CentraleSupélec

Apprentissage Automatique

Prédiction de la potabilité de l'eau

Étudiants:
Benoit Sioc'han de Kersabiec, benoit.dekersabiec@student-cs.fr
Babacar Toure, babacar.toure@student-cs.fr
Nathan Rougier, nathan.rougier@student-cs.fr

Sommaire

1 Introduction

Le dataset contient des métriques sur la qualité de l'eau dans 3276 points d'eau. Pour chaque point de donnée, 10 caractéristiques différentes sont données:

- **pH** : le caractère acide ou basique de l'eau
- **Hardness** : la quantité de sels mineraux (calcium et magnesium) présents dans l'eau
- **Solids** : la minéralisation de l'eau
- **Chloramines** : la concentration de chlorine et de chloramine dans l'eau
- **Sulfate** : la concentration de sulfate dans l'eau
- **Conductivity** : le caractère conducteur de l'eau
- **Organic Carbon** : le nombre de composés organiques dans l'eau
- **Trihalomethanes** : le nombre de ppm de trihalométhane dans l'eau
- **Turbidity** : la propriété de l'eau à absorber la lumière
- **Potability** : indique si l'eau est potable ou non

Le but de ce projet est de trouver et d'entraîner le meilleur algorithme de machine leaening capable de déterminer à partir des 9 premières caractéristiques si une eau est potable ou non.

2 Analyse exploratoire des données

Par un apercu rapide du jeu de données, on remarque qu'il manque certaines valeurs. Dans un premier temps, nous allons étudier ces valeurs manquantes

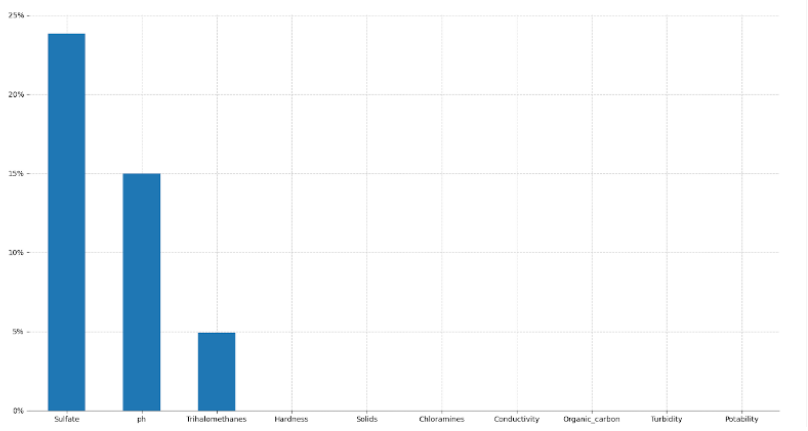

Figure 1: Pourcentages des valeurs manquantes par caractéristique

Comme le montre le graphique ci-dessus, il y a des valeurs manquantes dans trois différentes caractéristiques (Sulfate, ph et Trihalométhane). Ces données manquantes ne sont pas négligeables. Par exemple, 24% des données sur le sulfate sont absentes

