Université de Lille

Faculté des Sciences et Technologies
Département de Mathématiques

Projet de Méthode d'apprentissage

Master2 Ingénierie Statistique et Numérique

Construction d'un prédicateur discriminant les eaux potables et non potables

---

Réalisé par :

Martin DESMAZIERES
El Moustapha MALICK

19 décembre 2022

Table des matières

Liste des tableaux

Table des figures

1 Introduction

Le but de ce projet et de construire un classifieur capable de distinguer une eau potable d'une eau non potable. Pour construire ce classifieur on pourra s'appuyer sur une base de donnée de 7 999 individus évalués en 20 variables. Au cours de ce travail nous testerons et comparerons les performances de différents algorithmes.

Après un rapide examen des données on se rend compte qu'il nous manque la valeur d'une variable est les labels pour trois individus. On pourrait essayer de compléter ces trous par exemple en leur attribuant le label des individu qui leur ressemble le plus d'après un critère de distance euclidienne mais cela reste une opération assez périlleuse. On préfère ici considérer que la suppression de 3 individus sur 7999 ne changera pas grand chose à l'efficacité de nos classifieurs et on supprime ces trois individus. On considère donc désormais une base de données de 7 996 individus pour 20 variables.

Pour commencer ce travail nous procéderons a une analyse statistique plus poussée des données. Dans un second temps nous appliquerons différents algorithme de classification. Enfin, dans une troisième partie nous tenteront de combiner les performances des algorithmes en construisant un **super-learner**.

2 Étude préliminaire

2.1 Statistiques descriptives

L'étude statistique sert à mettre en évidence les caractéristiques des variables qui pourraient compromettre l'efficacité des algorithmes. Les algorithmes peuvent être sensibles aux données extrêmes, à la multicolinéarité et à la proportion de chaque label. Nous examinerons tour à tour chacune de ces caractéristiques.

2.1.1 Valeurs extrêmes

Les données extrêmes peuvent être à l'origine d'une instabilité des résultats pour les algorithmes dont les coefficients s'expriment comme une fonction des données (notamment une moyenne, très sensible aux valeurs extrêmes). En fonction de si elles sont présentes dans l'échantillon d'entraînement ou non la valeur des coefficients peut fortement varier. Pour les détecter on représente les histogrammes des variables. La distribution de deux d'entre eux se démarque. On représente les histogrammes des variables "arsenic" et "aluminium" dans la figure 1.

Les deux variables comportent de nombreuses valeurs "extrêmes". Mais celle-ci sont tellement nombreuses et proches qu'elles ne représentent potentiellement pas une menace pour la stabilité de nos résultats : la proportion de valeurs élevées devrait être similaire et assez stable dans les échantillons d'entraînement et de test.

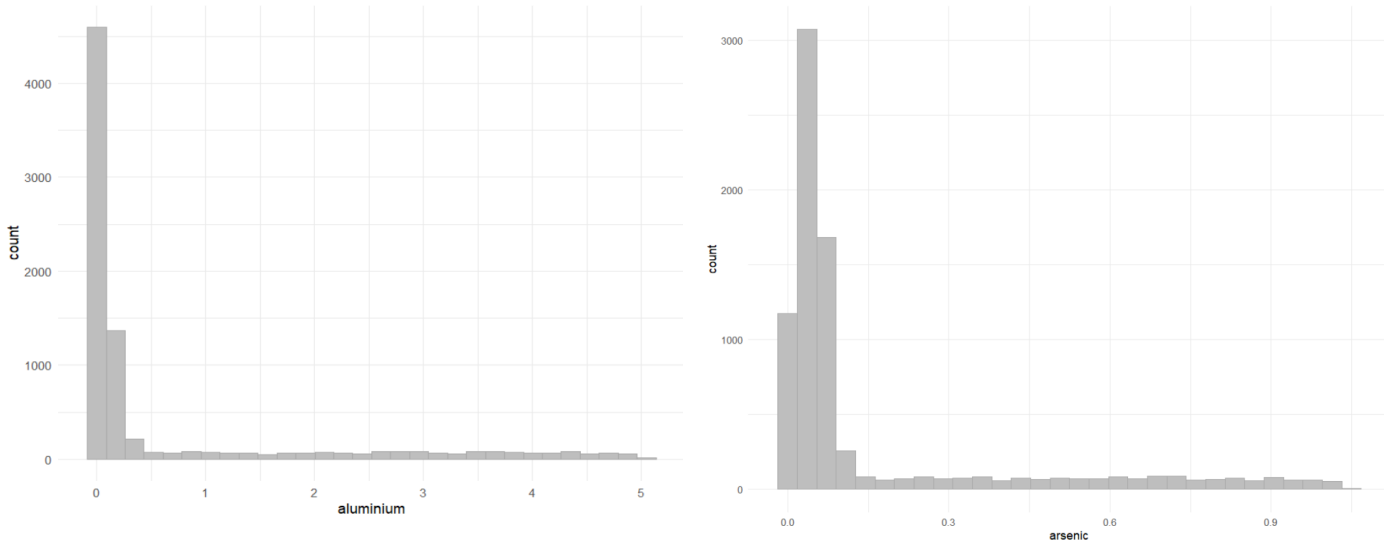
2.1.2 Multicoliéarité

La multicolinéarité est un problème qui survient lorsque certaines variables de prévision du modèle mesurent le même phénomène. Une multicolinéarité prononcée s'avère problématique, car elle peut augmenter la variance des coefficients et les rendre instables et difficiles à interpréter. Deux variables colinéaires sont fortement corrélées. On analyse donc la matrice de corrélation des variables représentée dans la figure 2.

