HEC MONTRÉAL

Projet MATH60603

Prévisions de perte de clientèle

David Garson, 11262368
Jean Guirguis, num-matricule

Projet présenté à

.Professeure Aurélie Labbé

Table des matières

Note :

baggin + svm + regression linéaire

équilibrer VS non équilibré (3 méthode)

https://www.slideshare.net/yogesh_khandelwal/churn-modelling

https://www.erpublication.org/published_paper/IJETR032129.pdf

1. Introduction : présentation des objectifs de l'étude

L'objectif de cette étude est de faire une prédiction sur le désbonnement d'un client pour un service de télécommunication. Cette simulation a été fait à partir d'une base de données présent sur kaggle à l'adresse suivante : https://www.kaggle.com/mnassrib/telecom-churn-datasets?select=churn-bigml-80.csv. L'analyse a été effectué sur les client d'Orange Telecom aux États-Unis et plusieurs variables on été analysés tels que lieu géographique, le type de plan que le client possède, le nombre de minutes utilisé pour des appels durant le jour et le soir, etc. Ce rapport contient en premier lieu, une présentation détaillé de notre jeu de donnée : présentation des variables et des observations, quelques analyses de corrélations entre les variables et de traintement et réencodage préalable du jeu de donnée. Ensuite, plusieurs méthode et stratégie de prédiction seront testés afin de déterminer le modèle le mieux adapté à notre échantillion d'utilisateurs. Tels que la régressions, les abres et forêt élague, etc. Finalement, les caractéristiques de chacun des modèeles seront comparés afin de déterminer le meilleur modèele de prédiction pour notre jeu de donnée.

2. Exploration des données

Cette section présentera d'abord les variables du jeu de donnée. La répartition des observations dans chacune des variables est rapidement survolé. Cette section explique aussi les différents traitements, réencodage et validation exécuté sur les variables et les observations avant de débuter l'analyse des différents modèeles de simulations.

2.1. Présentation du jeu de donnée

Ci-dessous est présenté l'ensemble des variables. L'ensemble des données est donc constitué de 20 variables énuméré ci-dessous. Le jeu de donnée est donc initialement constitué de 4 variables de type chaîne de caractères ("State","International.plan","Voice.mail.plan" et "Churn"). La variable Area.code est catégorielle, tandis que tous les autres varibales sont continuent.

- State : États américaine de l'observation, valeur de type "character"
- Account.length : Valeur entière depuis combien de temps le compte existe-t-il
- Area.code : Indicatif régionale (415,408 ou 510)
- International.plan : Adhésion au plan international (Yes ou No)
- Voice.mail.plan : Adhésion au plan de messagerie (Yes ou No)
- Number.vmail.messages : Nombre de message vocal
- Total.day.minutes : Nombre de minute utilisé durant le jour
- Total.day.calls : Nombre d'appel exécuté durant le jour
- Total.day.charge : Coût total pour l'utilisation de jour
- Total.eve.minutes : Nombre de minute utilisé durant le soir
- Total.eve.calls : Nombre d'appel exécuté durant le soir
- Total.eve.charge : Coût total pour l'utilisation de soir
- Total.night.minutes : Nombre de minute utilisé durant la nuit
- Total.night.calls : Nombre d'appel exécuté durant la nuit
- Total.night.charge : Coût total pour l'utilisation la nuit

— Total.intl.minutes : Nombre de minute utilisé à l'international
— Total.intl.calls : Nombre d'appel exécuté à l'international
— Total.intl.charge : Coût total pour l'utilisation à l'international
— Customer.service.calls : Nombre d'appel exécuté pour le service au client
— Churn : Est-ce que le client à quitté (True, False)

```
summary(train)
```

```
##     State           Account.length    Area.code     International.plan
##  Length:2666        Min.   :  1.0   Min.   :408.0   Length:2666
##  Class :character   1st Qu.: 73.0   1st Qu.:408.0   Class :character
##  Mode  :character   Median :100.0   Median :415.0   Mode  :character
##                     Mean   :100.6   Mean   :437.4
##                     3rd Qu.:127.0   3rd Qu.:510.0
##                     Max.   :243.0   Max.   :510.0
##  Voice.mail.plan    Number.vmail.messages Total.day.minutes Total.day.calls
##  Length:2666        Min.   : 0.000        Min.   :  0.0     Min.   :  0.0
##  Class :character   1st Qu.: 0.000        1st Qu.:143.4     1st Qu.: 87.0
##  Mode  :character   Median : 0.000        Median :179.9     Median :101.0
##                     Mean   : 8.022        Mean   :179.5     Mean   :100.3
##                     3rd Qu.:19.000        3rd Qu.:215.9     3rd Qu.:114.0
##                     Max.   :50.000        Max.   :350.8     Max.   :160.0
##  Total.day.charge Total.eve.minutes Total.eve.calls Total.eve.charge
##  Min.   : 0.00    Min.   :  0.0     Min.   :  0     Min.   : 0.00
##  1st Qu.:24.38    1st Qu.:165.3     1st Qu.: 87     1st Qu.:14.05
##  Median :30.59    Median :200.9     Median :100     Median :17.08
##  Mean   :30.51    Mean   :200.4     Mean   :100     Mean   :17.03
##  3rd Qu.:36.70    3rd Qu.:235.1     3rd Qu.:114     3rd Qu.:19.98
##  Max.   :59.64    Max.   :363.7     Max.   :170     Max.   :30.91
##  Total.night.minutes Total.night.calls Total.night.charge Total.intl.minutes
##  Min.   : 43.7       Min.   : 33.0     Min.   : 1.970     Min.   : 0.00
##  1st Qu.:166.9       1st Qu.: 87.0     1st Qu.: 7.513     1st Qu.: 8.50
##  Median :201.2       Median :100.0     Median : 9.050     Median :10.20
##  Mean   :201.2       Mean   :100.1     Mean   : 9.053     Mean   :10.24
##  3rd Qu.:236.5       3rd Qu.:113.0     3rd Qu.:10.640     3rd Qu.:12.10
##  Max.   :395.0       Max.   :166.0     Max.   :17.770     Max.   :20.00
##  Total.intl.calls Total.intl.charge Customer.service.calls    Churn
##  Min.   : 0.000   Min.   :0.000     Min.   :0.000          Length:2666
##  1st Qu.: 3.000   1st Qu.:2.300     1st Qu.:1.000          Class :character
##  Median : 4.000   Median :2.750     Median :1.000          Mode  :character
##  Mean   : 4.467   Mean   :2.764     Mean   :1.563
##  3rd Qu.: 6.000   3rd Qu.:3.270     3rd Qu.:2.000
##  Max.   :20.000   Max.   :5.400     Max.   :9.000
```

