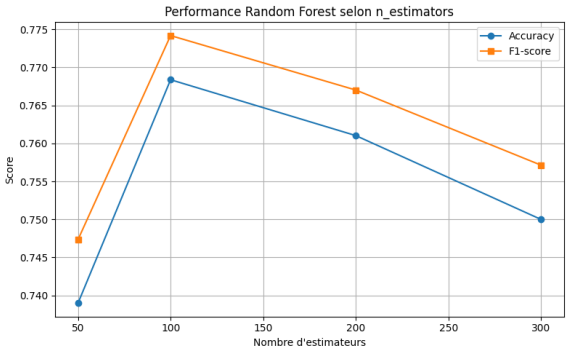
Interprétation

Le kernel RBF capture des frontières non linéaires ; la standardisation est essentielle. SVM atteint des performances proches de Random Forest et Gradient Boosting.

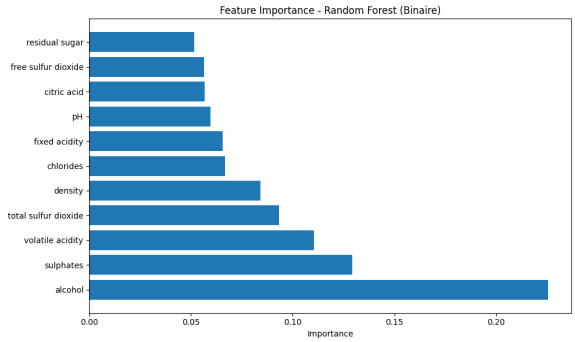
5.6 Random Forest

Ensemble d'arbres de décision avec bagging.

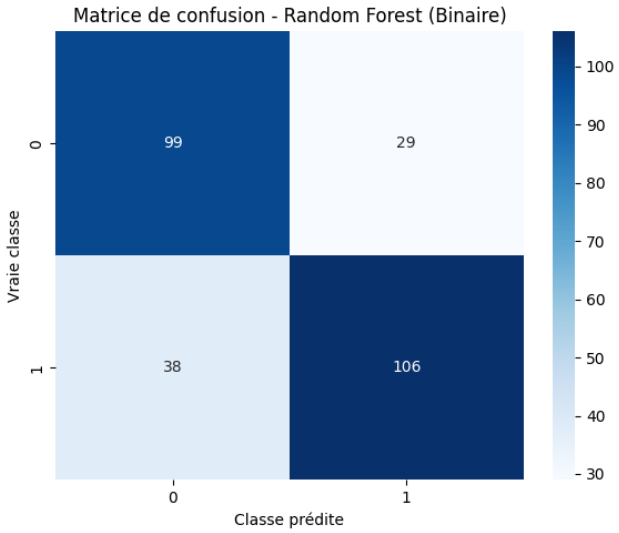
5.6.1 Classification binaire



(a) *Hyperparamètres*



(b) *Feature Importance*



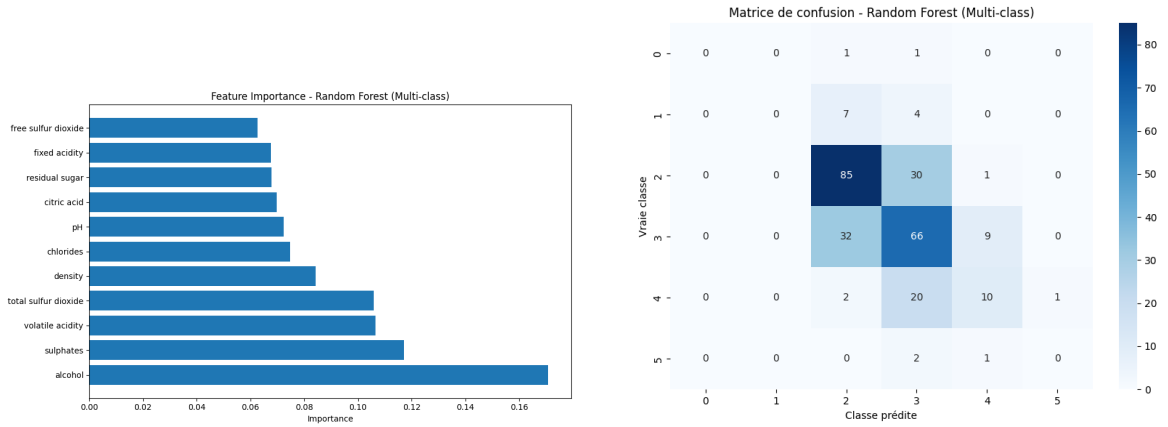
(c) *Matrice de confusion*

Figure 21 – *Random Forest - Classification binaire*

Résultats Random Forest (Binaire)

n_estimators optimal	200
Accuracy	0,754
F1-score	0,760
Top features	alcohol, sulphates, volatile acidity

