Projet - Analyse de Données

Projet KikiCkisenVa - Prédiction

Recuperation des donnees

```
data_train <- read.csv2("spreadsheets/data_train.csv", sep = ",")
data_train <- na.omit(data_train)
data_train <- fact.data(data_train)
dim(data_train)
```

```
## [1] 784  32
```

```
head(data_train)
```

```
##   Age Attrition BusinessTravel DailyRate             Department
## 1  50        No  Travel_Rarely      1126 Research & Development
## 2  36        No  Travel_Rarely       216 Research & Development
## 3  21       Yes  Travel_Rarely       337                  Sales
## 4  52        No  Travel_Rarely       994 Research & Development
## 5  33       Yes  Travel_Rarely      1277 Research & Development
## 6  47        No  Travel_Rarely      1001 Research & Development
##   DistanceFromHome Education EducationField EmployeeNumber
## 1                1         2        Medical            997
## 2                6         2        Medical            178
## 3                7         1      Marketing           1780
## 4                7         4  Life Sciences           1118
## 5               15         1        Medical            582
## 6                4         3  Life Sciences           1827
##   EnvironmentSatisfaction Gender HourlyRate JobInvolvement JobLevel
## 1                       4   Male         66              3        4
## 2                       2   Male         84              3        2
## 3                       2   Male         31              3        1
## 4                       2   Male         87              3        3
## 5                       2   Male         56              3        3
## 6                       3 Female         92              2        3
##                     JobRole JobSatisfaction MaritalStatus MonthlyIncome
## 1         Research Director               4      Divorced         17399
## 2    Manufacturing Director               2      Divorced          4941
## 3      Sales Representative               2        Single          2679
## 4 Healthcare Representative               2        Single         10445
## 5                   Manager               3       Married         13610
## 6    Manufacturing Director               2      Divorced         10333
##   MonthlyRate NumCompaniesWorked OverTime PercentSalaryHike PerformanceRating
## 1        6615                  9       No                22                 4
## 2        2819                  6       No                20                 4
## 3        4567                  1       No                13                 3
## 4       15322                  7       No                19                 3
## 5       24619                  7      Yes                12                 3
## 6       19271                  8      Yes                12                 3
##   RelationshipSatisfaction StockOptionLevel TotalWorkingYears
## 1                        3                1                32
## 2                        4                2                 7
## 3                        2                0                 1
## 4                        4                0                18
## 5                        4                0                15
## 6                        3                1                28
##   TrainingTimesLastYear WorkLifeBalance YearsAtCompany YearsInCurrentRole
## 1                     1               2              5                  4
## 2                     0               3              3                  2
## 3                     3               3              1                  0
## 4                     4               3              8                  6
## 5                     2               4              7                  6
## 6                     4               3             22                 11
##   YearsSinceLastPromotion YearsWithCurrManager
## 1                       1                    3
## 2                       0                    1
## 3                       1                    0
## 4                       4                    0
## 5                       7                    7
## 6                      14                   10
```

```
data_test <- read.csv2("spreadsheets/data_test.csv", sep = ",")
data_test <- na.omit(data_test)
data_test <- fact.data(data_test)
dim(data_test)
```

```
## [1] 332  31
```

```
head(data_test)
```

```
##   Age BusinessTravel DailyRate             Department DistanceFromHome
## 1  53  Travel_Rarely      1084 Research & Development               13
## 2  24  Travel_Rarely       240        Human Resources               22
## 3  45  Travel_Rarely      1339 Research & Development                7
## 4  34  Travel_Rarely       204                  Sales               14
## 5  39  Travel_Rarely      1431 Research & Development                1
## 6  45     Non-Travel      1052                  Sales                6
##   Education   EducationField EmployeeNumber EnvironmentSatisfaction Gender
## 1         2          Medical            250                       4 Female
## 2         1  Human Resources           1714                       4   Male
## 3         3    Life Sciences             86                       2   Male
## 4         3 Technical Degree            666                       3 Female
## 5         4          Medical            332                       3 Female
## 6         3          Medical            302                       4 Female
##   HourlyRate JobInvolvement JobLevel                JobRole JobSatisfaction
## 1         57              4        2 Manufacturing Director               1
## 2         58              1        1        Human Resources               3
## 3         59              3        3     Research Scientist               1
## 4         31              3        1   Sales Representative               3
## 5         96              3        1  Laboratory Technician               3
## 6         57              2        3        Sales Executive               4
##   MaritalStatus MonthlyIncome MonthlyRate NumCompaniesWorked OverTime
## 1      Divorced          4450       26250                  1       No
## 2       Married          1555       11585                  1       No
## 3      Divorced          9724       18787                  2       No
## 4      Divorced          2579        2912                  1      Yes
## 5                        2232       15417                  7       No
## 6        Single          8865       16840                  6       No
##   PercentSalaryHike PerformanceRating RelationshipSatisfaction StockOptionLevel
## 1                11                 3                        3                2
## 2                11                 3                        3                1
## 3                17                 3                        3                1
## 4                18                 3                        4                2
## 5                14                 3                        3                3
## 6                12                 3                        4                0
##   TotalWorkingYears TrainingTimesLastYear WorkLifeBalance YearsAtCompany
## 1                 5                     3               3              4
## 2                 1                     2               3              1
## 3                25                     2               3              1
## 4                 8                     3               3              8
## 5                 7                     1               3              3
## 6                23                     2               3             19
##   YearsInCurrentRole YearsSinceLastPromotion YearsWithCurrManager
## 1                  2                       1                    3
## 2                  0                       0                    0
## 3                  0                       0                    0
## 4                  2                       0                    6
## 5                  2                       1                    2
## 6                  7                      12                    8
```

