UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

PRÉDICTION DE LA QUALITÉ DES VINS BLANCS ET ROUGES

Analyse des données en grandes dimensions

Projet dans le cadre du Master 2 TIDE

Khadija BEN TALHA - Elodie HUTIN - Célia LAZIZI

10 février 2024

Table des matières

1 Introduction **1**

2 Analyse exploratoire **3**
2.1 Présentation du jeu de données 3
2.2 Statistiques du jeu de données 5
2.3 Distributions des variables 11

3 Régression **19**
3.1 Régression Linéaire 19
3.2 Régression LASSO 22
3.3 Régression RIDGE 25
3.4 Random Forest Regressor 28
3.5 Ensemble des résultats 29
3.6 Analyse en Composantes Principales (ACP) 29

4 Classification **33**
4.1 Régression Logistique 35
4.2 Arbre de décision 36
4.3 Forêt Aléatoire 38
4.4 k-plus proches voisins (kNN) 39
4.5 Support Vector Machine (SVM) 41
4.6 Boosting 43
4.6.1 AdaBoost 44
4.6.2 Gradient Boosting 45
4.6.3 XGBoost 47
4.7 Ensemble des résultats 48

5 CONCLUSION **49**

1 Introduction

En 2023, la France a retrouvé sa position de premier producteur mondial de vin, dépassant l'Italie avec une production estimée à près de 47 millions d'hectolitres, selon les données du ministère français de l'Agriculture. Cette pôle position témoigne de l'importance du vin dans la culture française. Au-delà d'une simple tradition, la consommation de vin en France se présente comme une véritable célébration de l'art de vivre et de la gastronomie. Que l'on opte pour un rouge robuste et plein de caractère ou un blanc délicat et rafraîchissant, le vin accompagne très souvent les moments de convivialité des Français, enrichissant ainsi ces instants par sa qualité ou sa saveur unique.

La qualité du vin, élément essentiel de cette expérience sensorielle, demeure une quête perpétuelle pour les amateurs et les experts. Le vin est une boisson alcoolisée, qui est produite par la fermentation des raisins. La fermentation définit le processus par lequel les sucres naturels présents dans les raisins sont transformés en alcool et en dioxyde de carbone par l'action des levures. Ainsi la croissance de la vigne, la sélection du raisin, la mise en oeuvre de la fermentation et la conservation des vins, sont tous des facteurs qui influent directement sur le type et sur la qualité du vin final obtenu. Le vin rouge est produit à partir de raisins rouges foncés et noirs. La couleur varie généralement entre différentes nuances de rouge, de brun et de violet. Il est produit à partir de raisins entiers, y compris la peau, ce qui ajoute à la couleur et à la saveur des vins rouges et leur donne un goût riche. Tandis que le vin blanc est produit à partir de raisins blancs sans peau ni pépins. La couleur est généralement jaune paille, jaune-vert ou jaune-or. La plupart des vins blancs ont une saveur légère et fruitée par rapport aux vins rouges plus riches.

Chaque vin présente des propriétés sensorielles différentes qui sont définis par son profil chimique, caractérisé par des composants chimiques et des caractéristiques physiques. Ces propriétés physico-chimiques peuvent être exploitées pour élaborer des modèles de prédiction de la qualité du vin.

C'est donc dans ce contexte que s'inscrit notre analyse et étude sur la prédiction de la qualité d'un vin. Nous disposons de deux ensembles de données, détaillant respectivement 11 caractéristiques chimiques des vins blancs et rouges. Nous utiliserons donc ces jeux de données afin de comprendre les facteurs qui influent sur la qualité du vin et d'améliorer les prédictions de qualité à l'aide de méthodes d'apprentissage statistique. L'objectif de notre étude est de mener une recherche approfondie, en mettant en avant les réussites et les échecs. Notre question centrale explore la possibilité de prédire la qualité d'un vin en fonction de ses caractéristiques, et nous nous efforcerons de répondre de manière rigoureuse en ajustant notre analyse en fonction des résultats obtenus.

Nous allons ainsi débuter notre étude par une analyse exploratoire appronfondie de nos données. Cette démarche est nécessaire afin de comprendre la complexité des caractéristiques chimiques présentes dans les vins blancs et rouges que nous étudions. Elle nous permettra d'identifier des tendances, des corrélations et des points d'intérêt initiaux, jetant ainsi les bases d'une investigation plus approfondie sur les facteurs qui influent sur la qualité du vin.

Le paramètre de qualité que nous essayons de prédire est une variable ordinale (une variable catégorielle ordonnée). Une variable ordinale est un type de variable catégorielle ; cependant, elle se situe quelque peu entre une variable catégorielle et une variable quantitative continue en raison de l'aspect ordonné. Ains une variable ordinale peut souvent être traitée comme une variable continue, d'un point de vue statistique. Cette ambiguïté apparente soulève immédiatement la question de savoir s'il faut utiliser un modèle de classification ou de régression pour prédire la qualité. Pour examiner cette question, nous allons alors réaliser les deux types de modèle, afin de déterminer

lequel est le plus adaptée pour notre prédiction.

Ainsi la première phase de notre étude sera dédiée à l'analyse des facteurs influencant la qualité du vin grâce à une régression. L'objectif fondamental sera d'établir les variables qui jouent un rôle dans la détermination de la qualité. Pour y parvenir, nous appliquerons des méthodes linéaire telles que la régression linéaire, le LASSO, et le RIDGE et des méthodes non linéaires telles que la forêt aléatoire afin de tenter d'ajuster un modèle de régression à nos données. Les régressions régularisées nous permettront d'identifier de manière rigoureuse les variables les plus déterminantes dans la prédiction de la qualité du vin. Nous comparerons l'ensemble des résultats obtenus avec ces différentes méthodes, grâce à des métriques que nous présenterons dans ctte partie. De plus, nous réaliserons une ACP afin de mieux visualiser les relations entre nos variables, et tenter de réduire nos dimensions pour améliorer nos modèles de prédiction.

La seconde phase de notre étude se concentrera sur les modèles de classification, avec une approche où la variable cible sera traitée comme binaire. Au cœur de cette section, nous aborderons une nouvelle problématique : la capacité à classifier les vins de haute qualité comme étant excellents. Pour cela, nous explorerons plusieurs modèles de classification et évaluerons leurs performances, dans le but de sélectionner celui qui répond le mieux à notre problématique.

En conclusion, nous synthétiserons l'ensemble de nos résultats et nous efforcerons de répondre à notre problématique initiale concernant la prédiction de la qualité du vin.

2 Analyse exploratoire

2.1 Présentation du jeu de données

```
RangeIndex: 4898 entries, 0 to 4897
Data columns (total 12 columns):
 \#   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   fixed acidity         4898 non-null   float64
 1   volatile acidity      4898 non-null   float64
 2   citric acid           4898 non-null   float64
 3   residual sugar        4898 non-null   float64
 4   chlorides             4898 non-null   float64
 5   free sulfur dioxide   4898 non-null   float64
 6   total sulfur dioxide  4898 non-null   float64
 7   density               4898 non-null   float64
 8   pH                    4898 non-null   float64
 9   sulphates             4898 non-null   float64
 10  alcohol               4898 non-null   float64
 11  quality               4898 non-null   int64

dtypes: float64(11), int64(1)


RangeIndex: 1599 entries, 0 to 1598
Data columns (total 12 columns):
 \#   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   fixed acidity         1599 non-null   float64
 1   volatile acidity      1599 non-null   float64
 2   citric acid           1599 non-null   float64
 3   residual sugar        1599 non-null   float64
 4   chlorides             1599 non-null   float64
 5   free sulfur dioxide   1599 non-null   float64
 6   total sulfur dioxide  1599 non-null   float64
 7   density               1599 non-null   float64
 8   pH                    1599 non-null   float64
 9   sulphates             1599 non-null   float64
 10  alcohol               1599 non-null   float64
 11  quality               1599 non-null   int64

dtypes: float64(11), int64(1)
```

Le jeu de données du vin rouge contient 1599 observations tandis que celui du vin blanc contient 4898 observations. Les 2 jeux de données présentent les mêmes variables, ainsi en ajoutant une colonne 'type' qui différenciera les vins rouges des vins blancs, nous pouvons les concaténer.

Afin de mieux comprendre nos jeux de données, et les résultats des modèles que nous allons implémenter, voici une description de l'ensemble des variables :

Fixed acidity (Acidité fixe, en g/dm³) : Le vin contient deux types d'acidité : fixe et volatile. L'acidité fixe représente la partie de l'acidité du vin qui s'évapore pas facilement et qui reste relativement stable au cours du temps.

Volatile acidity (Acidité volatile, en g/dm³) : Cette variable représente la partie de l'acide qui est non fixe, elle s'évapore plus facilement produisant ainsi des arômes volatils. Une forte teneur dans le vin peut donner un goût de vinaigre désagréable.

Citric acid (Acide citrique, en g/dm³) : L'acide citrique est un acide organique présent naturellement dans les raisins. Ce composant agit comme conservateur en augmentant l'acidité. Présent en petites quantités, il peut ajouter de la fraîcheur et de la saveur aux vins.

Residual sugar (Sucre résiduel, en g/dm³) : Il s'agit de la quantité de sucre restant après l'arrêt de la fermentation. La fermentation transforme les sucres présents dans le jus de raisin en alcool et dioxyde de carbone, sous l'action des levures. Cependant, si la fermentation est arrêtée avant que tous les sucres ne soient convertis en alcool alors une partie du sucre reste dans le vin ce qui forme le sucre résiduel. L'essentiel est d'obtenir un équilibre parfait entre le sucré et l'acidité. Il est rare de trouver des vins avec moins de 1 g/litre et les vins avec plus de 45 g/litre sont considérés comme sucrés.

Chlorides (Chlorures, en g/dm³) : Les chlorures font référence à la concentration d'ions chlorure présentes dans le vin, issus du processus de vinification. Cette composante représente une façon de mesurer la teneur en sel d'un vin. Ce sel provient de la présence naturelle dans les raisins qui varie en fonction des conditions géologiques et climatiques du vignoble, et sa teneur dans le vin dépend du processus de fermentation.

Total sulfur dioxide (Dioxyde de soufre total, en mg/dm³) : Le dioxyde de soufre (ou SO) est un additif important dans les vins où il agit comme conservateur, à la fois comme antioxydant et comme antimicrobien. Il est présent dans le vin à la fois sous forme libre et sous forme liée. Le dioxyde de soufre total est simplement la somme des formes libres et liées. Les vins rouges nécessitent généralement moins de dioxyde de soufre, contrairement aux vins blancs, car la peau des raisins est conservée lors de leur production, et celle-ci contient du tanin, qui est un antimicrobien naturel.

Free sulfur dioxide (Dioxyde de soufre libre, en mg/dm³) : Cette variable représente la forme libre du dioxyde de soufre, qui est sous forme gazeuse ou dissoute dans le vin. Elle agit en tant que conservateur, et empêche la croissance de micro-organismes indésirables et l'oxydation du vin.

Density (Densité, en g/cm³) : La densité du vin est proche de celle de l'eau en fonction du pourcentage d'alcool et de la teneur en sucre. les vins plus sucrés ont une densité plus élevée.

pH : Le Ph décrit le niveau d'acidité sur une échelle de 0 à 14. La plupart des vins se situent toujours entre 3 et 4 sur l'échelle du pH.

Alcohol (Alcool, en % vol.) : Cette variable représente le pourcentage d'alcool contenu dans le vin.

Sulfure (Sulfates, en g/dm³) : Le sulfate est un additif pour le vin qui contribue aux niveaux de dioxyde de soufre (SO) et agit comme un antimicrobien et un antioxydant.

Quality (Qualité) : Cette variable, qui est notre variable cible, représente une note entre 0 et 10 sur la qualité du vin, basé sur des données sensorielles.

2.2 Statistiques du jeu de données

Voici les statistiques concernant le jeu de données du vin blanc :

	Informations	Valeurs
0	Nombre de lignes	4898
1	Nombre de colonnes	12
2	Variables quantitatives	12
3	Variables qualitatives	0
4	Valeurs manquantes	0
5	Valeurs nulles	19
6	Doublons	937

Voici les statistiques concernant le jeu de données du vin rouge :

	Informations	Valeurs
0	Nombre de lignes	1599
1	Nombre de colonnes	12
2	Variables quantitatives	12
3	Variables qualitatives	0
4	Valeurs manquantes	0
5	Valeurs nulles	132
6	Doublons	240

Le jeu de données des vins blancs comprend environ 4900 observations, tandis que celui des vins rouges en compte 1350, avec chacun comprenant 12 variables distinctes. Les deux jeux de données ne présentent aucune valeurs qualitatives, ou valeurs manquantes. Cependant il y a 937 observations en doublon pour le vin blanc et 240 pour le vin rouge, que l'on décide de supprimer ici car notre but est de prédire la qualité du vin pour une nouvelle observation, et nous souhaitons construire un modèle généralisé. Conserver ces doublons pourraient introduire un biais dans nos modèles ou bien provoquer un surajustement.

Afin de mieux comprendre les caractéristiques de nos jeux de données, nous dressons les statistiques descriptives pour chacune de nos variables. Cela nous permettra d'avoir une première vue des caractéristiques de nos données, en visualisant la tendance centrale, la dispersion, la forme de la distribution, et d'autres propriétés importantes. De plus, les statistiques descriptives nous aident à identifier la nécessité de nettoyer ou de prétraiter nos données. Cela peut inclure la gestion des valeurs manquantes, la transformation de variables, ou la normalisation des données pour rendre l'analyse plus robuste.

	fixed acidity	volatile acidity	citric acid	residual sugar	\
count	3961.00	3961.00	3961.00	3961.00	
mean	6.84	0.28	0.33	5.91	
std	0.87	0.10	0.12	4.86	
min	3.80	0.08	0.00	0.60	
25\%	6.30	0.21	0.27	1.60	
50\%	6.80	0.26	0.32	4.70	
75\%	7.30	0.33	0.39	8.90	
max	14.20	1.10	1.66	65.80	

	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	\
count	3961.00	3961.00	3961.00	3961.00	3961.00	
mean	0.05	34.89	137.19	0.99	3.20	
std	0.02	17.21	43.13	0.00	0.15	
min	0.01	2.00	9.00	0.99	2.72	
25\%	0.04	23.00	106.00	0.99	3.09	
50\%	0.04	33.00	133.00	0.99	3.18	
75\%	0.05	45.00	166.00	1.00	3.29	
max	0.35	289.00	440.00	1.04	3.82	

	sulphates	alcohol	quality
count	3961.00	3961.00	3961.00
mean	0.49	10.59	5.85
std	0.11	1.22	0.89
min	0.22	8.00	3.00
25\%	0.41	9.50	5.00
50\%	0.48	10.40	6.00
75\%	0.55	11.40	6.00
max	1.08	14.20	9.00

Pour les vins blancs, on note que les valeurs maximales du chlorides, du sucre résiduel et du dioxyde de soufre libre sont bien supérieurs à la moyenne de ces variables. Cela suggère la présence de valeurs très extrêmes. De plus, les variables du sucre résiduel et du dioxyde de soufre libre et total présentent une dispersion élevée, ce qui suggère une variabilité significative dans les niveaux de ces variables, allant de faibles à élevés. On remarque que l'alcool dans le vin blanc a un taux moyen de 10.59%, et une valeur minimale de 8% et maximale de 14.20%. Les notes de qualité du vin blanc sont entre 3 et 9, avec une moyenne autour de 5.85. La moyenne de l'acidité fixe est d'environ 6.84, avec une faible dispersion - écart type de 0.87. Cela suggère une concentration moyenne d'acidité fixe dans les vins analysés.

	fixed acidity	volatile acidity	citric acid	residual sugar	\
count	1359.00	1359.00	1359.00	1359.00	
mean	8.31	0.53	0.27	2.52	
std	1.74	0.18	0.20	1.35	
min	4.60	0.12	0.00	0.90	
25\%	7.10	0.39	0.09	1.90	
50\%	7.90	0.52	0.26	2.20	
75\%	9.20	0.64	0.43	2.60	
max	15.90	1.58	1.00	15.50	

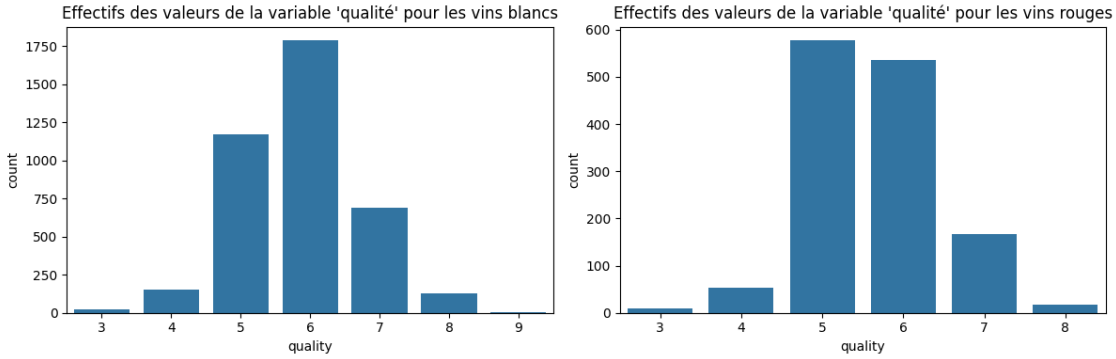
	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	\
count	1359.00	1359.00	1359.00	1359.00	1359.00	
mean	0.09	15.89	46.83	1.00	3.31	
std	0.05	10.45	33.41	0.00	0.16	
min	0.01	1.00	6.00	0.99	2.74	
25\%	0.07	7.00	22.00	1.00	3.21	
50\%	0.08	14.00	38.00	1.00	3.31	
75\%	0.09	21.00	63.00	1.00	3.40	
max	0.61	72.00	289.00	1.00	4.01	

	sulphates	alcohol	quality
count	1359.00	1359.00	1359.00
mean	0.66	10.43	5.62
std	0.17	1.08	0.82
min	0.33	8.40	3.00
25\%	0.55	9.50	5.00
50\%	0.62	10.20	6.00
75\%	0.73	11.10	6.00
max	2.00	14.90	8.00

Pour les vins rouges, les variables chlorides, sucre résiduel, dioxyde de soufre libre et dioxyde de soufre total présentent des tendances similaires à celles des vins blancs. Néanmoins, il est important de noter que les valeurs moyennes des dioxydes de soufre libre et total sont significativement inférieures à celles observées pour les vins blancs, renforçant ainsi notre affirmation antérieure : les vins rouges exigent une moindre quantité de dioxyde de soufre. On remarque que les vins rouges présentent un taux moyen d'alcool équivalent à celui des vins blancs, avec des valeurs minimales et maximales légérement plus élevées. Les notes de qualités du vin rouge vont de 3 à 8. Les valeurs d'acidité fixe et volatile sont différentes entre les deux types de vins tandis que celles de la densité et du pH sont très similaires.