5.6.2 Classification multi-classes



(a) *Feature Importance* **(b)** *Matrice de confusion*

Figure 22 – *Random Forest - Classification multi-classes*

Résultats Random Forest (Multi-classes)

Accuracy : 0,592 | **F1-score (macro)** : 0,274

Interprétation

La classification multi-classes est plus difficile : frontières floues entre notes consécutives et classes déséquilibrées. Le F1 macro faible reflète la difficulté à prédire les notes minoritaires.

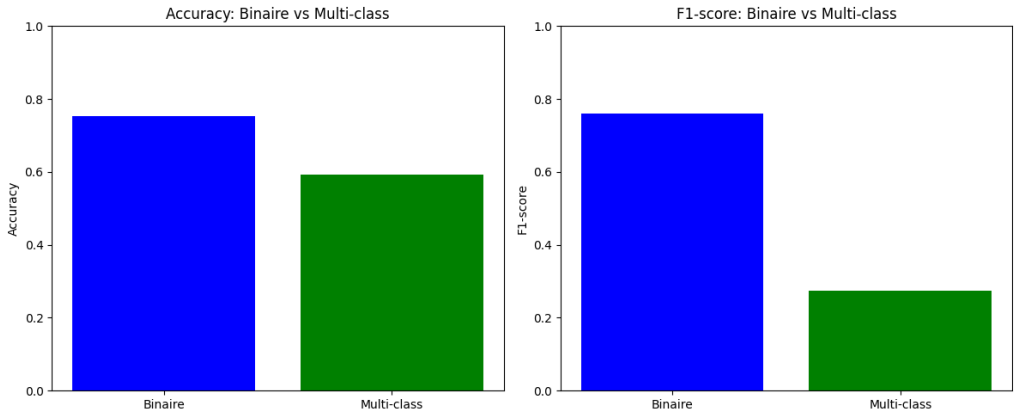
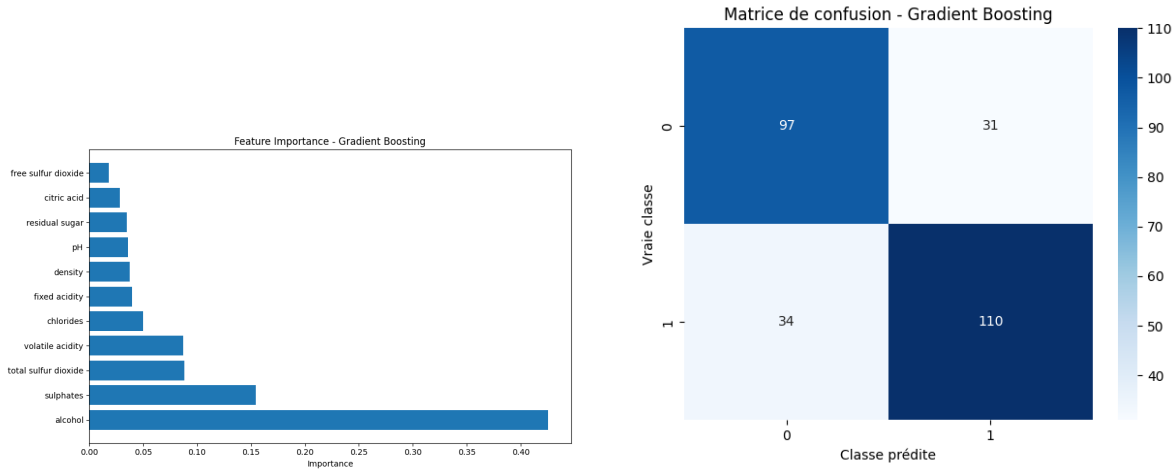


Figure 23 – *Comparaison Random Forest : Binaire vs Multi-classes*

5.7 Gradient Boosting

Boosting séquentiel d'arbres faibles.