Figure 1: Histogrammes des variables aluminium et arsenic

(a) Variable aluminium

(b) Variable arsenic



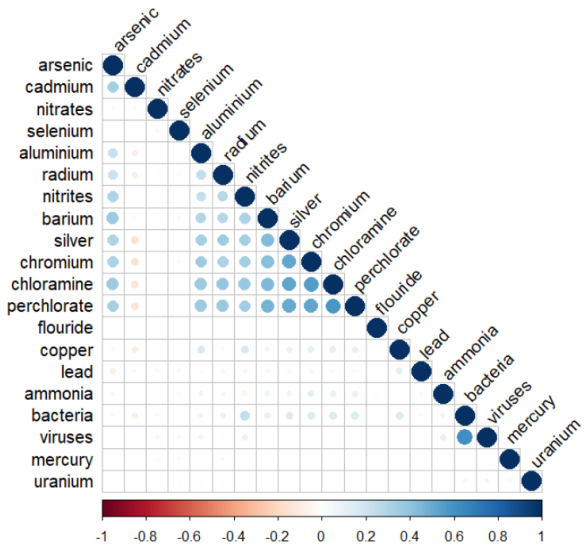


Figure 2: Matrice de corrélation des variables

La plus grande corrélation est celle entre les variables "viruses" et "bacteria", de 66%. Il est donc assez raisonnable de conclure à une absence de colinéarité entre les variables.

2.1.3 Cardinal des classes

Les données déséquilibrées peuvent aboutir à une perte de performance des algorithmes de classification. Les algorithmes sont construits de façon à minimiser le taux de mauvais classement, taux auquel la classe minoritaire contribue peu. Les méthodes risquent donc de se focaliser sur la classe majoritaire.

Effectivement nos données sont assez déséquilibrées. On observe seulement 11.41% d'échantillons d'eau potable. Avec une telle disproportion nos modèles risquent de minimiser le taux de mauvais classement de l'eau non potable, c'est-à-dire l'erreur consistant à prédire comme potable de l'eau non potable. Dans le cas présent ce n'est peut-être pas si grave : dans la plupart des situations il vaut mieux se tromper en prédisant de l'eau potable comme non potable que l'inverse (par exemple pour des raisons sanitaires pour minimiser les intoxications). Il n'est

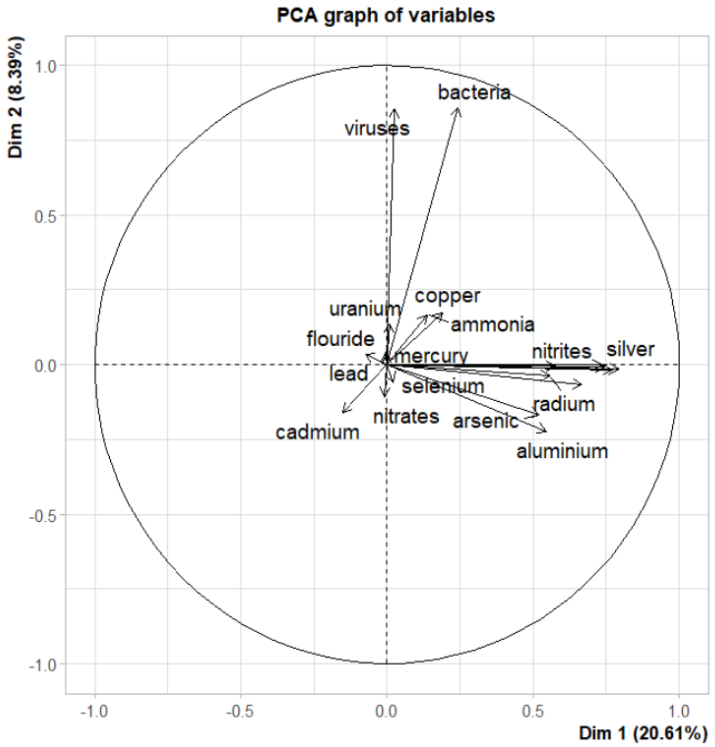
donc pas forcément nécessaire de rééquilibrer les données.

2.2 Analyse en Composantes Principales (ACP)

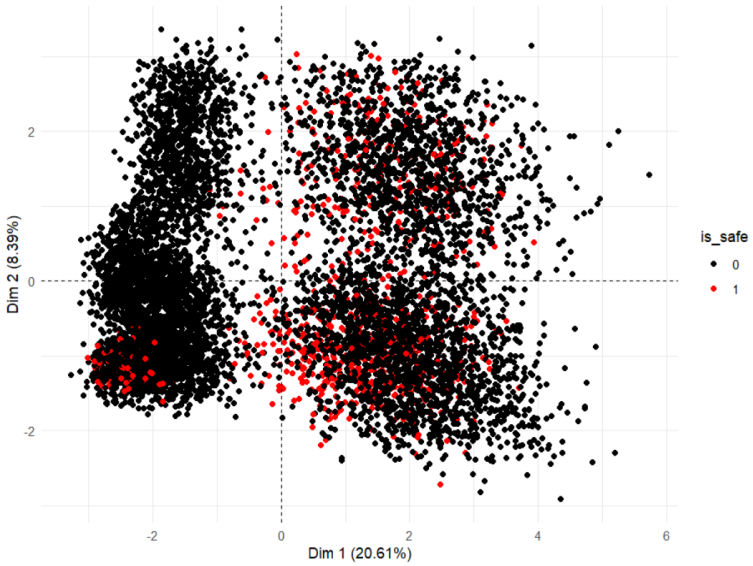
Réaliser une ACP permet de donner des intuitions sur la classification que l'on va réaliser, de savoir si les variables vont bien discriminer les labels, d'anticiper l'implication de chaque variable et leur potentielle influence sur chaque classe. Le graphe 3 montre les graphes des individus et le graphe des variables de l'ACP sur les deux premiers axes.

