PRÉDICTION DE LA POTABILITÉ DE L'EAU

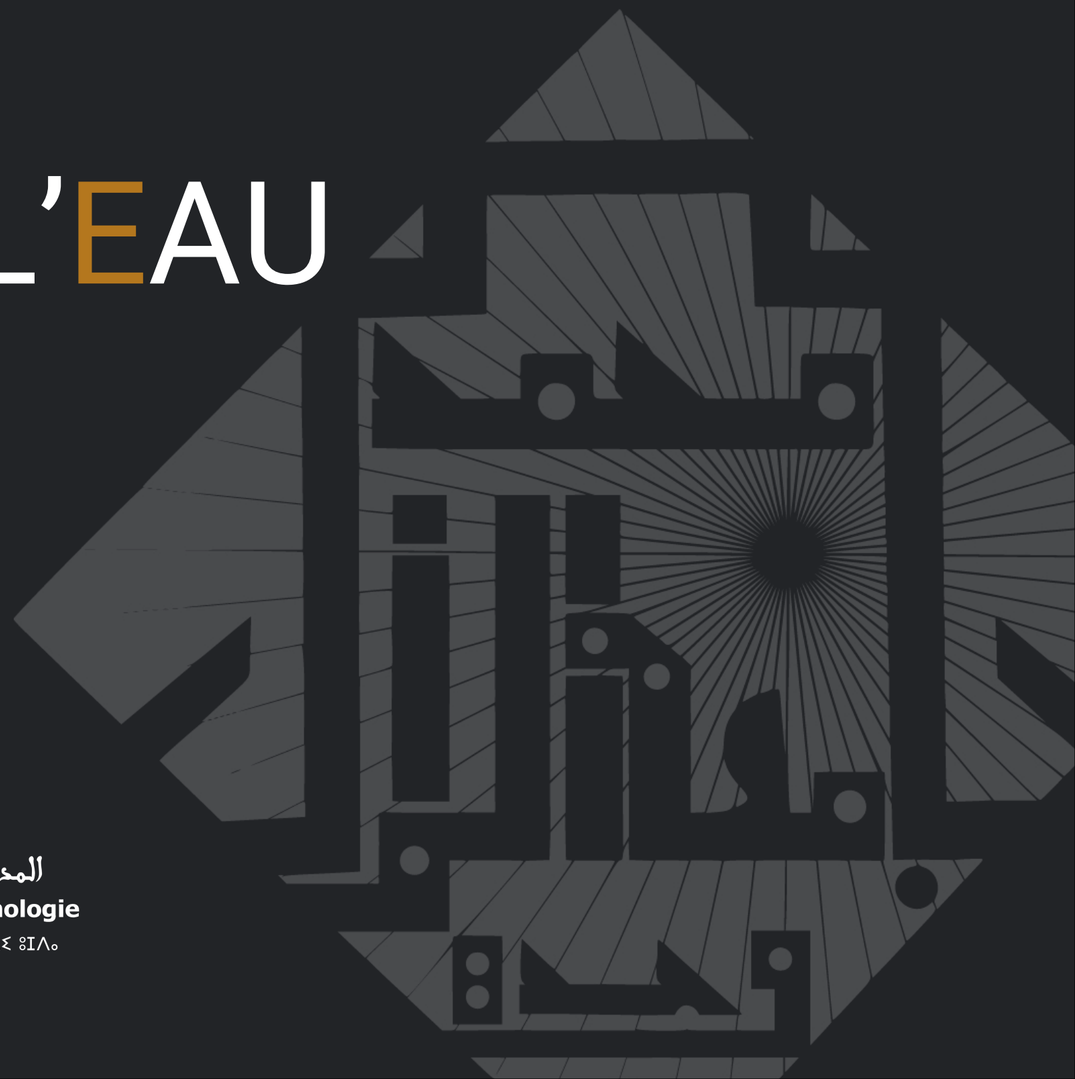
ARBRE DE DECISION

REALISE PAR :
Mohammed JABRI | Ayman MAKHOUKHI

ENCADRE PAR :
Mr. Mohammed BADAOUI

Annee Universitaire : 2020/2021

المدرسة العليا للتكنولوجيا
École Supérieure de Technologie
ⵜⴰⵖⵍⵍⴰⵢⵜ ⵏ ⵜⵓⵙⵙⵉⵔⵉⵏ ⴰ ⵜⵓⵙⵙⵉⵔⵜ ⴰⵣ ⵙⵍⴰⴰ



PLAN

INTRODUCTION

Dans le domaine d’apprentissage automatique (Machine Learning), on distingue entre deux principaux types d’apprentissages : supervisées et non supervisées. où dans notre cas les Arbres de Décision (DT) sont un apprentissage supervisé donc nous discuterons ce dernier concept.