Recupération des variables numériques

```
data_train_num <- data_train[, unlist(lapply(data_train, is.numeric))]
data_train_num[16] <- data_train["Attrition"]
dim(data_train_num)
```

```
## [1] 784  16
```

```
head(data_train_num)
```

```
##   Age DailyRate DistanceFromHome EmployeeNumber HourlyRate MonthlyIncome
## 1  50      1126                1            997         66         17399
## 2  36       216                6            178         84          4941
## 3  21       337                7           1780         31          2679
## 4  52       994                7           1118         87         10445
## 5  33      1277               15            582         56         13610
## 6  47      1001                4           1827         92         10333
##   MonthlyRate NumCompaniesWorked PercentSalaryHike TotalWorkingYears
## 1        6615                  9                22                32
## 2        2819                  6                20                 7
## 3        4567                  1                13                 1
## 4       15322                  7                19                18
## 5       24619                  7                12                15
## 6       19271                  8                12                28
##   TrainingTimesLastYear YearsAtCompany YearsInCurrentRole
## 1                     1              5                  4
## 2                     0              3                  2
## 3                     3              1                  0
## 4                     4              8                  6
## 5                     2              7                  6
## 6                     4             22                 11
##   YearsSinceLastPromotion YearsWithCurrManager Attrition
## 1                       1                    3        No
## 2                       0                    1        No
## 3                       1                    0       Yes
## 4                       4                    0        No
## 5                       7                    7       Yes
## 6                      14                   10        No
```

```
data_test_num <- data_test[, unlist(lapply(data_test, is.numeric))]
dim(data_test_num)
```

