Water Quality

François-Xavier Arthaut, Ayoub Bhija
Achraf Hanini, Alexandre Em

August 2, 2021

1 Introduction

Access to safe drinking-water is essential to health, a basic human right and a component of effective policy for health protection. This is important as a health and development issue at a national, regional and local level. In some regions, it has been shown that investments in water supply and sanitation can yield a net economic benefit, since the reductions in adverse health effects and health care costs outweigh the costs of undertaking the interventions.

Problématique: Prédire si l'eau est potable ou non pour sa consommation.

2 Exploration

Nous allons travailler sur un jeu de données contenant **3276 valeurs**, dont 1998 sont classées *non potables* et 1278 classées *potable*

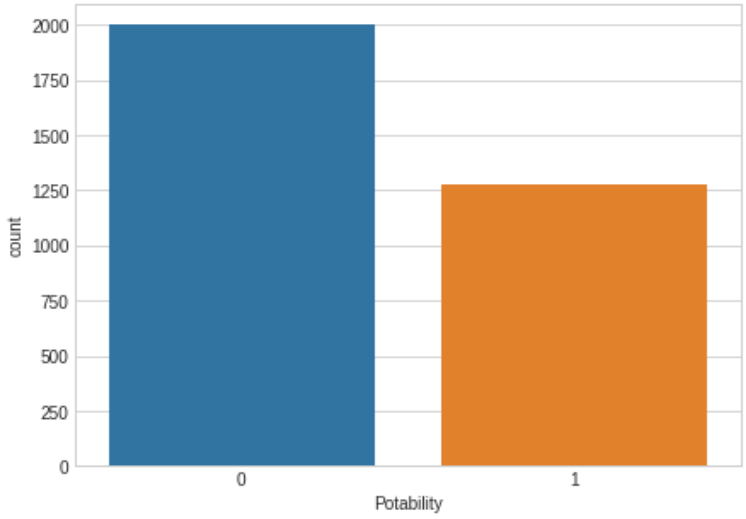


Figure 1: Nombre total d'eau potable(1) et non potable(0)

Ce jeu de données comporte 10 colonnes:

- ph: *flottant* - le pH mesure l'acidité ou la basicité d'une solution (0 to 14)
- Hardness: *flottant* - Montant de calcium et de Magnesium dans l'eau en mg/L
- Solids: *flottant* - le total de solide dissous dans l'eau en ppm
- Chloramines: *flottant* - montant de Chloramines en ppm
- Sulfate: *flottant* - montant de sulfate dissous en ppm
- Conductivity: *flottant* - la conductivité de l'électricité de l'eau en $\mu S/cm$
- Organic_carbon: *flottant* - montant de Carbone organique en ppm
- Trihalomethanes: *flottant* - montant de Trihalomethanes en $\mu g/L$
- Turbidity: *flottant* - Mesure de l'opacité de l'eau en NTU
- Potability: *entier* - Indique si l'eau est potable ou non

On peut voir que l'on a dans notre dataframe, que des variables quantitatives et que notre cible est un Integer dont les valeurs sont compris entre $[0, 1]$ et peut être considérer comme des booleens. Nous n'aurons donc pas besoin de l'encoder.
On peut aussi voir dans la figure 2, qu'il n'y a pas beaucoup de différence entre les courbes des eaux potables et non potables de chaque variable, sauf pour le **ph** et **Sulfate**. On peut donc supposer que **ph** et **Sulfate** sont peut être des variables qui influence sur la potabilité de l'eau ou non.