Supervised Learning :

DEFINITION : un algorithme d'apprentissage où nous avons une connaissance préalable de ce que devraient être les valeurs de sortie de nos échantillons.

BUT : l'apprentissage supervisé est d'entraîner le modèle afin qu'il puisse prédire la sortie lorsqu'il reçoit de nouvelles données.

L’exploration des données peut se regrouper en deux catégories principales : Méthodes Descriptives/ Méthodes Prédictives.

Méthodes Prédictives :

BUT : consistent à estimer la valeur d’une variable cible .

CLASSEMENT : Les algorithmes de classement sont utilisés pour prédire une valeur discrète catégorielle (sexe, vrai ou faux …)

LE MODÈLE ETUDIÉ

Arbre de décision

-> Une méthode d’apprentissage supervisée.

-> Un schéma représentant les résultats possibles d'une série de choix interconnectés.

-> Elle permet de faire des prédictions sur des variables (quantitatives ou catégorielles).

-> Elle peut être utilisée pour des problèmes de classement comme pour la régression.

qu’il a l'avantage d'être lisible et rapide à exécuter.

ARBRE DE DECISION

Dans le domaine d’apprentissage automatique (Machine Learning), on distingue entre deux principaux types d’apprentissages : supervisées et non supervisées. où dans notre cas les Arbres de Décision (DT) sont un apprentissage supervisé donc nous discuterons ce dernier concept.

Exemple d’utilisation :

-> **Sécurité et fouille de données**

-> **Médecine :**

- Prédire la maladie du patient.
- Prédire la sensibilité des médicaments.

-> **Finance :**

- Prévoir les résultats futurs et attribuer des probabilités à ces derniers.
- Prédictions binomiales des prix et analyse des options réelles.
- Approbation du prêt.

ARBRE DE DECISION

Principe et méthodologie:

-> Un arbre de décision commence généralement par un noeud d'où découlent plusieurs résultats possibles.

-> Chacun de ces résultats mène à d'autres noeuds, d'où émanent d'autres possibilités.

-> Chacun de ces résultats mène à d'autres noeuds, d'où émanent d'autres possibilités.

-> Il existe trois types de noeuds différents :

- Des noeuds de hasard : représenté par un cercle, montre les probabilités de certains résultats.
- Un noeud de décision : représenté par un carré, illustre une décision à prendre
- Un noeud terminal le résultat final d'un chemin de décision.

ARBRE DE DECISION

Avantages :

-> Ils sont faciles à comprendre.

-> Ils permettent de sélectionner l'option la plus appropriée parmi plusieurs.

-> Il est facile de les associer à d'autres outils de prise de décision.

-> La fiabilité d'un arbre peut être testée et quantifiée.

-> Ils tendent à être précis, même si les hypothèses des données source ne sont pas respectées.

...

Inconvénients :

-> Lors de la gestion de données de catégorie comportant plusieurs niveaux, le gain d'information est biaisé en faveur des attributs disposant du plus de niveaux.

-> Les calculs peuvent devenir compliqués lorsqu'une certaine incertitude est de mise et que de nombreux résultats sont liés entre eux.

-> Les conjonctions entre les noeuds sont limitées à l'opérateur « ET » , alors que les graphiques décisionnels permettent de connecter des noeuds avec l'opérateur « OU »

...

PROBLEMATIQUE

Comment peut-on prédire une eau potable?

Étudier un ensemble de données de mesures de la qualité de l'eau pour construire un modèle qui prédit si le risque et les differents caracteristiques d’avoir une eau potable

Objectif :

Prédire la possibilité si une eau donnée est potable ou non, selon des facteurs de leur qualité.

DESCRIPTION DES DONNEES

La base de données utilisé pendant cette étude est nommée « **water_potability.csv** », et contient **10 attributs.**

L'attribut « Potability » c'est la variable cible, elle désigne la potabilité de l'eau.
Cet attribut regroupe de catégories de valeur :
- 0 : L'eau donné n'est pas potable.
- 1 : L'eau donné est potable.

