CentraleSupélec

Rapport de projet

Prédiction de la potabilité de l'eau

BENAIJA Ismail, BADDOU Othmane, BAKKOURY Ayoub

1 Introduction

1.1 Contexte et motivation du projet :

L'accès à l'eau potable est essentiel à la santé, un droit humain fondamental, et une composante d'une politique efficace de protection de la santé. Il s'agit d'une question importante de santé et de développement au niveau national, régional et local. Dans certaines régions, il a été démontré que les investissements dans l'approvisionnement en eau et l'assainissement peuvent apporter un bénéfice économique net, puisque la réduction des effets néfastes sur la santé et des coûts des soins de santé est supérieure aux coûts des interventions. L'eau contaminée et le manque d'assainissement sont liés à la transmission de maladies. L'absence, l'insuffisance ou la mauvaise gestion des services d'eau et d'assainissement exposent les individus à des risques sanitaires évitables. C'est particulièrement le cas dans les établissements de soins de santé, où les patients et le personnel sont exposés à des risques supplémentaires d'infection et de maladie lorsque les services d'eau, d'assainissement et d'hygiène font défaut.

1.2 Problématique et solution retenue :

Notre objectif est donc de modéliser un algorithme de machine learning permettant au mieux de prédire si l'eau d'un cours d'eau est potable. Ainsi, cela permettrai d'investir dans un projet d'assainissement de cette eau. Ce modèle est donc un classificateur binaire qui associe à « 1 » le paramètre « potabilité » si une eau est potable, et « 0 » sinon. Pour ceci, nous nous basons sur les 9 variables numériques suivantes :

Feature	**Type**	**Recommendation OMS**
ph	float	$6.5 < \text{pH} < 8.5$
sulfate	float	Souhaitable : < 500 mg/L | Maximale : < 1000 mg/L
trihalomethanes	float	< 80 ppm
hardness	float	x
solids	float	< 4 mg/L ($= 4$ ppm)
chloramines	float	$3 <$ Sulfate < 30 mg/L
conductivity	float	< 400 µS/cm
organic carbon	float	< 2 mg/L
turbidity	float	$= 5.00$ NTU

Il existe de nombreuses façons d'évaluer la performance d'un classificateur. Une des plus répandus est l'accuracy. Bien que nous l'utilisons dans notre rapport, cette dernière ne suffit pas dans notre cas. En effet, il serait désastreux de prédire qu'une est est potable alors qu'en réalité elle ne l'est pas. Ainsi nous utiliserons aussi la précision (taux de vrais positifs) pour décider de l'algorithme à retenir. Ainsi nous avons conclu qu'un SVM (avec un paramètre de régularisation à 5, un kernel rbf et un paramètre alpha de 0.046) est le plus adéquat pour notre problématique.

2 Résolution du problème

2.1 Méthodologie :

- Analyse des données :
  - Analyse des variables (distributions et visualisation graphique)
  - Corrélation et pair-plot (Identification potentielle des variables explicatives)
- Preprocessing des données :

— Gestion des valeurs aberrantes
— Gestion des valeurs manquantes
— Mise à l'échelle des données
— PCA
— Autres méthodes expérimentales de réduction de dimensions non linéaires (TSNE,KPCA)

- Modélisation de 6 classificateurs et classement de leurs performances
- Selection du modèle final et tuning des hyperparamètres par nested cross validation
- Analyse des performances du modèle final