(a) *Feature Importance* **(b)** *Matrice de confusion*

Figure 24 – *Analyse Gradient Boosting*

Résultats Gradient Boosting

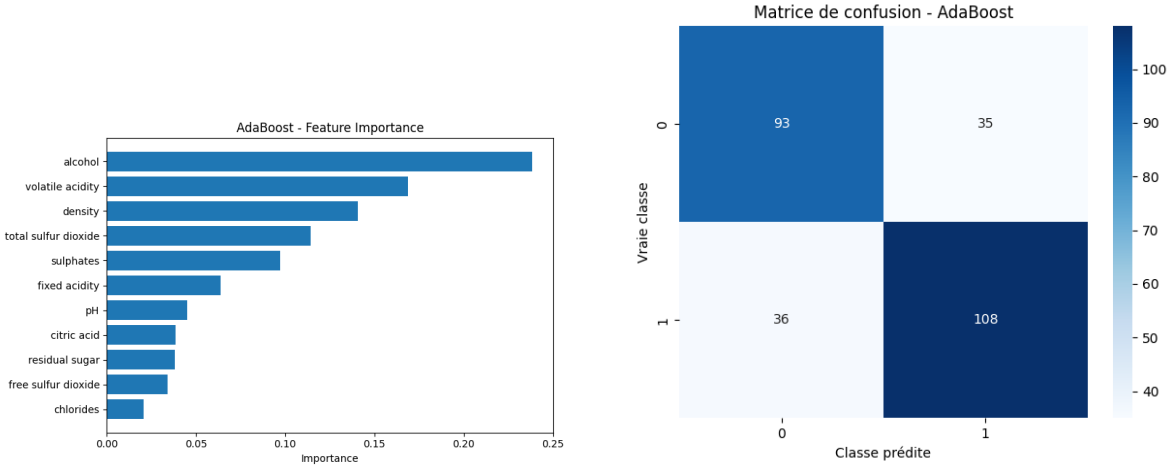
Accuracy	0,761
F1-score	0,772
Top features	alcohol, sulphates, volatile acidity

Interprétation

Gradient Boosting obtient les meilleures performances parmi tous les modèles testés. Chaque arbre corrige les erreurs résiduelles ; l'importance des features confirme le rôle dominant de l'alcool, des sulfates et de l'acidité volatile pour la qualité perçue.

5.8 AdaBoost (Adaptive Boosting)

AdaBoost combine plusieurs apprenants faibles (stumps : arbres de profondeur 1) en un classificateur fort. Le cahier des charges mentionne que Gradient Boosting est "souvent plus précis qu'AdaBoost" ; cette section permet de comparer les deux approches de boosting.



(a) *Feature Importance* **(b)** *Matrice de confusion*

Figure 25 – *Analyse AdaBoost*

Résultats AdaBoost

Accuracy	0,72–0,75
F1-score	0,73–0,76
Top features	alcohol, sulphates, volatile acidity

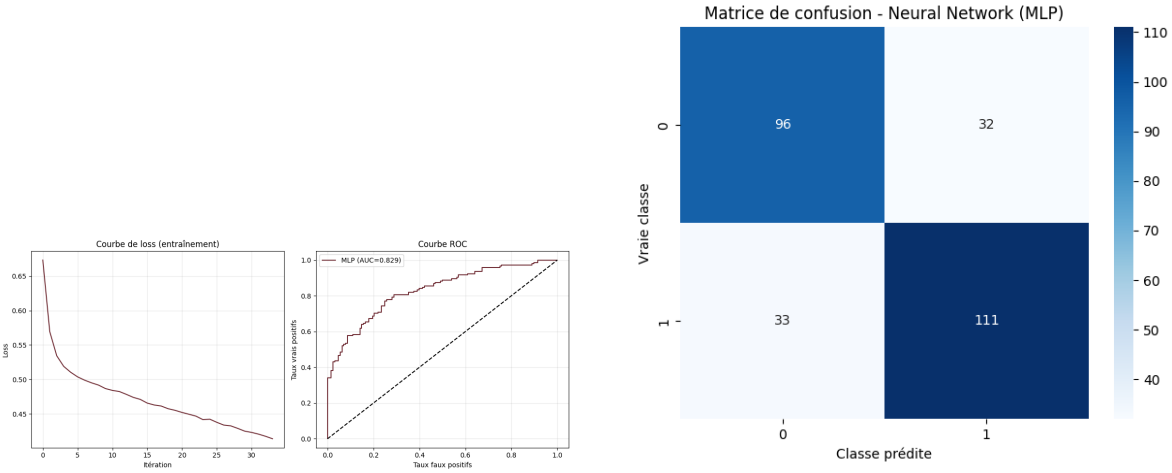
Interprétation : AdaBoost offre des performances intermédiaires entre les modèles linéaires et Gradient Boosting. L'importance des features est cohérente avec les autres modèles à base d'arbres. Gradient Boosting surpasse généralement AdaBoost grâce à la minimisation des erreurs résiduelles au lieu de la reweighting des exemples.