En résumé, nous observons que la plupart des caractéristiques physiques et chimiques ne présentent pas de variations significatives en fonction du type de vin (rouge ou blanc). Les rares variables qui présentent des variations ne sont pas fortement divergentes. Il semble donc envisageable de mener une analyse combinée des deux types de vins. Cependant, avant de prendre une décision définitive, nous allons poursuivre notre exploration des variables, en particulier en examinant la relation entre la variable cible et chacune des variables indépendantes.

Après avoir explorer les caractéristiques de la variable "qualité", nous allons approfondir l'analyse de notre variable cible. À cette fin, nous examinons la répartition des valeurs de la variable 'quality' tant pour les vins blancs que pour les vins rouges.



Les données de notre variable cible, présentent un déséquilibre, avec certaines valeurs plus fréquentes que d'autres, aussi bien pour les vins rouges que les vins blancs. Afin de mieux représenter la distribution de cette variable, il peut être bénéfique de regrouper certaines classes pour obtenir une répartition plus équilibrée.

Nous avons regroupé les vins des catégories 3, 4 et 5 dans une première catégorie, tandis que les catégories 7, 8 et 9 ont été regroupées dans une troisième catégorie. Puis, en raison de son effectif important, la catégorie 6 a été maintenue dans une catégorie distincte. Voici comment les observations des deux jeux de données sont réparties entre les trois nouvelles catégories de qualité de vin :

	Quality	Count_Vin_Blanc	Count_Vin_Rouge
1	1	0.340318	0.470935
0	2	0.451401	0.393672
2	3	0.208281	0.135394

Les trois catégories sont plus équilibrées que les sept catégories initiales, et nous conserverons celles-ci pour la suite de notre analyse. Les proportions de vins dans chaque catégorie de qualité sont relativement similaires entre les deux types de vin. Ainsi, la distribution de la qualité semble uniforme, indiquant une similitude dans la manière dont le type de vin influence la qualité attribuée par les experts. Il n'y a pas de distinction marquée entre la qualité et le type de vin, suggérant que chaque type de vin n'est pas systématiquement associé à une meilleure qualité. Cette observation suggère donc qu'une analyse combinée des types de vin pour la prédiction de la qualité est possible.

Pour tester cette observation de manière statistique, nous utiliserons le test de Kruskal-Wallis. Cette technique statistique non paramétrique est employée pour comparer les moyennes de trois groupes ou plus, permettant de déterminer s'il existe des différences significatives entre eux. Le test de Kruskal-Wallis classe les données de tous les groupes ensemble, attribue des rangs, puis calcule une statistique de test basée sur les rangs. La statistique de test suit une distribution de chi carré. Dans notre cas, notre variable cible se compose de trois groupes, et nous souhaitons vérifier si le type de vin est significatif par rapport à la qualité du vin.

La p-value obtenue par le teste de Kruskal-Wallis indique la probabilité d'observer les différences entre les groupes par simple hasard. Si la valeur p est inférieure à notre seuil défini à 0.05, nous rejetterons l'hypothèse nulle, suggérant ainsi l'existence de différences significatives entre au moins deux groupes.

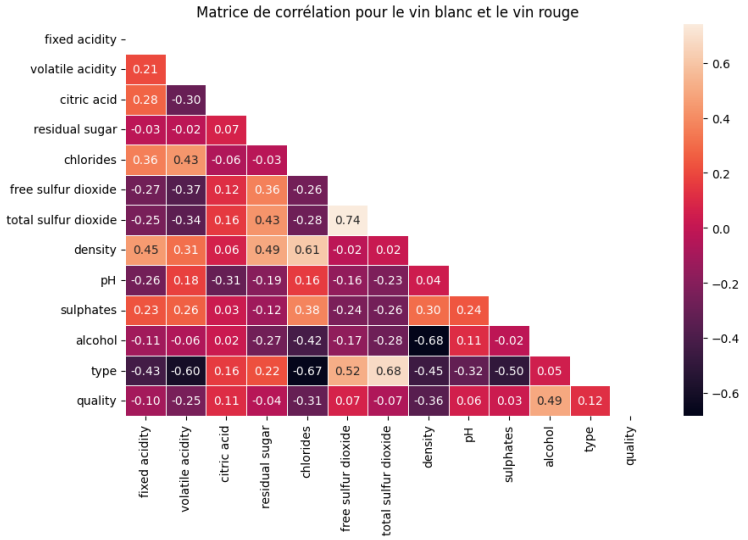
Après avoir réuni les deux jeux de données sur le vin blanc et le vin rouge en ajoutant une variable 'type' encodée à 1 pour le vin blanc et à 0 pour le vin rouge, nous avons mis en place le test de Kruskal-Wallis sur l'ensemble de nos variables indépendantes :

```
fixed acidity: p-value Kruskal-Wallis = 3.0292509154858305e-13
volatile acidity: p-value Kruskal-Wallis = 5.012014562527055e-81
citric acid: p-value Kruskal-Wallis = 2.6088693506487543e-15
residual sugar: p-value Kruskal-Wallis = 0.004346026036980001
chlorides: p-value Kruskal-Wallis = 2.3993271454955904e-113
free sulfur dioxide: p-value Kruskal-Wallis = 4.900699980377465e-08
total sulfur dioxide: p-value Kruskal-Wallis = 4.0766202035192086e-07
density: p-value Kruskal-Wallis = 7.530218089147531e-153
pH: p-value Kruskal-Wallis = 7.230313546747315e-05
sulphates: p-value Kruskal-Wallis = 0.043432503581295016
alcohol: p-value Kruskal-Wallis = 1.619293786635417e-283
type: p-value Kruskal-Wallis = 1.382447761902912e-18
```

Nous constatons que toutes nos variables, en particulier la variable "type", ont un impact significatif sur notre variable cible, puisque l'ensemble des p-values sont inférieures à 0.05. Cela suggère que toutes les variables peuvent être maintenues à ce stade, et que nous avons la possibilité d'utiliser le "type" de vin en tant que prédicteur supplémentaire pour évaluer la qualité du vin. Nous travaillons donc maintenant sur une base de données unifiée qui inclut les deux types de vins, distingués par l'introduction de la variable "type".

Matrice de corrélation entre les caractéristiques : Désormais, nous pouvons tenter de visualiser les relations linéaires entre nos variables, grâce au coefficient de corrélation de Pearson. Ce coefficient, dont la formule est $r = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum (X_i - \bar{X})^2 \sum (Y_i - \bar{Y})^2}}$, mesure la force de la relation entre deux variables et leur association l'une avec l'autre. Il prend ses valeurs dans l'intervalle [-1,1] où -1 et 1 représentent une corrélation parfaite respectivement négative et positive entre les variables considérées, et 0 l'absence de corrélation.

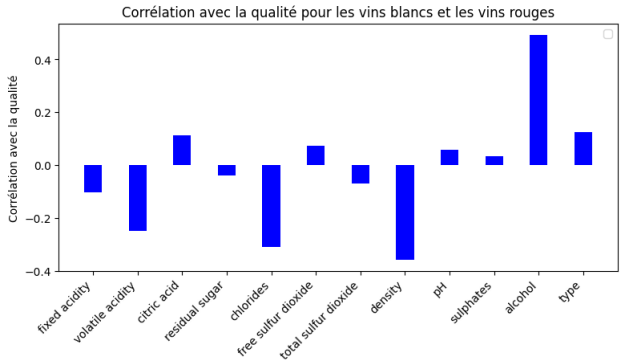
Nous présentons ainsi les coefficients de Pearson dans une matrice de corrélation afin de visualiser la présence de relation linéaire entre les caractéristiques du vin, et également de détecter la multicolinéarité, qui peut poser problème dans les modèles linéaires.



On observe une corrélation très importante entre :

- Densité et Alcool : -0.68
- Densité et les Chlorures : 0.61
- Type et Acidité Volatile : -0.60
- Type et Chlorides : -0.67
- Type et Dioxyde de soufre total : 0.68
- Type et Sulfure : -0.50
- Dioxyde de soufre libre et Dioxyde de soufre total : 0.74. Ceci est sûrement dû à la relation de ces deux variables dans la fabrication du vin, comme nous l'avons expliqué précédemment. Ainsi il est logique que ces deux variables soient très corrélées, sans pour autant signifier qu'il y a une relation de cause à effet, mais une association statistique.

De plus, étant donné que notre objectif est de prédire la qualité, il est crucial d'observer les corrélations des caractéristiques avec notre variable cible. Toutefois, comme nous l'avons observé dans le graphique précédent, il n'y a pas de dépendance substantielle entre la plupart des caractéristiques et la qualité. Regardons tout de même ces corrélations plus en détails :



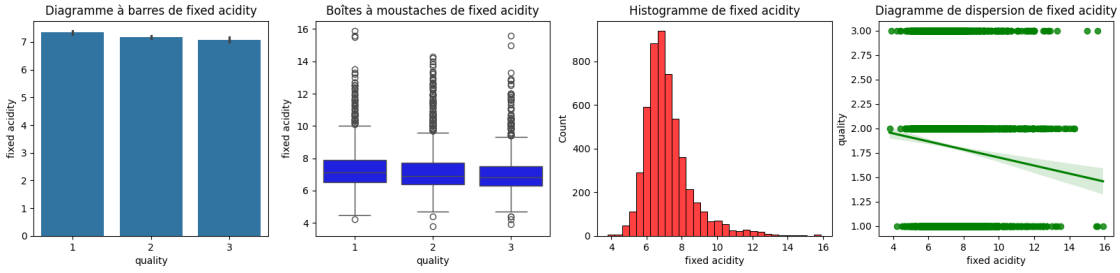
La variable alcool semble présenter la plus grande corrélation avec la qualité du vin (0.49). Les trois autres caractéristiques qui possédent une corrélation légérement plus élevée que les autres sont la densité (-0.36), les chlorures (-0.31) et l'acidité volatile (-0.25).

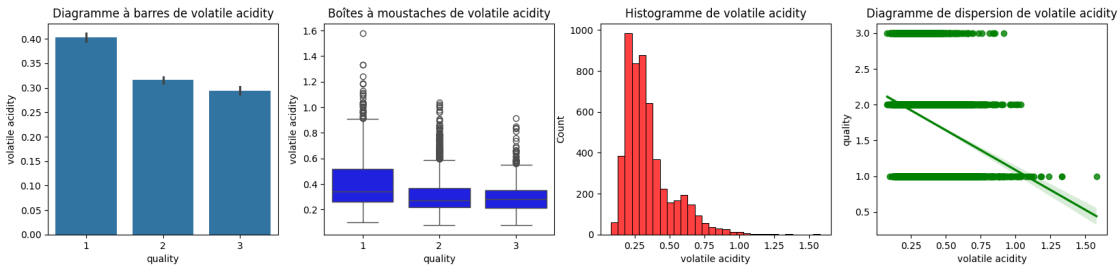
Ces corrélations semblent indiquer qu'il existe une relation entre la variable qualité et les caractéristiques mentionnées. Cependant, étant donné l'absence de corrélations très fortes, on peut se questionner sur la pertinence d'une représentation linéaire pour décrire au mieux l'interaction entre la qualité et les 11 caractéristiques d'un vin.

2.3 Distributions des variables

Pour continuer notre analyse exploratoire, nous avons créé une fonction permettant d'avoir la distribution de chaque variable, le diagramme de dispersion entre la variable dépendante et chacune des variables indépendantes, les boîtes à moustaches, et les diagrammes à barre pour notre jeu de données. L'analyse visuelle de nos données à travers chacun de ces graphiques est indispensable, et voici ce que chaque graphique permet d'observer :

— **Le diagramme en barres** représente la moyenne des valeurs de chaque variable dépendante pour chacune des catégories de la variable cible 'quality'. Cela nous permet d'identifier d'éventuelles variations significatives dans les valeurs des caractéristiques du vin entre les différentes qualités de vin.
— **La boîte à moustache** représente la distribution des données, à travers les statistiques générales telles que la médiane, les quartiles supérieurs et inférieures, et les valeurs aberrantes. Ici chaque box-plot permet visualiser la distribution des valeurs de chaque variable indépendante pour les trois catégories de qualité de vin, et on peut ainsi identifier les variations dans la distribution des caractéristiques en fonction de la qualité. De plus, il est facile de détecter quelle variable présente des valeurs aberrantes ou extrêmes pour chaque catégorie de qualité.
— **L'histogramme** est un outil de visualisation important pour l'analyse exploratoire de nos données. Nous pouvons l'utiliser ici afin d'obtenir une représentation visuelle des distributions de nos variables.
— **Le diagramme de dispersion** permet de visualiser la relation entre la variable dépendante (quality) et chacune des variables indépendantes. Il permet de voir rapidement s'il existe une linéarité, une tendance ou une corrélation.





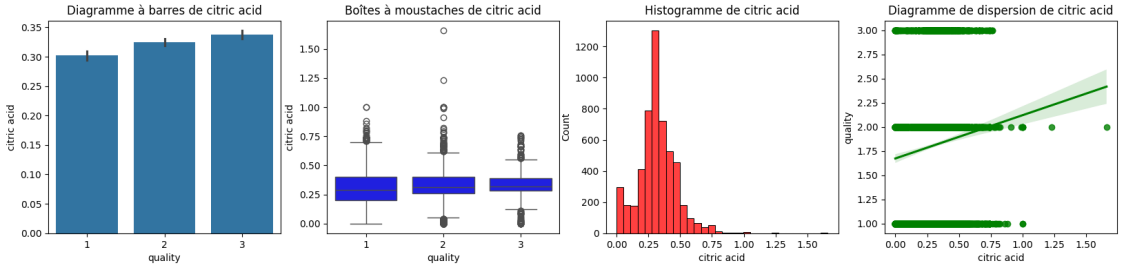


Diagramme à barres de citric acid
citric acid
quality
Boîtes à moustaches de citric acid
citric acid
quality
Histogramme de citric acid
Count
citric acid
Diagramme de dispersion de citric acid
quality
citric acid

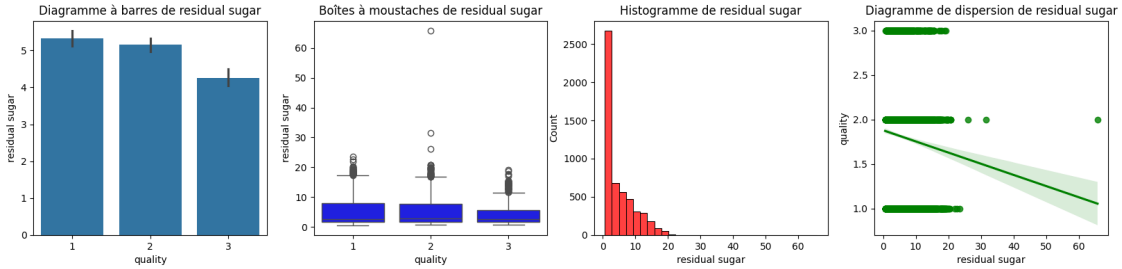


Diagramme à barres de residual sugar
residual sugar
quality
Boîtes à moustaches de residual sugar
residual sugar
quality
Histogramme de residual sugar
Count
residual sugar
Diagramme de dispersion de residual sugar
quality
residual sugar

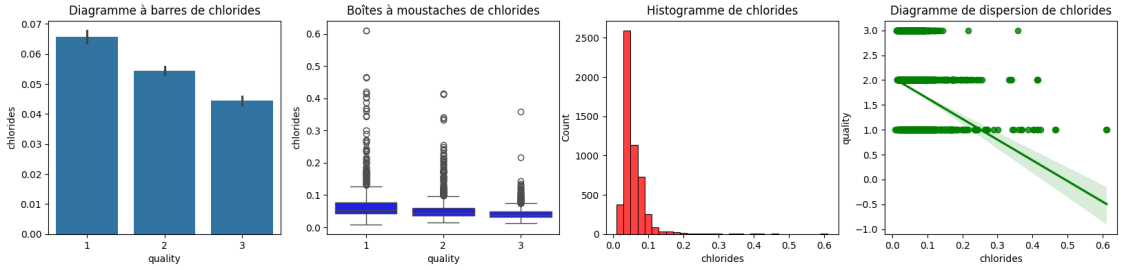


Diagramme à barres de chlorides
chlorides
quality
Boîtes à moustaches de chlorides
chlorides
quality
Histogramme de chlorides
Count
chlorides
Diagramme de dispersion de chlorides
quality
chlorides

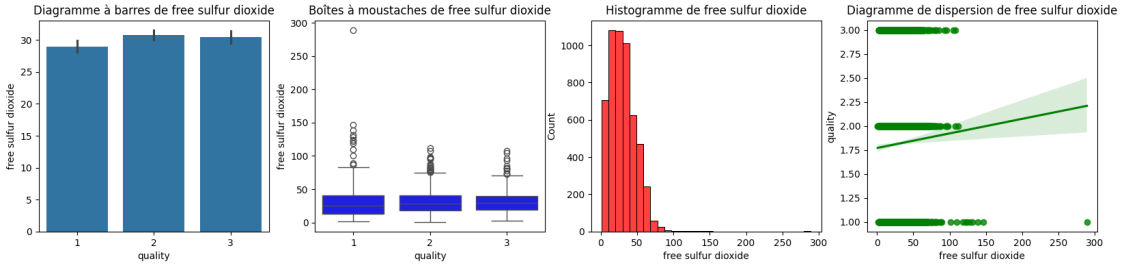


Diagramme à barres de free sulfur dioxide
free sulfur dioxide
quality
Boîtes à moustaches de free sulfur dioxide
free sulfur dioxide
quality
Histogramme de free sulfur dioxide
Count
free sulfur dioxide
Diagramme de dispersion de free sulfur dioxide
quality
free sulfur dioxide