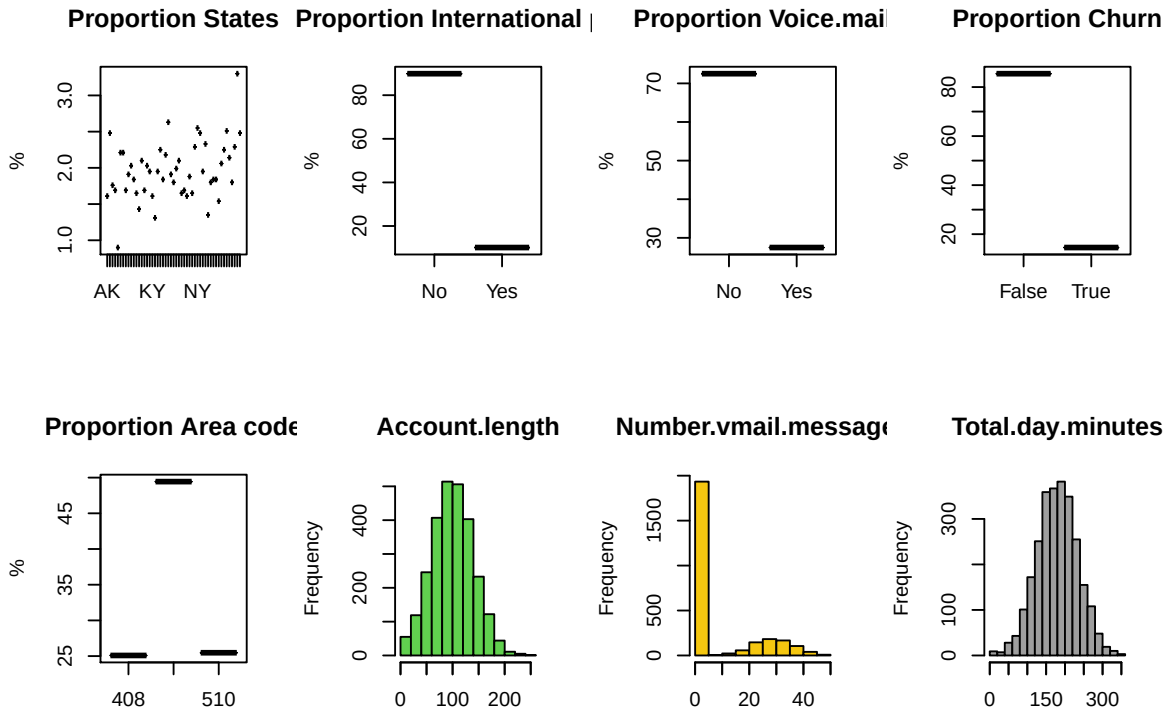
2.2. Présentation des observations

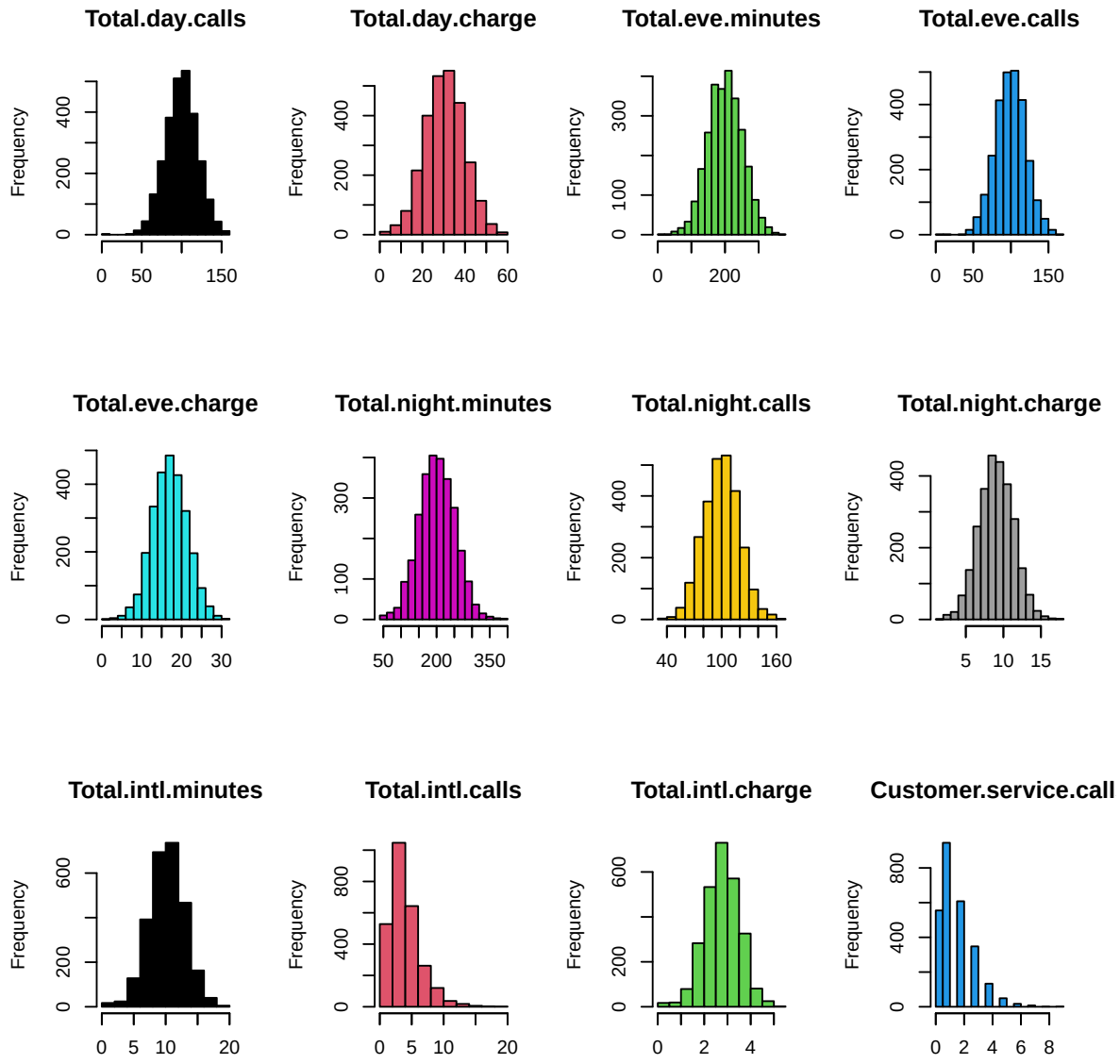
Ci-dessous les graphiques présente la répartition des valeurs pour chacune des variables du jeu de donnée. Les 5 premiers graphique présentes la fréquence en % des observations pour les variabes de type chaîne de caractères et catégorielle. Tandis que les variables de type continue sont présenté sous forme d'histogramme.

À partir des graphiques ci-dessous quelques interprétations intéressantes peuvent être faites :

— Le nombre d'obsertion fait sont similaire d'un États èa l'autre

— Le jeu de données est principalement composé d'observation sur des personnes n'ayant pas adhéré à un plan international et ni à un plan de messagerie vocal.
— La grande majorité des observation n'ont pas quitté leurs compagnie de télécomunication et la majorité des observations habitent le secteur 415.
— À partir des graphiques d'histogrammes, toutes les variables de type continue suivent une courbe normale à l'exeption des variables Customer.service.calls et total.intl,calls




Total.day.calls
Frequency
0 200 400
0 50 150
Total.day.charge
Frequency
0 200 400
0 20 40 60
Total.eve.minutes
Frequency
0 100 300
0 200
Total.eve.calls
Frequency
0 200 400
0 50 150
Total.eve.charge
Frequency
0 200 400
0 10 20 30
Total.night.minutes
Frequency
0 100 300
50 200 350
Total.night.calls
Frequency
0 200 400
40 100 160
Total.night.charge
Frequency
0 200 400
5 10 15
Total.intl.minutes
Frequency
0 200 600
0 5 10 20
Total.intl.calls
Frequency
0 400 800
0 5 10 20
Total.intl.charge
Frequency
0 200 600
0 2 4
Customer.service.call
Frequency
0 400 800
0 2 4 6 8