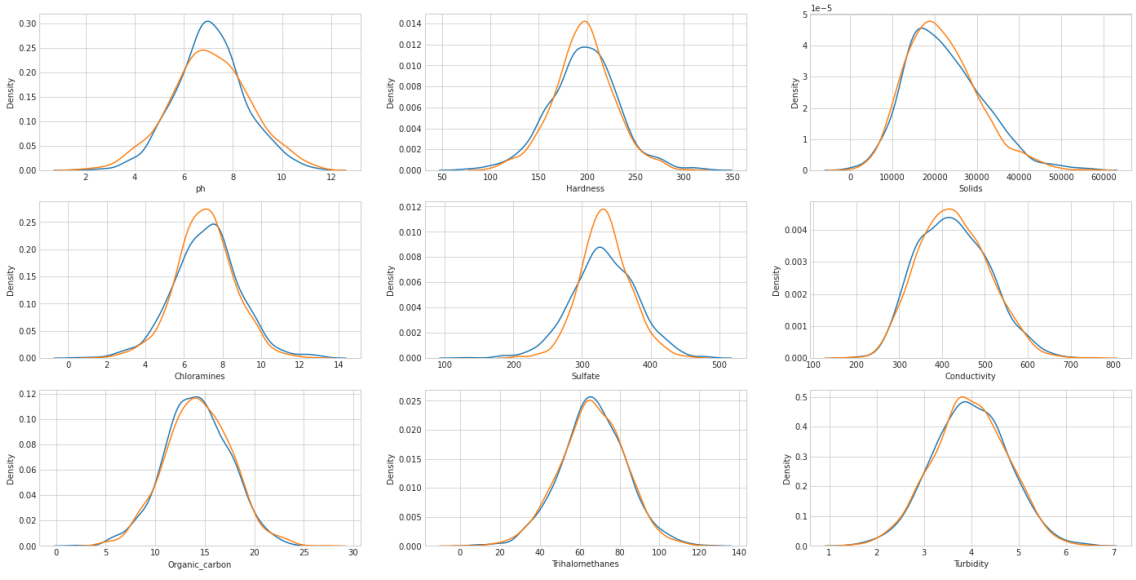


Figure 2: Répartition des eaux potable et non potable

2.1 Quels problèmes avez-vous rencontré avec les données ?

2.1.1 Valeurs nulles

En executant la commande `data.isnull().sum()`, nous avons pu voir que nous avions 3 variables qui contiennent des valeurs nulles (NaN):

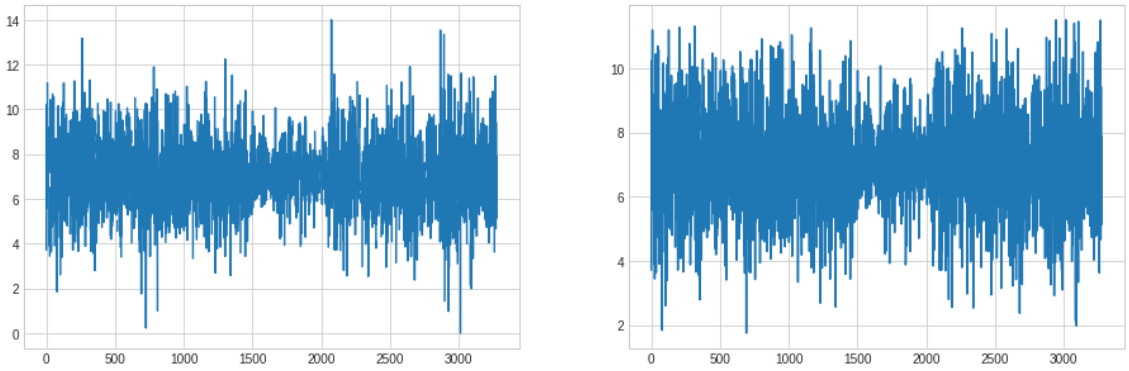
Table 1: Variables contenant des valeurs nulles (NaN)

Variable	**Occurrence de valeurs nulles**
ph	491
Sulfate	781
Trihalomethanes	162

Pour un total de 1434 lignes, soit environ 44% des données. Ce qui est assez conséquent si on supprime toutes ces lignes. Nous avons donc fait plusieurs cas d'essais incluant la suppression des lignes contenant les valeurs nulles. Nous avons donc remplis ces valeurs nulles par interpolation (cas 1) et par la moyenne de chaque variable (cas 2). Nous pourrons donc comparer ces 3 cas pour ne prendre uniquement celui qui obtient le meilleur accuracy.