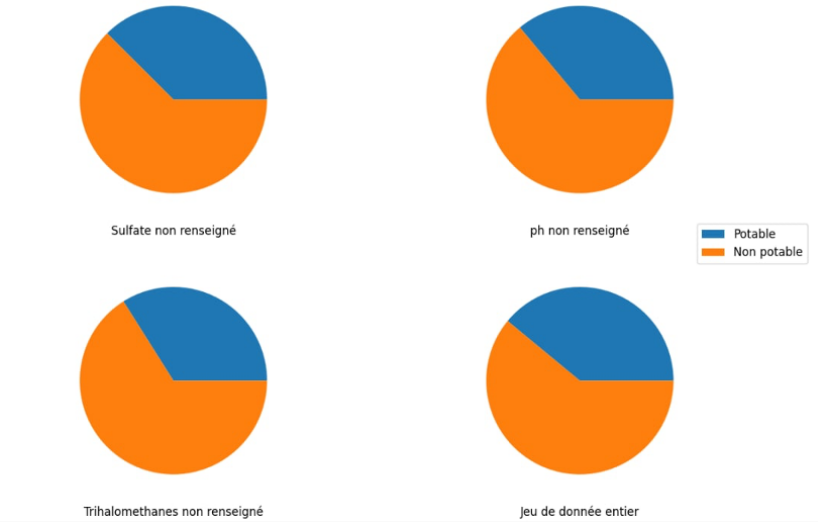

Figure 2: Analyse du type des données manquantes

D'après les diagrammes circulaires ci-dessus, 40% des points de données sont potables. Ce ratio est conservé dans les points avec des données manquantes. Les légères fluctuations peuvent être expliquées par la taille relativement petite de l'ensemble de point utilisé pour construire les graphiques. Ainsi, on en conclut que la probabilité d'avoir une caractéristique manquante dans un point de donné est indépendant de la probabilité que ce point de donné est potable.
De plus, on observe sur le graphique ci-dessus que lorsqu'une valeur est manquante pour une caractéristique, la probabilité d'avoir une donnée manquante pour une autre caractéristique est sensiblement similaire aux ratios de données manquantes par caractéristiques montrés plus haut.

Pour conclure, les données manquantes dans ce jeu de données semble être de type *Missing Completely at random (MCAR)*

3 Prétraitement des données

Le prétraitement de données est la partie délicate d'un projet de machine learning. C'est une tâche très chronophage mais qu'il faut réaliser avec soin si l'on souhaite garantir de bonnes prédictions. Sur ce dataset, l'analyse exploratoire des données à montrer que les tâches de prétraitement nécéssaires sont la standardisation, la normalisation et la gestion des données manquantes.

3.1 Standardisation et normalisation

La standardisation est une technique de mise à l'echelle des données. Les données sont centrées et réduite ($X = \frac{X-\mu}{\sigma}$) pour avoir un écart-type de 1. Nous avons décidé d'appliquer une standardisation plutot qu'une normalisation car l'analyse des données a montré que les features suivent une distrubion gaussienne. En pratique, la standardisation correspond juste à un transformation dans nos pipelines de transformation de données.

Dans cette étude, des approches telles que les SVM, les KNN et la régression logistique sont employées; il était donc obligatoire de standardiser. Pour la régression, les données non standardisées créent des différences de vitesse de convergence du gradient (qui n'est plus lisse). Sur les SVM et les

KNN, l'échelle crée un déséquilibre dans la distance euclidienne car certains features pèseront plus. Par contre les méthodes basées sur les arbres sont insensibles à l'échelle.

3.2 Gestion des données manquantes

Pour la gestion des données manquantes, plusieurs techniques ont été abordées. La toute première et la plus simple a été de supprimer toutes les données du dataset avec des valeurs manquantes et nous avons obtenu des résultats convenables. Nous avons ensuite appliqué la moyenne et la médiane avec et sans classification et avec et sans bruit.

Nous avons pu ainsi comparer pour nos différents modèles les résultats selon la technique employée afin de déterminer la meilleur gestion de ces données manquantes.

4 Sélection des modèles

4.1 Régression logistique

La regression logistique est la version classification de la regression linéaire. Elle permet donc de trouver les relations entre la variable à prédire y et nos différents prédicteurs X_i. Le passage de la regression au discret se fait grace à la fonction sigmoide qui ramène la sortie dans l'intervalle [0,1]. Ceci va correspondre à la probabilité d'appartenance à une classe. La fonction sigmoide s'ecrit $\sigma(x) = \frac{1}{1+e^{-x}}$.

Ainsi la "fonction" de la régression logistique est: $\forall X \in R^n, h(X) = \sigma(\theta X)$ avec σ la fonction sigmoide. Résultats obtenues avec cette méthode:

	precision	recall	f1-score	Support
Non potable	1	0.62	0.76	816
Potable	0.01	1.00	0.02	3
accuracy			0.62	819

Cet algorithme n'est pas du tout adapté à notre problème car nos données ne sont pas linéairement séparables.

4.2 Support vector machines

Les machines SVM (Machine à vecteurs de support) permettent de traiter des problemes de discrimination et de régression et sont donc adaptés dans notre problème de classification de potabilité.

