# rpart

Thomas Servant

07/01/2021

## R Markdown

Dans le cadre de notre partiel, nous devons réaliser un total de 12 travaux retracant notre parcours et notre travail durant les 30 heures de cours. Le travail à faire est le suivant : - Une entête comportant un titre, un lien Github avec le ou les noms des auteurs. - Une synthese de ce travail - Un extrait commenté avec des parties de codes clé avec explication et commentaire. - Une évalutation du travail avec nos 5 criteres. - Une conclusion du travail

## Definition des 5 critères de notations :

1) Présentation et lisibilité du RMD. 2) Knit opérationnel. 3) Contenue facilement compréhensible. 4) Facilité de réutilisation du code. 5) Explication des outils utilisés.

## Travail n°2 : "RPART PACKAGE"

Travail réalisé par "Maxime & Siva".

https://github.com/mallaker/PSB_X/blob/main/Package%20Rpart/Rpart%20package.Rmd

## Synthese :

Ce RMD porte sur le package RPART (Recursive And Regression Trees), il permet de construire des modèles de classification ou de régresssion d'une très générale. Les modèles étudiés sont représentés sous forme d'arbres de decision.

La méthode d'arbre de décision est une technique d'apprentissage automatique prédictif puissant et populaire, aussi connu sous le nom de CART.

Dans leur Markdown, les étudiants ont procédé a l'explicatin des differents avantages et inconveniennt de cette méthode et ont ensuite donné un exemple de code.

## Extrait commenté du code :

Dans leur introduction , les auteurs ont utilisé le code ci-dessous pour illustrer leur exemple :

```
library(rpart)
library(rpart.plot)

data(ptitanic)
summary(ptitanic)
```

```
##  pclass        survived       sex           age              sibsp
##  1st:323   died    :809   female:466   Min.   : 0.1667   Min.   :0.0000
##  2nd:277   survived:500   male  :843   1st Qu.:21.0000   1st Qu.:0.0000
##  3rd:709                               Median :28.0000   Median :0.0000
##                                        Mean   :29.8811   Mean   :0.4989
##                                        3rd Qu.:39.0000   3rd Qu.:1.0000
##                                        Max.   :80.0000   Max.   :8.0000
##                                        NA's   :263
##      parch
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :0.000
##  Mean   :0.385
##  3rd Qu.:0.000
##  Max.   :9.000
##
```

```
lapply(ptitanic,class)
```

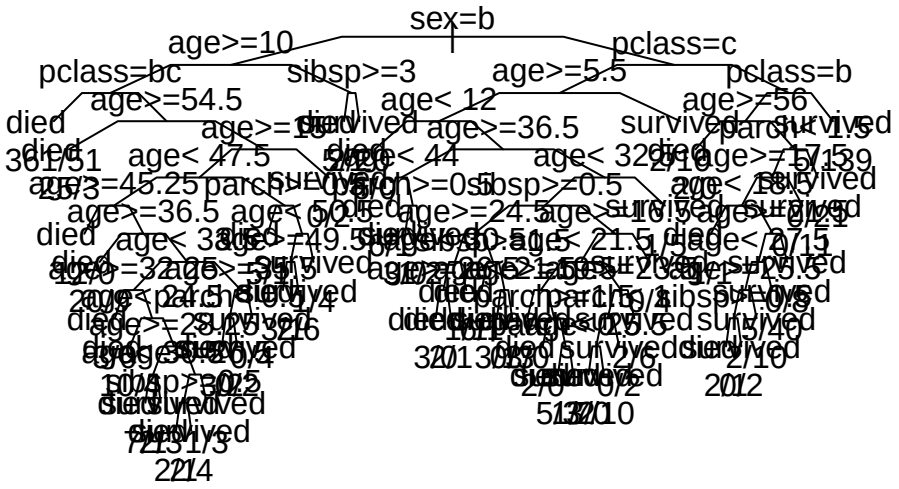
```
## $pclass
## [1] "factor"
##
## $survived
## [1] "factor"
##
## $sex
## [1] "factor"
##
## $age
## [1] "labelled"
##
## $sibsp
## [1] "labelled"
##
## $parch
## [1] "labelled"
```

```
attr(ptitanic$age,"class") <- NULL
class(ptitanic$age)
```