Cette variable cible est estimée en fonction de plusieurs autres variables explicatives qui constitues les 9 attributs restants :

Attribut	***Description***
PH	La valeur du pH (0 => 14)
Hardness	Capaciter de l'eau à précipiter le savon
Solids	Solides dissous totaux en ppm
Chloramines	Quantité de chloramines en ppm
Sulfates	Quantité de sulfates dissous en mg/L
Conductivity	Conductivité électrique de l'eau en µS/cm
Organic_carbon	Quantité de carbone organique en ppm
Trihalomethanes	Quantité de Trihalomethanes en µg/L
Turbidity	Mesure de la propriété d'émission de lumière de l'eau en NTU

APPLICATION & INTERPRETATION

les étapes suivies

01

ENVIRONEMENT

téléchargement des packages et l'importation des bibliothèques nécessaires.

02

EXPLORATION

l'étude des données ainsi que sa structure et description.

03

PRÉPARATION DU DATASET

la vérification s'il y'a des valeurs manquantes dans les observations et preparer la forme catégorielle pour le modele predictif de la classification.

APPLICATION & INTERPRETATION

les étapes suivies

04

DEVISION DE LA DATASET

Entrainer le modèle sur la base d’apprentissage et vérifier ses performances sur la base de test.

05

IMPORTANCE

Etude de l'importance de chaque variable.

06

CREATION DU MODELE DT

Creation du modele d'arbre de decision avec les donnees d'apprentissage

APPLICATION & INTERPRETATION

les étapes suivies

07

PREDICTION

L’effectuation de la prédiction sur la base de test

08

EVALUATION DU MODELE

calcul de l'erreur , et l'évaluation du modèle avec la matrice de confusion et etudier la performance du modele.

09

VISUALISATION

Visualisation de l ’arbre de décision graphiquement.

APPLICATION & INTERPRETATION

ENVIRONEMENT

L'installation des packages nécessaires

L'installation des packages nécessaires

```{r}

install.packages("FSelector")
install.packages("party")
install.packages("rpart.plot")
install.packages("data.tree")
install.packages("ggthemes")
```

Le chargement des librairies nécessaire

```{r}
library(FSelector)
library(rpart)
library(caret)
library(rpart.plot)
library(data.tree)
library(dplyr)

library(randomForest)
library(psych)
library(pROC)
library(Amelia)

library(ggplot2)
library(plotly)
library(ggthemes)

library(rattle)
```

APPLICATION & INTERPRETATION

EXPLORATION

on commence par l'étude des données ainsi que sa structure et description, après on vérifie s'il y'a des valeurs manquantes dans les observations comme indiqué ci-dessous :