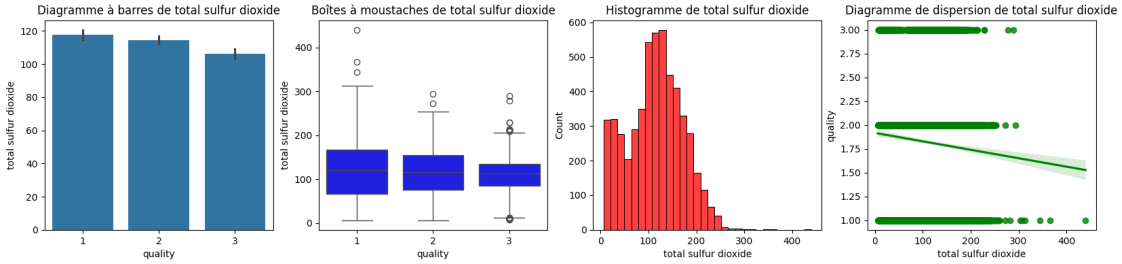


Diagramme à barres de total sulfur dioxide
Boîtes à moustaches de total sulfur dioxide
Histogramme de total sulfur dioxide
Diagramme de dispersion de total sulfur dioxide
total sulfur dioxide
quality
Count

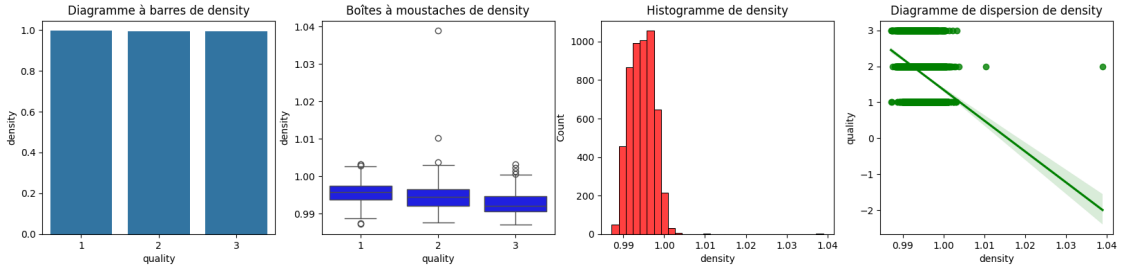


Diagramme à barres de density
Boîtes à moustaches de density
Histogramme de density
Diagramme de dispersion de density
density
quality
Count

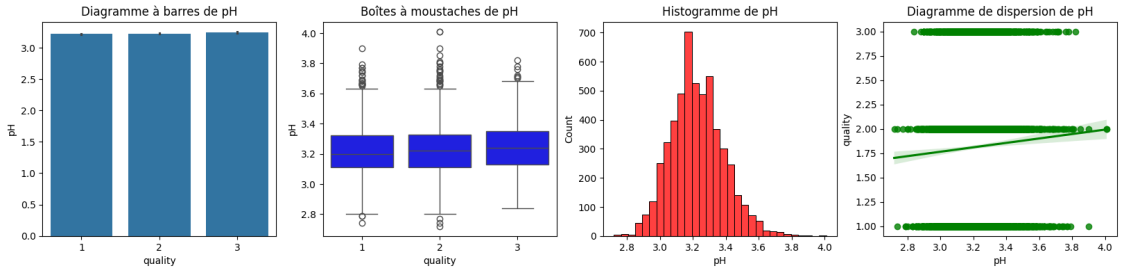


Diagramme à barres de pH
Boîtes à moustaches de pH
Histogramme de pH
Diagramme de dispersion de pH
pH
quality
Count

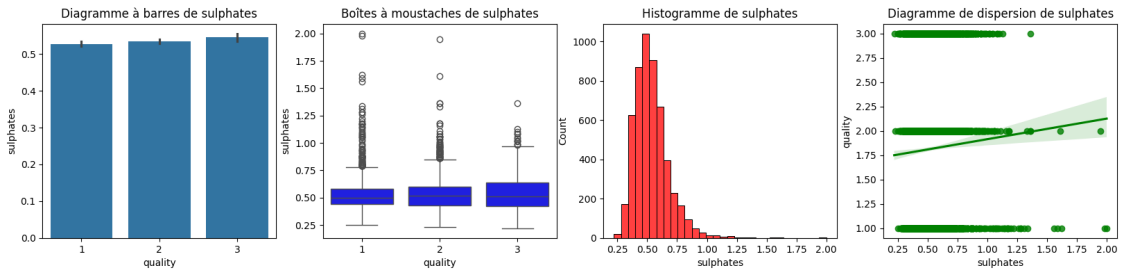
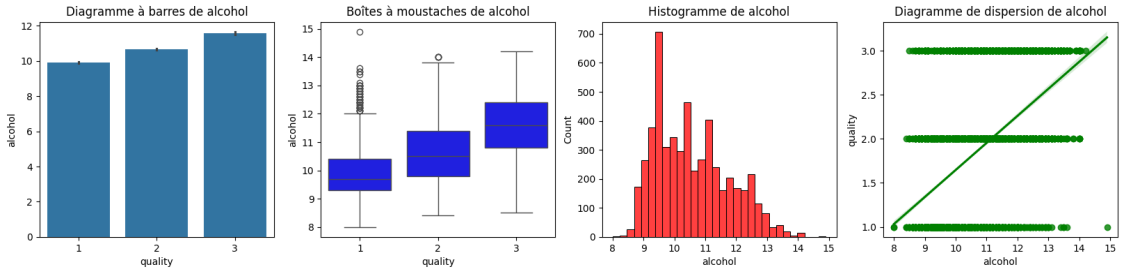


Diagramme à barres de sulphates
Boîtes à moustaches de sulphates
Histogramme de sulphates
Diagramme de dispersion de sulphates
sulphates
quality
Count



A partir de ces visualisations, voici les conclusions sur les distributions du jeu de données du vin que nous pouvons établir :

- **Le diagramme en barres** : La plupart des variables ont une moyenne a peu près équivalente pour chaque qualité de vin, sauf pour l'alcool, le dioxyde de soufre total, les chlorures, l'acidité volatile et le sucre résiduel.
- **La boîte à moustache** : On peut noter ici que toutes nos variables, sauf la densité présentent un très grand nombre de valeurs aberrantes. Il sera par la suite nécessaire de les traiter.
- **L'histogramme** : On observe ici que nos données ont des échelles de valeurs différentes, et qu'il sera nécessaire, pour les modèles basés sur la distance, de standardiser nos données. De plus, on observe que pour la plupart de nos variables (exceptés alcool, pH, acide citrique, dioxyde de sulfure total), les distributions sont étirées vers la droite, et présentent donc une asymétrie positive. Ceci peut s'expliquer par la présence de valeurs aberrantes Nous devrons donc étudier la skewness, et faire potentiellement des changements
- **Le diagramme de dispersion** : Dans notre cas, nous ne pouvons déterminer de relation linéaire claire entre la variable cible qualité et les caractéristiques. Cependant, une association très légèrement négative semble perceptible entre la qualité et les chlorides, et une autre positive entre la qualité et l'alcool. Ce graphique nous permet également de visualiser les valeurs aberrantes ou extrêmes qui sont représentés par les points isolés en dehors de la masse principale.

On remarque donc que les distributions des caractéristiques que les variables indépendantes ne présentent pas clairement une relation linéaire avec la variable dépendante.

Les histogrammes présentant des distributions asymétriques, nous nous intéressons aux valeurs de la skewness des variables de nos données :

	skewness
fixed acidity	1.649952
volatile acidity	1.504133
citric acid	0.484172
residual sugar	1.706069
chlorides	5.336732
free sulfur dioxide	1.362335
total sulfur dioxide	0.063596
density	0.666138
pH	0.389859
sulphates	1.808944
alcohol	0.545542
type	-1.121489
quality	0.297236

On observe que toutes les variables, exceptés l'acide citrique, le dioxyde de sulfure total, le pH et l'alcool, ont une skewness élevée (>0.5), ce qui équivaut à une asymétrie positive dans la distribution des données. Ce phénomène peut être dû à la présence de valeurs positives extrêmes, qu'il faudra certainement que l'on traite.

Une asymétrie importante peut entraîner des biais dans les estimations des paramètres. Nous faisons donc le choix de limiter cette asymétrie, en la corrigeant. Cela nous permettra d'obtenir des estimations plus précises et moins biaisées.

Nettoyage du set de données : À partir des box-plots précédents, nous avons pu observer qu'un grand nombre de variables indépendantes présentaient des valeurs aberrantes ou extrêmes. Il est alors important de les traiter, car dans notre cas de prédiction, elles peuvent avoir un impact sur la capacité du modèle à généraliser à de nouvelles données. Plusieurs choix pour le traitement des valeurs aberrantes s'offrent à nous : les exclure, les remplacer, les transformer ou recourir à des modèles robustes. Notre décision reposera principalement sur le type de données que nous avons, et sur l'objectif que nous souhaitons atteindre. Dans notre contexte, les valeurs aberrantes ou extrêmes correspondent à des vins présentant des caractéristiques peu communes. Cette situation peut résulter d'une erreur de mesure des propriétés chimiques ou bien être attribuable à un type de vin très rare (médiore ou exceptionnelle), affichant un profil atypique. Afin de prendre la meilleure décision, commencons par détecter plus précisement nos valeurs aberrantes, en utilisant la méthode de l'écart interquartile (IQR). Pour cela, nous définissons les valeurs du premier et du troisième quartile, grâce auxquelles on définit l'écart interquartile. En utilisant cette mesure et en définissant un seuil de sensibilité aux valeurs aberrantes, nous pouvons établir des limites inférieures et supérieures, au-delà desquelles toutes les observations seront considérées comme aberrantes.

```
Il y a 260 lignes avec des valeurs aberrantes en utilisant l'écart interquartile
avec un seuil de 3.
Cela représente 4.89% d'observations de notre jeu de données.
```

Afin d'évaluer si nos valeurs aberrantes/extrêmes sont dû à des erreurs de mesure ou bien à des particularités rares, nous allons examiner chaque observation contenant une valeur aberrante. Le tableau suivant permet d'identifier la caractéristique à laquelle cette valeur est associée. Ainsi, nous allons pouvoir comparer les valeurs aberrantes aux statistiques descriptives réalisées précédemment et à nos recherches.

```
      fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
4187            7.7              0.41         0.76            1.80      0.611
4096            9.2              0.52         1.00            3.40      0.610
4057            7.8              0.41         0.68            1.70      0.467
4036            7.8              0.43         0.70            1.90      0.464
4560            8.6              0.49         0.51            2.00      0.422
...             ...               ...          ...             ...        ...
3261            6.1              1.10         0.16            4.40      0.033
2860            7.7              0.43         1.00           19.95      0.032
2955            6.8              0.45         0.28           26.05      0.031
1501            7.7              0.49         1.00           19.60      0.030
628             7.4              0.20         1.66            2.10      0.022

      free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
4187                  8.0                  45.0  0.99680  3.06       1.26
4096                 32.0                  69.0  0.99960  2.74       2.00
4057                 18.0                  69.0  0.99730  3.08       1.31
4036                 22.0                  67.0  0.99740  3.13       1.28
4560                 16.0                  62.0  0.99790  3.03       1.17
...                   ...                   ...      ...   ...        ...
3261                  8.0                 109.0  0.99058  3.35       0.47
2860                 42.0                 164.0  0.99742  3.29       0.50
2955                 27.0                 122.0  1.00295  3.06       0.42
1501                 28.0                 135.0  0.99730  3.24       0.40
628                  34.0                 113.0  0.99165  3.26       0.55

      alcohol  type  quality  outlier    source_outlier
4187      9.4     0        1     True         chlorides
4096      9.4     0        1     True       citric acid
4057      9.3     0        1     True         chlorides
4036      9.4     0        1     True         chlorides
4560      9.0     0        1     True         chlorides
...       ...   ...      ...      ...               ...
3261     12.4     1        1     True  volatile acidity
2860     12.0     1        2     True       citric acid
2955     10.6     1        2     True    residual sugar
1501     12.0     1        2     True       citric acid
628      12.2     1        2     True       citric acid
```

	index	source_outlier
0	chlorides	0.438462
1	fixed acidity	0.280769
2	volatile acidity	0.134615
3	sulphates	0.061538
4	citric acid	0.038462
5	free sulfur dioxide	0.030769
6	residual sugar	0.007692
7	pH	0.007692

On observe que la variable du chlorures présentent le plus de valeurs aberrantes (plus de 40% d'entre elles). Les chlorures font référence à la concentration d'ions chlorure présents dans le vin. Ils sont simplement une façon de mesurer la teneur en sel d'un vin. Le sel est naturellement présent dans les raisins, bien que la quantité puisse varier en fonction des conditions géologiques et climatiques du vignoble. De plus, le processus de fermentation peut influer sur la teneur en chlorures.

Nos ensembles de données indiquent des concentrations moyennes de 5 mg/L (équivalent à 0.05 g/L = 0.05 g/dm³) pour les vins blancs et de 9 mg/L pour les vins rouges. Ces moyennes sont cohérentes avec les conclusions d'articles scientifiques, qui suggèrent que ces niveaux peuvent augmenter significativement en raison de la salinité naturelle des sols et de l'eau d'irrigation, notamment pour les vins récoltés près de la mer.

	fixed acidity	volatile acidity	citric acid	residual sugar	\
count	5320.00	5320.00	5320.00	5320.00	
mean	7.22	0.34	0.32	5.05	
std	1.32	0.17	0.15	4.50	
min	3.80	0.08	0.00	0.60	
25%	6.40	0.23	0.24	1.80	
50%	7.00	0.30	0.31	2.70	
75%	7.70	0.41	0.40	7.50	
max	15.90	1.58	1.66	65.80	

	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	\
count	5320.00	5320.00	5320.00	5320.00	5320.00	
mean	0.06	30.04	114.11	0.99	3.22	
std	0.04	17.81	56.77	0.00	0.16	
min	0.01	1.00	6.00	0.99	2.72	
25%	0.04	16.00	74.00	0.99	3.11	
50%	0.05	28.00	116.00	0.99	3.21	
75%	0.07	41.00	153.25	1.00	3.33	
max	0.61	289.00	440.00	1.04	4.01	

	sulphates	alcohol	type	quality
count	5320.00	5320.00	5320.00	5320.00
mean	0.53	10.55	0.74	1.82
std	0.15	1.19	0.44	0.73
min	0.22	8.00	0.00	1.00
25%	0.43	9.50	0.00	1.00

50%	0.51	10.40	1.00	2.00
75%	0.60	11.40	1.00	2.00
max	2.00	14.90	1.00	3.00

Nous les traiterons grâce à la fonction “winsorizer”, qui consiste à modifier les valeurs extrêmes d’une distribution en les remplaçant par des valeurs moins extrêmes. On utilise l’écart interquartile pour déterminer les seuils bas et haut. Les valeurs en dessous de Q1 - 1.5xIQR sont remplacées par Q1, et celles au-dessus de Q3 + 1.5xIQR sont remplacées par Q3.

De plus, comme observé grâce aux histogrammes et aux tableaux de skewness, la plupart des variables des deux jeux de données présentent un distribution asymétrique positive, et il est alors nécessaire d’appliquer une transformation logarithmique. En effet, celle-ci permettra d’aider à stabiliser la variance des données, mais également à linéariser les relations non linéaires entre les variables, facilitant ainsi l’ajustement par un modèle de régression linéaire.

	skewness
fixed acidity	0.734332
volatile acidity	1.039136
citric acid	0.484172
residual sugar	0.505370
chlorides	1.566298
free sulfur dioxide	-0.690710
total sulfur dioxide	0.063596
density	0.068148
pH	0.389859
sulphates	0.769803
alcohol	0.375748
type	-1.121489
quality	0.297236

On remarque que le traitement de ces valeurs aberrantes influent directement sur la distribution de nos variables car on obtient des skewness plus faibles pour l’ensemble de nos variables. Cependant, certaines variables conservent toujours une asymétrie, probablement en raison du fait que la transformation des valeurs aberrantes n’a pas réussi à corriger toutes les occurrences atypiques. Cependant, cette information demeure importante à conserver, car elle reflète la réalité malgré les ajustements effectués.

3 Régression

Préparation des données Pour préparer la sélection du modèle et l'ajustement des hyperparamètres, l'ensemble de données doit être divisé en deux ensembles, l'un pour l'entraînement et l'autre pour le test. Notre variable cible, "qualité", est affecté à y et séparée des caractéristiques, affectées à X. La division est ensuite effectuée à l'aide de la fonction train_test_split, avec une taille de test de 30 %. Après la division de nos données en deux ensembles, nous standardisons les variables puisqu'elles possédent des échelles de valeurs bien différentes.

Maintenant que l'analyse exploratoire de nos données est terminée et que nous disposons de deux ensembles de données correctement nettoyés et préparés, nous pouvons entamer la première partie de notre projet de prédiction en utilisant des méthodes de régression.

Les méthodes de régression représentent un ensemble d'approches statistiques utilisées pour estimer les relations entre une variable dépendante et une ou plusieurs variables indépendantes. Dans notre analyse de régression, les variables indépendantes (les 11 caractéristiques) sont utilisées pour estimer la variable dépendante (la qualité du vin).

3.1 Régression Linéaire

Pour amorcer cette analyse, nous allons mettre en œuvre le modèle de régression linéaire multiple en tant que point de départ, étant donné que nous n'avons pas clairement déterminé la nature des relations entre la variable dépendante et les variables indépendantes. Nous conservons l'ensemble des variables indépendantes afin d'évaluer la performance du modèle global, et d'examiner plus en profondeur les relations des variables avec la qualité du vin, et leur impact sur celle-ci.

Lorsque nous ajustons un modèle statistique à nos données à l'aide de la méthode des moindres carrés ordinaires (OLS), nous souhaitons obtenir un résumé complet des résultats du modèle ajusté. Dans notre cas, ce résumé comprend plusieurs éléments tels que : - **L'estimation des coefficients :** ces valeurs quantifient la relation entre les variables indépendantes et la variable dépendante - **La t-statistique associée à chaque coefficient :** cette statistique mesure le rapport entre la différence de la valeur estimée d'un paramètre par rapport à sa valeur hypothétique et son erreur standard. Elle est cruciale dans les tests d'hypothèse, en particulier dans le cadre du test t de Student. - **La p-valeur associée à chaque coefficient :** cette valeur est calculée en fonction de la t-statistique. Elle représente la probabilité, sous l'hypothèse nulle, d'obtenir une valeur au moins aussi extrême que celle observée pour la t-statistique. Une p-valeur plus faible suggère une plus grande significativité du coefficient dans le contexte des tests d'hypothèse. - **Le coefficient de détermination (R^2) :** cette mesure évalue la proportion de la variance totale de la variable dépendante expliquée par le modèle. Un R^2 élevé indique une meilleure adéquation du modèle aux données observées. - **L'erreur quadatrique moyenne (MSE) :** cette métrique quantifie la moyenne des carrés des différences entre les valeurs prédites par le modèle et les valeurs réelles. Une MSE plus faible indique une meilleure précision du modèle. - **L'erreur absolue moyenne (MAE) :** cette métrique quantifie la moyenne des valeurs absolues des différences entre les valeurs prédites par le modèle et les valeurs réelles. Une MAE plus faible indique une meilleure adéquation du modèle aux données en termes d'erreurs absolues moyennes.