2.2 Analyse des données :

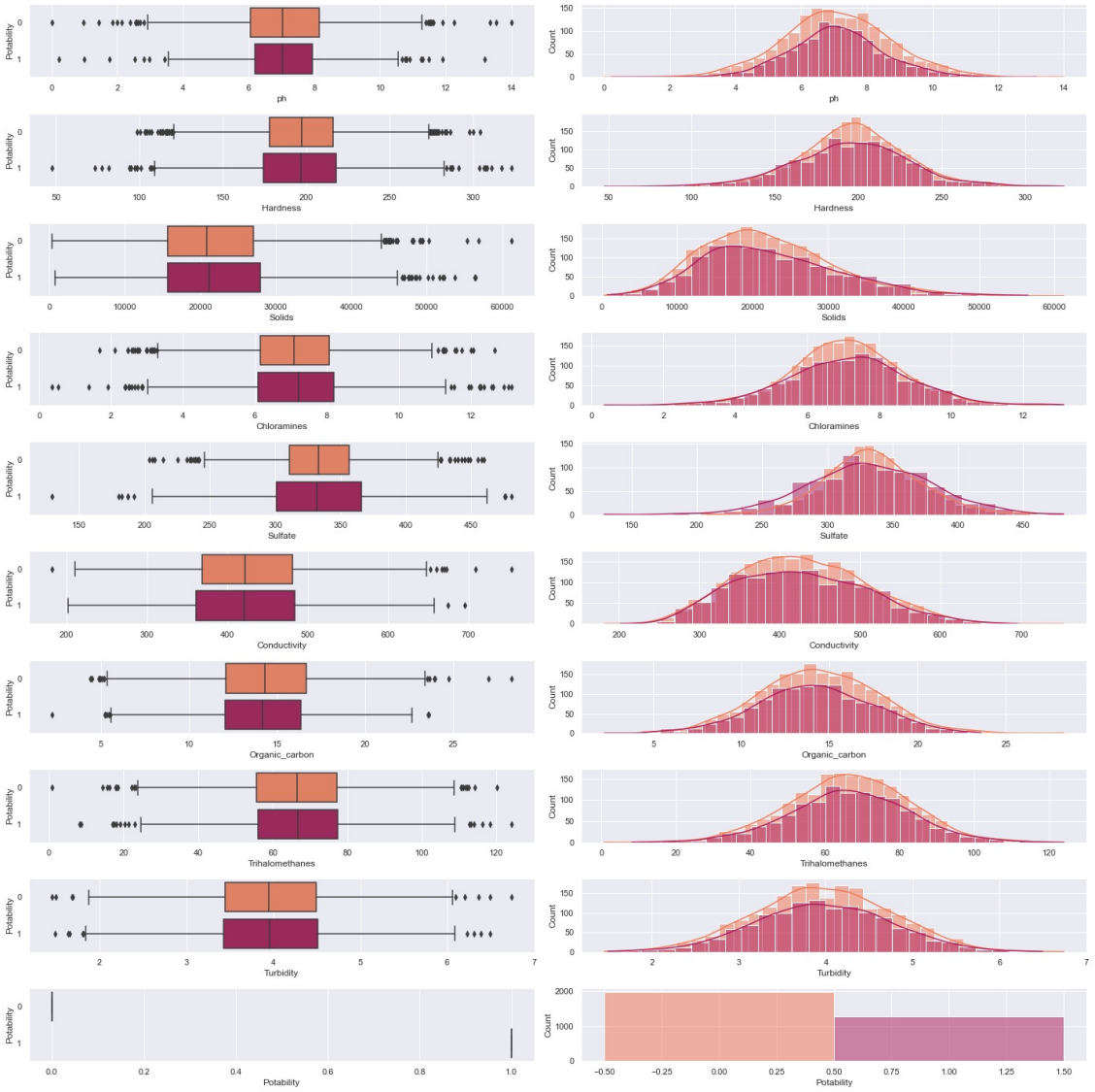

FIGURE 1 – Distribution des features selon la valeur de potabilité

Comme on peut le constater, toutes les variables explicatives suivent une loi Gaussienne. Ainsi, par la suite, nous pouvons utiliser un standard-scaler pour mettre les données à l'échelle. De plus, on remarque que pour toutes les variables, il ne semble pas y avoir une séparation évidente des classes à prédire. Les écarts types et moyennes sont très proches dans tous les cas.

On note tout de même que l'on peut déjà identifier certains outliers. En effet, des valeurs de PH extrêmes correspondent à des eaux labélisées comme potable.