Importation et étude des données

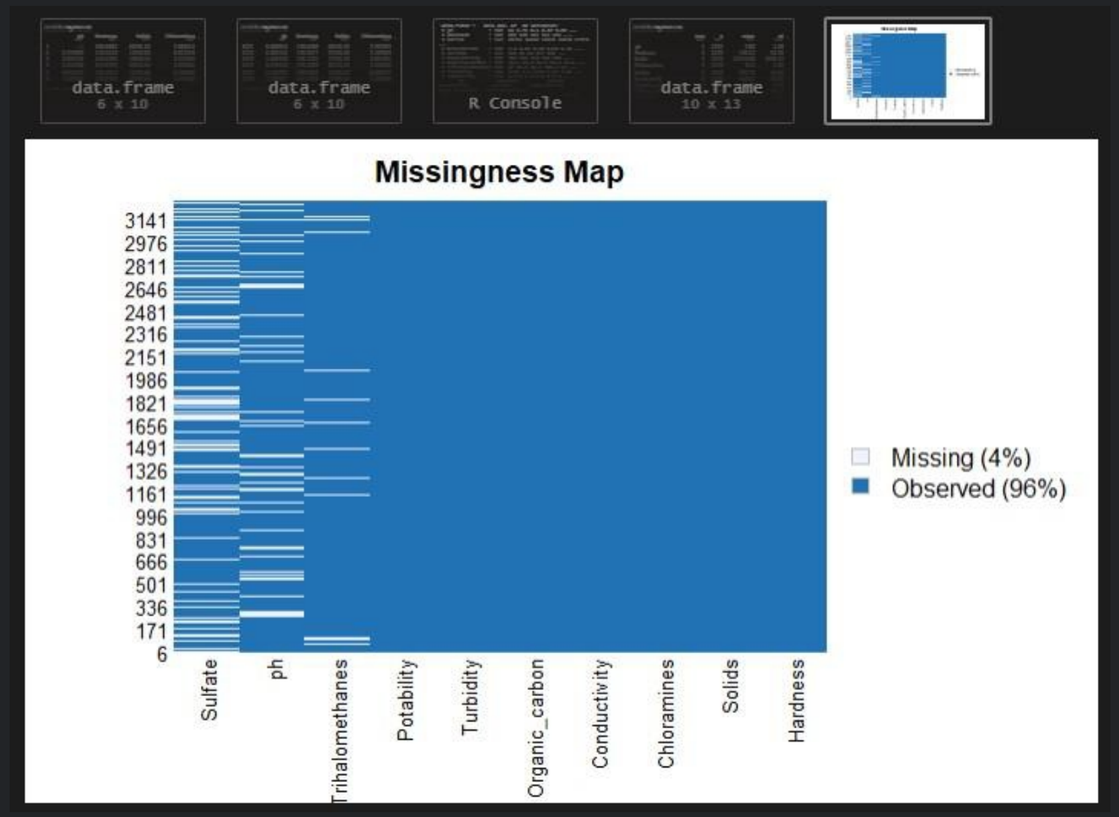
```{r}
# Importer les donnée qui ont dans le fichier water_potability.csv qui est dans la même répertoire
data <- read.csv("./water_potability.csv")

# les premières observations des données
head(data)

# les dernières observations des données
tail(data)

# La structure des attributs
str(data)

# Statiques de bases sur l'ensemble des attributs
summary(data)
describe(data)

# Les valeurs non-observés pour chaque attribut
missmap(data)

table(data$Potability)
```

APPLICATION & INTERPRETATION

EXPLORATION

Le plot de la fonction missmap() nous montre qu'il y a souvent beaucoup

de valeurs manquants dans plusieurs observations de la base de données, surtout lequel de l'attribut « Sulfate » et « pH ».



APPLICATION & INTERPRETATION

PRÉPARATION DU DATASET

Tout d'abord on va convertir la variable cible de la forme numérique à la forme catégorielle pour éviter les problèmes avec la classification, et ensuite on va supprimer toutes les observations qui ont des valeurs manquantes :

Pre-Processing, correction et nettoyage des données

```{r}

# generer une liste des indexes aléatoires
shuffle_index <- sample(1:nrow(data))

# On va utiliser ses indexes pour mélanger les donnée
data <- data[shuffle_index, ]

# Conversion de la variable cible d'une forme numérique au forme catégorielle
data <- mutate(data, Potability = factor(Potability, levels = c(0, 1), labels = c('No', 'Yes')))

# Supprimer tout les observation avec des valeurs manquantes
data <- na.omit(data)
glimpse(data)

```

APPLICATION & INTERPRETATION

DEVISION DE LA DATASET

Après avoir étudier les données, ainsi que les nettoyer, on va diviser la base de données en base d'apprentissage « training data », et une base de test « testing data » : On divise les données en 80% pour l'apprentissage et 20% pour le test, et après ont vérifié les pourcentages de potabilité dans chaque base.

Divise les données entre données d'apprentissage et données de test

```r
# Les données sont divisées : 80% d'apprentissage, et 20% de test
set.seed(123)
sample = sample.split(data$Potability, SplitRatio = .80)

# Données d'apprentissage
train_data = subset(data, sample==TRUE)

# Données de test
test_data = subset(data, sample==FALSE)

# Pourcentage de potabilité dans les données d'apprentissage et les données de test
prop.table(table(train_data$Potability))
prop.table(table(test_data$Potability))
```

APPLICATION & INTERPRETATION

DEVISION DE LA DATASET

L’objectif principal de cette division est de vérifier les performances du modèle sur des données invisibles. Ou, en d’autres termes d’entrainer le modèle sur l’ensemble d’apprentissage et vérifier ses performances sur l’ensemble de test :

```
# Pourcentage de potabilité dans les données d'apprentissage et les données de test
prop.table(table(train_data$Potability))
prop.table(table(test_data$Potability))
```

```
        0         1
0.6099237 0.3900763

        0         1
0.6097561 0.3902439
```