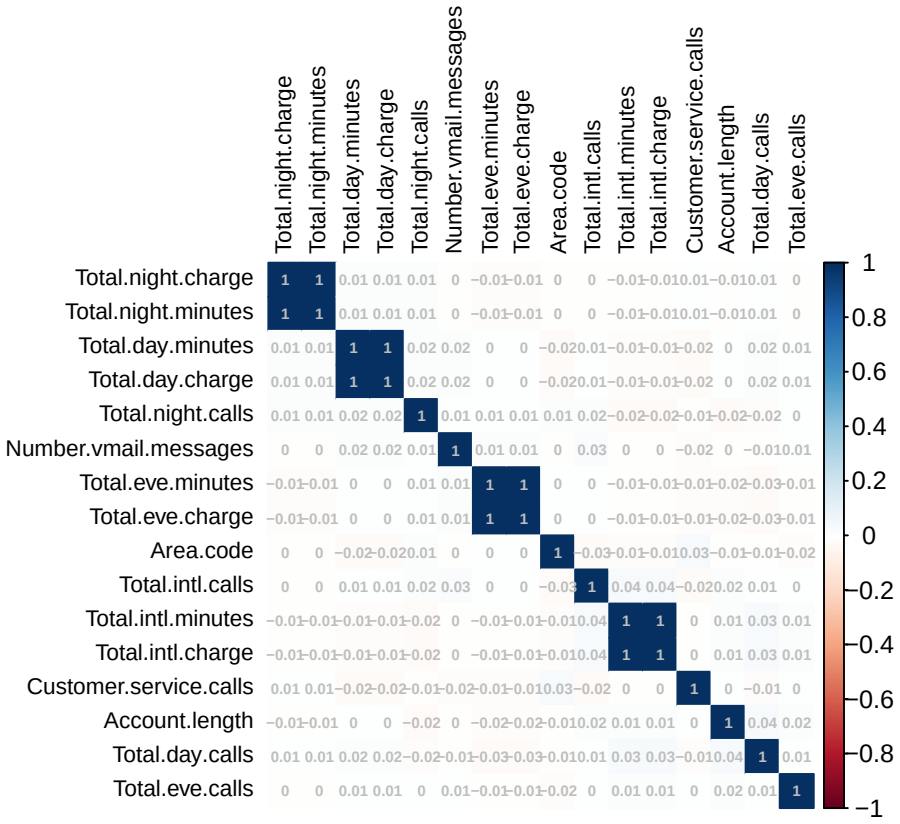
2.3. Réencodage ou traitment préparatoire des données

Tel que présenté dans le jeu de donnée et pour la suite des simulations plusiuers valeurs ont été d'abord réencodé:

- La variable "State" a été convertie en une variable Facteur
- Les variables "International.plan" , "Voice.mail.plan" et "Churn" en variable logique.
- Le dataset d'entraînement ne contient aucune valeur manquante.
- Le log normale des variables a été calculé sur les variables Customer.service.calls et total.intl,calls, car leurs courbent d'histogramme étaient décentré vers la gauche.

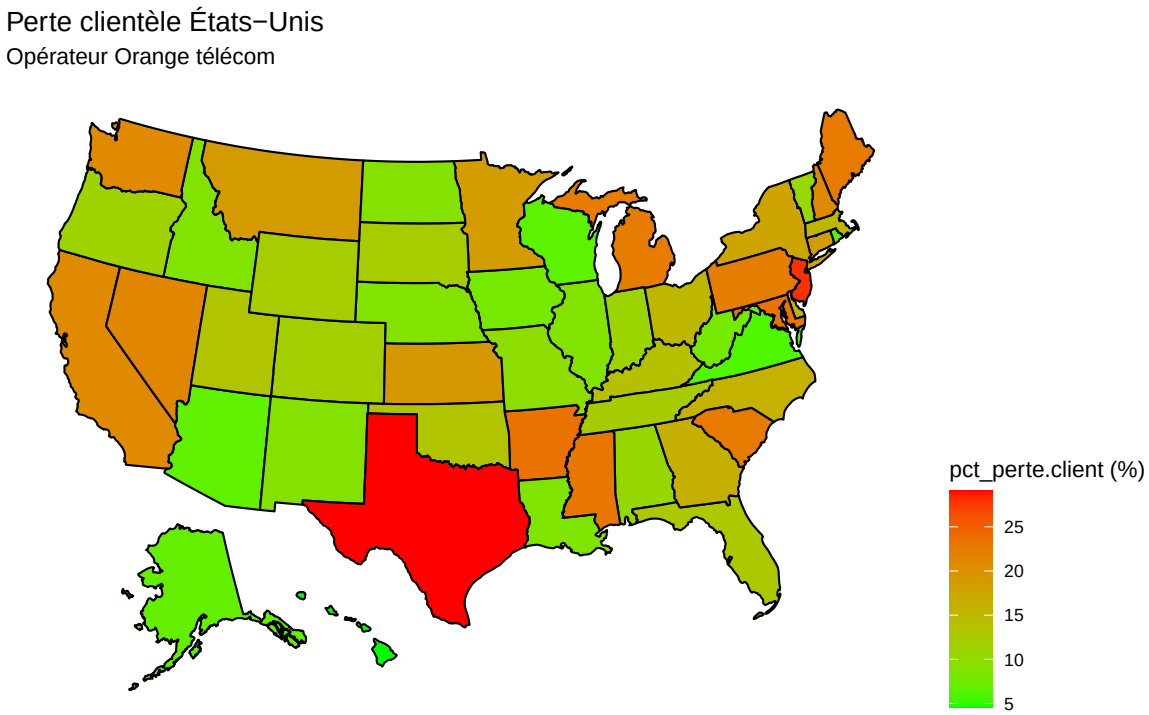
2.3.1 Présentation de la corrélation entre les variables