2.1.2 Valeurs aberrantes

Nous avons ensuite analyser la répartition des valeurs pour chaque variable afin d'enlever toutes les valeurs aberrantes, comme nous pouvons le voir dans la figure 3a, on sait que le ph d'un liquide peut s'étendre de 0 à 14, mais nous recherchons le ph d'une eau potable et nous considérons que le ph de l'eau ne peut atteindre des valeurs extrémités, donc nous les avons supprimés (figure 3b).



(a) Répartition de base (b) Répartition après suppression

Figure 3: Répartition des valeurs du ph

2.1.3 Incohérence avec les recommandations

Nous avons analyser et comparer pour chaque variable, toutes les valeurs avec les recommandations données dans l'énonce et il y a beaucoup d'incohérence entre eux.

Par exemple le Sulfate a une accuracy de $\sim$ 40%, mais il n'y a aucune eau non potable qui rempli les recommandations qui sont d'être compris dans $[3, 1000]$ mg/L.

La recommandation la plus proche de la réalité est la Conductivité de l'eau avec une accuracy de $\sim$ 50% comme on peut le voir dans la figure 4

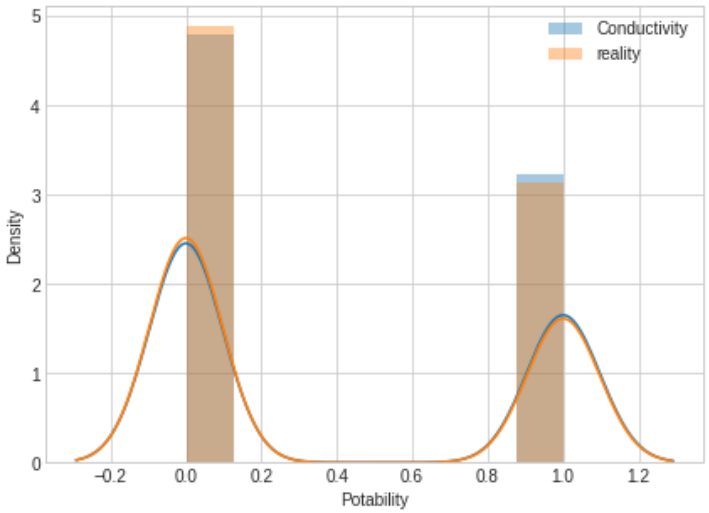


Figure 4: Répartition des eaux potable et non potable

Nous avons ensuite comparé chaque variable avec la cible pour voir si il y avait une corrélation entre elles (figure 5) et nous avons remarqué que la correlation entre chaque variable et la cible est très faible, c'est à dire que chaque valeur s'approche de 0 et qu'aucune valeur à l'absolue n'est supérieure à 0.5 et on peut donc emettre l'hypothèse qu'ils sont independent et qu'il y a aucune relation lineaire entre elles.