Figure 3: Graphes des indidus et des variables de l'ACP

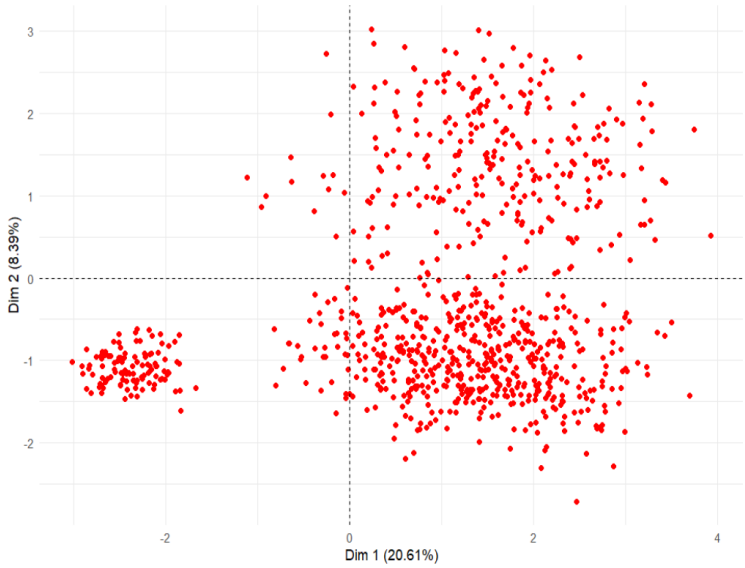
(a) Variables de l'ACP



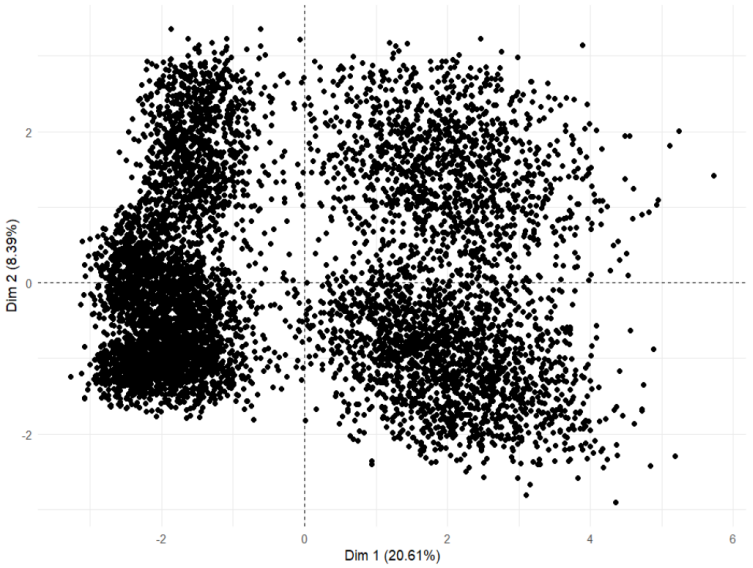
(b) Individus de l'ACP



(c) Individus de label potable



(d) Individus de label non potable



Ces deux axes représentent à eux deux 29% de la variabilité totale.

- L'axe 1 porte 20.61% de la variabilité. Il discrimine les individus présentant de fortes valeurs pour les variables chloramine, perchlorate, chromium et silver des autres (variables dont les coordonnées et le cos2[1] sont supérieures à 0.5 sur l'axe1).
- L'axe 2 porte 8.39% de la variabilité. Il discrimine les individus présentant de fortes valeurs pour les variables bacteria et viruses des autres (variables dont les coordonnées et le cos2 sont supérieures à 0.5 sur l'axe2).

On remarque que les données forment deux clusters plus ou moins équivalents le long du 1^{er} axe, celui restituant la plus grande part de la variabilité. Mais alors que les eaux non potables semblent assez bien réparties dans chaque cluster, les eaux potables sont beaucoup moins présentes dans le cluster le plus à gauche sur l'axe. Plus notable encore, les eaux potable sont quasiment totalement absentes du carré en haut à gauche du graphe.

On peut donc conclure, avec prudence car on a analysé moins d'un tiers de la variabilité, que les variables permettent de différencier les labels. Par exemple on peut affirmer que les eaux potable ont peu de chance de combiner de faibles valeurs pour les variables chloramine, perchlorate, chromium et silver et de fortes valeurs pour les variables viruses et bacteria. Les eaux potables ont de manière générale tendance à présenter de fortes valeurs pour les variables chloramine, perchlorate, chromium et silver.

3 Algorithmes de classifications

Dans cette section on fait tourner différents algorithmes de classification. On appliquera tour à tour une classification logistique, une forêt aléatoire et une méthode SVM.

Avant de lancer tout algorithme on divise une fois pour toute notre échantillon en échantillon *train* sur lequel nous entraînerons notre modèle et en échantillon *test* sur lequel nous mesurerons les performances de notre classifieur pour le comparer aux autres. On évaluera les performances de chaque algorithme en mesurant la proportion totale d'individus mal classés (faux négatifs et faux positifs confondus).