En analysant ces différents éléments, nous obtenons une vision approfondie de la robustesse des coefficients du modèle, de leur significativité statistique et de la fiabilité des prédictions générées par le modèle ajusté. Les métriques telles que la MSE, la MAE et le coefficient de détermination R^2 nous permettront d'évaluer la performance et la précision du modèle. Ces indicateurs nous

offrent une base solide pour mesurer l'adéquation du modèle aux données, permettant ainsi des comparaisons significatives avec d'autres modèles.

Regardons les résultats de l'ajustement du modèle de régression linéaire sur notre ensemble de données :

OLS Regression Results

Dep. Variable:	quality	R-squared:	0.352
Model:	OLS	Adj. R-squared:	0.350
Method:	Least Squares	F-statistic:	167.9
Date:	Sun, 04 Feb 2024	Prob (F-statistic):	0.00
Time:	18:38:28	Log-Likelihood:	-3292.6
No. Observations:	3724	AIC:	6611.
Df Residuals:	3711	BIC:	6692.
Df Model:	12		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025- 0.975]
const	1.8158	0.010	188.826	0.000	1.797 1.835
fixed acidity	0.0964	0.019	5.081	0.000	0.059 0.134
volatile acidity	-0.1773	0.014	-12.374	0.000	-0.205 -0.149
citric acid	0.0138	0.012	1.132	0.258	-0.010 0.038
residual sugar	0.1741	0.023	7.624	0.000	0.129 0.219
chlorides	-0.0256	0.014	-1.813	0.070	-0.053 0.002
free sulfur dioxide	0.1393	0.015	9.318	0.000	0.110 0.169
total sulfur dioxide	-0.1417	0.019	-7.388	0.000	-0.179 -0.104
density	-0.2084	0.037	-5.601	0.000	-0.281 -0.135
pH	0.0758	0.014	5.257	0.000	0.048 0.104
sulphates	0.0941	0.012	7.986	0.000	0.071 0.117
alcohol	0.2418	0.021	11.382	0.000	0.200 0.283
type	-0.0630	0.025	-2.566	0.010	-0.111 -0.015

Omnibus:	68.713	Durbin-Watson:	2.001

```
Prob(Omnibus):                  0.000   Jarque-Bera (JB):                48.602
Skew:                           0.171   Prob(JB):                     2.79e-11
Kurtosis:                       2.557   Cond. No.                         10.4
==============================================================================
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
Erreur Absolue (MAE) : 0.49255958767253877
Erreur Résiduelle (MSE) : 0.3549302769496651
```

Dans un premier temps, nous observons que la grande majorité des p-valeurs associées aux coefficients sont considérablement faibles voire nulles, tandis que celles associées aux variables de l'acide citrique et des chlorides dépassent le seuil de risque de 0.05, que nous avons défini comme notre α. Les coefficients associés à des p-valeurs supérieures à 0.05 sont considérés comme non significatifs, ainsi l'acide citrique et les chlorides ne semblent pas contribuer de manière significative à notre modèle, suggérant qu'elles pourraient être éliminée de notre modèle. En ce qui concerne les variables liées aux coefficients significatifs, il semble que l'alcool, la densité, l'acidité volatile et le sucre résiduel ont les plus grandes influences sur la qualité du vin. Le coefficient attribué à la variable alcool révèle une relation positive entre le taux d'alcool et la qualité d'un vin : toute choses étant égales par ailleurs, plus le taux d'alcool augmente dans un vin, plus la qualité de celui-ci sera haute. A l'inverse, les variables dioxyde de sulfure total et densité exercent une forte influence négative sur la qualité, agissant comme des facteurs déterminants négatifs pour cette variable.

Le coefficient de détermination que nous avons obtenu est relativement bon, atteignant 0.352. Cela indique que notre modèle est capable d'expliquer environ 35% de la variance totale de notre variable dépendante. Le modèle de régression linéaire ne semble donc pas être le plus ajusté à notre ensemble de données puisqu'il ne parvient pas à expliquer de manière significative une portion substantielle de la variance de notre variable dépendante. Cependant, il est à noter que certaines métriques d'évaluation, telles que la MSE et la MAE, sont modérément basses, suggérant que notre modèle présente des erreurs de prédiction qui pourraient être réduites par des ajustements supplémentaires.

Nos jeux de données semblent présenter des caractéristiques complexes, notamment un nombre de variables qui ne contribuent pas toutes de manière significative à la prédiction, ainsi que des relations complexes entre les variables.

En outre, les résultats suivants des tests de White indiquent la présence d'hétéroscédasticité dans les résidus des modèles de régression linéaire (p-value très faible).

```
P-value du test de White : 6.087557483196252e-20
```

L'hétéroscédasticité, détectée par le test de White, remet en question l'hypothèse d'homoscédasticité, une condition importante dans le modèle de régression linéaire. Ces résultats soulignent donc la nécessité de considérer des approches plus avancées, telles que des méthodes de régression régularisées, pour tenter de pallier les défis complexes que présentent nos données.

3.2 Régression LASSO

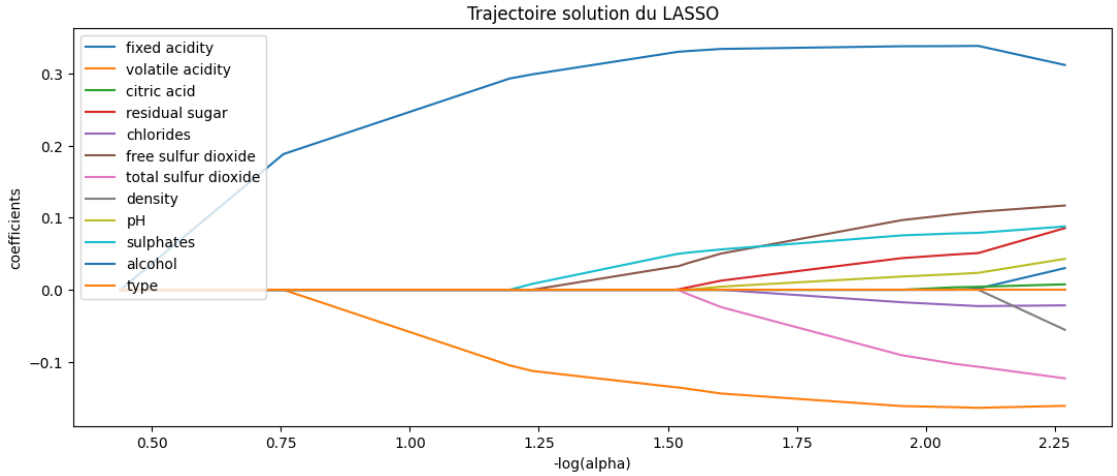
Les régressions régularisées, telles que la régularisation L1 (Lasso) et L2 (Ridge), peuvent être particulièrement bénéfiques dans des situations où plusieurs variables sont présentes et où certaines d'entre elles peuvent ne pas être significatives, et également lorsqu'il existe des interactions complexes entre les variables. Ces méthodes de régularisation imposent des contraintes sur les coefficients des variables, ce qui peut conduire à une sélection automatique des caractéristiques et à une amélioration de la généralisation du modèle.

Dans un premier temps, nous allons utiliser la méthode LASSO (Least Absolute Shrinkage and Selection Operator), qui est une technique de régression linéaire régularisée qui introduit une pénalité L1 sur les coefficients du modèle. Cette pénalité L1 est ajoutée à la fonction de coût de la régresson linéaire, et elle permet ainsi de forcer certains coefficients à devenir égaux à zéro, conduisant à une sélection automatique de variables. Ainsi certaines caractéristiques de vin peuvent être totalement exlues du modèle, le rendant plus interprétable. La régression de Lasso consiste donc à minimiser la somme des valeurs absolues des coefficients, le tout pondéré par un facteur (noté λ), ce qui revient à minimiser l'expression suivante : $RSS + \lambda \sum_{i=1}^{p} ||\beta_i||_1$

Cette régression est particulièrement utile lorsque nous sommes en présence d'un jeu de données en grandes dimensions, c'est-à-dire possédant un très grand nombre de variables par rapport au nombre d'observations. Dans notre cas, nous possédons 12 caractéristiques pour prédire la qualité, ce qui ne représente pas nécessitement un grand nombre de variables, mais il est possible que certaines caractéristiques ne contribuent pas à la prédiction. De plus, elle est très utile lorsque les variables sont très corrélées. D'après la matrice des corrélations, plusieurs caractéristiquent présentent, entre elles, de forte corrélations ce qui peut venir fausser les résultats d'une régression linéaire. Mettre en place une régression Lasso semble donc pertinent, afin de procéder à la sélection de variables et d'appliquer une régularisation. Ainsi c'est en évaluant les résultats obtenus, que nous pourrons déterminer si cette approche permet d'améliorer la performance de notre régression.

Tout d'abord, nous allons représenter le chemin de la trajectoire du LASSO, qui permet de comprendre comment les coefficients du modèle évoluent en fonction du paramètre de régularisation alpha. Certaines caractéristiques peuvent devenir exactement égales à zéro, et d'autres peuvent rester non nulles, et c'est ainsi qu'on définira les variables qui seront incluses dans le modèle à un niveau particulier de régularisation.

Ensuite, nous devons trouver le paramétre de régularisation alpha optimal, celui qui donne le meilleur compromis entre l'ajustement du modèle aux données d'entraînement et la généralisation du modèle sur de nouvelles données. Pour cela, nous utiliserons une validation croisée, qui évalue la performance du modèle pour chaque valeur de alpha issu de la grille défini par le LASSO. L'alpha optimal sera donc celui qui minimise l'erreur de validation croisée. Enfin, nous pourrons mettre en place notre modèle de LASSO, avec le α optimal, et évaluer sa performance grâce au coefficient de détermination, et mesurer sa précision grâce aux métriques, que nous comparerons avec le modèle de régression linéaire.



En observant la trajectoire de la solution ci-dessus dans la régression LASSO, on constate que à mesure que le niveau de régularisation augmente (c'est-à-dire que la valeur de $-\log(\alpha)$ diminue), de plus en plus de coefficients deviennent nuls et restent ainsi lorsque le niveau de régularisation est accru davantage.

Certaines variables demeurent non nulles pour des valeurs élevées de α. Il est intéressant de noter que la dernière variable à devenir nulle est "alcool", suggérant ainsi que cette variable possède une relation importante avec notre variable cible, et qu'elle est donc importante dans notre prédiction, comme nous l'avions déjà constaté avec le modèle de régression linéaire. Il en est de même pour la variable de l'acidité

Les nombreuses autres variables s'annulent plus rapidement, pour des valeurs de α plus petites. Cela suggère que le modèle LASSO, avec une régularisation plus prononcée, tend à favoriser la parcimonie dans la prédiction de la qualité du vin. Une grande valeur de α entraîne donc une forte réduction du nombre de coefficients non nuls ce qui indique que le modèle accorde une importance significative à un sous-ensemble restreint de variables, tandis qu'une valeur de α plus petit signifie qu'un nombre relativement plus important de variables ayant une influence dans le modèle,est conservé.

`Paramètre de régularisation optimal est : 0.0001`

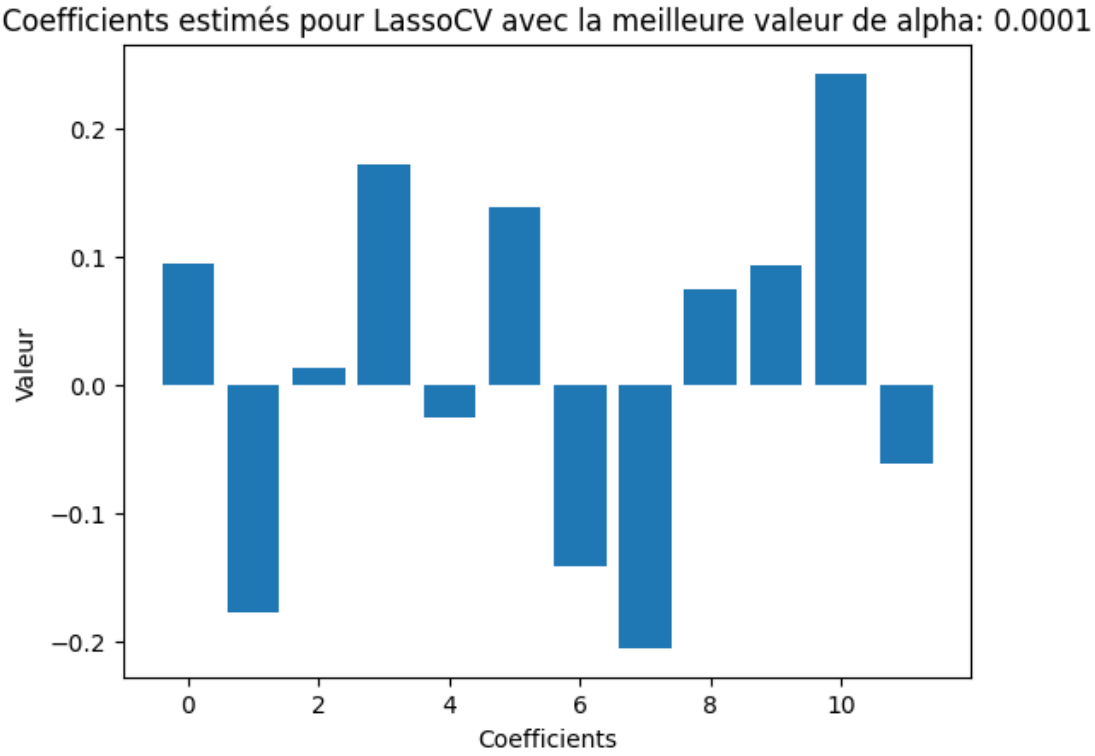
L'alpha optimal trouvé par la méthode de validation croisée pour le jeu de données 0.0001. Cela signifie que, parmi les valeurs testées dans la grille, 0.0001 a conduit à la meilleure performance moyenne du modèle LASSO sur les ensembles de validation de chaque jeu de données. Une faible valeur de α signifie une pénalisation légère, et les coefficients ont donc moins de pression pour être réduits à zéro.

Avec une valeur aussi basse pour le paramètre de régularisation, le modèle peut conserver un nombre significatif de coefficients non nuls, ce qui peut conduire à un modèle relativement complexe. Cette valeur d'alpha obtenue correspond à une valeur très élevée de $-\log(\alpha)$. L'analyse des trajectoires du LASSO que nous avons examiné précédemment confirme bien que cette valeur de α correspond à un modèle peu parcimonieux, favorisant la conservation d'un grand nombre de variables.

Nous mettons donc en place le modèle LASSO sur nos données du vin avec un α de 0.0001, et nous trouvons les résultats suivants :

```
Erreur absolue moyenne (MSE) : 0.4925921553373132
Erreur quadratique moyenne (MSE) : 0.35492505484557185
Coefficient de détermination (R^2) : 0.3518389689076843

Coefficients attribués par le modèle Lasso :
Intercept: 1.8157894736841989
fixed acidity: 0.09531782267460906
volatile acidity: -0.1772356843115473
citric acid: 0.01357742131098614
residual sugar: 0.17237004879840004
chlorides: -0.025446701066427567
free sulfur dioxide: 0.13906296505235644
total sulfur dioxide: -0.1414436287365038
density: -0.20562193357888864
pH: 0.07508232047428094
sulphates: 0.09383734624577668
alcohol: 0.2431080042870207
type: -0.06195421469915245
```



Les résultats sont remarquablement similaires à ceux obtenus avec la régression linéaire, principalement en raison de la très faible valeur d'alpha. Cette faible valeur n'a engendré ni pénalisation

significative ni sélection de variables. Les estimations des coefficients associés à chaque variable sont très similaires à celles obtenues avec la régression linéaire, étant donné que seule une pénalisation très faible a été appliquée. Ces résultats, plutôt étonnants, suggèrent donc que toutes les variables conservent leur importance dans la prédiction de notre variable cible au sein des modèles linéaires.

Les résultats de l'estimation des coefficients liés aux variables présentent quelques petites différences par rapport à ceux obtenus avec la régression linéaire multiple. On observe, par exemple, une légère augmentation du coefficient associé à l'alcool (passant de 0.2418 avec la régression linéaire à 0.2431 avec la régression LASSO), suggérant que la pénalisation de norme L1 a favorisé cette variable au détriment du sulfure dont le coefficient a diminué (passant de 0.0941 à 0.0938), réduisant ainsi son impact dans la prédiction de la qualité. Dans l'ensemble, la majorité des coefficients ont été rapprochés vers 0. Cependant, cette tendance ne semble pas être optimale, car elle se traduit par un R^2 plus petit et des erreurs résiduelles plus importantes.

En conclusion pour ce modèle, l'application de la régression LASSO ne semble pas apporter d'amélioration par rapport à la régression linéaire. La méthode LASSO a pénalisé très peu de variables de manière significative, préservant ainsi la quasi-totalité des variables et conduisant à des résultats très similaires à ceux de la régression linéaire multiple. Cette similitude découle de la faible valeur des paramètres de régularisation obtenus grâce à la validation croisée.

Par ailleurs, le LASSO est plus efficace lorsque les variables présentent une multicolinéarité importante. Cependant, comme nous l'avons vu grâce à la matrice de corrélation, seul un petit nombre de variables indépendantes affichent des corrélations significatives entre elles, limitant ainsi l'efficacité de la pénalisation.