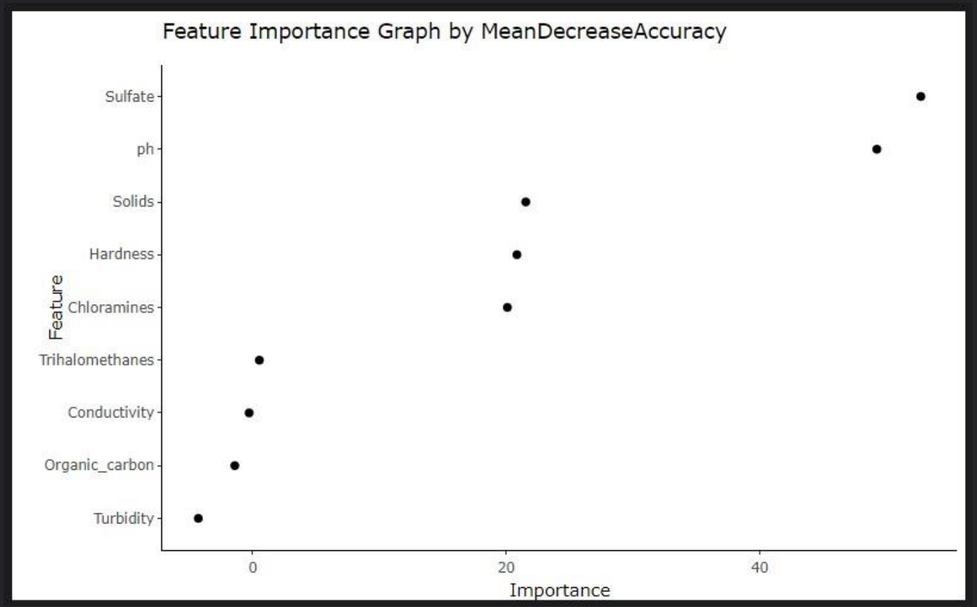
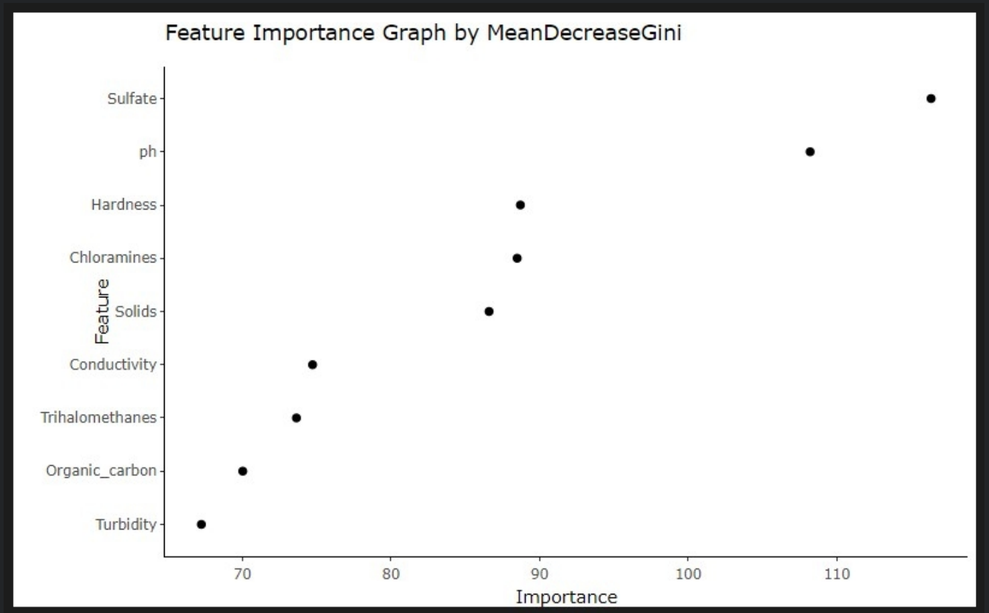
APPLICATION & INTERPRETATION

Etude de l'importance chaque variable selon

« Accuracy » avec le plot « MeanDecreaseAccuracy »,

et aussi « Gini » avec le plot « MeanDecreaseGini » :

IMPORTANCE

```r
```{r}
# Importance des attributs
# L'utilisation des forêts aléatoires juste pour visualiser l'importance de chaque variable

rf_tmp <- randomForest(Potability ~ .,
                       data=train_data, ntree=1000,
                       keep.forest=FALSE,
                       importance=TRUE)

# varImpPlot(rf_tmp, main = "Importance des variables")
# importance(rf_tmp)

# GGploot Plots

feat_imp_df <- importance(rf_tmp) %>%
    data.frame() %>%
    mutate(feature = row.names(.))