3.1 Classification logistique

On a vu en première partie que le jeu de données ne présentait pas de données extrêmes ou de multicolinéarité. On peut donc appliquer une classification logistique qui offre le double avantage de ne pas demander de paramètre exogène à optimiser et d'être facilement interprétable grâce au calcul des odd-ratios.

3.1.1 Prévisions

On fait tourner notre algorithme. On obtient une erreur de 9.60%. On affiche la matrice de prédiction dans la figure 4.

Les lignes de la matrice de confusion représente les labels réels quand les colonnes représentent les prédictions. Une matrice parfaite est donc diagonale. Ainsi, dans notre matrice, 40 eaux non potables ont été classées en eau potable (faux positifs) quand 152 eaux potables on été classées comme eau non potable (faux négatifs). Équilibrer les données fait augmenter le taux de faut positifs on préfère donc conserver nos résultats actuels.

[1]Le cos2 est un indicateur de la qualité de la représentation d'une variable sur le plan. Une variable de cos2 faible ne peut être analysée sans risquer de commettre des erreurs

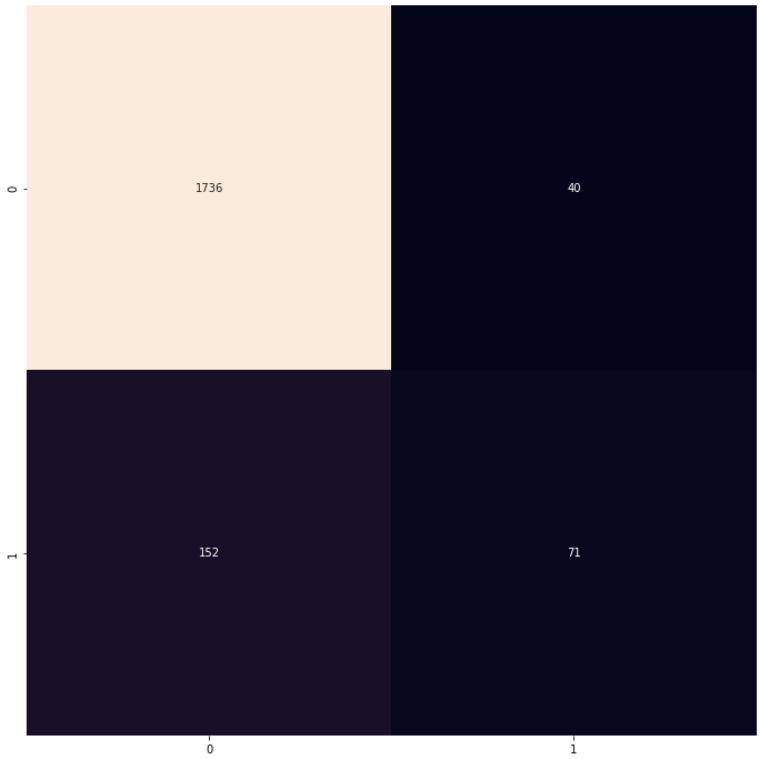


Figure 4: Matrice de confusion classification logistique

3.1.2 Stabilité et interprétation des résultats

On effectue 100 simulations de notre algorithme en faisant attention à retirer des échantillons *train* et *test* à chaque itération. On obtient une erreur moyenne de 9.40 (IC(95%) = [8.26 ; 10.48]) et une très faible variance de 3.19e-05. On a donc un algorithme assez stable. Les résultats sont facilement interprétable grâce aux odds-ratios. Les odds-ratio sont des coefficient affiliés à chaque variable. Dans le cas d'une classification binaire un odd-ratio représente les chances d'appartenir à la classe 1 (eau potable) putôt qu'à la classe 0 (eau non potable) si la variables prédictive augmente d'une unité, toutes les autres demeurant constantes. On affiche dans la figure 2 la moyenne et l'intervalle de confiance des odds-ratio les plus éloignés de 1.

Variable	aluminium	arsenic	cadmium	chromium	bacteria	mercury	selenium	uranium
Odd-ratio	2.0539	0.0483	1.57e-09	3.488	2.231	0.0422	0.0103	3.11e-06
borne inf	1.9786	0.0287	0	2.6789	1.8056	0	0	0
borne sup	2.1293	0.068	4.75e-09	4.297	2.6563	0.283	0.0276	9.53e-06

Table 1: Odds-ratio de la régression logistique

Si on prend l'exemple de l'aluminium, on constate un odds-ratio moyen de 2.05. Autrement dit, quand la variable aluminium augmente d'une unité, on a 2.05 fois plus de chance de considérer une eau potable qu'une eau non potable. À partir de ses odd-ratio on peut donc déterminer quelles variables on le plus d'influence sur notre classification en regardant celles dont l'odd-ratio est très différent de 1 : "chromium", "bacteria" et "aluminium" qui augmentent les chance de considérer une eau potable et "cadmium", "uranium", "selenium", "mercury" et "arsenic"

qui augmentent celles de considérer une eau non potable.

3.2 Forêt aléatoire

Le deuxième algorithme que nous utilisons est celui de la forêt aléatoire. Cette méthode n'est ni sensible aux données extrêmes ni à la multicolinéarité et comme la classification logistique elle présente l'avantage d'être facilement interprétable. Contrairement à la première méthode elle requière des paramètre exogènes a optimiser comme le nombre d'arbres qui constitueront la forêt ainsi que la profondeur maximum de chaque arbre (une profondeur trop élevée engendre du sur-apprentissage).