Le graphique ci-dessous présente la corrélation entre les variables. Les varaibles du dataset sont faiblement corrélées à l'exception des variables indiquant le nombre de muinutes consommés et les frais chargés associés, comme les varaibles : "Total.night charge" et "Total.night.minutes". Comme ces variables ont une corrélation parfaites, nous décidons de supprimer du dataset les variables "charge". Ce qui revient à supprimer 4 variables du dataset.



2.3.2 Présentation du taux de Churn selon l'État

Ci-desosus un map est présenté avec sn taux de perte de client. On constate que les états du Texas et New Jersey sont les états ayant perdus le plus de clientèle, avec un pourcentage de perte supérieur à 25%.



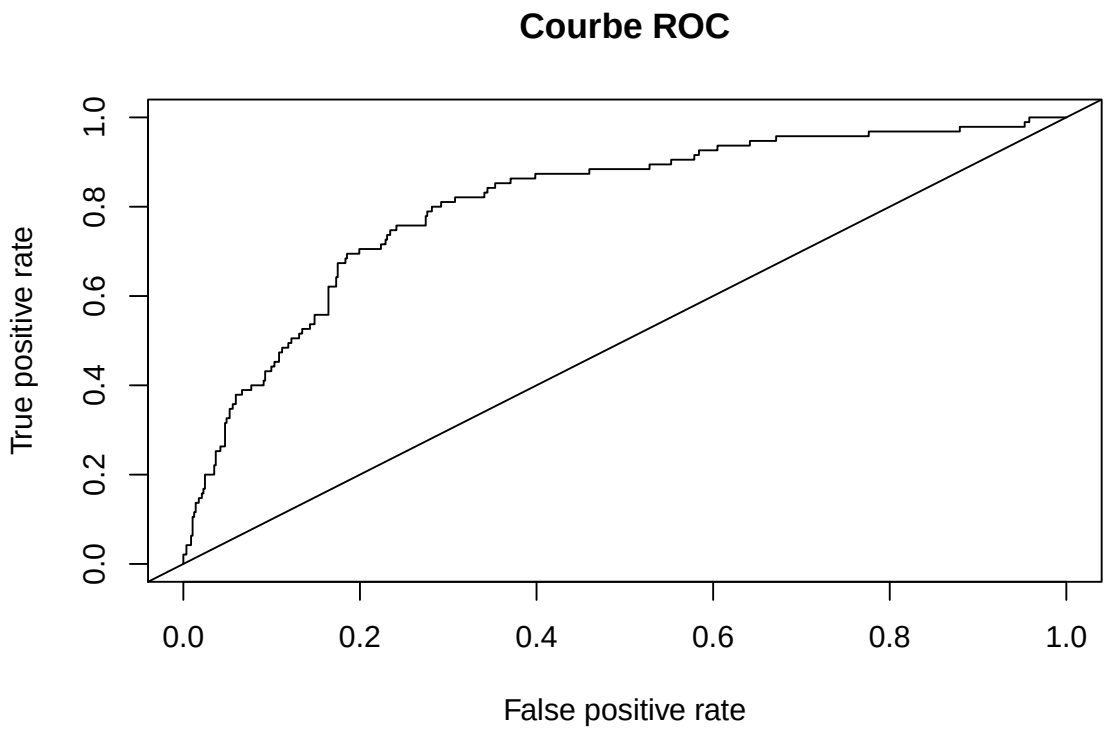
3. Modèle de simulation

Cette section présentera plusieurs simulations pour porédire la valeur du churn selon les variables de notre jeu de donnée. Les simulations qui serons testés sont les suivantes :

- Régression logistiqu
- Arbre de classification élagé
- forêt aléatoire approche bagging et boosting
- La méthode SVM