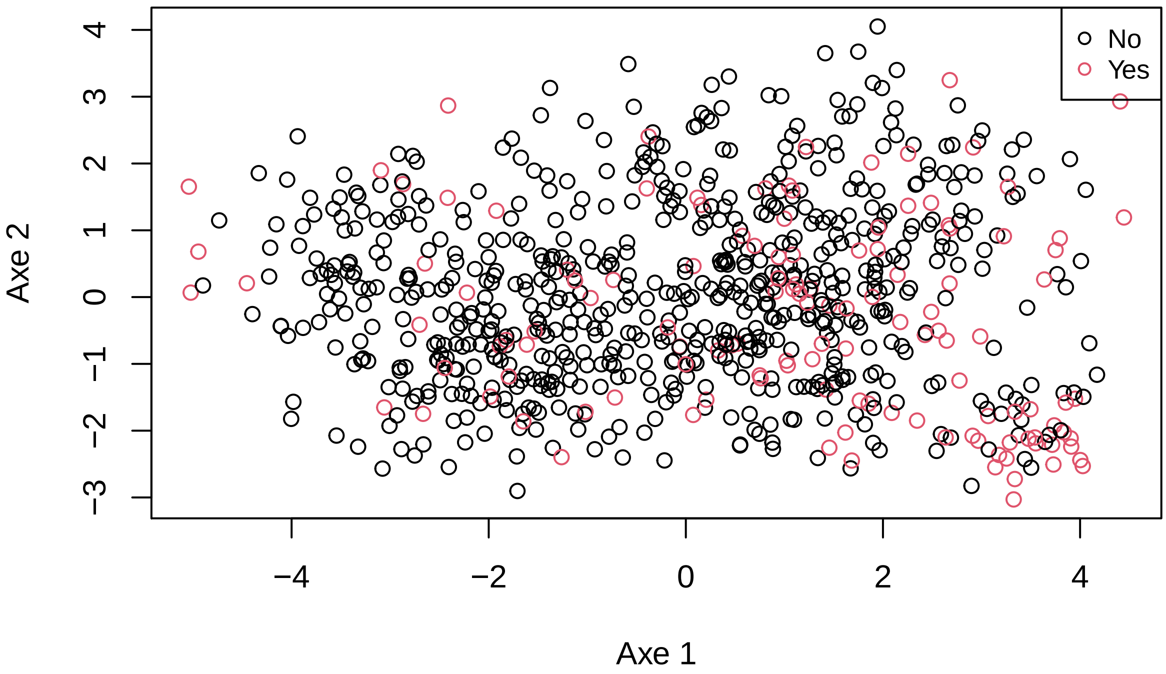
```
## [1] 332  15
```

```
head(data_test_num)
```

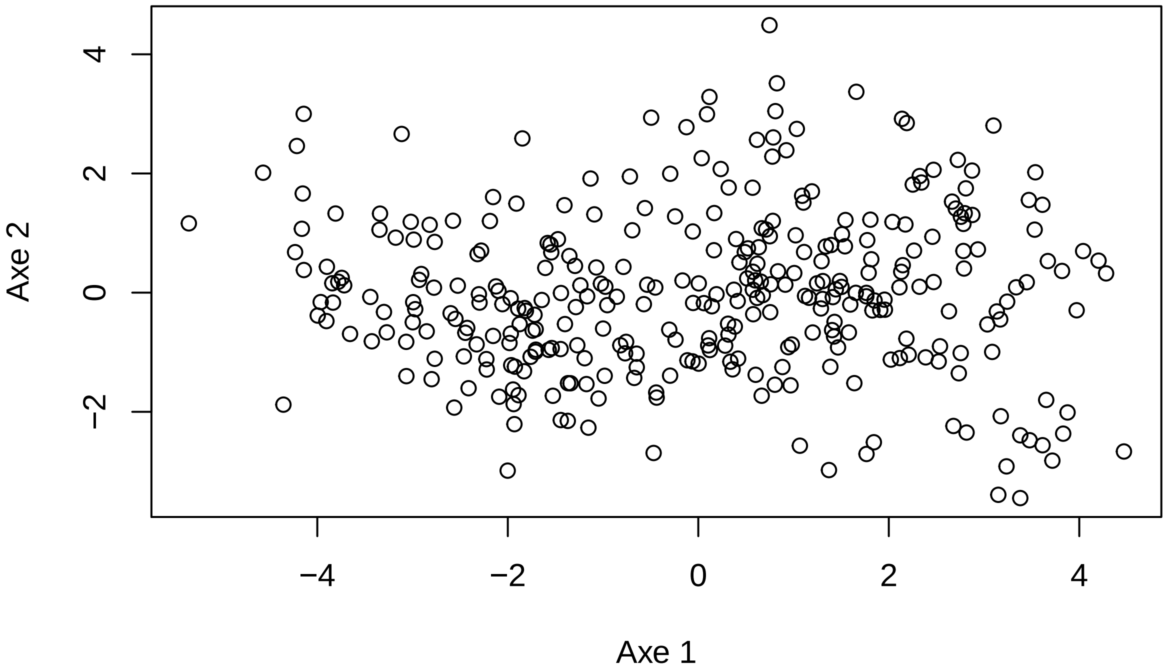
```
##   Age DailyRate DistanceFromHome EmployeeNumber HourlyRate MonthlyIncome
## 1  53      1084               13            250         57          4450
## 2  24       240               22           1714         58          1555
## 3  45      1339                7             86         59          9724
## 4  34       204               14            666         31          2579
## 5  39      1431                1            332         96          2232
## 6  45      1052                6            302         57          8865
##   MonthlyRate NumCompaniesWorked PercentSalaryHike TotalWorkingYears
## 1       26250                  1                11                 5
## 2       11585                  1                11                 1
## 3       18787                  2                17                25
## 4        2912                  1                18                 8
## 5       15417                  7                14                 7
## 6       16840                  6                12                23
##   TrainingTimesLastYear YearsAtCompany YearsInCurrentRole
## 1                     3              4                  2
## 2                     2              1                  0
## 3                     2              1                  0
## 4                     3              8                  2
## 5                     1              3                  2
## 6                     2             19                  7
##   YearsSinceLastPromotion YearsWithCurrManager
## 1                       1                    3
## 2                       0                    0
## 3                     0                     0
## 4                     0                     6
## 5                     1                     2
## 6                    12                     8
```