On effectue notre classification avec une forêt de 200 arbres (pas à optimiser, plus il y a d'arbres mieux c'est) de profondeur maximum égale à 23 arbres (paramètre obtenu par validation croisée grâce à python).

3.2.1 Prévisions

On donne les prévisions d'un algorithme présent dans l'intervalle de confiance (évalué pour les mêmes échantillons que celui de la régression logistique). Après une simulation de notre algorithme on obtient un taux d'erreur de 4.85%. On affiche le graphe de la matrice de prédiction dans la figure 5.

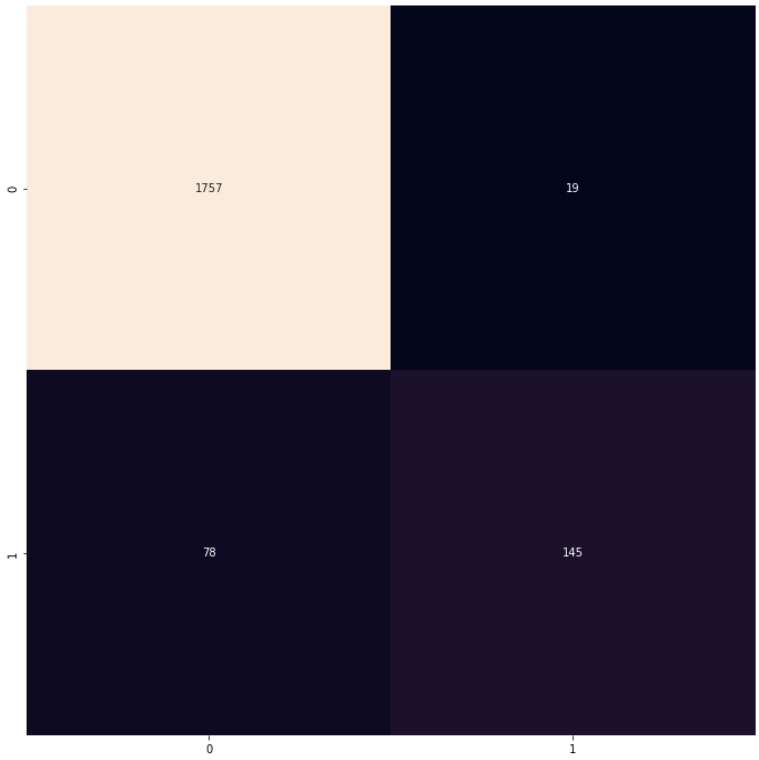


Figure 5: Matrice de confusion algorithme forêt

On constate donc la présente ce 19 eaux non potables classées potables et 78 eaux potables classées non potables. Ainsi, la forêt aléatoire est plus adapté à nos données que la fonction logistique

3.2.2 Stabilité et interprétation

Comme pour la régression logistique on effectue notre algorithme 100 fois pour différents échantillons. On obtient une moyenne de erreurs de 4.19% (IC(95%) = [3.25, 5.13]) et une faible variance de 2.30e-05.

Pour savoir quelles variables on le plus comptée dans l'algorithme on représente un arbre de classification évalué avec les même paramètre. La figure 6 nous montre le schéma de séparation d'un arbre.