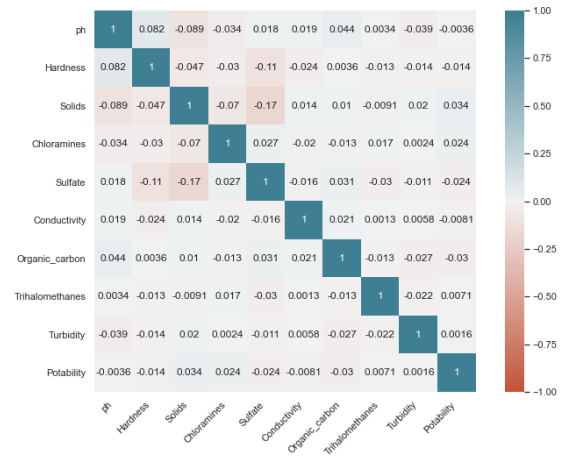

FIGURE 2 – Matrice de corrélation

On peut constater d'après cette matrice de corrélation de Pearson qu'il n'y a aucune corrélation linéaire évidente entre les variables explicatives et la potabilité de l'eau. Ceci est déjà un indice qu'il faudra utiliser un modèle capable de d'expliquer les relations non linéaires.

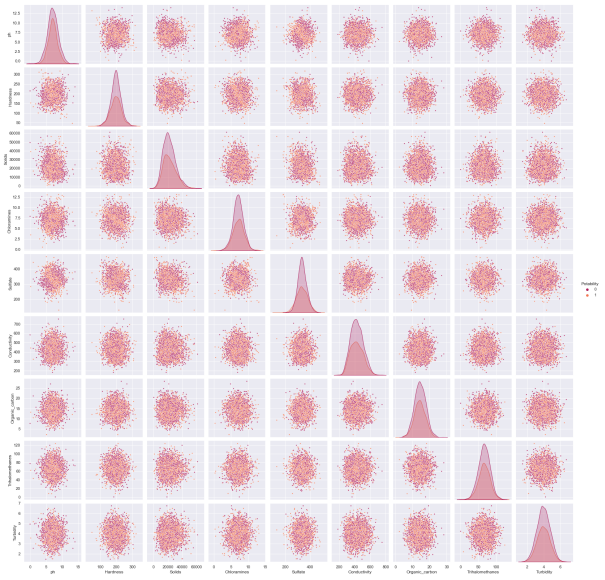

FIGURE 3 – Pairplots

Sur ce pair-plot, nous pouvons confirmer notre hypothèse tirée de la matrice de correlation. En effet, il n'y a aucun pair-plot montrant graphiquement qu'il existe une façon de séparer linéairement sur ces 2 dimensions, les valeurs d'eau potable et non potable.

2.3 Preprocessing des données :

Valeurs aberrantes

Concernant les données aberrantes, nous avons décidé d'utiliser la méthode vue en cours basée sur IQR (en dehors de Q1 1.5 IQR et Q3 + 1.5 IQR). Ceci représente environ 10% de notre dataset.

Valeurs manquantes et standardisation

Lors de notre exploration on constate que seulement trois variables contiennent des valeurs manquantes, le ph avec 14.98% de valeurs manquantes, le Sulfate avec 23.84% et enfin Trihalomethanes avec 4.95%.

Il ne serait pas intéressant de supprimer tout simplement ces valeurs manquantes par ce que l'on perdrait trop de données. De plus, comme le montre la matrice de corrélation, il n'existe pas de corrélation importante entre les variables. Nous ne pourrions donc pas utiliser de méthodes d'imputation basées sur les autres variables.

Ainsi nous avons décidé de remplacer ces valeurs manquantes par la moyenne des variables.

Finalement, nous utilisons un standard-scaler pour toutes les variables. En effet, comme expliqué précédemment dans le rapport, chacun des variables suit une loi Gaussienne. Cette méthode est donc plus adéquate que le min-max.

PCA

L'analyse en composantes principales (PCA pour Principal Component Analysis) est une méthode de réduction de dimension qui consiste à transformer des variables corrélées en nouvelles variables décorrélées les unes des autres.