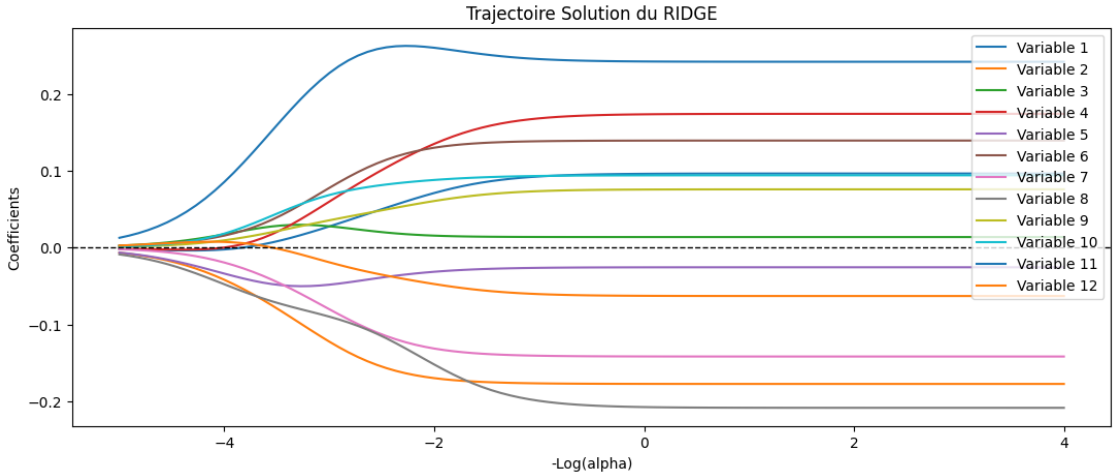
Enfin, notre jeu de données comporte un nombre restreint de variables indépendantes, seulement 11 caractéristiques, comparé au nombre d'observations. Ainsi, la méthode LASSO peut être moins efficace pour fournir une sélection adéquates de variables susceptibles d'améliorer les performances du modèle. De plus, en raison de la complexité de nos ensembles de données, ce modèle ne semble pas être adapté à nos besoins.

3.3 Régression RIDGE

Nous allons tout de même tenter d'obtenir de meilleurs résultats avec une autre méthode : la régression Ridge, qui se caractérise par une pénalité plus élevée que le Lasso. Contrairement au Lasso, le modèle Ridge utilise une régularisation L2 plus forte, ce qui peut favoriser une modélisation plus robuste en réduisant davantage les coefficients des variables moins importantes tout en contrôlant le surajustement.

La régression Ridge consiste à minimiser la somme des carrées des coefficients, le tout pondéré par un facteur (noté λ). Cela revient à minimiser l'expression suivante : $RSS + \lambda \sum_{i=1}^{p} ||\beta_i||_2$ Le facteur λ contrôle la force de régularisation. Plus λ est élevé, plus la régularisation est forte. Le terme de régularisation encourage les coefficients à être petits, mais ils ne sont pas annulés complètement. Ainsi, la régression Ridge préserve toutes les caractéristiques, mais elle peut réduire l'importance des caractéristiques moins importantes en diminuant leurs coefficients, contrairement à la régression Lasso qui a la particularité de conduire à la sparsité des coefficients.

Pour déterminer les modèles de régression Ridge optimaux pour chaque ensemble de données, nous procédons de la même manière que pour la régression LASSO en suivant les mêmes étapes.



À travers l'analyse de la trajectoire des solutions de la régression Ridge, on observe que les coefficients associés aux variables ne sont pas nuls pour de petites valeurs de α et restent constants. Cependant, au-delà d'une certaine valeur de α, ces coefficients tendent à se rapprocher de zéro, sans jamais l'atteindre. Ce phénomène, qui est appelé "rétrécissement" devient plus prononcé à mesure que le paramètre de régularisation augmente, indiqué par une baisse de la valeur de $\log(\alpha)$. En d'autres termes, à des niveaux élevés de régularisation, les coefficients associés aux variables sont davantage restreints vers zéro, soulignant l'effet significatif de la pénalité de régularisation sur la magnitude des coefficients.

```
Paramètre de régularisation optimal est : 15.199110829529332
```

L'alpha optimal trouvé par la méthode de validation croisée est 15.19. L'alpha est le paramètre de régularisation qui contrôle la force de la pénalité appliquée aux coefficients du modèle. Plus α est élevé, plus la régularisation sera forte.

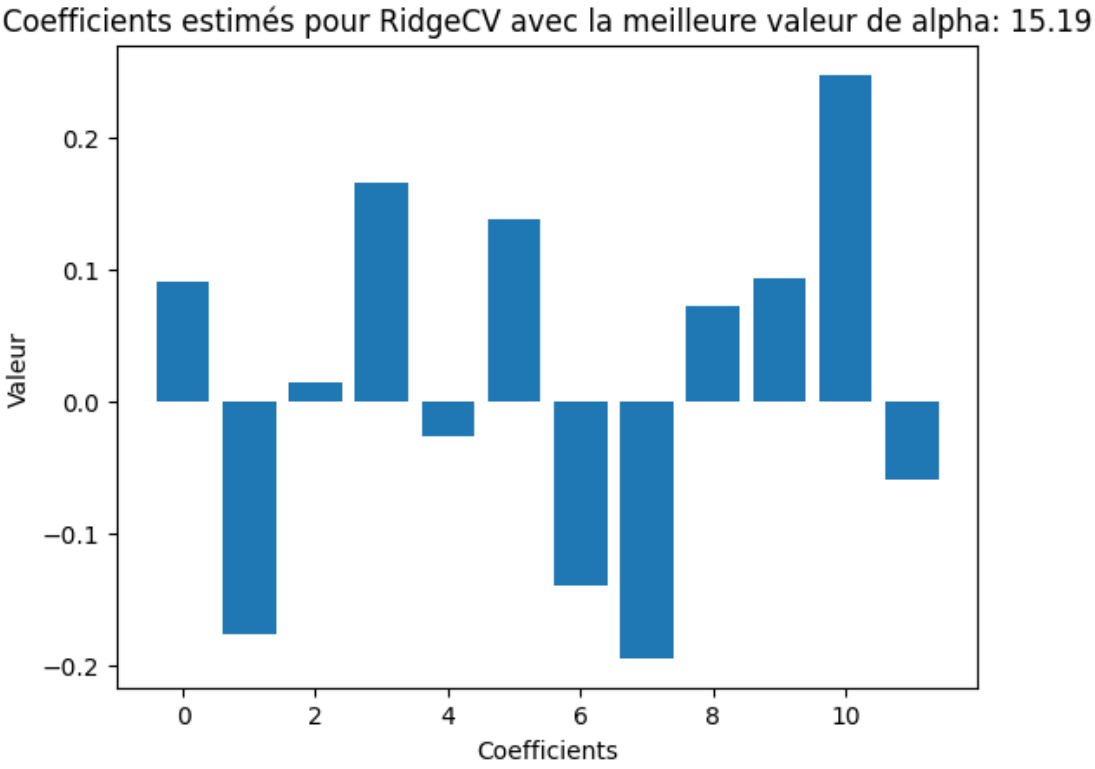
Lorsque α est fixé à 53,36, cela implique l'application d'une pénalité forte. Dans le contexte du jeu de données sur les vins, cette valeur spécifique de α indique que le modèle accorde une importance significative à la régularisation. Cette régularisation conduit alors à des coefficients de régression plus modestes, contribuant ainsi à minimiser le potentiel de surajustement du modèle aux données d'entraînement.

Nous mettons en place le modèle Ridge sur nos données avec un α de 15.19, et nous trouvons les résultats suivants :

```
Erreur absolue moyenne (MSE) : 0.4927336754466488
Erreur quadratique moyenne (MSE) : 0.3548803716079747
Coefficient de détermination (R^2) : 0.35180613344194167

Coefficients attribués par le modèle Lasso :
Intercept: 1.8157894736841995
fixed acidity: 0.09077746372319154
volatile acidity: -0.1762116478019553
citric acid: 0.013931124148145279
```

```
residual sugar: 0.16542449368047854
chlorides: -0.026642213483084192
free sulfur dioxide: 0.13795926437470013
total sulfur dioxide: -0.1401137325316225
density: -0.1951066134851676
pH: 0.07241016202351996
sulphates: 0.09283066950735816
alcohol: 0.2468870031042981
type: -0.05930564786251465
```



On observe quelques différences dans les valeurs des coefficients estimés, mais elles sont très faibles. La régression Ridge a accordé davantage de poids à certaines variables tout en réduisant l'influence d'autres. Cependant, malgré ces ajustements, le modèle ne semble pas s'être amélioré en termes d'ajustement aux données, comme en témoigne le coefficient de détermination.

Après avoir réalisé ces trois régressions, nous pouvons conclure que les modèles linéaires ne semblent pas performants et pas adapatés à notre jeu de données, très certainement dû à la structure spécifique de nos données. En effet, plusieurs hypothèses essentielles à la régression linéaire ne semblent pas être respectées, entraînant ainsi des ajustements insatisfaisants des modèles aux données. Par conséquent, ces approches ne se révèlent pas efficaces pour prédire la qualité du vin.

3.4 Random Forest Regressor

Néanmoins, il existe d'autres modèles d'apprentissage automatique qui peuvent être utilisés lorsque la relation entre les variables indépendantes et la variable cible est non linéaire ou complexe. Nous allons donc tenter de les mettre en place, afin de voir si éliminer l'hypothèse de linéarité permet d'obtenir des modèles plus performants.

Pour commencer, nous allons implémenter une forêt aléatoire. Cette méthode est construite en agrégeant les prédictions de plusieurs arbres, ce qui permet de réduire le surajustement possible associé aux arbres de décision individuels. Bien que les forêts aléatoires soient moins faciles à interpréter, elles ont souvent une meilleure prévision que les arbres de décision, en raison de cette agrégation et de la réduction du surajustement. De plus, cette méthode n'impose pas de structure de données ou d'hypothèses spécifiques sur leurs distributions.

Comme un grand nombre de modèle, celui de la forêt aléatoire possède des paramètres, qui affectent la façon dont il construira les arbres de décision et agit sur l'ensemble de données. Il est important de trouver la valeur des meilleures paramètres, que nous établissons ici grâce à la méthode de recherche GridSearchCV. Elle explore l'ensemble des combinaisons possibles des paramètres définis, et évalue chaque modèle en utilisant une technique de validation croisée. La combinaison de paramètres fournissant la meilleure performance de modèle est alors sélectionnée.

Nous cherchons les valeurs optimales de n_estimators (le nombre d'arbres dans la forêt), de max_depth (la profondeur maximale de chaque arbre), de min_samples_split (le nombre minimum d'échantillons requis pour diviser un nœud interne), de min_samples_leaf (le nombre minimum d'échantillons requis pour former une feuille), de max_features (le nombre maximum de caractéristiques à considérer pour la division d'un nœud) et de bootstrap (qui indique si les échantillons sont tirés avec remplacement ou non).

La méthode de GridSearchCV appliquée aux données du vin blanc a permis d'identifier les hyperparamètres optimaux suivants pour le modèle de forêt aléatoire :

- n_estimators' : 200
- 'min_samples_split' : 5
- 'min_samples_leaf' : 2
- 'max_features' : 'log2'
- 'max_depth' : 30
- 'bootstrap' : False

En utilisant ces paramètres, nous instaurons le modèle de forêt aléatoire :

```
R2 0.40349422855107864
MSE 0.31589760857535426
MAE 0.46266327276524644
```

Les performances de ce modèle surpassent significativement celles des modèles précédents. Le coefficient de détermination a augmenté, indiquant ainsi que ce modèle explique une plus grande part de la variance. De plus, la MSE a diminué, ce qui traduit une réduction des erreurs de prédiction. À titre d'illustration, voici un extrait des valeurs prédites par le modèle sur l'ensemble des données :

	Vraies valeurs	Valeurs prédites
4878	1	1.040833
1434	2	1.995417
2989	2	2.515833
5235	2	1.998750

3.5 Ensemble des résultats

Pour finaliser cette première partie de notre étude, voici un tableau récapitulatif des différents modèles de régression que nous avons mis en place, ainsi que leurs performances respectives mesurées par le R^2 et leurs précisions mésurées par les erreurs quadratiques et absolues moyennes :

	Modèle de régression	R2	MAE	MSE
3	Forêt Aléatoire	0.403494	0.462663	0.315898
0	Régression Linéaire	0.351840	0.492560	0.354930
1	Régression LASSO	0.351839	0.492592	0.354925
2	Régression RIDGE	0.351806	0.492734	0.354880

Le modèle de forêt aléatoire se distingue nettement en affichant les meilleures performances, avec un coefficient de détermination particulièrement élevé (0.40) et des erreurs quadratiques et absolues moyennes minimes (MSE de 0.32 et MAE de 0.46). Ces résultats suggèrent que la forêt aléatoire est le choix le plus adapté à nos données pour prédire la qualité d'un vin en se basant sur ses caractéristiques physiques et chimiques.

Cette efficacité accrue peut être attribuée à la complexité inhérente de nos données, qui ne se prêtent pas de manière optimale à l'ajustement par des modèles linéaires traditionnels tels que la régression linéaire, la régression Lasso et la régression Ridge. Ces modèles peuvent être limités par des relations linéaires rigides et ne parviennent peut-être pas à saisir la structure sous-jacente complexe de nos caractéristiques.

En optant pour la forêt aléatoire, un modèle non linéaire et basé sur des arbres de décision, nous sommes en mesure de mieux capturer les nuances et les interactions complexes entre les variables, ce qui se traduit par des performances supérieures. Ainsi, ce choix de modèle semble être le plus approprié pour notre ensemble de données, dépassant les limites des modèles linéaires traditionnels.

3.6 Analyse en Composantes Principales (ACP)

Les performances du meilleur modèle, la forêt aléatoire, peuvent encore être améliorées, puisque le coefficient de détermination (R^2) est encore loin de 1. Le coefficient de détermination mesure la proportion de la variance totale de la variable cible qui est expliquée par le modèle. Un R2 proche de 1 indiquerait que le modèle explique la quasi-totalité de la variabilité de la qualité des vins à partir de leurs caractéristiques physiques et chimiques. Actuellement, bien que notre modèle soit performant, il y a encore une certaine part de variabilité non expliquée que nous pourrions chercher à réduire pour rendre nos prédictions plus précises et robustes. Dans la suite de notre analyse, nous chercherons à optimiser davantage ces résultats.

Rappellons que notre ensemble de données comprend 12 variables indépendantes permettant de prédire la qualité d'un vin. Ceci est un nombre important et nous avons vu que l'introduction d'une sélection de variable par régressiion régularisée n'est pas pertinente pour notre analyse, crobablement en raison de la pertinence de toutes les variables. Cependant, il est tout de même possible de réduire la dimension du problème, en cherchant les dépendances entre variables. Cette

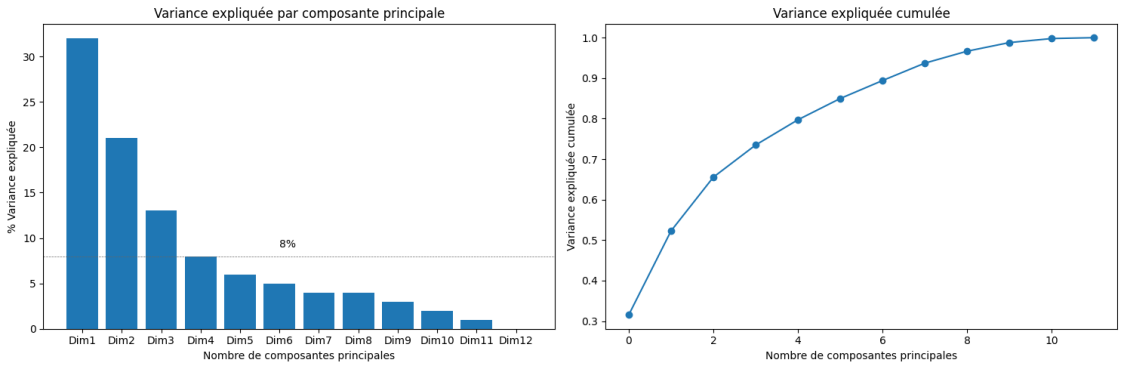
réduction ne consiste pas à exclure certaines variables du modèle, mais plutôt à identifier de nouvelles variables (en nombre inférieur, m < p) par une transformation bien choisie. Ces nouvelles variables doivent conserver les propriétés "géométriques" des variables initiales afin de conserver autant d'informations que possible.

L'une des techniques bien connues pour la réduction de dimension est l'Analyse en Composantes Principales (ACP). L'ACP vise à transformer des variables corrélées en nouvelles variables décorrélées en projetant les données dans le sens de la variance croissante. Les variables avec la variance maximale sont sélectionnées comme composantes principales. Pour ce faire, il est nécessaire de trouver une nouvelle base orthonormée dans laquelle représenter nos données, de manière à maximiser la variance selon ces nouveaux axes. Cela implique le calcul des valeurs propres et des vecteurs propres à partir de la matrice de variance-covariance de nos données. Après avoir trié les valeurs propres par ordre décroissant, on choisit le nombre de vecteurs propres k conservé pour former une nouvelle matrice. Ces nouvelles coordonnées correspondent alors à une représentation des observations dans une dimension réduite.

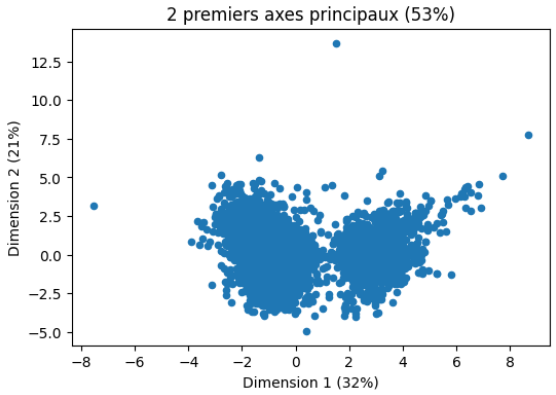
Il est crucial de souligner que le choix de k, représentant le nombre de composantes nécessaires pour décrire les données, revêt une importance particulière. Une méthode courante d'estimation du nombre de composantes principales consiste à examiner le ratio de variance expliquée cumulatif en fonction du nombre de composantes. Le ratio de la variance expliquée est le pourcentage de la variance attribuée à chacune des composantes sélectionnées.