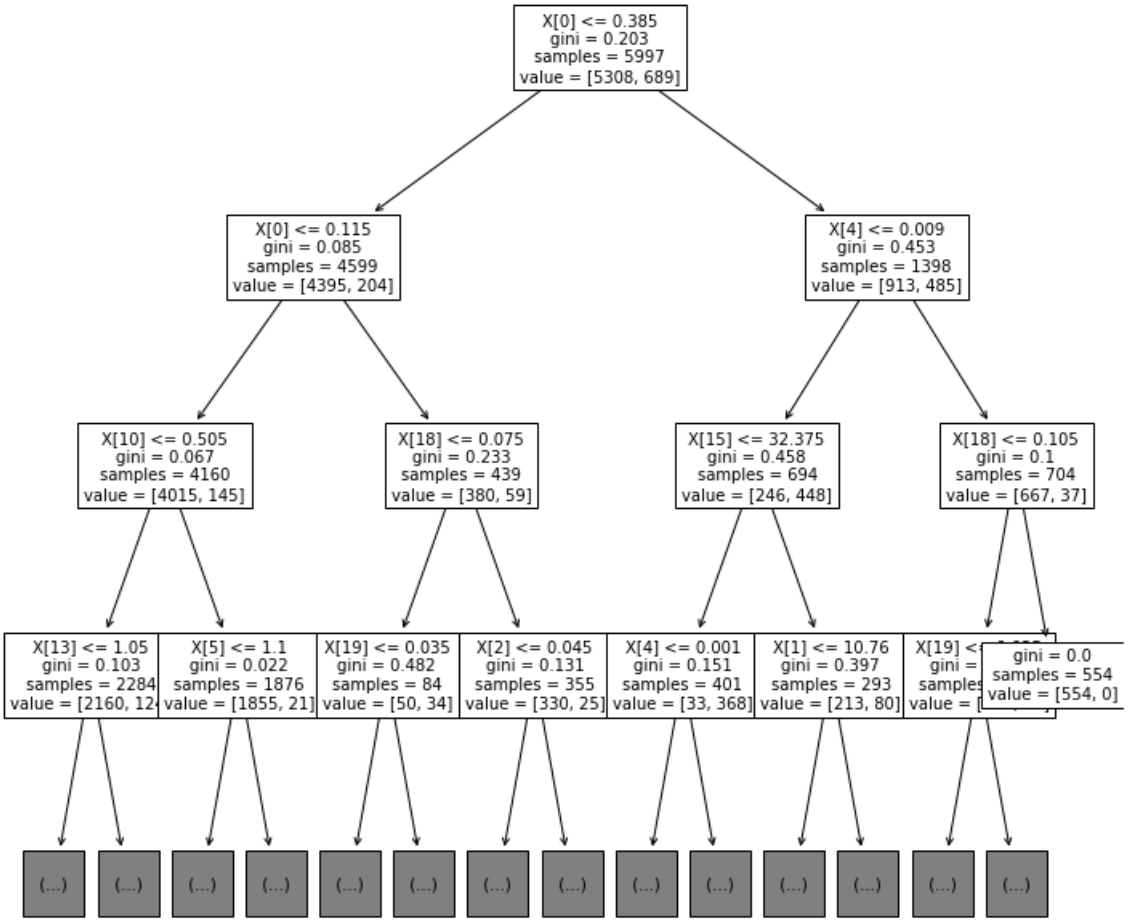


Figure 6: Un arbre de décision

Les variables qui ressortent en premier sont celles qui discriminent le mieux nos individus. Si on s'en réfère aux trois premières divisions, les variables les plus importantes sont les variables "aluminium", "cadmium", "viruses", "silver" et "perchlorate". On remarque que les variables "aluminium" et "cadmium" étaient déjà celle qui expliquaient le mieux notre classification pour notre premier modèle.

3.3 Support Vecteur Machine (SVM)

Le troisième algorithme que nous utilisons est le SVM, une méthode consistant à chercher une dimension supérieure dans laquelle nos données seraient séparables par une droite linéaire. Comme la forêt aléatoire, elle est insensible aux données extrêmes et à la multicolinéarité. Cependant elle nécessite l'apport et donc le choix de nombreux paramètres exogènes. De plus elle requière de nombreuses données d'apprentissage pour bien apprendre et n'est pas interprétable.

3.3.1 Choix des paramètres

Pour réaliser notre classification on choisit d'optimiser trois paramètres : le type de noyau de probabilité (gaussien, linéaire, etc.), son coefficient γ et un paramètre de régularisation C. Il n'est pas faisable de tester toutes les valeurs possibles à cause du manque de puissance de calcul mais après plusieurs essais on choisit un noyau gaussien de coefficient $\gamma = 0.01$ et une constante de régularisation $C = 90$.

Les paramètres sont très sensibles au tirage des échantillons d'entraînement et de test. On ne peut pas se permettre de réaliser 100 simulations et de les optimiser pour calculer un intervalle de confiance des résultats. On ne réaliser donc qu'une seule simulation de l'algorithme.

Par expérience on sait qu'il vaut mieux appliquer la méthode des SVM aux données standardisées. On choisit ici de considérer la moyenne et l'écart type de chaque variable comme un paramètre de l'algorithme : c'est à dire à dire que la standaridisation des données *train* sera appliquée aux données *test*. Notre motivation est que comme les données *train* sont plus nombreuses, elle offrent une meilleure estimation de la moyenne et de la variance.

3.3.2 Prévisions

Après simulation on obtient une erreur de classification de 5.05%. La matrice de confusion obtenue est présentée dans la graphe 7.

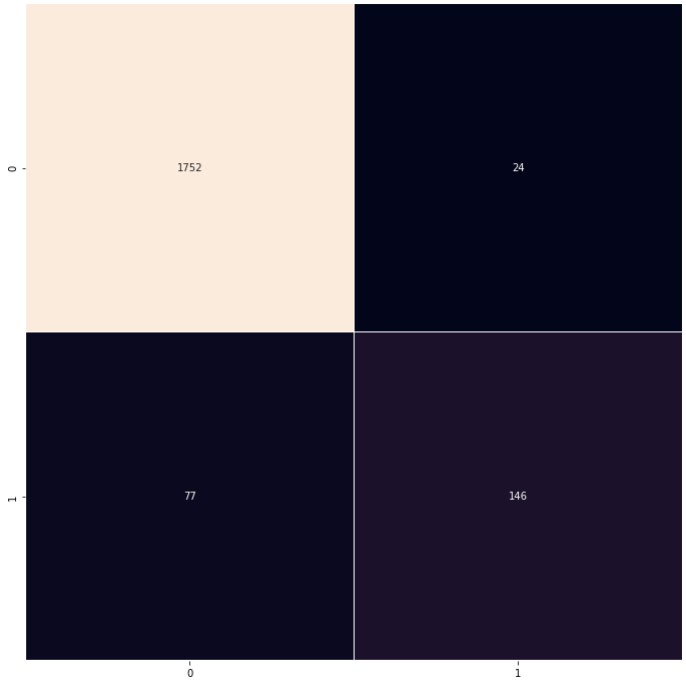


Figure 7: Matrice de prédiction SVM

On obtient donc 24 faux positifs et 77 faux négatifs soit un meilleur résultat que pour la foncion logistique mais moins bien que pour la forêt aléatoire.

3.3.3 Courbe ROC, comparaison des méthodes

On trace les courbes ROC (Receiver Operating Characteristic) de chaque algorithme elle représente le taux de vrais positifs (TVP) en fonction du taux de vrais négatifs (TVN). Le TVP est le nombre de vrais positifs sur le nombre total de positifs (vrais et faux) et respectivement pour le TVN. L'AUC (Area Under the Curve) est l'aire sous la courbe ROC et correspond donc à la moyenne du taux de vrais positifs et du taux de vrais négatifs (AUC = (TVP + TVN)/2). On trace les courbes ROC de chaque algorithme dans la figure 8.