Il s'agit de résumer l'information contenue dans un ensemble de données en un certain nombre de variables synthétiques, combinaisons linéaires des variables originelles : ce sont les Composantes Principales. Si l'on adopte un point de vue un peu plus mathématique, l'idée est de projeter l'ensemble des données sur l'hyperplan le plus proche des données. Les vecteurs directeurs de cet hyperplan sont les Composantes Principales.

Le choix du nombre de facteurs est important. L'enjeu est généralement de réduire de manière significative la dimension du jeu de données tout en conservant au maximum l'information véhiculée par les données. On parle de part de variance expliquée.

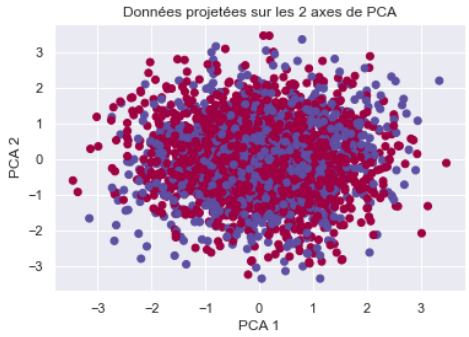

FIGURE 4 – Projection selon les deux premières composantes principales

On peut voir sur cette image les points projetés sur les deux composantes principales. Il est clair qu'il n'y a aucune séparabilité possible dans cet espace dimensionnel. En effet, la part de la variance expliquée est de 0.26.

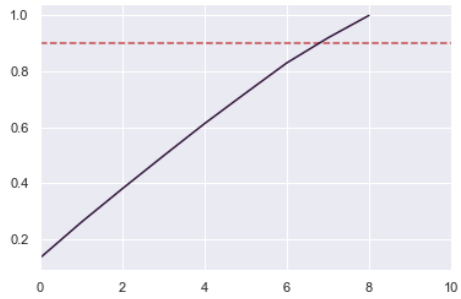

FIGURE 5 – Part de variance expliquée cumulée en fonction du nombre de composantes de la PCA

Sur les graphiques ci-dessus, nous pouvons constater qu'il nous faut 7 composantes principales pour obtenir 90% d'explication de la variance. En d'autres termes, il n'est pas pertinent d'utiliser un PCA dans notre cas. Ceci est certainement lié au fait que la relation entre la variable à expliquer et les variables explicatives est non linéaire

TSNE

Le TSNE est une autre technique de réduction de dimension. Sa particularité comparativement au PCA, est le fait qu'elle soit non linéaire. Ainsi nous avons projeté les données sur les deux axes de TSNE. Encore une fois, nous pouvons constater qu'il n'est pas possible de séparer linéairement les données dans cet espace dimensionnel. Le TSNE n'est donc pas une solution à notre cas et nous allons travailler sur toutes les features.

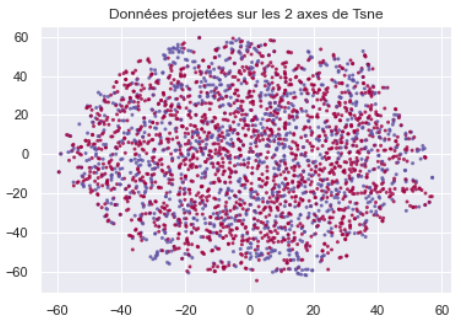

FIGURE 6 – Projection selon les deux premières composantes de TSNE

2.4 Modélisation :

Nous avons décidé de modéliser 6 algorithmes de classification pour notre problème. Les 4 métriques d'évaluations utilisées sont :

- L'accuracy qui est simplement la mesure du taux de prédiction correctes
- Le recall et la précision qui sont des mesures relatives au taux de True Positive. En effet dans notre cas ces métriques sont très importantes puisque l'on ne veut pas classer une eau comme étant potable alors qu'elle ne l'est pas.
- F1-score qui est une mesure hybride permettant de prendre en compte le recall ainsi que la précision en une seule métrique.