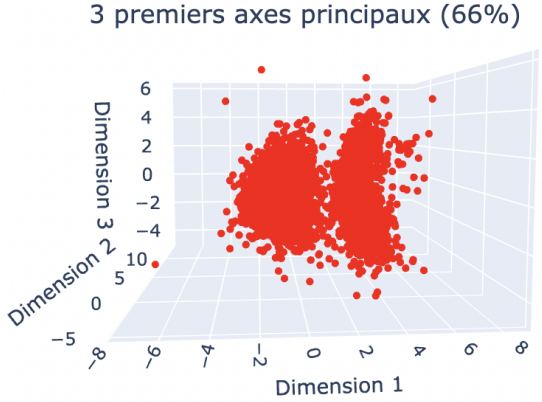
Les résultats de la recherche du nombre optimal de composantes pour notre jeu de données sont les suivants :

	Dimension	Variance expliquée	% variance expliquée	% cum. var. expliquée
0	Dim1	3.782530	32.0	32.0
1	Dim2	2.488885	21.0	52.0
2	Dim3	1.592479	13.0	66.0
3	Dim4	0.953528	8.0	73.0
4	Dim5	0.747896	6.0	80.0
5	Dim6	0.630265	5.0	85.0
6	Dim7	0.532616	4.0	89.0
7	Dim8	0.516030	4.0	94.0
8	Dim9	0.354521	3.0	97.0
9	Dim10	0.255394	2.0	99.0
10	Dim11	0.119601	1.0	100.0
11	Dim12	0.026255	0.0	100.0



On observe que les quatres premières composantes expliquent, pour chacune d'entre elles, la partie minimale requise définie par 100% / nombre de variables : 100/12=8%. En dessous de 8%, il n'est pas judicieux d'ajouter une composante au vu du % d'informations qu'elle apporte. De plus, grâce à la coubre représentant la variance expliquée cumulée, nous pouvons voir que les quatres premières composantes principales expliquent pour plus de 70% de la variance totale. Ce seuil des 70/75% est généralement utilisé pour définir le nombre de composantes principales. Nous l'utiliserons pour notre étude car il semble bien plus optimal que choisir 2 ou 3 composantes comme le montre les graphiques ci dessous :

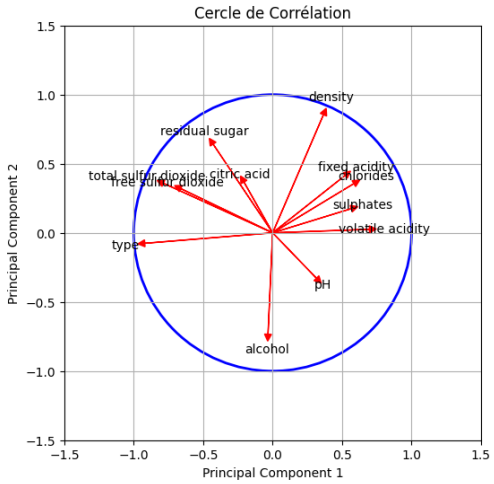




On observe clairement que le nuage de points dans l'ACP à deux composantes est dense et rapproché, indiquant ainsi que cette réduction de dimension n'a pas réussi à capturer efficacement la variance. De plus, cette représentation en 2D révèle la présence de deux "paquets" distincts de points, ce qui pourrait indiquer une certaine structure ou sous-groupement au sein des données.

La visualisation en 3D de l'ACP avec trois composantes révèle également une densité significative du nuage de points, suggérant que ce nombre de composantes n'est pas optimal pour une réduction efficace de nos données. Ainsi, nous opterons bien pour une ACP avec quatre composantes principales.

Pour une meilleure appréhension de la composition des composantes principales, nous pouvons examiner le cercle des corrélations. Ce dernier offre une représentation visuelle des corrélations entre les variables initiales et les composantes principales, en projetant le nuage de points des variables sur le plan formé par les deux premières composantes principales. Les points d'intérêt sont typiquement ceux situés à proximité d'un des axes et éloignés de l'origine. Ces points dénotent une corrélation significative avec l'axe correspondant, les rendant ainsi des indicateurs explicatifs importants pour cette composante spécifique.



Le cercle des corrélations ci-dessus révèle les relations les plus significatives entre les variables et les composantes principales de l'ACP. Pour la première composante principale, le type de vin, le dioxyde de soufre total et libre affichent des corrélations négatives marquées, tandis que l'acidité volatile, le sulfure, les chlorures et l'acidité fixe présentent des corrélations positives notables. En ce qui concerne la deuxième composante principale, la densité et le sucre résiduel dévoilent des corrélations positives prédominantes, tandis que l'alcool affiche une forte corrélation négative. Ces informations suggèrent quelles variables ont une influence significative sur chacune des composantes principales extraites par l'ACP.

Bien que la réduction de dimension par l'ACP semble ne pas convenir à nos données, nous procéderons tout de même à l'évaluation des performances des modèles de régression sur les nouvelles coordonnées de l'espace de dimensions réduites. Cette démarche vise à comparer les résultats obtenus avec ceux précédemment obtenus sur les données originales.

Préalablement à cette évaluation, nous devons effectuer une transformation inverse afin de ramener les données à leur forme d'origine dans l'espace initial de dimensions élevées. Cette étape est cruciale pour faciliter l'interprétation, l'application et l'analyse des résultats des modèles d'apprentissage statistique dans le contexte des caractéristiques d'origine.

Voici les résultats de performance des quatre modèles de régression que nous avons choisis, évalués sur les nouvelles données définies par l'ACP :

	Modèle de régression	R2	MAE	MSE
3	Forêt Aléatoire	0.291183	0.512421	0.375376
0	Régression Linéaire	0.238254	0.533674	0.408798
1	Régression LASSO	0.238253	0.533721	0.408810
2	Régression RIDGE	0.238252	0.533723	0.408795

On constate que, même pour ces données, la forêt aléatoire semble être le modèle le plus approprié, avec le coefficient de détermination le plus élevé et l'erreur quadratique moyenne la plus basse. Cependant, l'ensemble des résultats est moins satisfaisant que ceux des données initiales, ce qui suggère que la réduction de dimension par l'ACP n'est pas adaptée et n'a pas amélioré la prédiction de la qualité du vin.

4 Classification

Nous avons précédemment tenté de construire un modèle de prédiction de la qualité du vin basé sur des algorithmes de régression linéaire et régularisée, avec ou sans une réduction de dimension. Les résultats de ces modèles ne sont pas satisfaisants, et un meilleur modèle pourrait être trouvé. Le manque de performance des algorithmes de régression présentés précédemment pourrait être dû à la présence de relations non linéaires entre les variables, ou à l'absence de multicolinéarité entre les caractéristiques. Il est alors possible d'utiliser des modèles plus complexes, capable de capturer la relation entre la variable cible et les variables indépendantes, afin de prédire la qualité d'un vin.

Nos jeux de données contiennent des vins dont la qualité est comprise entre 3 et 9. Nous décidons ici de faire de la classification binaire, afin de tenter de prédire si le vin considérée est de "très bonne qualité". La classification multiclasse, bien que puissante, peut être complexe à interpréter dans certaines situations pratiques. C'est pourquoi nous avons délibérément opté pour une classification binaire, en définissant un objectif précis.

Notre objectif est de réussir à prédire si le vin considéré est de très haute qualité, et qu'il sort donc du lot. Notre critère de qualité varie de 1 à 10, où la note minimale de 3 suggère l'absence de vins véritablement médiocres. La catégorie la plus représentée de la qualité (6), englobe probablement les vins de bonne qualité, abordables, et appréciés par tous. Dans cette perspective, nous considérons les vins de très haute qualité à partir d'une note de 7, clairement au-dessus de la moyenne.

Ainsi nous transformons notre variable cible en variable binaire où 0, la classe négative, représente les vins de qualité moyenne, tandis que 1, la classe positive, identifie les vins de haute qualité. Nous considérons plus grave qu'un vin de moyenne qualité soit classé comme escellent plutôt que l'inverse, car notre objectif est d'assurer la certitude lorsque nous qualifions un vin d'excellent. Les algorithmes de régression logistique, de fôrets aléatoire, de kNN peuvent ainsi être appliqués. Nous allons donc essayer de trouver le meilleure modèle de classification afin de tenter de prédire la qualité d'un vin à partir de ses caractéristiques, et nous sélectionnerons celui qui présente les meilleurs performances.

Préparation des données

	Qualité	Effectif
0	0	0.791719
1	1	0.208281

On remarque que la classe négative comporte plus de 79 % des observations tandis que la classe positive contient les 20% observations restantes. On constate donc que les deux classes sont très déséquilibrées. En effet, la classe minoritaire correspond souvent aux individus positifs, car c'est la survenance de l'événement rare qui nous intéresse. Il est nécessaire de gérer ce problème de déséquilibre entre les classes, car cela peut poser des défis lors de la mise en place des modèles de Machine Learning. Ces derniers peuvent avoir tendance à être biaisés en faveur de la classe majoritaire, notamment en la prédisant davantage que la classe minoritaire.

Dans le contexte de notre problème de classification binaire, notre principal objectif est d'obtenir des prédictions précises pour la classe minoritaire. Il est donc important de remédier à ce déséquilibre et de prévenir le surajustement. Nous optons pour une stratégie de suréchantillonnage de la classe minoritaire. Cette approche consiste à fournir au modèle un nombre accru d'exemples de la classe minoritaire, renforçant ainsi sa capacité à discerner les caractéristiques spécifiques à cette

classe. Il est important de souligner que, dans notre cas, la décision de suréchantillonner la classe minoritaire découle de notre volonté de préserver la diversité des exemples associés à cette classe. Cette approche vise à garantir une meilleure généralisation du modèle et à préserver son efficacité dans la prédiction de la classe minoritaire. Il est important de ne suréchantillonner uniquement les données de notre ensemble d'entraînement car cela permet d'éviter toute fuite d'informations provenant de l'ensemble de test et garantit une évaluation impartiale de la performance du modèle sur des données inédites.

Nous allons donc tester plusieurs modèles de classification, afin de comparer leurs performances, et de sélectionner celui qui semble être le meilleur. Pour cela nous allons utiliser plusieurs métriques :

- **L'accuracy :** elle est une mesure de la performance d'un modèle de classification qui représente le pourcentage de prédictions correctes parmi l'ensemble total des prédictions. Si elle est proche de 1, alors le modèle a une bonne capacité à faire des prédictions correctes. Cette métrique peut être trompeuse en présence de classes déséquilibrées. Néanmoins, grâce à notre rééchantillonnage, nous pourrons maintenir l'accuracy tout en adoptant une approche prudente dans son interprétation.
- **La précision :** elle est une mesure de performance d'un modèle de classification qui se concentre sur les individus positifs. Elle correspond au taux de prédictions correctes parmi les prédictions positives : $\frac{TP}{TP+FP}$. Elle mesure la capacité du modèle à ne pas faire d'erreur lors d'une prédiction positive.
- **La sensibilité (ou recall) :** c'est également une mesure de performance qui se concentre sur les individus positifs. Elle correspond au taux d'individus positifs détectés par le modèle : $\frac{TP}{TP+FN}$. Elle permet ainsi de mesurer la capacité du modèle à détecter l'ensemble des individus positifs.
- **Le F1-score :** il est une mesure de performance d'un modèle de classification qui prend en compte à la fois la précision et la sensibilité (=taux de vrais positifs). Le F1-score atteint donc sa valeur maximale de 1 lorsque la précision et la sensibilité sont toutes deux à leur maximum, indiquant ainsi un modèle avec un équilibre optimal entre la précision et la sensibilité.
- **Le score de la validation croisée :** il est un outil d'évaluation robuste de la performance d'un modèle et notamment sa capacité de généralisation, c'est-à-dire comment le modèle se comporterait sur de nouvelles données en simulant l'apprentissage sur des sous-ensembles des données d'entraînement et en évaluant le modèle sur les sous-ensembles restants. Ce score est calculé en moyennant les performances du modèle sur chaque pli, avec ici la précision comme métrique.
- **La matrice de confusion :** elle est également un outil d'évaluation de la performance d'un modèle de classification. Elle permet de comparer les prédictions du modèle avec les véritables valeurs de la classe cible sur un ensemble de données de test.

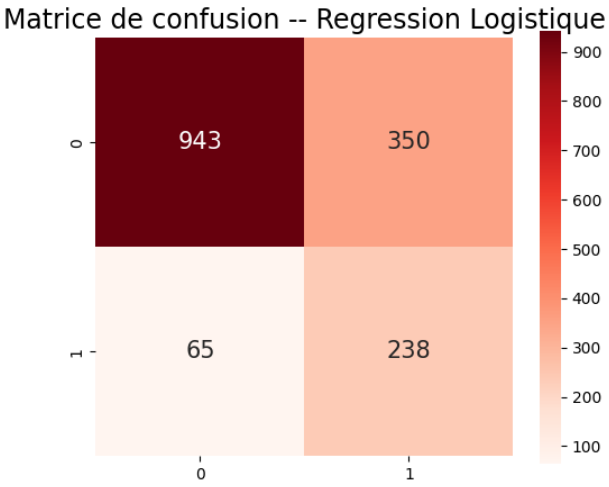
Pour résumer, afin d'évaluer la performance des modèles pour notre problème de classification, nous utiliserons la matrice de confusion afin d'avoir le détail des prédicitions pour chaque classe, mais également l'accuracy, la précision, la sensibilité, le F1-score.

4.1 Régression Logistique

Désormais que nous sommes dans un cadre de prédiction d'une variable binaire, l'outil idéal pour initier notre analyse de classification est la régression logistique. Cette méthode vise à modéliser la probabilité d'appartenir à la classe positive ($P[Y = 1]$) en utilisant une fonction logistique. Lorsque cette probabilité dépasse significativement 0.5, le modèle attribue l'observation à la classe positive ; sinon, elle est assignée à la classe négative. Le fonctionnement de la régression logistique repose sur la transformation de la somme pondérée des variables indépendantes à l'aide de la fonction logistique, qui est définie de la sorte : $p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$

Cette fonction logistique joue un rôle crucial dans la modélisation des probabilités. Elle prend en entrée la somme pondérée des variables, souvent appelée logit, et produit en sortie une probabilité compris entre 0 et 1. Cette transformation permet de capturer les relations complexes entre les variables indépendantes et la variable cible, tout en générant des probabilités interprétables pour la classification binaire. En ajustant les poids des variables, la régression logistique cherche à maximiser la vraisemblance des observations et à estimer les paramètres optimaux du modèle. Ainsi, en choisissant judicieusement les seuils de probabilité, la régression logistique offre un moyen efficace de prendre des décisions de classification dans un contexte de prédiction binaire.

Les résultats de ce premier modèle sont les suivants :



======Pour Regression Logistique======

	precision	recall	f1-score	support
0	0.94	0.73	0.82	1293
1	0.40	0.79	0.53	303
accuracy			0.74	1596
macro avg	0.67	0.76	0.68	1596
weighted avg	0.83	0.74	0.77	1596

Pour ce premier modèle de classification, nous allons approfondir l'analyse des résultats afin de les comprendre en détail. Cette compréhension approfondie nous permettra d'appliquer plus rapidement les interprétations aux modèles suivants que nous allons mettre en place.

Concernant la précision, on observe que 94% des observations prédites comme qualité moyenne (classe 0) sont effectivement de qualité moyenne. Seulement 40% des observations prédites comme très haute qualité (classe 1) sont effectivement de très haute qualité.

La valeur du recall pour la classe 0 est de 0.73 et pour la classe 1 est de 0.79, ce qui signifie que 73% des observations de qualité moyenne ont été correctement prédites,tandis que 79% des observations de très haute qualité sont correctement prédites.

La valeur du F1-score fournit une moyenne pondérée de la précision et du rappel, ce qui donne une mesure globale de la performance du modèle. Pour la classe 0, le F1-score est de 0.82, et pour la classe 1, il est de 0.53. Un F1-score de 0.82 pour la classe 0 est relativement élevé, ce qui suggère que le modèle a une bonne harmonie entre précision et rappel pour cette classe. Le modèle est donc capable de bien identifier et classifier les instances de la classe 0, tandis qu'il a plus des difficultés à bien prédire les instances de la classe 1, soit en les manquant (faux négatifs), soit en classant incorrectement d'autres instances comme appartenant à cette classe (faux positifs).

Enfin, la précision globale du modèle pour toutes les classes est de 74%. Cela représente le pourcentage total d'observations correctement prédites.

En conclusion, la régression logistique présente une performance relativement plus faible pour prédire la classe 1 (très haute qualité) que la classe 0 (qualité moyenne). Étant donné que notre objectif est d'identifier correctement les vins de haute qualité, nous accordons une priorité particulière à la précision dans la classification des individus positifs. Il est préférable de catégoriser un vin exceptionnel comme de qualité moyenne plutôt que de qualifier un vin moyen comme exceptionnel. Cependant, dans ce contexte, la précision est seulement de 40%, un niveau qui est jugé trop bas et inacceptable pour répondre à notre problématique de prédiction.

Afin d'identifier un modèle qui soit potentiellement plus performant que celui-ci, nous allons mettre en place différents modèles d'apprentissage supervisé, et comparé leurs performances grâce aux métriques adaptés à notre problématique. Pour garantir une clarté dans notre présentation et éviter des redondances, chaque section suivante se concentrera sur un modèle particulier, détaillant son fonctionnement et exposant ses résultats. L'interprétation complète de l'ensemble des résultats sera réalisée dans la conclusion de cette partie.

4.2 Arbre de décision

Parmi les modèles d'apprentissage supervisé existant, l'arbre de décision se distingue comme l'un des plus largement adoptés, pouvant être appliqué tant à des problèmes de classification que de régression. C'est un modèle qui prend des décisions en se basant sur les caractéristiques d'un ensemble de données. À chaque niveau, il sélectionne la meilleure caractéristique pour diviser les données. Ce processus se répète jusqu'à former un arbre complet.

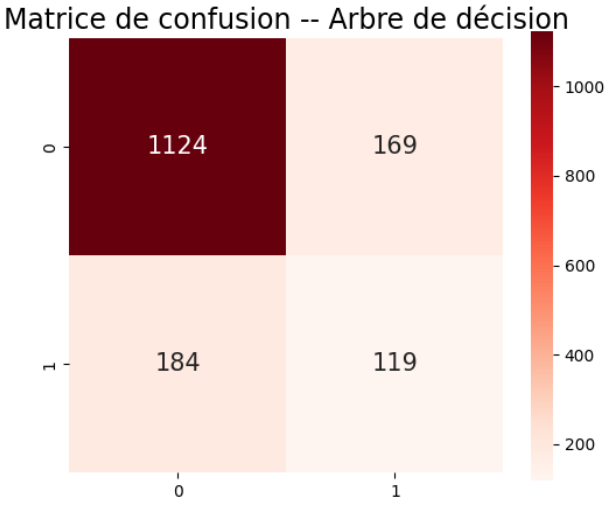
Lorsqu'une nouvelle donnée est introduite, elle suit le parcours décisionnel de l'arbre permettant de prédire la classe, dans le cas de la classifiation, ou la valeur, dans le cas de la régression.