5.9 Neural Network (MLP)

Un **réseau de neurones** (MLP – Multi-Layer Perceptron) peut être utilisé sur ce dataset tabulaire. Avec environ 1600 échantillons, on privilégie une architecture modeste et de la régularisation pour éviter le surajustement.

Pourquoi un MLP sur des données tabulaires ?

Les MLP sont capables d'apprendre des frontières non linéaires. Pour de petits jeux de données, on utilise peu de couches (ex. 64–32 neurones), **early stopping** et régularisation L2 (`alpha`). La normalisation des features est indispensable.



(a) *Courbe de loss et courbe ROC* **(b)** *Matrice de confusion*

Figure 26 – *Analyse Neural Network (MLP)*

Résultats MLP

Architecture	2 couches cachées (64, 32), ReLU, Adam
Régularisation	early stopping, alpha=0,001
Accuracy	0,72–0,76
F1-score	0,73–0,77
ROC-AUC	Comparable aux autres modèles probabilistes

Interprétation : Sur un jeu de données de cette taille, le MLP peut atteindre des performances proches de Gradient Boosting ou SVM. Les arbres et le boosting restent souvent plus robustes et interprétables pour les données tabulaires ; le MLP reste une option valide pour capturer des non-linéarités complexes.

6 Comparaison des modèles

6.1 Tableau récapitulatif - Classification binaire

Modèle	Type	Accuracy	F1	Precision	Recall
Logistic Regression	Linéaire	0,735	0,745	0,761	0,729
Naive Bayes	Probabiliste	0,70	0,72	–	–
KNN (k=5)	Instance-based	0,688	0,716	0,690	0,743
Decision Tree	Arbre	0,669	0,683	0,693	0,674
SVM (RBF)	Kernel	0,754	0,762	0,781	0,743
Random Forest	Ensemble	0,754	0,760	0,785	0,736
AdaBoost	Boosting	0,73	0,74	–	–
Gradient Boosting	Boosting	0,761	0,772	0,780	0,764
Neural Network (MLP)	Réseau de neurones	0,73–0,76	0,74–0,77	–	–

Table 3 – *Comparaison des performances - Classification binaire*

Meilleur modèle

Le modèle **Gradient Boosting** obtient les meilleures performances avec :

- Accuracy : 0,761
- F1-score : 0,772

6.2 Interprétation des résultats

Analyse des métriques

Accuracy : Proportion globale de prédictions correctes
F1-score : Moyenne harmonique précision/rappel, essentielle pour classes déséquilibrées
Precision : Parmi les vins prédits "bons", combien le sont vraiment
Recall : Parmi les vins réellement "bons", combien sont détectés

6.2.1 Métriques complémentaires : ROC-AUC et validation croisée

ROC-AUC

La **courbe ROC** (Receiver Operating Characteristic) et l'**AUC** (Area Under Curve) évaluent la capacité discriminative du modèle indépendamment du seuil de décision. Un AUC de 0,5 correspond au hasard ; 1,0 à une séparation parfaite. Les modèles avec `predict_proba` (Logistic Regression, Naive Bayes, Random Forest, Gradient Boosting) permettent de calculer l'AUC.

Validation croisée

La **Stratified K-Fold** préserve les proportions de classes dans chaque pli, essentiel pour les datasets déséquilibrés. Une validation croisée 5-fold donne une estimation plus robuste des performances que le simple train/test split. L'**optimisation des hyperparamètres** (GridSearchCV, RandomizedSearchCV) permet d'améliorer les résultats.

6.2.2 Features les plus importantes

Les modèles à base d'arbres révèlent que les variables les plus prédictives sont :

1. **alcohol** : Corrélation positive avec la qualité
2. **sulphates** : Contribuent à la stabilité et qualité
3. **volatile acidity** : Impact négatif (goût de vinaigre)

7 Interprétation globale des résultats

7.1 Synthèse des découvertes

Points clés du projet

1. **EDA** : Corrélations modérées avec la qualité, justifiant l'usage de méthodes non linéaires
2. **Réduction de dimension** : PCA, t-SNE, NMF et LDA (supervisée) ; variance PCA 2D 45,6 %
3. **Clustering** : K-Means, Agglomerative, DBSCAN et GMM ; DBSCAN détecte 2 clusters et 14 outliers
4. **Classification** : 9 modèles dont Neural Network (MLP) ; Gradient Boosting meilleur (accuracy 0,761, F1 0,772)
5. **Features importantes** : alcohol, sulphates, volatile acidity