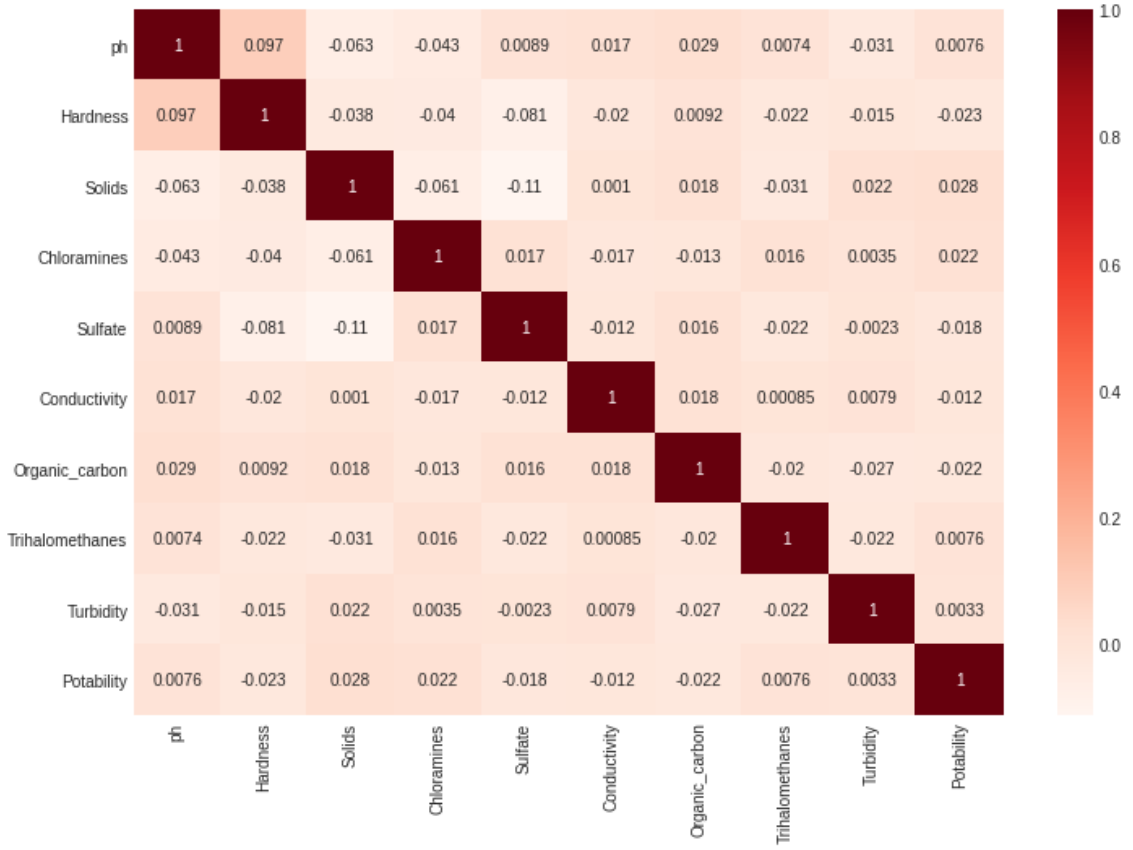


Figure 5: Matrice de corrélation entre variable

Afin de comparer au résultat final, nous avons donc fait plusieurs essai sur tout les cas de traitement (suppression des valeurs nulles, interpolation, moyennes) et sur différents modèles.

3 Quel(s) modèle(s) avez-vous choisi ?

Le but de notre projet est de prédire si une eau est potable ou non et nous voulons donc une classification de l'eau.
Après avoir réaliser un traitement de nos données, nous avons étudié la matrice de corrélation pour ne sélectionner que les variables dont la corrélation est pertinente et ensuite appliquer le dataset sur un modèle de régression logistique.