3.1. Régression logistique (validation croisé)

```
## [1] 13.94303
```



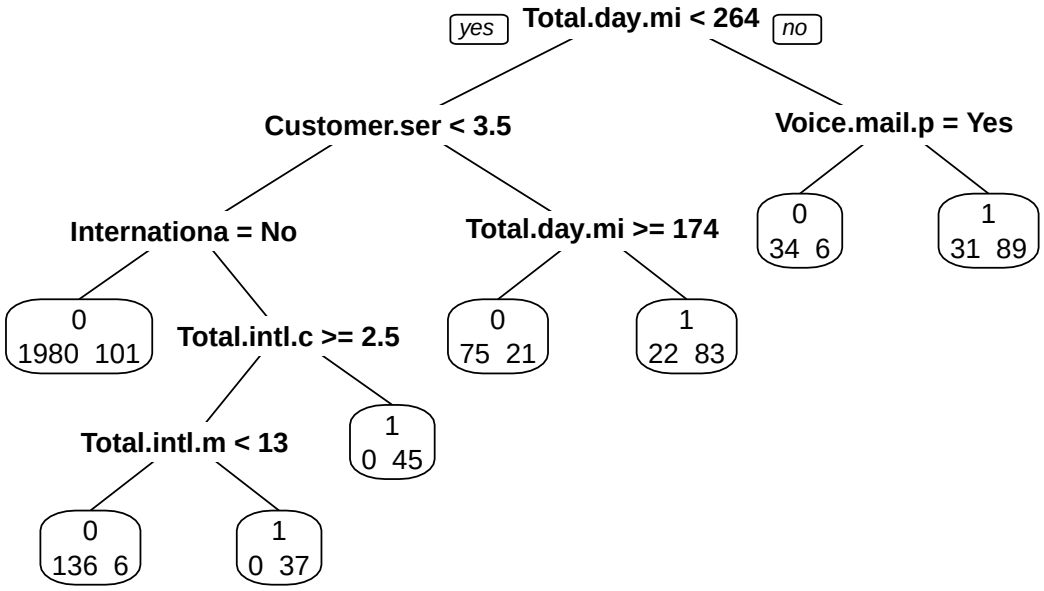
3.2. Arbre de classification élagé

```
train = train0
test = train0

train$Churn[train$Churn == "False"] = 0
train$Churn[train$Churn == "True"] = 1
train$Churn = as.numeric(train$Churn)

test$Churn[test$Churn == "False"] = 0
test$Churn[test$Churn == "True"] = 1
test$Churn = as.numeric(test$Churn)

set.seed(400)

# Création de l'arbre :
library(rpart.plot)
```

```
## Loading required package: rpart
```

```
mytree = rpart(Churn~., data=train , method = "class")
cp_optimal=mytree$cptable[which.min(mytree$cptable[,4]),1]
mytree_optimal = prune(mytree,cp=cp_optimal)
prp(mytree_optimal,extra=1,roundint=FALSE, main = "Arbre avec élagage")
```

Arbre avec élagage




```
mytable=table(test$Churn, predict(mytree_optimal,test, type="class"))
names(dimnames(mytable))= c("Observed", "Predicted")
M = mytable

a=M[1,1]
b=M[1,2]
c=M[2,1]
d=M[2,2]


# Taux de mauvaises classifications
((b+c)/(a+b+c+d))*100
```

```
## [1] 7.014254
```

```
# Taux de faux positifs
round(((b+c)/(a+b+c+d))*100, digit = 2)
```

```
## [1] 7.01
```

```
# Taux faux négatif
round((1-(d/(b+d)))*100, digit = 2)
```