Le tableau ci-dessous présente la performance de chacun d'entre eux :

	name	accuracy_score	f1_score	precision_score	recall_score
3	RandomForestClassifier	0.640791	0.387160	0.590496	0.289061
5	GradientBoostingClassifier	0.638145	0.351106	0.554172	0.258736
1	SVC	0.633382	0.537221	0.519058	0.557337
2	KNeighborsClassifier	0.609081	0.403498	0.483929	0.347174
4	AdaBoostClassifier	0.602750	0.312219	0.466368	0.235163
0	LogisticRegression	0.525066	0.455529	0.405258	0.521341

FIGURE 7 – Scores

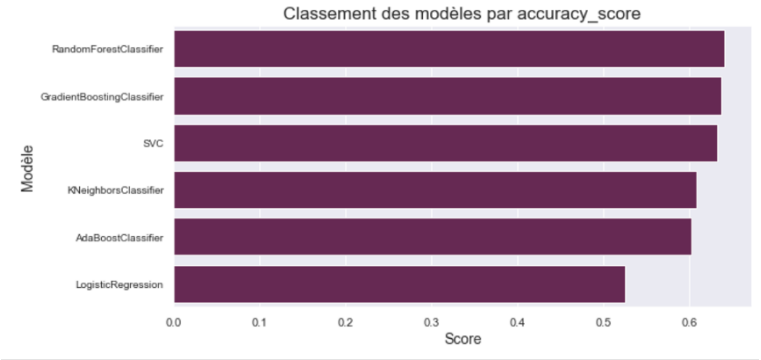

FIGURE 8 – Classement par accuracy

Nous avons donc classé ces algorithmes en fonction de leur accuracy. Les trois modèles les plus performants sont donc le Random Forest, le GrandientBoosting Classifier et le SVM.

Cependant, lorsque l'on prend en compte le F1-score, le grandient boosting classifier est l'un des moins bons algorithmes pour répondre à notre problème, contrairement au SVM et au Random forest.

Ainsi pour la suite de notre démarche, nous allons garder le SVM et le Random Forest pour sélectionner un modèle final.

2.5 Sélection du modèle final :

Avant d'expliquer comment nous avons procéder à la sélection, faisons un petit rappel de ce représentent ces deux modèles.

Random Forest :

Les algorithmes de forêts aléatoires sont un cas particulier du Bagging appliqué aux arbres de décision. En plus du principe de Bagging, les forêts aléatoires ajoutent de l'aléa au niveau des variables. Pour chaque arbre on sélectionne un sous-échantillon par bootstrap d'individus et à chaque étape, la construction d'un noeud de l'arbre se fait sur un sous-ensemble de variables tirées aléatoirement. Le principe de fonctionnement des forêts aléatoire est simple : de nombreux petits arbres de classification sont produits sur une fraction aléatoire de données. Random Forest fait ensuite voter ces arbres de classification peu corrélés afin de déduire l'ordre et l'importance des variables explicatives.

SVM :

Les séparateurs à vastes marges sont des classificateurs qui reposent sur deux idées clés, qui permettent de traiter des problèmes de discrimination non linéaire, et de reformuler le problème de classement comme un problème d'optimisation quadratique.

La résolution du problème de classification passe par la construction d'une fonction hh qui à un vecteur d'entrée xx fait correspondre une sortie yy : y=h(x)y=h(x) . Le principe des SVM est le même pour la classification à l'aide de la régression logistique.

La particularité du SVM est le choix du Kernel. En effet, en fonction du kernel choisit, il sera possible ou non de séparer les données même si la relation est non linéaire (à l'aide d'un changement d'espace dimensionnel).

Nested Cross-Validation :

Pour obtenir le meilleur modèle, nous avons procédé à ce que l'on appelle une grid-search. Cette méthode consiste à faire varier tous les paramètres du model pour obtenir un grand nombre de combinaisons de ces derniers et ainsi comparer les performances pour choisir la meilleure configuration.