3.1 Regression logistique

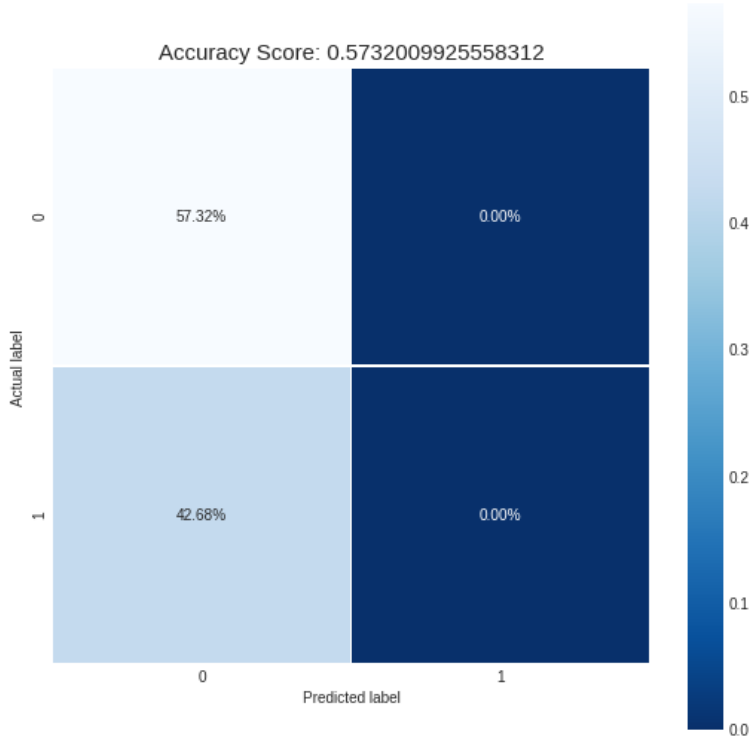
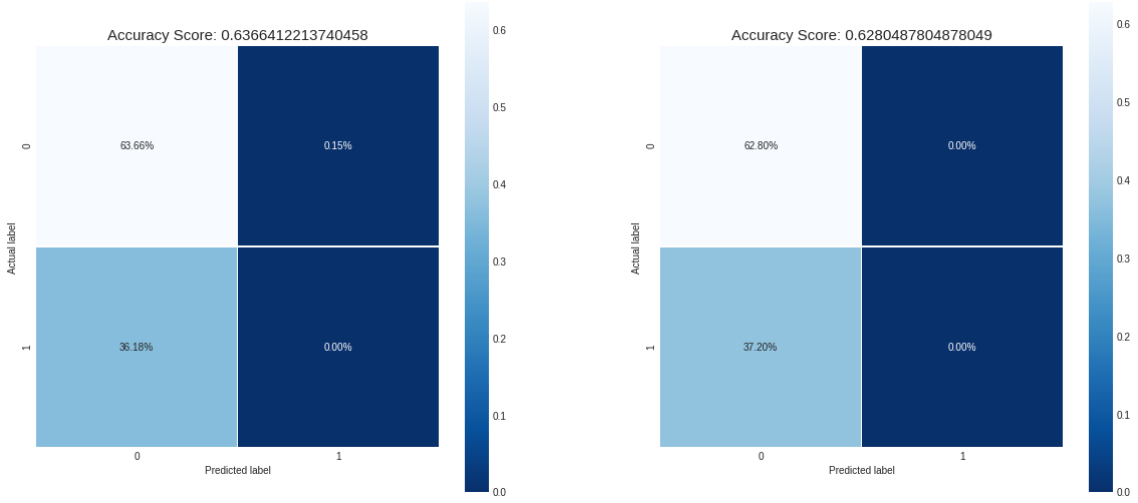


Figure 6: Matrice de confusion (Pearson) dans le cas suppression des NaN

On peut remarquer que ce modèle n'est pas performant, en effet on a une accuracy de $\sim 57\%$ (puis $\sim 62\%$ après remplissage des valeurs nulles), mais surtout en observant la matrice de confusion, aucun positif (eau potable) n'a pas été prédit correctement. On remarque la même chose après traitement des données nulles (figure 7)



(a) Cas de remplissage par interpolation des données nulles

(b) Cas de remplissage par moyenne des données nulles

Figure 7: Matrices de confusion (Pearson)

3.2 Decision tree

Afin d'avoir une meilleure visualisation, nous allons utiliser un arbre de decision et calibrer les hyperparamètres afin d'avoir un accuracy optimal sans tombé dans le cas de sur-entrainement ni de sous-entrainement, pour cela nous allons tout d'abord limiter la hauteur et le nombre de feuille de notre arbre de decision.