Récupération des coordonnées

```
library(FactoMineR)
data_train_log <- log(data_train_num[-16])
data_train_log[data_train_log == -Inf] <- 0
data_train_log <- t(scale(t(data_train_log)))
data_train_log <- as.data.frame(data_train_log)
data_train_log[16] <- data_train["Attrition"]
coord_data_train <- PCA(data_train_log, scale.unit = TRUE, graph = FALSE, quali.sup = 16)$ind$coord[,1:2
plot(coord_data_train[,1], coord_data_train[,2], col = data_train$Attrition, xlab = "Axe 1", ylab = "Axe
legend('topright', legend = levels(data_train$Attrition), col = 1:2, cex = 0.8, pch = 1)
```



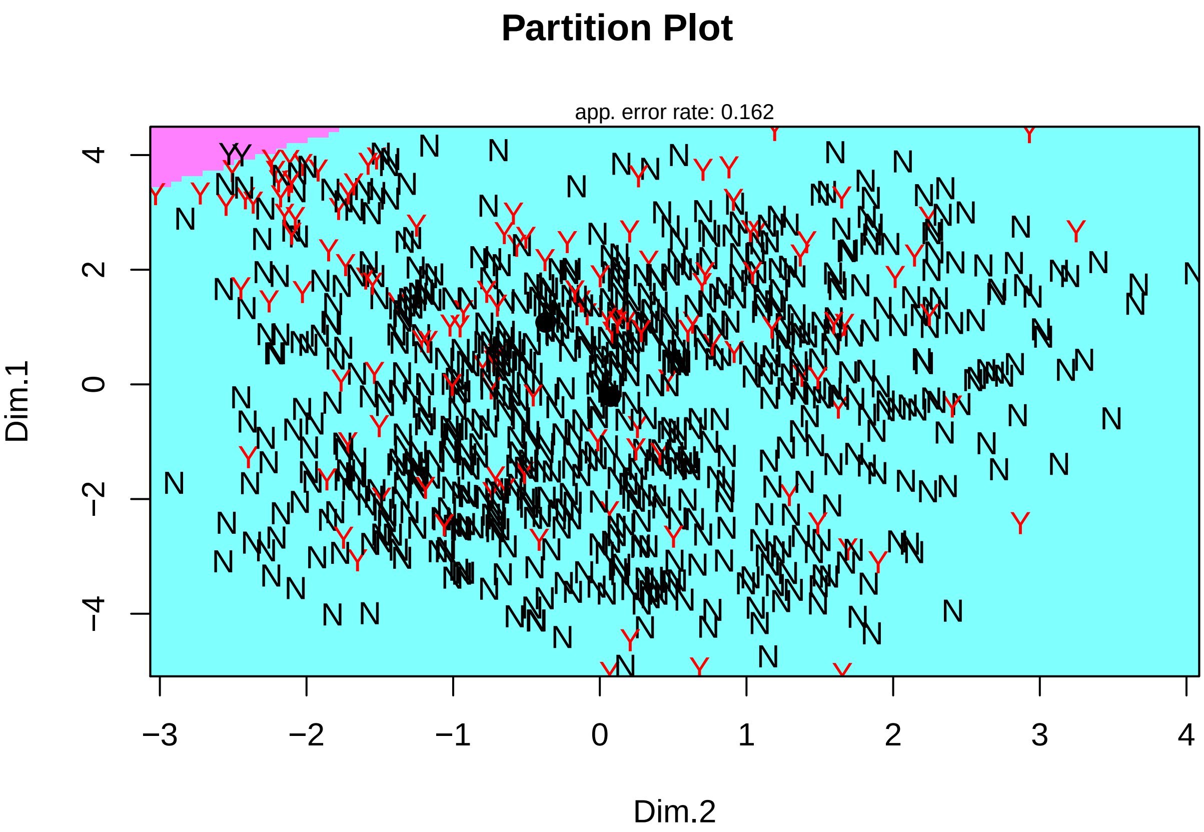
```
data_test_log <- log(data_test_num)
data_test_log[data_test_log == -Inf] <- 0
data_test_log <- t(scale(t(data_test_log)))
data_test_log <- as.data.frame(data_test_log)
coord_data_test <- PCA(data_test_log, scale.unit = TRUE, graph = FALSE)$ind$coord[,1:2]
plot(coord_data_test[,1], coord_data_test[,2], xlab = "Axe 1", ylab = "Axe 2")
```