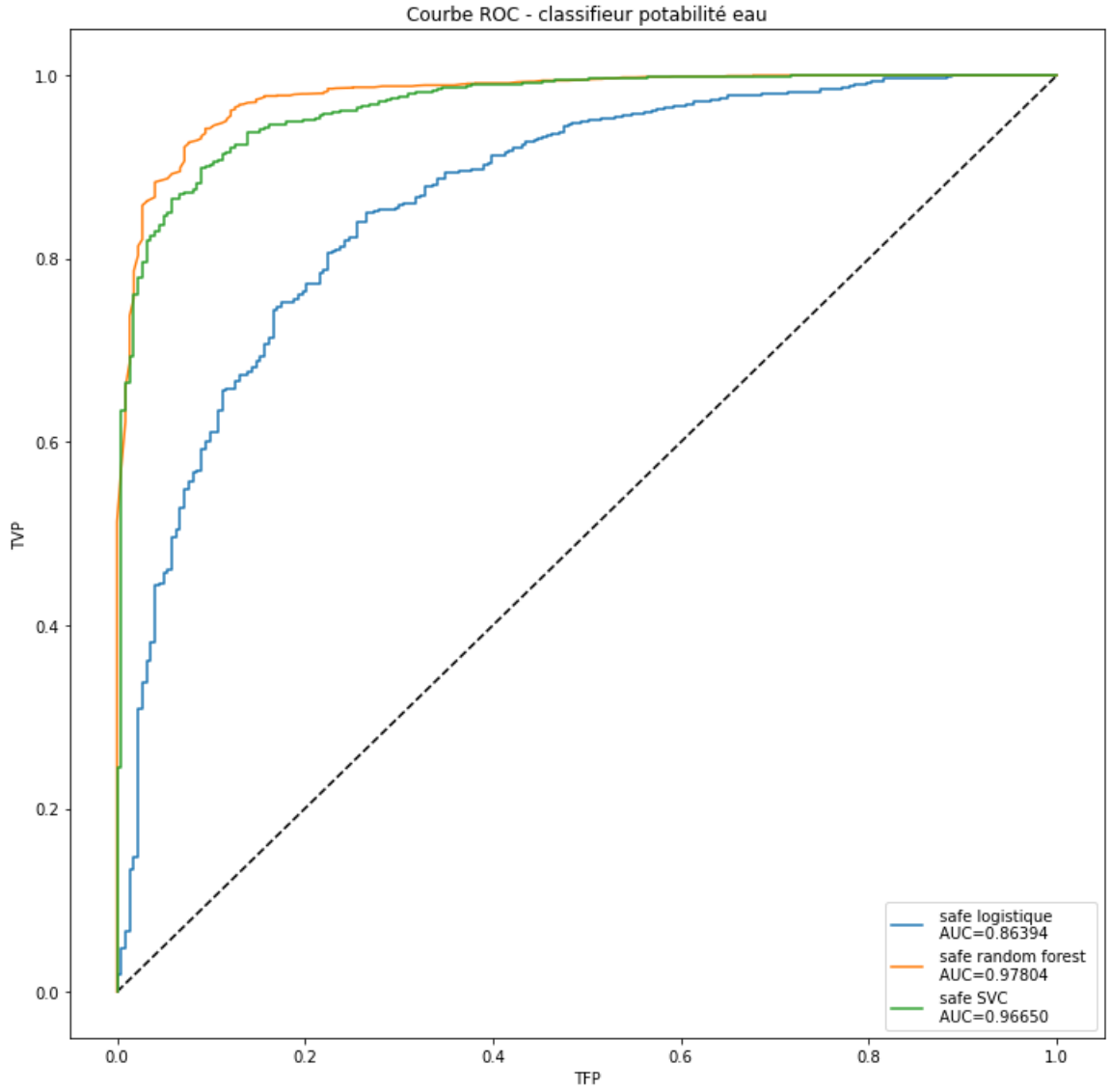


Figure 8: Coubes ROC des trois algorithmes

Les résultats sont évidemment en accord avec ce à quoi on s'attendait : d'après le critère de l'AUC le meilleur algorithme est la forêt aléatoire (AUC = 0.98) suivie de la méthode des SVM (AUC = 0.97) et de la régression logistique (AUC = 0.86).

4 Super-learner

Dans cette partie on de créer un super learner, un algorithme qui se servira des trois algorithmes que l'on a obtenu pour améliorer les prédictions. Il consiste simplement en une régression logistique sur les probabilités d'obtenir la classe 1 pour chaque algorithme. On obtient une erreur de 4.45% soit une meilleure erreur que celle de la forêt aléatoire, la plus faible jusqu'à présent (4.85%).

Grâce aux odds-ratio on peut se rendre compte de quel algorithme prend la plus grande place dans la classification. On présente les odds-ratio et la matrice de confusion dans la Figure2.

Variable	Proba Logit	Proba Forêt	Proba SVM
Odd-ratio	1.13	1.2e-05	0.11

Table 2: Odds-ratio Super-learner

Les odds-ratio s'interprète exactement de la même manière que pour la première régression logistique. Nous voyons sans surprise que c'est la forêt qui a le plus contribué à la classification suivie du SVM. La fonction logistique n'a quasiment pas contribué. La classe référence étant la classe "potable" on peut facilement lire l'odd ratio : si la probabilité de la variable forêt augmente d'une unité alors on a 1.2e-05 fois plus de chance de considérer une eau non potable ce qui est assez logique puisque la variable forêt concerne les probabilité d'obtenir 1. On représente les quatre courbes ROC dans la figure 9.