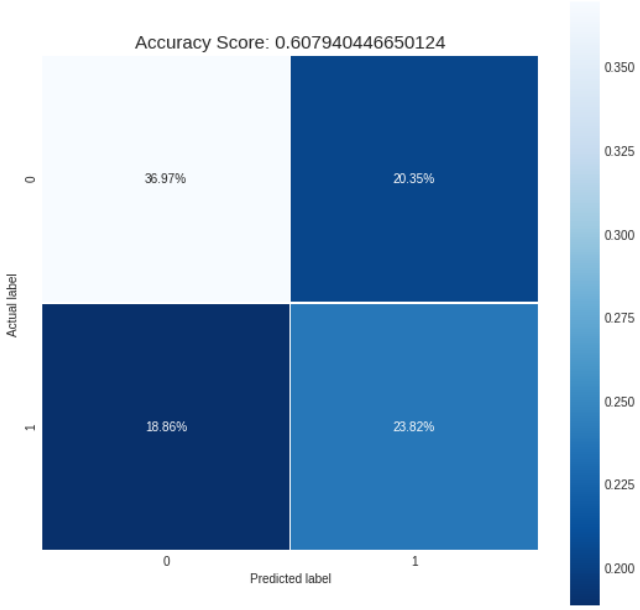
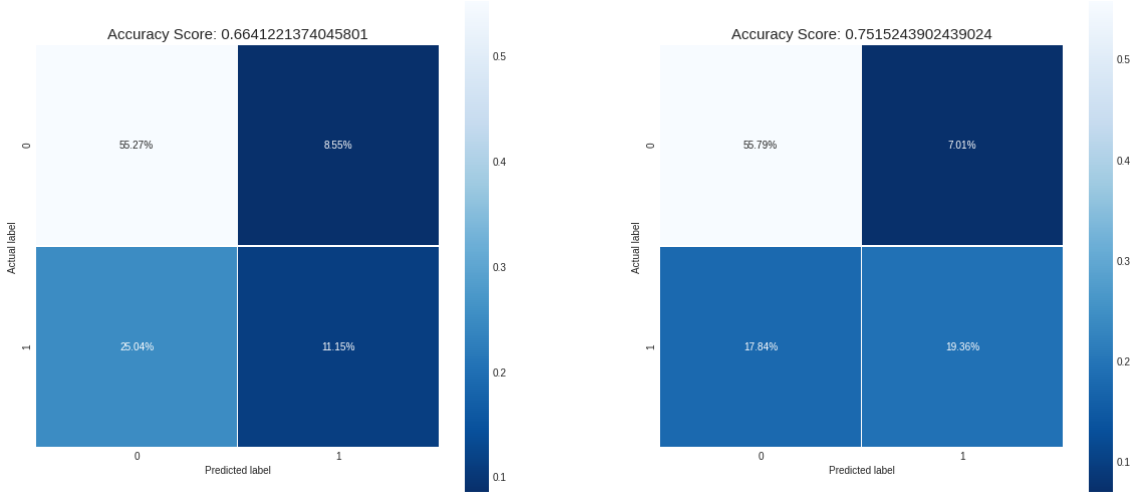


Figure 8: Nombre total d'eau potable(1) et non potable(0)

On peut remarquer une amélioration par rapport au modèle de régression logistique. Notamment après traitement des données on peut remarquer une nette amélioration allant jusqu'à $\sim 75\%$ d'accuracy avec $\sim 19\%$ de vrai positif (f. 9).



(a) Cas de remplissage par interpolation des données nulles

(b) Cas de remplissage par moyenne des données nulles

Figure 9: Matrices de confusion (Pearson)

Nous allons ensuite essayer d'améliorer l'accuracy compte tenu de la problématique, en essayant avec un autre modèle la forêt aléatoire.

3.3 Random tree

Ayant des variables indépendantes et que ses valeurs sont éparpillés, nous allons ensuite utiliser le modèle Random forest.