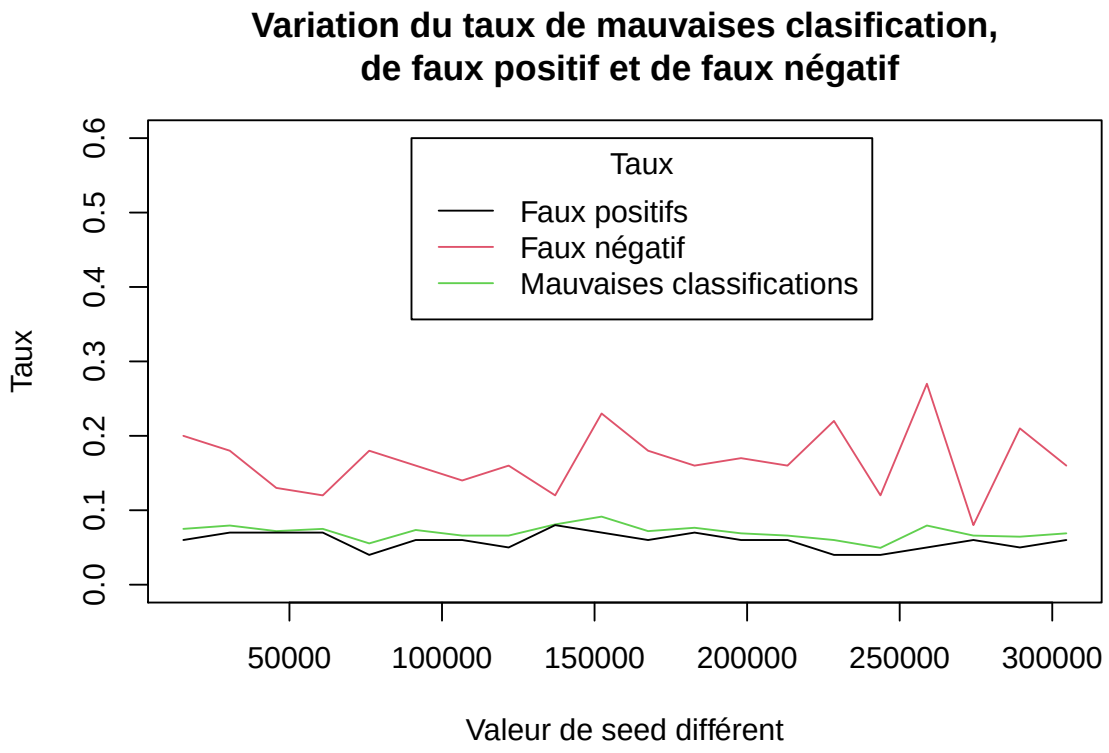
```
## [1] 17.26
```

```r
# Paramètre de complexité maximale
cp_optimal
```

```
## [1] 0.02835052
```

```r
# Boucle for

nb_boucle = 20

Taux_mauvaise_classificaiton=matrix(0,nb_boucle,1)
Taux_faux_positifs=matrix(0,nb_boucle,1)
Taux_faux_negatif=matrix(0,nb_boucle,1)
valeur_seed = matrix(0,nb_boucle,1)

par(mfrow=c(4,3))

for (i in 1:nb_boucle)
{
n=nrow(mydata)
set.seed(i*15231)
id.train=sample(1:n,size=nrow(train))
id.test=setdiff(1:n,id.train)

mydata.train= mydata[id.train,]
mydata.test = mydata[id.test,]

library(rpart.plot)
set.seed(i*15231)
valeur_seed[i]=i*15231

mytree = rpart(Churn~., data=mydata.train, method = "class")

cp_optimal=mytree$cptable[which.min(mytree$cptable[,4]),1]
mytree_optimal = prune(mytree,cp=cp_optimal)

#prp(mytree_optimal,extra=1,roundint=FALSE)

mytable=table(mydata.test$Churn, predict(mytree_optimal,mydata.test, type="class"))
names(dimnames(mytable))= c("Observed", "Predicted")
M = mytable

a=M[1,1]
b=M[1,2]
c=M[2,1]
d=M[2,2]

# Taux de mauvaises classifications
Taux_mauvaise_classificaiton[i]= ((b+c)/(a+b+c+d))
# Taux faux positifs
Taux_faux_positifs[i]= round((1-(a/(a+c))), digit = 2 )
# Taux faux négatif
Taux_faux_negatif[i] = round((1-(d/(b+d))), digit = 2)
```

```
}

par(mfrow=c(1,1))

plot(valeur_seed,Taux_mauvaise_classificaiton, type = "l", ylim= c(0,0.6), col = 3, ylab = "Taux
lines(valeur_seed,Taux_faux_positifs,col=1)
lines(valeur_seed,Taux_faux_negatif,col=2)
legend(90000, 0.6, legend=c("Faux positifs","Faux négatif","Mauvaises classifications"), col=1:3
```




```
# Taux de faux positif
## Maximum
round(max(Taux_faux_positifs), digit =2)
```

```
## [1] 0.08
```

```
## Minimum
round(min(Taux_faux_positifs), digit =2)
```

```
## [1] 0.04
```

```
# Taux de mauvaises classifications
## Maximum
round(max(Taux_mauvaise_classificaiton), digit =2 )
```

```
## [1] 0.09
```

```r
## Minimum
round(min(Taux_mauvaise_classificaiton) , digit = 2)
```

```
## [1] 0.05
```

```r
# Taux faux négatif
## Maximum
round(max(Taux_faux_negatif), digit = 2)
```

```
## [1] 0.27
```

```r
## Minimum
round(min(Taux_faux_negatif), digit = 2)
```