Ainsi, grâce au super learner on a réussit assez simplement à améliorer l'AUC de la forêt aléatoire en passant de 0.978 à 0.979. On remarque tout de même que l'amélioration n'est pas très élevée.

5 Conclusion

Au cours de ce travail on a créé un classifieur permettant de distinguer une eau potable d'une eau non potable.

Nous avons commencé notre investigation par une analyser statistique qui nous a permis de mettre en évidence l'absence de problèmes pouvant engendrer une instabilité des algorithmes. Une ACP en fin d'analyse nous a permis de mettre en évidence que les variables discriminait bien les labels.

Par la suite nous avons tester trois algorithmes : une classification logistique offrant une taux d'erreur moyen de 9.40% dans un intervalle de confiance à 95% de [8.26% ; 10.48 %], une forêt aléatoire offrant un taux d'erreur moyen de 4.19% dans un intervalle de confiance de [3.25% ; 5.13 %] et enfin, un SVM engendrant un taux d'erreur de 5.05%. Grâce à ses résultat et aux courbes ROC associés nous avons pu nous rendre compte de la supériorité de la forêt aléatoire sur les deux autres algorithmes.

Enfin, nous avons mis au point un algorithme super learner permettant de combiner les résultat obtenus par les trois algorithme. Grâce a cet algorithme on parvient à construire un classifieur offrant un taux d'erreur meilleurs qui les trois algorithme qui on permis de le construire.

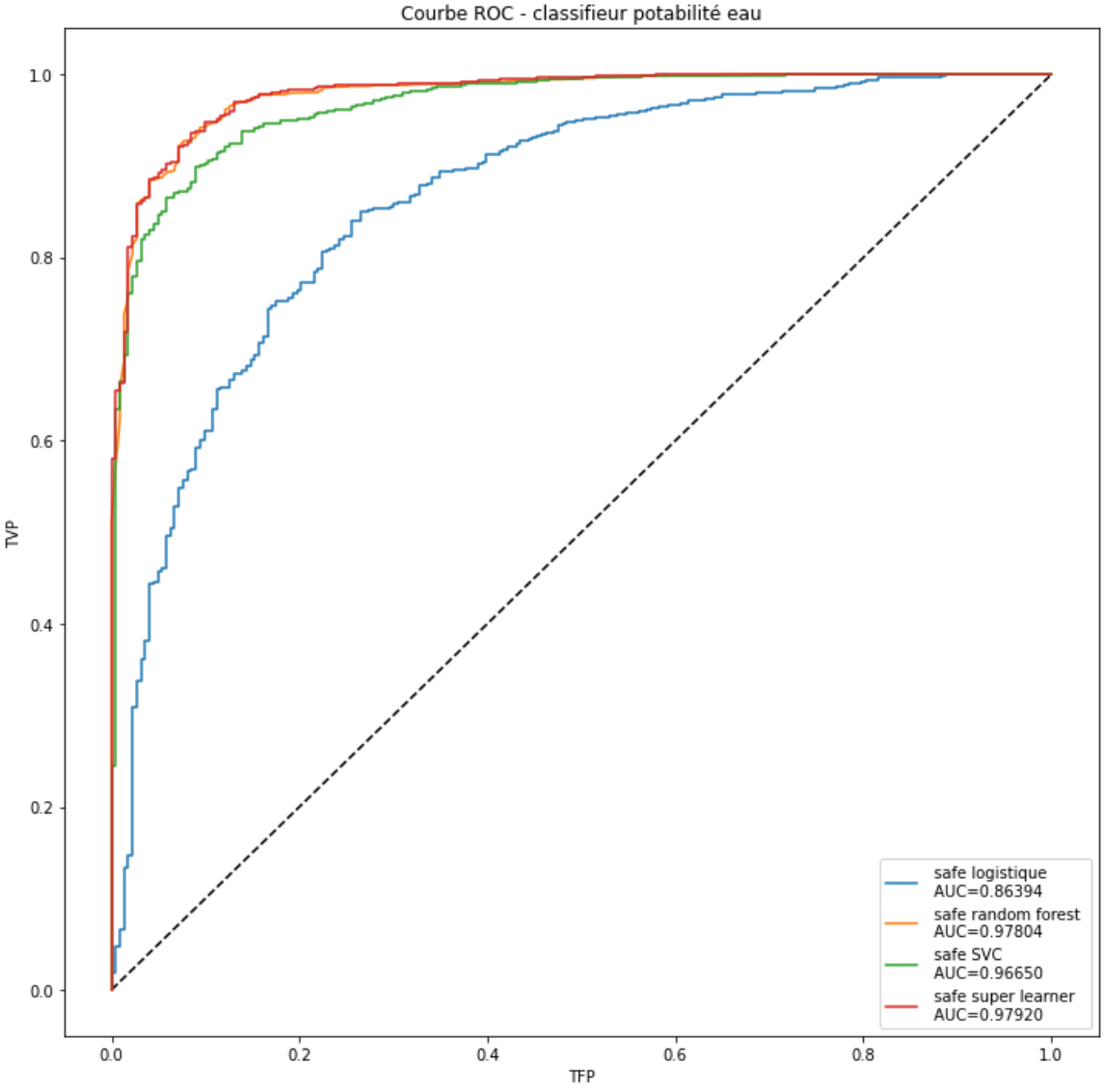


Figure 9: Coube ROC Super-learner