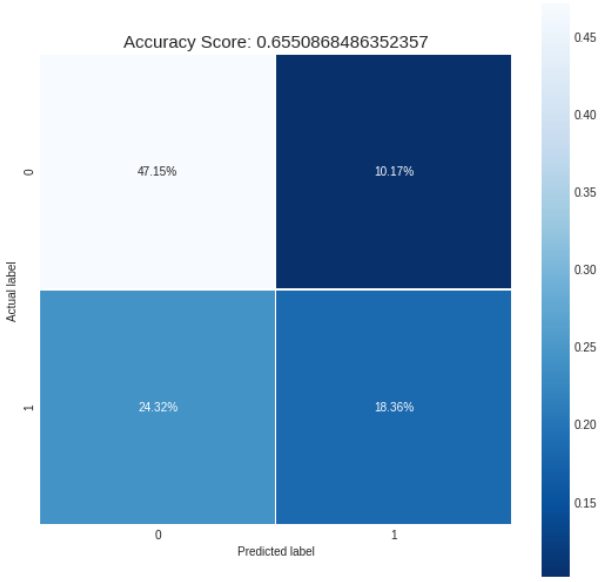
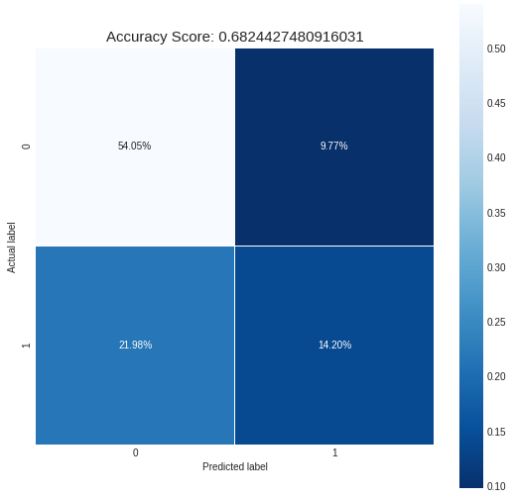
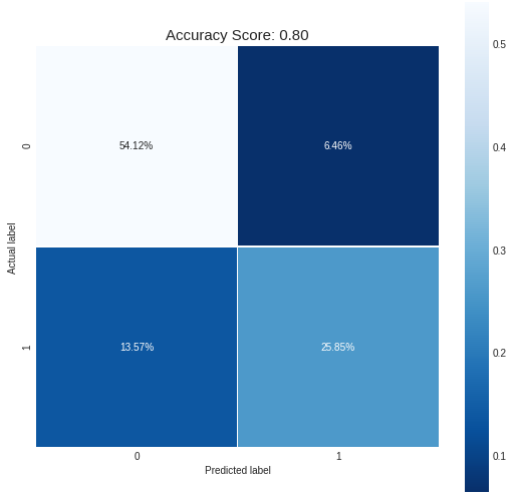


Figure 10: Nombre total d'eau potable(1) et non potable(0)

Encore une fois on peut remarquer une nette amélioration par rapport aux deux autres modèle. Notamment après traitement des données on peut remarquer une nette amélioration allant jusqu'à $\sim 80\%$ d'accuracy avec $\sim 26\%$ de vrai positif.



(a) Cas de remplissage par interpolation des données nulles



(b) Cas de remplissage par moyenne des données nulles

Figure 11: Matrices de confusion (Pearson)

A partir de ce modèle nous avons aussi tracé un histogramme des variables importantes (figure 12) et on comme prévu, d'après notre hypothèse énoncé au début de ce rapport. On peut voir que le Sulfate et le ph sont des variables qui sont bien importantes.