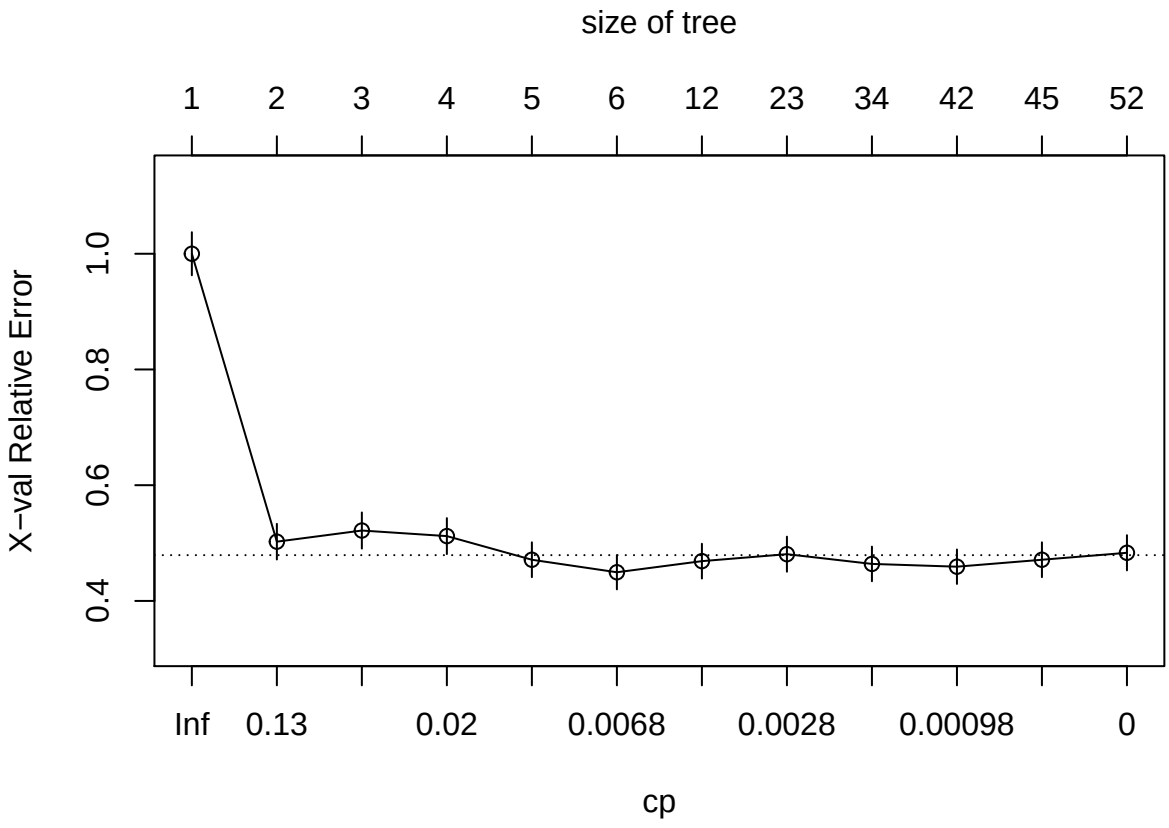
```
## [1] "numeric"
```

```
nb_lignes <- floor((nrow(ptitanic)*0.75)) #on selctionne le nombre de ligne pour notre echantillons d'a
ptitanic.apprt <- ptitanic[1:nb_lignes, ]#echantillon d'apprentissage
ptitanic.test <- ptitanic[(nb_lignes+1):nrow(ptitanic), ]#echantillon de test

#construction de l'arbre
ptitanic.Arbre <- rpart(survived~.,data= ptitanic.apprt,control=rpart.control(minsplit=5,cp=0))
#affichage de l'arbre
plot(ptitanic.Arbre, uniform=TRUE, branch=0.5, margin=0.1)
text(ptitanic.Arbre,all=FALSE, use.n=TRUE)
```




```
plotcp(ptitanic.Arbre)
```