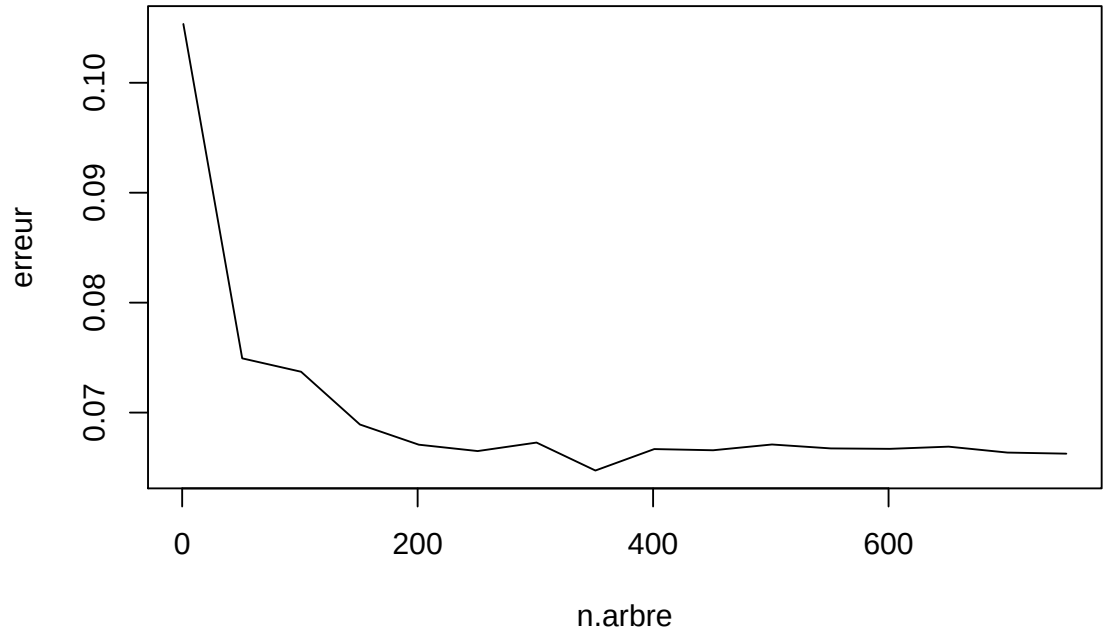
```
## [1] 0.08
```

3.3. forêt aléatoire approche bagging

```r
train = train0
test = train0

train$Churn[train$Churn == "False"] = 0
train$Churn[train$Churn == "True"] = 1
train$Churn = as.numeric(train$Churn)

test$Churn[test$Churn == "False"] = 0
test$Churn[test$Churn == "True"] = 1
test$Churn = as.numeric(test$Churn)


library(randomForest)
```

```
## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin

## The following object is masked from 'package:gridExtra':
##
##     combine
```

```
mydata$Churn <- as.factor(mydata$Churn)

set.seed(1234)
n.arbre=seq(1,800,by=50)
erreur=NULL
for (i in n.arbre)
{
  rf=randomForest(Churn~.,data=mydata,ntree=i,mtry=length(colnames(mydata))-1)
  erreur=c(erreur,sum(rf$err.rate[,1])/rf$ntree)
}
plot(n.arbre, erreur,type="l")
```




```
rf=randomForest(Churn~.,data=mydata,ntree=600,mtry=length(colnames(mydata))-1)
rf
```

```
##
## Call:
##  randomForest(formula = Churn ~ ., data = mydata, ntree = 600,      mtry = length(colnames(my
##                Type of random forest: classification
##                      Number of trees: 600
## No. of variables tried at each split: 15
##
##         OOB estimate of  error rate: 6.51%
## Confusion matrix:
##       FALSE TRUE class.error
## FALSE  2766   84  0.02947368
## TRUE    133  350  0.27536232
```

```
importance(rf,type=1)
```

```
##
## State
## Account.length
## Area.code
## International.plan
## Voice.mail.plan
## Number.vmail.messages
## Total.day.minutes
## Total.day.calls
## Total.eve.minutes
## Total.eve.calls
## Total.night.minutes
## Total.night.calls
## Total.intl.minutes
## Total.intl.calls
## Customer.service.calls
```

3.4. Boosting

```
library(randomForest)
library(adabag)
```

```
## Warning: package 'adabag' was built under R version 4.0.3
```

```
## Loading required package: caret
```

```
## Loading required package: foreach
```

```
## Warning: package 'foreach' was built under R version 4.0.3
```

```
## Loading required package: doParallel
```

```
## Loading required package: iterators
```

```
## Warning: package 'iterators' was built under R version 4.0.3
```

```
## Loading required package: parallel
```

```
train = train0
test = train0

train$Churn[train$Churn == "False"] = 0
train$Churn[train$Churn == "True"] = 1
train$Churn = as.numeric(train$Churn)

test$Churn[test$Churn == "False"] = 0
test$Churn[test$Churn == "True"] = 1
test$Churn = as.numeric(test$Churn)
```

```
train$Churn=as.factor(train$Churn)
test$Churn=as.factor(test$Churn)

set.seed(1234)

# boosting with trees of depth 10
myboost=boosting(Churn~., data=train, mfinal = 10, coeflearn = 'Freund', control=rpart.control(m
myboost$importance
```

```
##          Account.length              Area.code Customer.service.calls
##               4.4688932              0.3117404               7.4183039
##     International.plan  Number.vmail.messages                   State
##               5.0942122              0.4453988              34.5424419
##        Total.day.calls       Total.day.charge      Total.day.minutes
##               2.9945890              0.0000000              16.6550028
##        Total.eve.calls       Total.eve.charge      Total.eve.minutes
##               0.6688840              0.0000000               8.2465733
##       Total.intl.calls      Total.intl.charge     Total.intl.minutes
##               4.7813379              0.0000000               4.2572276
##      Total.night.calls     Total.night.charge    Total.night.minutes
##               3.3159080              0.0000000               4.2649704
##        Voice.mail.plan
##               2.5345167
```

```
pred=predict(myboost, newdata=test)

pred$error
```

```
## [1] 0.009002251
```

```
M=pred$confusion


#mytable=table(myboost$class,mydata$Churn)
# names(dimnames(mytable))= c("Predicted", "Observed")
# M = mytable
# M
# a=M[1,1]
# b=M[1,2]
# c=M[2,1]
# d=M[2,2]

# Taux de faux positifs
M[2,1]/(M[2,1]+M[1,1])
```

```
## [1] 0.0008779631
```

```
# Taux de faux négatifs
M[1,2]/(M[2,2]+M[1,2])
```

```
## [1] 0.05670103
```

3.5. SVM

4. Résultats : présentation des résultats sous forme de tableaux et figures (ne mettez pas de sortie R)

5. Conclusion/discussion : conclusion générale, limites de votre étude, qu'avez vous appris ?