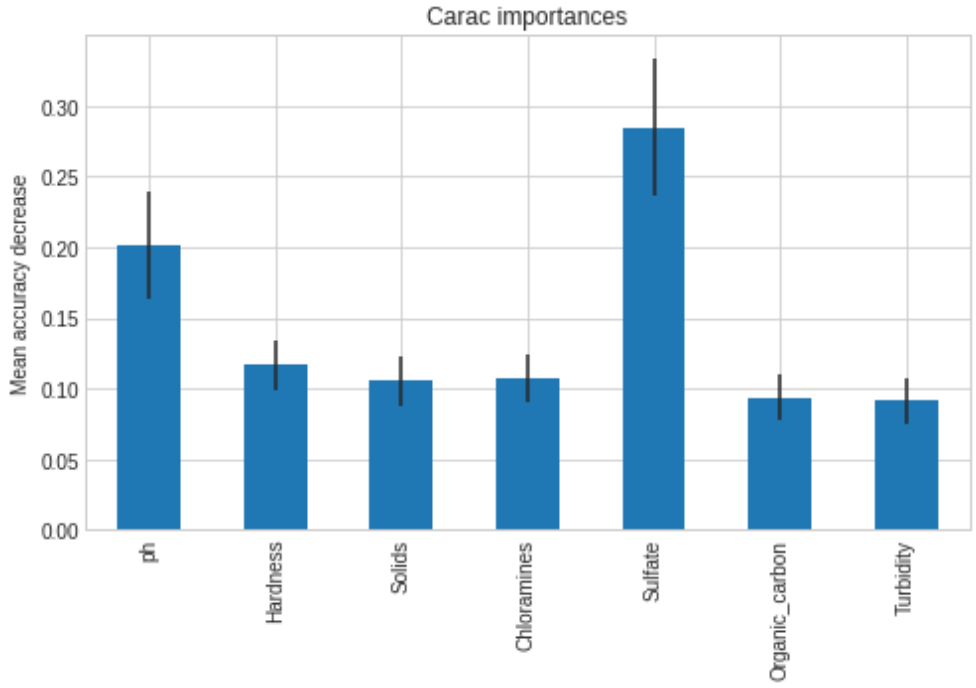


Figure 12: Histogramme des variables selon leur importance

Nous avons aussi comparé ces 3 modèles par une courbe de caractéristique de performance et de taux d'erreurs des faux positifs avec les faux négatifs.

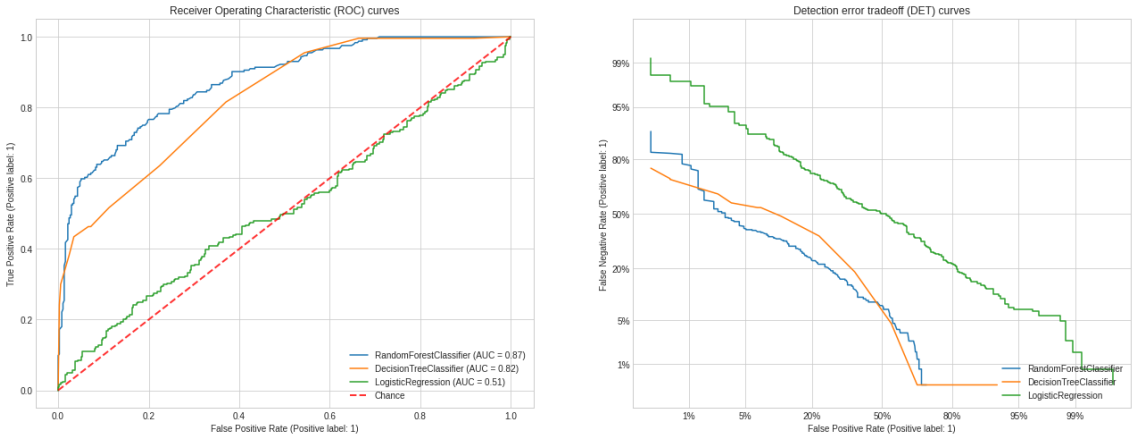


Figure 13: Histogramme des variables selon leur importance

On peut voir dans les graphes ci dessus que l'algorithme de Random Forest est bien le plus précis par l'allure des courbes et que la courbe de DecisionTreeClassifier et de RandomForestClassifier se rapprochent et ont presque la même allure dans les deux courbes

4 Conclusion

En comparant différent modèle et en traitant les données de façon différente, on est passé d'une accuracy de $\sim 60\%$ à une accuracy de $\sim 80\%$ avec un taux de vrai positif de $\sim 26\%$ grâce au model *RandomForestClassifier*. Ce qui n'est pas mal étant donné le jeu de donnée que l'on a mais insuffisant étant donné la problématique qui est un enjeu majeur pour l'homme. En effet, si une personne boit de l'eau considéré comme potable mais qui ne l'est en réalité pas, il pourrait développer des maladies plus ou moins grave. Pour toute les eaux considérées non potables alors qu'elles le sont, on priverait un village, une ville voir un pays de boire de l'eau potable.
On pourrait peut être améliorer l'accuracy en traitant les valeurs nulles et les valeurs aberrantes (qui ont été supprimé au cours de ce projet) plus efficacement, par exemple en comparant la ligne contenant la valeur nulle avec les autres lignes et l'affecter à la valeur de la ligne qui possède le maximum de similitude selon un degrès choisi.

References

[1] Water quality dataset

[2] Manipulation de données avec Pandas - MonCoachData

[3] Confusion matrix explained - Dhilip Subramanian

[4] How to tune a decision tree - Mukesh Mithrakumar