# Feature Importance Graph | MeanDecreaseAccuracy

importanceAccuracyGraph <- ggplot(feat_imp_df, aes(x = reorder(feature, MeanDecreaseAccuracy), y = MeanDecreaseAccuracy)) +
    geom_point() +
    coord_flip() +
    theme_classic() +
    labs(
      x     = "Feature",
      y     = "Importance",
      title = "Feature Importance Graph by MeanDecreaseAccuracy",
      color="Feature"
    )

ggplotly(importanceAccuracyGraph)


# Feature Importance Graph | MeanDecreaseGini

importanceGiniGraph <- ggplot(feat_imp_df, aes(x = reorder(feature, MeanDecreaseGini), y = MeanDecreaseGini)) +
    geom_point() +
    coord_flip() +
    theme_classic() +
    labs(
      x     = "Feature",
      y     = "Importance",
      title = "Feature Importance Graph by MeanDecreaseGini",
      color="Feature"
    )

ggplotly(importanceGiniGraph)
```
```

APPLICATION & INTERPRETATION

IMPORTANCE

MeanDecreaseAccuracy : Estimer à quel point l'absence d'une variable affecte la performance.

MeanDecreaseGini : Estimer la pureté des noeuds par division.

Selon le résultat ci-dessus on peut constater que les variables : « Sulfate » et « ph » ont une importance élevée.





APPLICATION & INTERPRETATION

CREATION DU MODELE DT

Ensuite on applique le classifieur « arbre de décision » avec la fonction rpart () :

```
Création du modèle d'arbre de décision avec les données d'apprentissage


```{r}
# Créattion du classifieur sur les donnée d'apprentissage
tree <- rpart(Potability ~.,
              data = train_data,
              method="class")

```
```

APPLICATION & INTERPRETATION

PREDICTION

Et finalement on effectue la prédiction sur la base de test « testing data » :

Prédiction avec l'arbre de décision sur les donnée de test

```{r}

# Prédiction sur les données de test
tree.Potability.predicted <- predict(tree, test_data, type='class')

# Calculer l'erreur de la prédiction sur les données de test
tab <- table(tree.Potability.predicted, test_data$Potability)
paste("Erreur sur le test_data :", round(1 - sum(diag(tab)) / sum(tab), digits = 2), "%")
cat("\n")

# Générer la courbe ROC
roc(test_data$Potability,
    as.numeric(tree.Potability.predicted),
    plot=TRUE, legacy.axes=TRUE, percent=TRUE, print.auc=TRUE)

# Evaluer le modèle avec la matrice de confusion
confusionMatrix(tree.Potability.predicted,test_data$Potability)
```

APPLICATION & INTERPRETATION

EVALUATION DU MODELE

Résultat de calcul de l'erreur, et l'évaluation du modèle avec la matrice de confusion :

La précision de ce modèle est 65%, l'erreur c'est 35%, et le P-Value = 0,014.

« 219 échantillons sont bien classés comme potable par contre 21 mal classés sur la colonne 1 ».