Nous choisissons de mettre en place un arbre de décision dans notre analyse de classification en raison de sa polyvalence dans la résolution de problèmes, et de sa facilité d'intéprétation. Afin d'obtenir le modèle optimal, il est crucial de rechercher les meilleurs hyperparamètres nécessaires à

la construction d'un arbre de décision. Cette recherche vise à déterminer les valeurs optimales des paramètres qui définissent la profondeur maximale d'un arbre, les nombres minimum d'échantillons requis pour diviser un nœud interne et pour former une feuille, le nombre maximal de caractéristiques à considérer lors de la meilleure division, ainsi que la fonction utilisée pour évaluer la qualité de chaque division. Cette recherche est réalisée à l'aide de la technique du GridSearchCV, et les valeurs obtenues pour les hyperparamètres sont donc :

— Profondeur maximale de l'arbre('max_depth') : None (c'est-à-dire que la profondeur maximale des nœuds de l'arbre n'est pas limitée)
— Nombre minimum d'échantillons ('min_samples_split') : 2
— Nombre ('min_samples_leaf') : 1 - Nombre maximal de caractéristiques ('max_features') : auto (c'est-à-dire que le nombre maximum de caractéristiques à considérer lors de la recherche de la meilleure division à chaque nœud est automatiquement déterminé)
— Critère ('criterion') : gini (cela signifie que c'est l'indice de Gini qui est utilisé comme critère pour mesurer la qualité d'une division. L'indice de Gini mesure la pureté des nœuds dans un arbre. Il mesure avec quelle fréquence un élément aléatoire de l'ensemble serait mal classé si son étiquette était choisie aléatoirement selon la distribution des étiquettes dans le sous-ensemble)

Les résultats de l'arbre de décision optimal pour notre jeu de données sont :



======Pour Arbre de décision======

	precision	recall	f1-score	support
0	0.86	0.87	0.86	1293
1	0.41	0.39	0.40	303
accuracy			0.78	1596
macro avg	0.64	0.63	0.63	1596
weighted avg	0.77	0.78	0.78	1596

4.3 Forêt Aléatoire

Suite à l'arbre de décision mis en place, nous pouvons tenter de mettre en place une forêt aléatoire, qui, en plus de répondre au problème de régression comme nous l'avons fait précédemment, est particulièrement bien adaptée à des problèmes de classification binaire. Cette méthode repose sur la combinaison des prédictions de multiples arbres de décision individuels pour améliorer la robustesse et la précision globale du modèle.

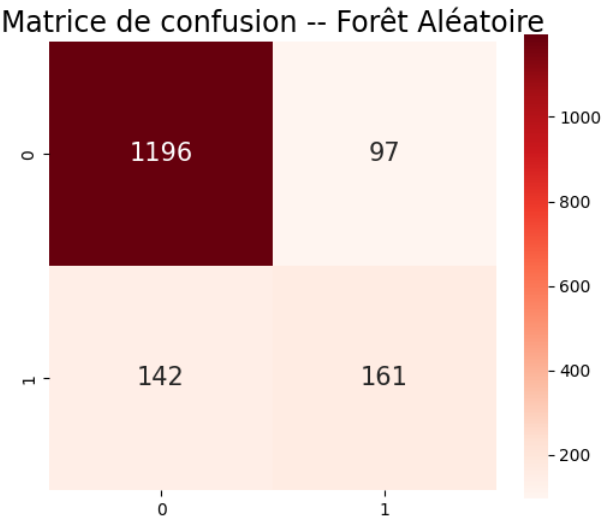
Une forêt aléatoire est constituée d'un ensemble d'arbres de décision, chacun formé sur un échantillon aléatoire des données d'entraînement. Cette diversification des données permet à chaque arbre d'apprendre des aspects différents du jeu de données, réduisant ainsi le risque de surajustement. Lorsqu'une nouvelle observation est introduite, chaque arbre de la forêt donne sa prédiction, et la classe majoritaire parmi ces prédictions est attribuée à l'observation.La forêt aléatoire est plus flexible et permet de traiter des ensembles de données complexes, ce qui la rend très performante pour les problèmes de classification binaire.

Nous implémentons une forêt aléatoire en utilisant les meilleurs hyperparamètres déterminés par GridSearchCV, de manière similaire à notre approche pour l'arbre de décision. De plus, nous explorons le nombre optimal d'arbres de décision qui seront inclus dans la forêt.

Ainsi, avec les hyperparamètres optimaux identifiés comme suit :

— Profondeur maximale (max_depth) : None
— Nombre minimal d'échantillons dans une feuille (min_samples_leaf) : 1
— Nombre minimal d'échantillons requis pour diviser un nœud interne (min_samples_split) : 2
— Nombre d'arbres de décision dans la forêt (n_estimators) : 50

Nous présentons ci-dessous les résultats obtenus par le modèle de forêt aléatoire :



```
======Pour Forêt Aléatoire======
              precision    recall  f1-score   support

           0       0.89      0.92      0.90      1293
           1       0.60      0.53      0.56       303

    accuracy                           0.84      1596
   macro avg       0.74      0.72      0.73      1596
weighted avg       0.84      0.84      0.84      1596
```

4.4 k-plus proches voisins (kNN)

Le k-Nearest Neighbors (kNN ou k-plus proche voisins) est un algorithme d'apprentissage supervisé utilisé pour la classification et la régression. C'est un algorithme dit "paresseux" qui n'apprend pas de modèle durant la phase d'entraînement, mais qui se contente de mémoriser les exemples d'entraînement pour les utiliser lors de la phase de prédiction. Ainsi, lorsqu'une nouvelle observation doit être classifiée (ou prédite), il va rechercher les k voisins les plus proches dans l'espace des caractéristiques en utilisant une mesure de distance. La prédiction sera basée sur les étiquettes des voisins trouvés, c'est-à-dire la classe majoritaire des k voisins dans le cas de la classification.

Cet algorithme ne fait donc aucune hypothèse sur la distribution des données, et peut capturer des relations non linéaires ou complexes entre la variable cible et les variables indépendantes. Il semble donc appliquable dans notre cas, car la relation entre la qualité du vin et les caractéristiques semble complexe, et ce modèle pourrait ainsi la capturer.

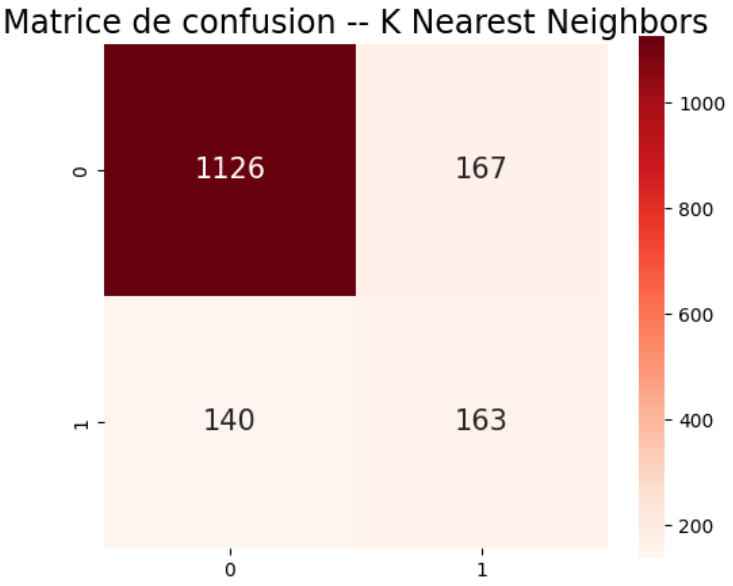
De plus, le choix du meilleur k, le nombre de voisins, est crucial dans le modèle kNN, car il peut avoir un impact significatif sur les performances du modèle. Il existe plusieurs méthodes pour le trouver, basé sur l'évaluation des performances du modèle pour différents k, ou bien sur la validation croisée. Notre cible présente un déséquilibre d'effectif entre ses deux classes. La précision, une métrique d'évaluation représentant le rapport du nombre de prédictions correctes sur le nombre total d'observations, n'est pas adapté lorsqu'il existe une classe majoritaire et une classe minoritaire. Une valeur élevée de la précision peut donner une fausse impression de performance, masquant des problèmes réels dans la prédiction de la classe moins fréquente. Il est alors préférable d'utiliser d'autres métriques d'évaluation telle que la F-mesure par exemple, qui prend en compte à la fois les faux positifs et les faux négatifs. Il donne une mesure équilibrée entre la précision et le rappel, aidant à évaluer la performance globale du modèle.

Nous allons donc chercher le meilleur k en entraînant un classificateur k-NN pour différentes valeurs de k, et évaluer ses performances sur l'ensemble de test en utilisant la F-mesure. On conservera la valeur de k associée au meilleur score F1 obtenu. Le k-NN est sensible à l'échelle des variables, et une normalisation peut être nécessaire lorsque ces échelles sont différentes, ou que la distribution des variabls n'est pas équilibrée. Dans notre cas la normalisation ne sera pas nécessaire car nous avons standardisé nos données lors de la préparation des ensembles de test et d'entrâinement, et nos variables présentent désormais les mêmes échelles de valeurs.

On constate que le meilleur score de la F-mesure est associé au modèle k-NN avec k=1. Nous

appliquons donc le modèle k-NN à nos données, et évaluons sa performance pour k=1.

```
Best Recall: 0.49393939393939396 Best K-Recall: 1
```




```
======Pour K Nearest Neighbors======
              precision    recall  f1-score   support

           0       0.89      0.87      0.88      1293
           1       0.49      0.54      0.52       303

    accuracy                           0.81      1596
   macro avg       0.69      0.70      0.70      1596
weighted avg       0.81      0.81      0.81      1596
```

4.5 Support Vector Machine (SVM)

Nous pouvons également tenté d'utiliser le SVM (Support Vector Machine), qui est un algorithme d'apprentissage supervisé utilisé principalement pour des problèmes de classification, mais également pour des tâches de régression. Étant dans un problème de classification binaire, nous présenterons l'utilisation et le fonctionnement du SVM dans ce cadre.

Son but principal est de trouver l'hyperplan dans un espace à haute dimension qui sépare le mieux les différentes classes de données. Ceci fonctionne donc très bien lorsque les classes sont dites "linéairement séparables", c'est-à-dire s'il existe un hyperplan B défini par $\{x \in \mathbb{R}^d \mid B_0 + B^T x = 0\}$ tel que $\forall i,\ Y_i = \text{sign}(B_0 + B^T X_i) \Leftrightarrow \begin{cases} Y_i = 1 & \text{si } B_0 + B^T X_i > 0 \\ Y_i = -1 & \text{sinon} \end{cases}$. Ainsi plusieurs hyperplans peuvent exister et le SVM effectue alors une optimisation pour en choisir un. Minimiser la distance entre l'hyperplan et les observations, en se limitant aux observations bien classées, $\min\left(\sum_i (y_i f_{\widetilde{\beta}}(x_i))_+\right)$, semble être un critère limité puisqu'il ne converge qu'avec des classes linéairement séparables. Le SVM cherche donc l'hyperplan qui maximise la marge entre les classes. La marge représente la distance maximale qui peut être tracée entre les points de données les plus proches des deux classes différentes. Elle permet à l'algorithme de déterminer une fontière de décision, ou un hyperplan, qui sépare de manière optimale les classes.

Cependant cette aproche est limitée aux classes linéairement séparables, ce qui n'est pas notre cas.

Dans ce cas, il est possible de mettre en place les "astuces du noyau", qui consiste à transformer les données afin de les projeter dans un espace de dimension supérieure où elles peuvent être séparés linéairement. Différents noyaux peuvent être utilisés pour le SVM, ce qui détermine la nature de la transformation des données.

- linéaire : c'est le noyau simple qui ne modifie pas les données, utilisé quandd les données sont déjà linéairement séparables : $k(x, y) = \langle x, y \rangle_{\mathbb{R}^d}$
- polynomial : il transforme les données en ajoutant des combinaisons polynomiales des caractéristiques, permettant une séparation linéaire dans l'espace transformé : $k(x, y) = (\langle x, y \rangle_{\mathbb{R}^d} + c)^d$
- RBF (=Radial Basis Function) ou Gaussian : il projette les données dans un espace de dimension infinie, créant une frontière autour de chaque point de données, permettant de séparer pratiquement n'importe quel ensemble de données : $k(x, y) = \exp\left(-\frac{(x-y)^2}{2}\right)$

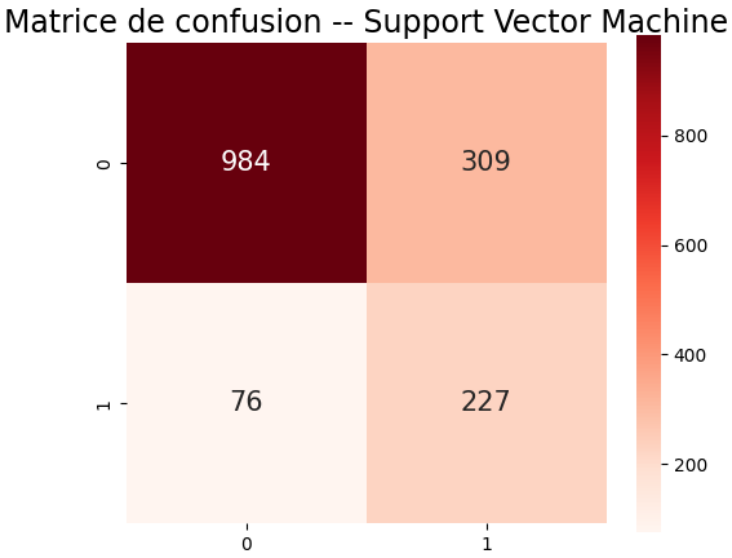
Comme pour l'ensemble des modèles, il est primordial de mettre en place un modèle de SVM avec des hyperparamètres optimisés. Les paramètres clés que nous allons chercher à optimiser grâce à une valdiation croisée sont :

- C : la paramètre C (coût) contrôle le compromis entre avoir une marge maximale et minimiser les erreurs de classification. Un C élevé permet une marge plus étroite mais peut ocnduire à un surajustement tandis qu'un C plus faible donne une marge plus large mais peut conduire à un sous ajustement.
- Kernel : comme nous l'avons détaillé précédemment, le choix du noyau est important afin de déterminer la fonction utilisée pour transformer les données dans un espace de caractéristiques de dimension supérieure.

Les valeurs optimales des hyperparamètres du SVM sont :

— Paramètre coût (‘C’) : 10
— Noyau (‘kernel’) : rbf

Voici les résultats du modèle SVM optimal :




```
======Pour Support Vector Machine======
              precision    recall  f1-score   support

           0       0.93      0.76      0.84      1293
           1       0.42      0.75      0.54       303

    accuracy                           0.76      1596
   macro avg       0.68      0.76      0.69      1596
weighted avg       0.83      0.76      0.78      1596
```

4.6 Boosting

Nos ensembles de données présentent une structure assez complexe, et il serait judicieux, pour terminer, d'utiliser des modèles de boosting. Les modèles de boosting sont une classe d'algorithmes d'apprentissage automatique qui combinent plusieurs modèles plus faibles pour créer un modèle global plus fort. Les "apprenants faibles" sont introduits de manière séquentielle, et exploitent les faiblesses des apprenants précédents. Un apprenant faible peut être utilisé plusieurs fois, et être choisi comme l'optimal à plusieurs reprises. Ainsi, contrairement à un modèle spécifique, le boosting est un algorithme générique, qui nécessite la spécification d'un modèle faible (tel qu'une régression, un arbre de décision, etc...) afin de l'améliorer ensuite.

L'idée fondamentale derrière le boosting est d'entraîner l'un après l'autre plusieurs modèles relativement faibles, c'est-à-dire pour lequel on est en situation de sous-ajustement des données, en demandant à chaque modèle d'essayer de corriger les erreurs résiduelles de son prédécesseur. Cette stratégie vise à améliorer les classificateurs qui ne fonctionnent pas bien, qui sont parfois à peine plus performants qu'une supposition aléatoire.

Les erreurs du modèle précédent sont corrigées grâce à un ajustement itératif du poids des observations dans l'ensemble d'entraînement, en donnant plus de poids aux observations mal classées lors de l'itérations précédente, et moins de poids à celles bien classées. Ainsi chaque modèle faible est construit en mettant l'accent sur les erreurs des modèles précédents, ce qui améliore progressivement la performance globale.

Mathématiquement voici ce que le boosting fait :

Supposons que nous avons un ensemble de données d'entraînement $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ où x_i sont les caractéristiques et y_i sont les étiquettes.

- Poids des observations :
  - Chaque observation (x_i, y_i) est associée à un poids $w_i \geq 0$
  - Les poids w_i sont initialisés uniformément à $w_{i_0} = \frac{1}{n}$ où n correspond au nombre total d'observations. Cela signifie donc qu'au départ, chaque observation possède un poids égal, ce qui est une initalisation sans a priori.
- Apprenants faibles :
  - Un apprenant faible est défini comme étant légèrement meilleur qu'une prédiction aléatoire. L'ensemble de nos apprenants faibles est défini par : $G = \{g_b \mid 1 \leq b \leq B\}$. Un premier modèle est donc ajusté à nos données, en utilisant les poids des observations initialisés.
- Quantification de l'erreur à partir des observations pondérées :
  - On mesure à chaque fois l'erreur de classification de l'apprenant faible à l'aide d'une fonction $R_n(g_b, w)$ qui dépend du modèle g_b et des poids w.
  - La formule de l'erreur empirique pondérée (R_n) de l'apprenant faible se définit comme la somme pondérée des indicateurs d'erreurs. Ces indicateurs d'erreurs, représentés par la fonction indicatrice, évaluent si la classe prédite par le modèle (g_b) diffère de l'étiquette réelle (y_i) pour chaque observation. Cette mesure tient compte des poids (w_i) attribués à chaque observation en fonction de sa classification, qu'elle soit correcte ou incorrecte.