Pour comparer les résultats, nous nous sommes basés sur l'accuracy des modèles en utilisant une Nested Cross-Validation. La validation croisée imbriquée (nested cross validation) est une méthode élaborée qui fonctionne ainsi :

Les données sont divisés en K ensembles plus petits. Chacun des K échantillon est mis de côté une fois. Pour chaque modèle d'apprentissage, nous effectuons ensuite une Validation croisée à K' échantillons, avec l'ensemble des K-1 échantillons restants, et avec sélection d'hyperparamètres, comme précedemment. Le meilleur ensemble d'hyperparamètres pour chaque algorithme est utilisé afin d'estimer son score de validation sur l'échantillon mis de côté. Ensuite, le score de validation moyen ainsi que l'écart type sont calculés sur les K échantillons et l'algorithme le plus performant est sélectionné. Par la suite, nous choisissons le meilleur ensemble d'hyperparamètres par GridSearch en utilisant l'ensemble d'entraînement complet et nous estimons l'erreur de généralisation en utilisant l'ensemble de test. Pour la sélection de l'algorithme, il n'est pas réellement important de trouver le 'meilleur' ensemble d'hyperparamètres pour notre échantillon d'entraînement. Il est préferable de choisir un algorithme qui se généralise bien, et qui ne change pas fondamentalement si nous utilisons des données légèrement différentes pour l'apprentissage. L'algorithme doit être stable, sinon, l'estimation de l'erreur de généralisation des modèles, pourrait varier en fonction des données d'entraînement, et cette estimation deviendrait erronnée.

Par conséquent, nous allons chercher le modèle ayant obtenu le meilleur score moyen, mais aussi une variance faible, sur la validation croisée externe.Ce qui ce traduit par des scores similaires sur les différents ensembles d'apprentissage. Si c'est le cas, il semble probable que l'apprentissage de l'algorithme sur les données d'apprentissage complètes produira à nouveau un modèle similaire.

Les paramètres testés sont les suivants :

```
param_grid_rf  = [{'n_estimators': [10, 50, 100, 250, 500, 1000],
                   "max_depth": [None, 5, 10, 20, 30, 40],
                   'max_features': ['sqrt', 'log2']}]

param_grid_svc = [{'kernel': ['rbf', 'linear'],
                   "C": [0.1, 5, 10, 20, 50],
                    'gamma': np.logspace(-4, 0, 4)}]
```

FIGURE 9 – Paramètres cross-validation

Les résultats ont été les suivants :

Modèle	Outer Accuracy
RF	65.37 +/- 0.16
SVM	65.50+/-0.61

Les deux algorithmes ont donc des performances très proches. Cependant comme vu précédemment le SVM est l'algorithme qui a la meilleur precision et le meilleur F1 score.

Nous allons donc garder le SVM comme étant l'algorithme à utiliser et configurer pour la suite.

2.6 Modèle Final et matrice de confusion :

En entrainant le modèle une dernière fois avec les paramètres trouvées précedemments $[' C' : 20,' gamma' : 0.046,' kernel' :' rbf']$, nous obtenons les performances suivantes sur le test set :

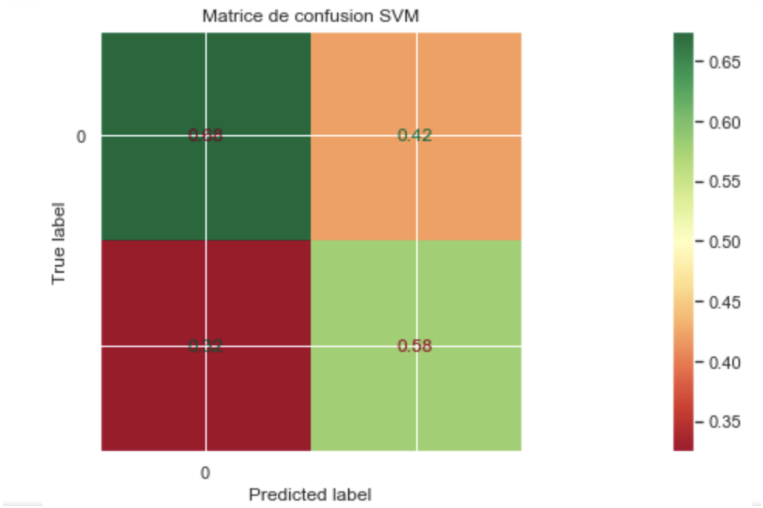

FIGURE 10 – Confusion Matrix

2.7 Feature importance :

Comme la performance du SVM et de RF est très proche, nous pouvons utiliser ce dernier pour étudier l'importance des features et nous obtenons les résultats suivants :