« 36 échantillons sont bien classés comme pas potable par contre 126 mal classés sur la colonne 2 ».

```
[1] "Erreur sur le test_data : 0.35 %"

Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No  212 112
       Yes  28  50

               Accuracy : 0.6517
                 95% CI : (0.6029, 0.6983)
    No Information Rate : 0.597
    P-Value [Acc > NIR] : 0.01388

                  Kappa : 0.2096

 Mcnemar's Test P-Value : 2.303e-12

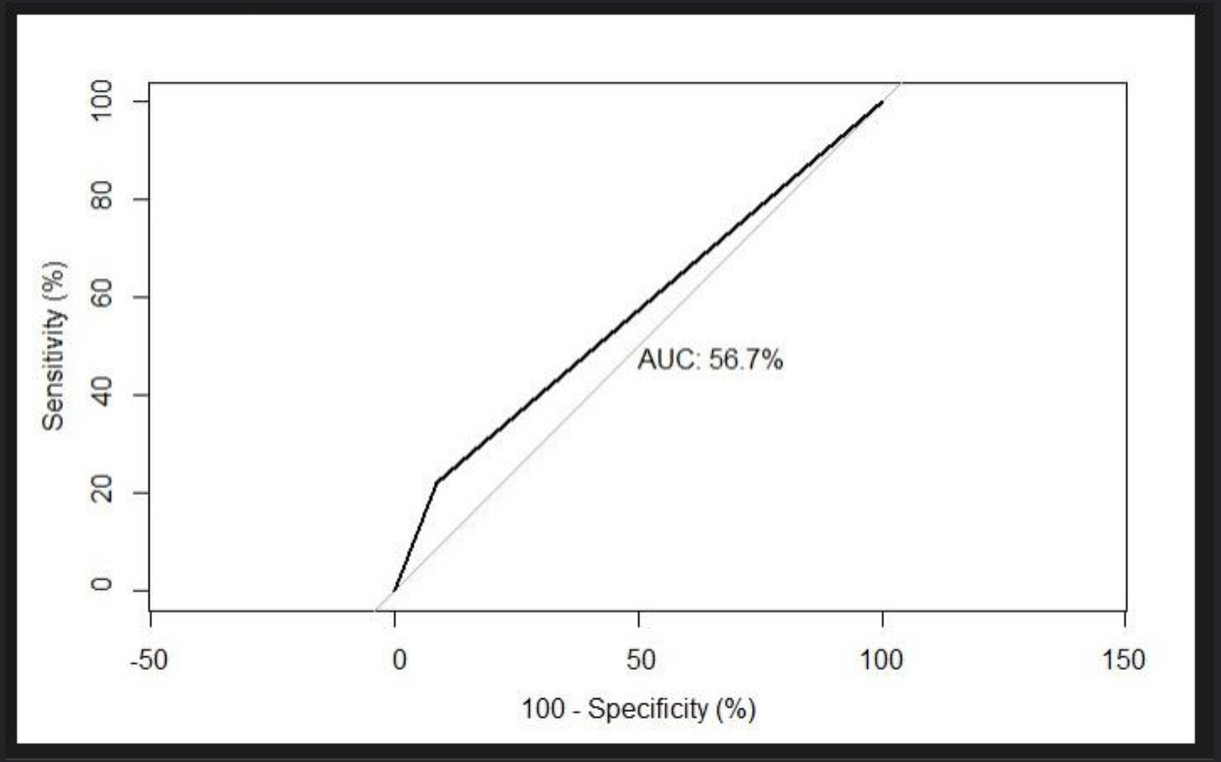
            Sensitivity : 0.8833
            Specificity : 0.3086
         Pos Pred Value : 0.6543
         Neg Pred Value : 0.6410
             Prevalence : 0.5970
         Detection Rate : 0.5274
   Detection Prevalence : 0.8060
      Balanced Accuracy : 0.5960

       'Positive' Class : No
```

APPLICATION & INTERPRETATION

EVALUATION DU MODELE

La courbe ROC : représente les performances du modèle, en traçant le taux des valeurs Faux Positives en fonction de valeurs Vrais Positives.



APPLICATION & INTERPRETATION

VISUALISATION

Visualisation de l’arbre de décision graphiquement :

```
Visualisation de l'arbre de décision graphiquement