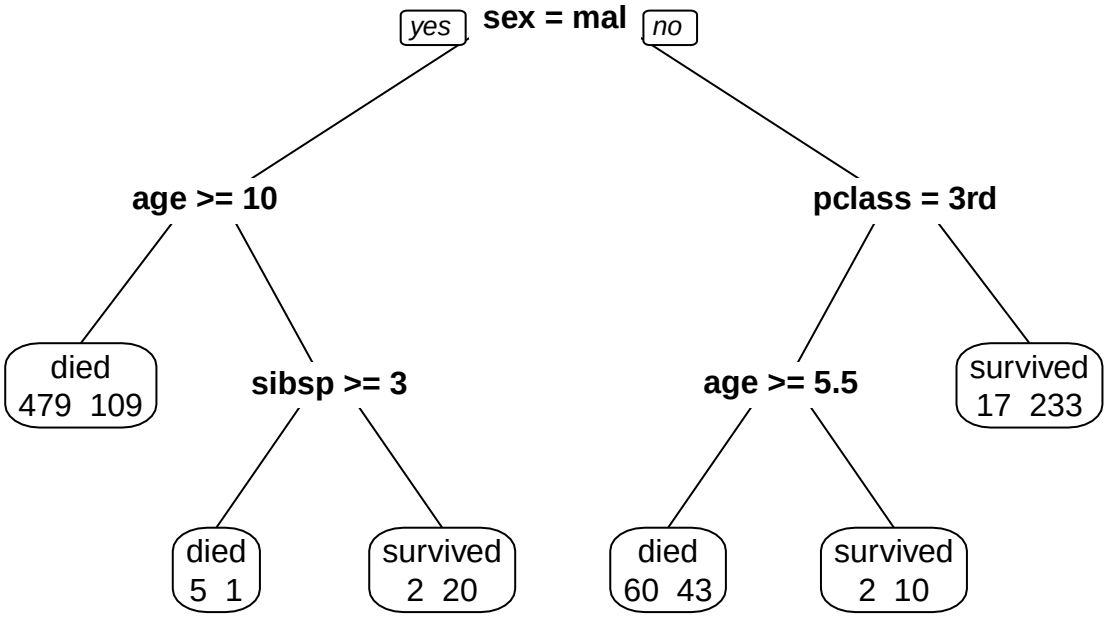



```
print(ptitanic.Arbre$cptable[which.min(ptitanic.Arbre$cptable[,4]),1])
```

```
## [1] 0.004807692
```

```
ptitanic.Arbre_Opt <- prune(ptitanic.Arbre,cp=ptitanic.Arbre$cptable[which.min(ptitanic.Arbre$cptable[,4
```

```
prp(ptitanic.Arbre_Opt,extra = 1)
```




Test et validation

```
#prediction du modele sur les données de test
ptitanic.test_predict <- predict(ptitanic.Arbre_Opt,newdata =ptitanic.test,type = "class")
#affichons juste la prediction faite sur les 10 premiers elements
print(ptitanic.test_predict[1:10])
```

```
##  982  983  984  985  986  987  988  989  990  991
## died died died died died died died died died died
## Levels: died survived
```

```
#Matrice de confusion
MC <- table(ptitanic.test$survived,ptitanic.test_predict)
print(MC)
```

```
##           ptitanic.test_predict
##            died survived
##   died      238        6
##   survived   77        7
```

```
#Erreur de classement
erreur <- 1.0-(MC[1,1]+MC[2,2]/sum(MC))
print(erreur)
```

```
## [1] -237.0213
```

```r
#Taux de prediction
prediction <- MC[2,2]/sum(MC[2,])
prediction
```

```
## [1] 0.08333333
```

```r
print(ptitanic.Arbre_Opt)
```

```
## n= 981
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 981 416 died (0.57594292 0.42405708)
##    2) sex=male 616 130 died (0.78896104 0.21103896)
##      4) age>=10 588 109 died (0.81462585 0.18537415) *
##      5) age< 10 28   7 survived (0.25000000 0.75000000)
##       10) sibsp>=3 6   1 died (0.83333333 0.16666667) *
##       11) sibsp< 3 22   2 survived (0.09090909 0.90909091) *
##    3) sex=female 365  79 survived (0.21643836 0.78356164)
##      6) pclass=3rd 115  53 died (0.53913043 0.46086957)
##       12) age>=5.5 103  43 died (0.58252427 0.41747573) *
##       13) age< 5.5 12   2 survived (0.16666667 0.83333333) *
##      7) pclass=1st,2nd 250  17 survived (0.06800000 0.93200000) *
```

## Evaluation du travail :

Ce tutoriel a pour but d'aborder les principes du package rpart et de la méthode d'arbre de décision, les explications sont clair et le code est très facilement réutilisables. Il m'a été très facile de le knit sur ma machine.

1) Présentation et lisibilité du RMD : RMD bien structuré et détaillé. 2) Knit opérationnel : RMD très facile a kniter. 3) Contenue facilement compréhensible : Bonnes explication avec un bon détail de chaque étape. 4) Facilité de réutilisation du code : Code très bien détaillé a chaque étape et semble très facile a réutiliser. 5) Explication des outils utilisés : Chaque chunk est très bien detaillé avec plein de commentaires pour expliquer étape par étape les différents outils utilisés.

## Conclusion :

On peux en conclure que c'est un très bon RMD. très bien expliqué, concis, organisé et très simple a réutiliser.