Plusieurs valeurs du paramètre de régularisation **C** ont été testés avec les deux kernels suivant : "**RBF** (Radial basis function)" et "**linéaire**". Nous avons obtenue comme meilleure résultat avec les hyperparamètres $C = 3$ et $kernel = RBF$ une précision de 68%, ce qui n'est pas un résultat du tout satisfaisant. Tester d'autres approches est nécessaire.

	precision	recall	f1-score	Support
Non potable	0.89	0.68	0.77	650
Potable	0.35	0.66	0.46	169
accuracy			0.68	819

4.3 KNN

La méthode des k plus proches voisins (KNN) consiste prédire la classe (potabilité), en regardant quelle est la classe majoritaire des k données voisines les plus proches. Le seul paramètre à fixer est k, le nombre de voisins à considérer.

Le choix de K, le seul hyperparamètre a choisir a été fait par GridSearch, ce qui marche assez bien en général. Les modèles ont été choisis pour k allant de 1 à 34. Le modèle élu retourne une précision de 66

Ce résultat est dans similaire à ceux obtenus par les autres méthodes (SVM etc) cette technique n'a été employé dans ce problème que par curiosté car l'analyse par composantes principales ne montrait pas d'aggrégat dans les données ni des groupes de voisins. Les données sont en fait très similaires.Le KNN est aussi pertinent qu'un classifier qui renvoie "Non potable" tout le temps. Voir si dessous le résultat des K plus proches voisins avec K=27 (d'après GridSearch):

	precision	recall	f1-score	Support
Non potable	0.92	0.66	0.77	708
Potable	0.23	0.66	0.34	111
accuracy			0.66	819

4.4 Random Forest

L'algorithme de *Random Forest* consiste à créer un grand nombre d'arbre de décision et à faire la moyenne des résultats obtenus pour donner une prédiction.

La création individuelle d'un arbre de décision se fait à partir d'un jeu de données créé par *bootstrapping* du jeu de données original (tirage de N éléments avec remise).

Nous avons utilisé le principe de *cross validation* pour évaluer les *Random Forest*. Nous avons divisé le jeu de données en 10 parties et avons construit les arbres à partir de 9 d'entre elles pour enfin les évaluer sur la dernière. L'algorithme construit 10 fois la *Random Forest* en changeant à chaque fois la partie utilisée pour l'evaluation. Nous considérons par la suite la précision moyenne obtenue lors de ces 10 itérations.

Nous avons évalué chacune des différentes façons de remplir les données manquantes dans le jeu de données mais aussi l'impact de travailler avec les données obtenue par une *PCA* à trois dimensions et enfin l'impact de la suppression des *outliers* dans le jeu de données (les 10% des valeurs les plus extrèmes sont retirées).

Data filling method	removing outliers	PCA	Accuracy
dropna	no	no	68.17%
dropna	yes	no	67.67%
dropna	yes	yes	58.12%
mean	no	no	67.51%
mean	yes	no	65.35%
mean	yes	yes	59.09%
median	no	no	66.96%
median	yes	no	65.98%
median	yes	yes	58.78%
mean by class	no	no	79.55%
mean by class	yes	no	78.71%
mean by class	yes	yes	58.94%
median by class	no	no	79.33%
median by class	yes	no	78.55%
median by class	yes	yes	59.17%
mean by class with noise	no	no	66.85%
mean by class with noise	yes	no	65.84%
mean by class with noise	yes	yes	58.06%
median by class with noise	no	no	66.47%
median by class with noise	yes	no	65.33%
median by class with noise	yes	yes	59.01%

Figure 3: Résultat obtenus à l'aide de *Random Forest*

Pour la ligne obtenant le meilleur résultat (mean by class avec les outliers et sans la PCA) on a les détails suivant:

	precision	recall	f1-score	Support
Non potable	0.92	0.74	0.82	211
Potable	0.65	0.89	0.75	117
accuracy			0.79	328

5 Conclusion

En conclusion, le meilleur résultat obtenu a été obtenu en utilisant les *Random Forest*, en remplacant les données manquantes par la moyenne de la classe à laquelles elles appartiennent et en gardant les outliers. En utilisant cette methode, nous avons obtenu 79.55% de précision sur la prediction de la potabilité des points d'eau