```{r}

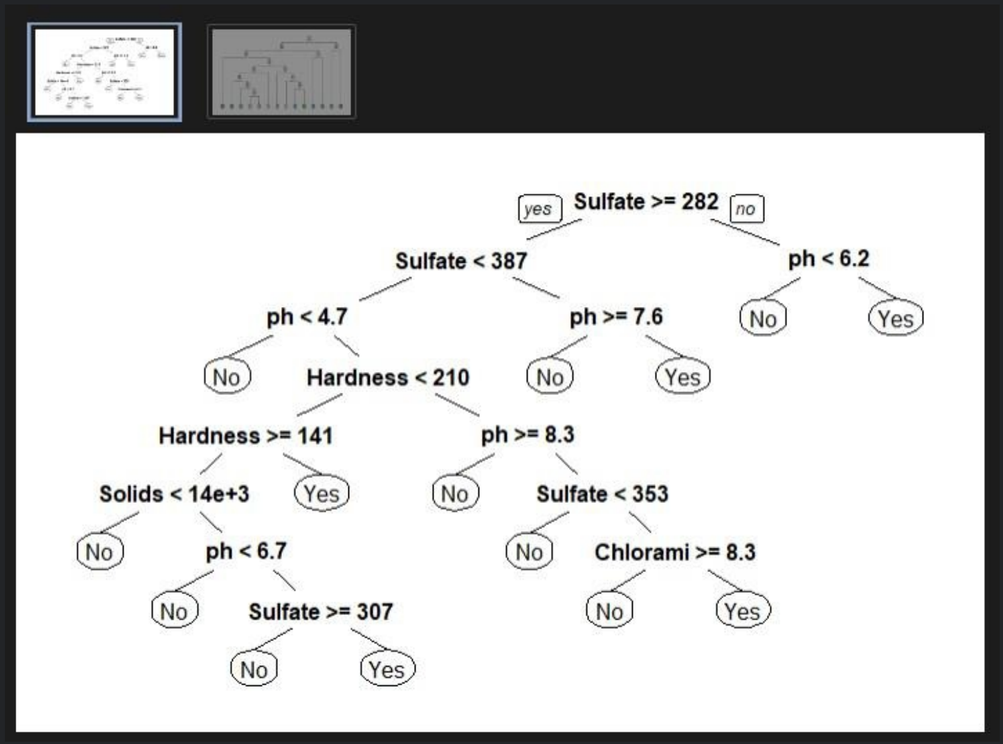
# Visualisation de l'arbre de décision
prp(tree)
rpart.plot(tree)

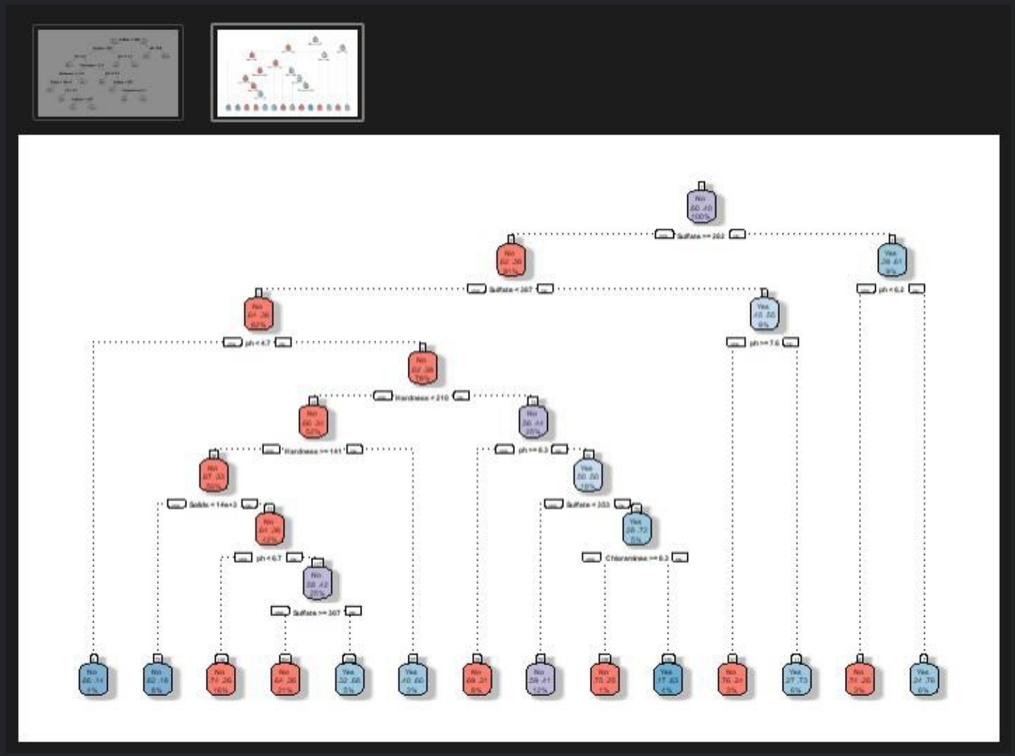
#print(tree2)
#plot(tree2)
```
```

APPLICATION & INTERPRETATION

VISUALISATION

Visualisation de l’arbre de décision graphiquement :





CONCLUSION

il n’existe pas un modèle parfait pour toutes les situations, chaque situation, chaque ensemble de données et chaque objectif souhaité exige un choix spécifique d’un modèle.

MERCI POUR VOTRE ATTENTION

DES QUESTIONS?