— Les poids w_i jouent un rôle crucial dans l'erreur empirique pondérée. Un poids nul ($w_i = 0$) signifie que l'observation (x_i, y_i) ne joue aucun rôle dans le modèle. Tandis qu'un poids élevée (w_i important) indique que commettre une erreur sur l'observation (x_i, y_i) est coûteux.

— Apprenants faibles suivants :

  — L'apprenant faible suivant est formé à partir du nouvel ensemble de données pondéré. Comme précédemment on mesure son erreur de classification, et on met à jour les poids associés aux observations en fonction de ses performances.

— Classificateur final :

  — L'ensemble des apprenants faibles sont combinés afin de former un apprenant fort. La classification finale prédite est obtenue en combinant les prédictions de tous les apprenants faibles. La combinaison se fait de manière pondérée, où chaque modèle faible contribue à la prédiction finale avec un poids déterminé par son niveau de performance.

Les modèles de boosting sont donc très utilisés pour des problèmes de classification et de régression, en raison de leurs performances élevées et de leur capacité à gérer des ensembles de données complexes. Il existe deux grands algorithmes de boosting : l'AdaBoost et le Gradient Boosting.

4.6.1 AdaBoost

L'un des premiers modèles de boosting introduit en classification est le AdaBoost (Adaptive Boosting). C'est un algorithme d'apprentissage ensembliste qui s'appuie sur des procédures d'apprentissage séquentielles de plusieurs arbres de décision. Sa technique présente donc des similitudes avec la forêt aléatoire mais dans lequel les modèles apprennent en parallèle grâce au bagging.

Il combine des classificateurs faibles, qui sont généralement des arbres de décisions à une seule division, de manière à créer un classificateur fort.

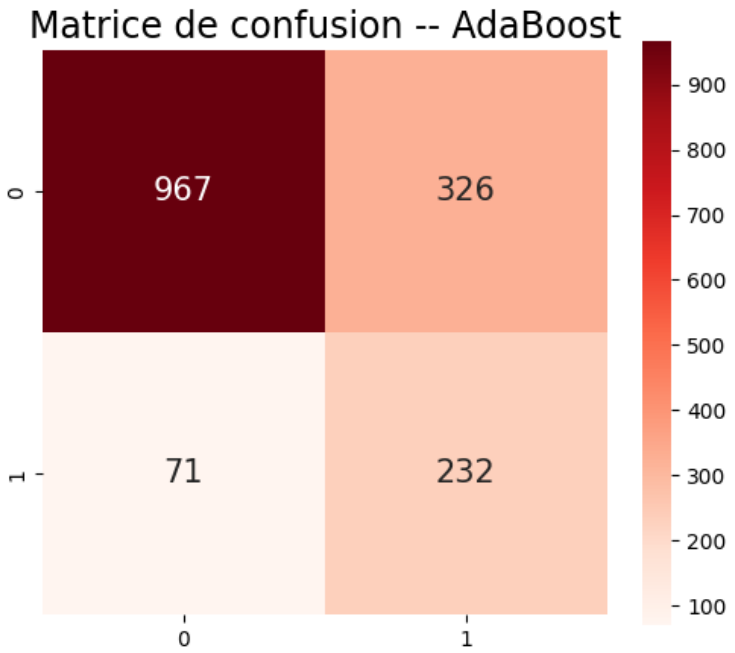
Pour mettre en œuvre cet algorithme, nous devons d'abord ajuster ses paramètres de manière optimale. L'étape clé de cette mise en place consiste à déterminer le nombre optimal d'arbres à entraîner dans le modèle Adaboost, ainsi que le moment optimal pour arrêter cet entraînement. Cette étape cruciale sera réalisée grâce à la validation croisée. Les deux paramètres essentiels à optimiser sont les suivants :

— Nombre d'estimateurs faibles ('nb_estimators') qui représente donc le nombre d'arbres de décisions à construire. Son optimisation revêt une importance cruciale pour éviter le surapprentissage en cas d'une valeur trop élevée ou le sous-apprentissage dans le cas contraire.
— Taux d'apprentissage ('learning_rate') qui détermine la contribution de chaque arbre de décision à la prédiction finale. Son ajustement approprié est fondamental pour influencer de manière équilibrée l'ensemble du modèle.

Pour notre jeu de données sur les vins, les hyperparamètres obtenus pour le modèle AdaBoost sont les suivants :

— Nombre d'estimateurs faibles (nb_estimators) 150
— Taux d'apprentissage (learning_rate) : 0.2

Les résultats de classification du modèle optimal AdaBoost sont :



======Pour AdaBoost======

	precision	recall	f1-score	support
0	0.93	0.75	0.83	1293
1	0.42	0.77	0.54	303
accuracy			0.75	1596
macro avg	0.67	0.76	0.68	1596
weighted avg	0.83	0.75	0.77	1596

Ce modèle de boosting peut être sensible au bruit et aux valeurs aberrantes. Des variantes plus avancées telles que Gradient Boosting, XGBoost, et LightGBM ont été développées pour améliorer les performances et la robustesse du boosting.

4.6.2 Gradient Boosting

Le Gradient Boosting est une technique de boosting, dont l'objectif principal est de minimiser la fonction de perte globale du modèle en ajustant de manière séquentielle les prédictions du modèle pour corriger les erreurs résiduelles du modèle précédent.

L'algorithme commence par initialiser un modèle de base afin de définir la prédiction initiale, qui dans le cas d'une classification binaire, représente la classe majoritaire. Les résidus du modèle, qui correspondent à l'erreur entre les prédictions actuelles et les vrais valeurs cibles sont calculées. Ainsi, à chaque étape, un nouvel arbre est entraîné sur les résidus calculés du modèle précédent, et on utilise une procédure de descente de gradient pour optimiser les poids de ce modèle faible. Les poids sont déterminés en minimisant une fonction de perte, souvent définie comme la somme des

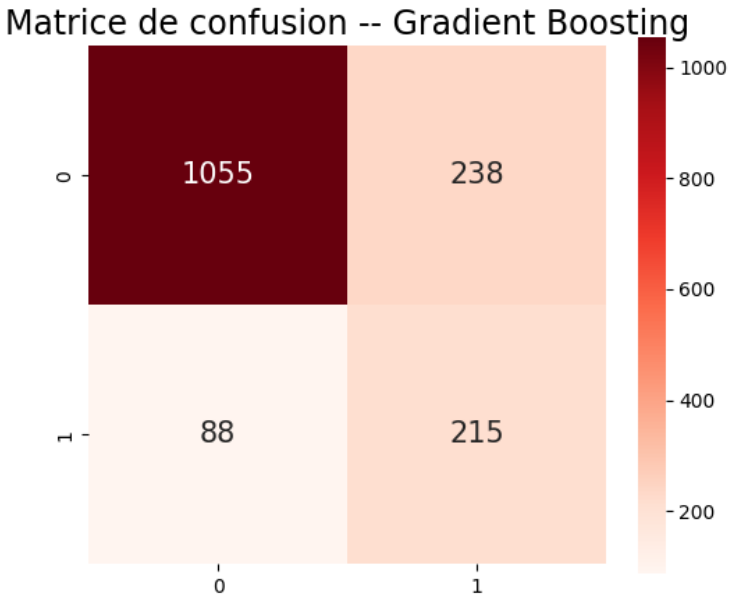
carrés des résidus pour la régression ou la déviance pour la classification. [mettre formule] Puis, les prédictions de cet arbre sont ajoutées aux prédictions existantes, et on met à jour le modèle global après chaque itération en ajoutant une version pondérée du modèle faible. Cette procédure itérative permet ainsi au modèle global de s'améliorer progressivement et de mieux capturer les relations complexes présentes dans les données, en évitant le surajustement. La prédiction finale est donc obtenue en sommant avec une pondération les prédictions individuelles de tous les modèles faibles, qui ont été entraînés successivement au cours du processus d'entraînement.

Comme pour l'algorithme du AdaBoost, il est primordial d'optimiser les paramètres du Gradient Boosting. En plus de trouver la meilleure valeur de 'nb_estimators' et de 'learning_rate', nous devons optimiser le paramètre 'loss' qui représente la fonction de perte utilisée pour mesurer l'écart entre les prédictions du modèle et les vraies valeurs.

Les meilleurs hyperparamètres obtenus sont :

— Nombre d'estimateurs faibles ('n_estimators') : 150
— Taux d'apprentissage ('learning_rate') : 0.2 - Fonction de perte ('loss') : deviance

Les résultats de classification du modèle optimal de Gradient Boosting sont :



======Pour Gradient Boosting======

	precision	recall	f1-score	support
0	0.92	0.82	0.87	1293
1	0.48	0.71	0.57	303
accuracy			0.80	1596
macro avg	0.70	0.76	0.72	1596
weighted avg	0.84	0.80	0.81	1596

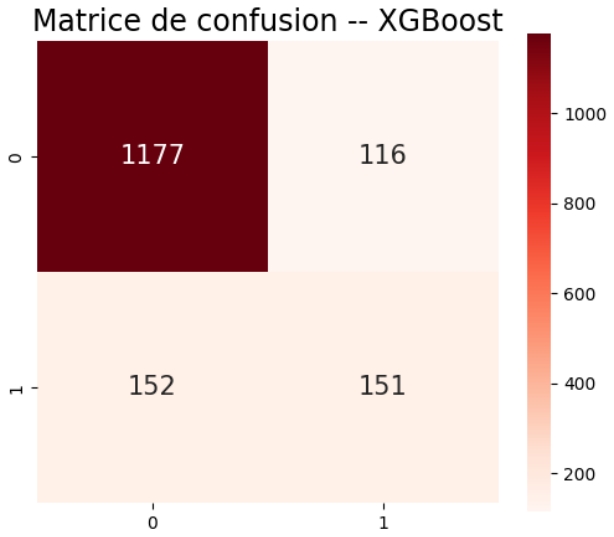
4.6.3 XGBoost

XGBoost (Extreme Gradient Boosting) est un modèle amélioré de l'algorithme d'amplification de gradient (le Gradient boosting). Il introduit des techniques de régularisation (tel que le LASSO, le RIDGE, ou l'Elastic Net) pour contrôler la complexité des arbres et gérer les valeurs manquantes de manière plus efficace. Ce modèle est plus efficace et performant que le Gradient Boosting car il utilise le parallélisme. Ses paramètres et son fonctionnement permettent également de mieux gérer le cas des classes déséquilibrées. Afin donc d'obtenir un modèle XGBoost le plus performant possible, nous devons définir ses meilleurs hyperparamètres dans notre cadre. Il existe dans ce modèle un très grand nombre de paramètres, et parmi ceux-ci on retrouve le taux d'apprentissage, le nombre d'estimateurs (le nombre d'arbres à ajouter à l'ensemble), la profondeur maximale des arbres, la fraction des échantillons d'entraînement utilisés pour ajuster chaque arbre et la fraction de caractéristiques à prendre en compte lors de la construction de chaque arbre.

Voici pour notre jeu de données les valeurs optimales des hyperparamètres :

— Taux d'apprentissage ('learning_rate') : 0.2 - Nombre d'estimateurs ('n_estimators') : 200
— Profondeur maximale des arbres ('max_depth') : 5
— Fraction des échantillons d'entraînement ('subsample') : 0.8
— Fraction de caractéristiques ('colsample_bytree') : 1.0

Les résultats de classification du modèle optimal XGBoost sont :



======Pour XGBoost======

	precision	recall	f1-score	support
0	0.89	0.91	0.90	1293
1	0.57	0.50	0.53	303
accuracy			0.83	1596
macro avg	0.73	0.70	0.71	1596
weighted avg	0.82	0.83	0.83	1596

4.7 Ensemble des résultats

Voici un tableau récapitulatif présentant les performances de classification de chaque modèle implémenté :

	Modèle de classification	F1 mesure	Précision	Recall	Accuracy
2	Forêt Aléatoire	0.560420	0.597015	0.528053	0.842732
7	XGBoost	0.529825	0.565543	0.498350	0.832080
3	K Nearest Neighbors	0.515008	0.493939	0.537954	0.807644
6	Gradient Boosting	0.570292	0.476718	0.709571	0.796992
4	Support Vector Machine	0.541120	0.423507	0.749175	0.758772
5	AdaBoost	0.538908	0.415771	0.765677	0.751253
1	Arbre de décision	0.402707	0.413194	0.392739	0.778822
0	Regression Logistique	0.534231	0.404762	0.785479	0.739975

Comme nous l'avons déjà dit, notre préoccupation principale dans notre classification est de minimiser les faux positifs (c'est-à-dire classer un vin de moyenne qualité dans la classe haute qualité), ainsi la métrique à privilégier est la précision. En se basant sur la précision, le modèle de Forêt Aléatoire semble être le choix optimal avec une précision de 60%, suivie par XGBoost avec une précision de 56%. Cependant, il est important de noter que le modèle de Forêt Aléatoire présente également un bon équilibre entre la précision et le recall (le F1-score) tandis que le modèle XGBoost a un recall inférieur. De plus, la Forêt Aléatoire présente une accuracy de 84% ce qui signifie que ce modèle a correctement prédit la bonne classe pour environ 84% de toutes les observations dans votre ensemble de données de test. Cela inclut à la fois les vrais positifs (cas où le modèle a correctement prédit haute qualité) et les vrais négatifs (cas où le modèle a correctement prédit moyenne qualité).

Par conséquent, si notre objectif est d'obtenir un modèle exigeant en ce qui concerne les observations classées positivement, afin de minimiser autant que possible les erreurs de type I, le modèle de Forêt Aléatoire semble être le choix le plus approprié.

En revanche, si notre priorité est de détecter le maximum de vins de haute qualité, même au risque de classer certains vins moyens comme excellents, nous optimisons nos modèles en fonction du rappel, ce qui nous donne les performances suivantes :

	Modèle de classification	F1 mesure	Précision	Recall	Accuracy
3	K Nearest Neighbors	0.506000	0.362984	0.834983	0.690476
0	Regression Logistique	0.534231	0.404762	0.785479	0.739975
4	AdaBoost	0.538908	0.415771	0.765677	0.751253
5	Gradient Boosting	0.569536	0.475664	0.709571	0.796366
6	XGBoost	0.565789	0.563934	0.567657	0.834586
2	Forêt Aléatoire	0.555759	0.622951	0.501650	0.847744
1	Arbre de décision	0.431579	0.460674	0.405941	0.796992

Le modèle kNN, avec k=47, affiche la meilleure performance en termes de rappel, atteignant 83% d'observations correctement classées (classe positive ou négative). En fonction de notre objectif, il devient évident que le choix d'un modèle plutôt qu'un autre dépendra de cette métrique spécifique. En conclusion, il semble évident que notre cadre d'analyse se prête davantage à une classification binaire, avec une classe "cible" bien définie, plutôt qu'à une régression. Il semble ainsi possible, tout en tenant compte d'une marge d'erreur, de pouvoir prédire si un vin est de qualité excellente ou non.

5 CONCLUSION

Pour rappel, notre analyse a débuté avec deux bases de données distinctes, l'une concernant le vin rouge et l'autre le vin blanc. Après une exploration détaillée, des tests statistiques et la constatation de la sginificativité de la variable 'type' dans la prédiction de la qualité, nous avons pris la décision de fusionner ces deux ensembles de données. L'objectif était de prédire la qualité du vin indépendamment de son type.

Notre variable cible, la qualité, qui peut être considérée comme une variable continue, peut tenter d'être prédite grâce à des modèles de régression, ce que nous avons tenté de faire dans la première partie de notre analyse. Nous avons commencé par une régression multiple simple et globale, puis avons exploré des régressions régularisées pour la sélection de variables, sans toutefois obtenir d'amélioration significative. Face à une complexité et une non-linéarité potentielles dans nos données, nous nous sommes tournés vers la forêt aléatoire, qui a montré des performances supérieures bien que toujours améliorables. Par la suite, nous avons donc envisagé la réduction de dimension par l'ACP pour potentiellement optimiser les performances des modèles. Cependant, cette approche n'a pas été concluante, principalement en raison de l'hypothèse de linéarité non respectée entre la variable cible et les variables indépendantes.

Face à la difficulté de prédire la qualité du vin par une régression, nous avons réorienté notre problématique vers une prédiction plus précise : déterminer si un vin est de très haute qualité ou non, établissant ainsi une classification binaire. Nous avons examiné divers modèles, dont la régression logistique, l'arbre de décision, la forêt aléatoire, le k-Plus Proche Voisin, le SVM, ainsi que des techniques de boosting telles que l'AdaBoost, le Gradient Boosting et le XGBoost.

Notre évaluation des performances s'est concentrée sur la précision et le F1-score, privilégiant la correction de la prédiction de la classe positive. Notre objectif était de minimiser les faux positifs et ainsi de prédire avec la plus grande précision possible les vins de haute qualité. La Forêt Aléatoire s'est avérée être le modèle le plus performant pour ces métriques, en assurant une classification précise des vins de haute qualité comme excellents. Cependant, si l'on considère uniquement le taux global de bonnes prédictions du modèle, le kNN se démarque comme le meilleur choix.

Bien que nos modèles démontrent des performances appréciables, il est essentiel de reconnaître qu'il reste difficile d'atteindre une prédiction parfaite de la qualité d'un vin en se basant uniquement sur ses composants physiques et chimiques. La notation de la qualité d'un vin, évaluée sur une échelle de 0 à 10, demeure intrinsèquement subjective. Même lorsque cette évaluation est effectuée par des experts, les préférences individuelles, les goûts personnels, et la sensibilité de chacun restent des biais humains difficiles à modéliser de manière exhaustive.

Une recherche portée sur l'origine du vin, tel que son cépage, pourrait être envisagée, car elle repose sur des données scientifiques plutôt que sur des impressions subjectives. Une étude menée conjointement par des chercheurs des universités de Genève et de Bordeaux a mis en lumière la possibilité de déterminer avec précision l'origine de plusieurs grands crus bordelais grâce à l'application de l'Intelligence Artificielle sur leurs composants chimiques. L'utilisation des informations relatives à la composition du vin, tant de notre part que de celle des chercheurs, ouvre ainsi de nouvelles perspectives. Elle offre la possibilité d'exploiter les composants du vin pour des exercices de prédiction ou de classification. Il devient désormais possible de transcender les frontières traditionnelles de l'appréciation sensorielle du vin, en intégrant des méthodes d'analyse et d'apprentissage statistique. Ces développements peuvent non seulement enrichir notre compréhension du vin, mais également ouvrir de nouvelles possibilités pour son utilisation et son appréciation.