On peut remarquer que le pH et le taux de sulfates sont les deux features les plus importantes et cela rejoint interprétation business qu'on peut se faire sur les caractéristiques les plus importantes pour la potabilité de l'eau notamment en regardant les recommandations de l'OMS.

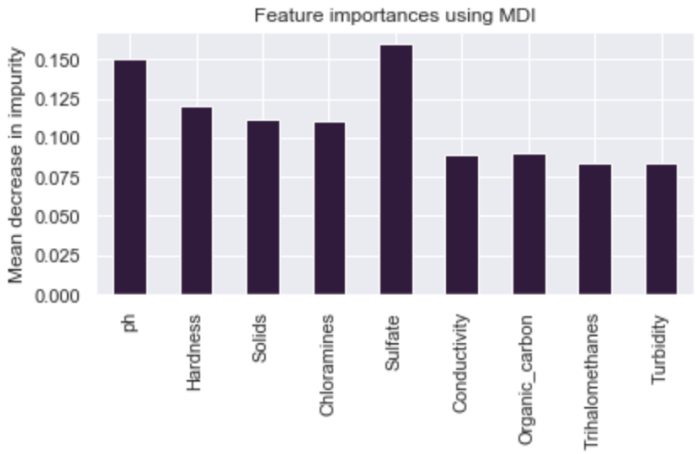

FIGURE 11 – Confusion Matrix

3 Pour aller plus loin :

Pour pouvoir respecter l'ordre de grandeur de la taille du rapport, nous avons omis de présenter de nombreuses démarches et méthodes. En effet, voici une liste non exhaustive de ce que nous avons entrepris mais qui n'apporte pas réellement de meilleur performance :

- Utilisation d'un KPCA (kernel-pca) pour traiter les données non linéaires. Même conclusion que pour le TSNE.
- Feature engineering : nous avons créé d'autres variables binaires à partir des recommandations de WHO. En effet, nous avons transformés les variables en « 1 » lorsque la valeur correspond à la recommandation pour une eau potable par WHO, et « 0 » lorsque la valeur n'est pas dans la plage recommandée. Cependant, utiliser ces autres variables n'a pas apporté de réelle amélioration.
- Remplacer les valeurs manquantes par une autre méthode tel que la médiane.
- Nous aurions pu instancier un voting classifier entre le SVM et le random forest mais leurs accuracy étaient similaire et le SVM présente un meilleur f1-score.
- Exploitabilité du modèle : en utilisant le « feature importance » du random forest, nous pouvons accéder à l'information concernant les variables explicatives les plus importantes du modèle.

4 Conclusion :

Nous avons pu appliquer ce que l'on a appris dans le cours de machine learning à un use-case réel. Ce projet s'est trouvé être très intéressant pour nous car il nous a permis de chercher des solutions qui vont au-delà de ce que l'on a vu en TD et en cours.

Nous pouvons considérer que l'algorithme auquel nous avons aboutis permet d'obtenir des résultats intéressants. Cependant, au vu de l'importance de la tâche, une amélioration pourrait être nécessaire.

En effet, obtenir des variables supplémentaires pourrait être un moyen d'améliorer les performances. De plus, il serait judicieux d'utiliser un tel algorithme comme première « couche décisionnelle », mais de laisser le dernier mot à un expert.

En effet, les algorithmes de machine learning sont très efficaces en tant qu'aide à la décision plutôt que de les utiliser comme seul et unique méthode.