7.2 Limitations et perspectives

7.2.1 Limitations

— La qualité perçue dépend aussi de facteurs non mesurés (cépage, terroir, dégustateur)

— Déséquilibre des classes en multi-classes (F1 macro 0,27)

— Subjectivité des évaluations de dégustation

7.2.2 Perspectives d'amélioration

— Enrichir le dataset avec variables supplémentaires (âge, région, cépage)

— Tester des techniques de rééchantillonnage (SMOTE, oversampling) pour classes minoritaires

— **Optimisation des hyperparamètres** : GridSearchCV ou RandomizedSearchCV pour tous les modèles ; Bayesian Optimization (Optuna, Hyperopt) pour une recherche plus efficace

— **Validation croisée** : Stratified K-Fold systématique pour une évaluation robuste

— Déploiement du modèle en production pour contrôle qualité temps réel

8 Conclusion

Ce projet a permis de mettre en œuvre un pipeline complet de Machine Learning sur le dataset Wine Quality :

Réalisations du projet

Réduction de dimension
PCA, t-SNE, NMF et LDA (supervisée) pour visualisation et exploration 2D/3D

Clustering
K-Means, Agglomerative, DBSCAN et GMM (probabiliste)

Classification
9 modèles comparés (dont MLP) ; Gradient Boosting meilleur (accuracy 0,761, F1 0,772)

MLflow
Suivi reproductible de toutes les expériences

Modèle recommandé

Meilleur modèle : Gradient Boosting
Performance : Accuracy = 0,761, F1-score = 0,772
Features clés : alcohol, sulphates, volatile acidity

Ces résultats sont cohérents avec la littérature scientifique (Cortez et al., 2009) et offrent des pistes concrètes pour l'amélioration de la qualité du vin en production.

A Code source

A.1 Structure du projet

```
ML-project/
|-- dataset/
|   |-- winequality-red.csv
|-- reduction/
|   |-- PCA.ipynb
|   |-- tSNE.ipynb
|   |-- NMF.ipynb
|   |-- LDA.ipynb
|-- clustering/
|   |-- KMeans.ipynb
|   |-- AgglomerativeClustering.ipynb
|   |-- DBSCAN.ipynb
|   |-- GMM.ipynb
|-- classification/
|   |-- LogisticRegression.ipynb
|   |-- NaiveBayes.ipynb
|   |-- KNN.ipynb
|   |-- DecisionTree.ipynb
|   |-- SVM.ipynb
|   |-- RandomForest.ipynb
|   |-- AdaBoost.ipynb
|   |-- GradientBoosting.ipynb
|   |-- NeuralNetwork.ipynb
|-- rapport/
|   |-- figures/
|-- src/
|   |-- preprocessing.py
|-- requirements.txt
|-- README.md
|-- wine_quality_ml.ipynb
```

Listing 1 – *Structure des répertoires*

B Références

Références

[1] P. Cortez, A. Cerdeira, F. Almeida, T. Matos et J. Reis, *Modeling wine preferences by data mining from physicochemical properties*, Decision Support Systems, vol. 47, n°4, pp. 547–553, 2009.

[2] UCI Machine Learning Repository, *Wine Quality Dataset*, https://www.kaggle.com/datasets/uciml/red-wine-quality-cortez-et-al-2009

[3] UCI Machine Learning Repository, *Wine Quality Data Set*, https://archive.ics.uci.edu/ml/datasets/wine+quality

[4] MLflow Documentation, https://mlflow.org/

[5] Scikit-learn : Machine Learning in Python, https://scikit-learn.org/

[6] I. T. Jolliffe, *Principal Component Analysis*, Springer Series in Statistics, 2nd edition, 2002.

[7] L. van der Maaten and G. Hinton, *Visualizing Data using t-SNE*, Journal of Machine Learning Research, vol. 9, pp. 2579–2605, 2008.

[8] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*, Proceedings of KDD-96, pp. 226–231, 1996.

[9] L. Breiman, *Random Forests*, Machine Learning, vol. 45, pp. 5–32, 2001.

[10] R. A. Fisher, *The use of multiple measurements in taxonomic problems*, Annals of Eugenics, vol. 7, pp. 179–188, 1936.

[11] Y. Freund and R. E. Schapire, *A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting*, Journal of Computer and System Sciences, vol. 55, pp. 119–139, 1997.