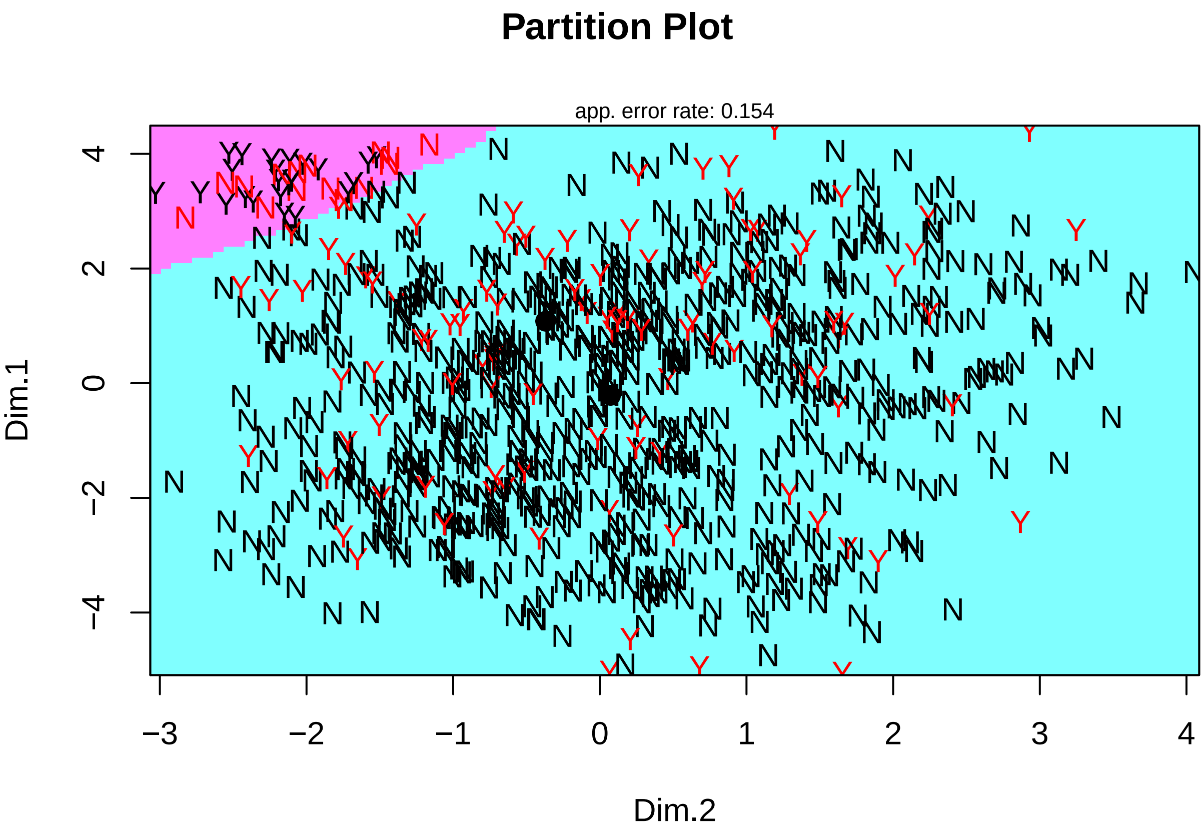
Classification

LDA - QDA

```
library(klaR)
partimat(coord_data_train, grouping = data_train_num$Attrition, method = "lda")
```

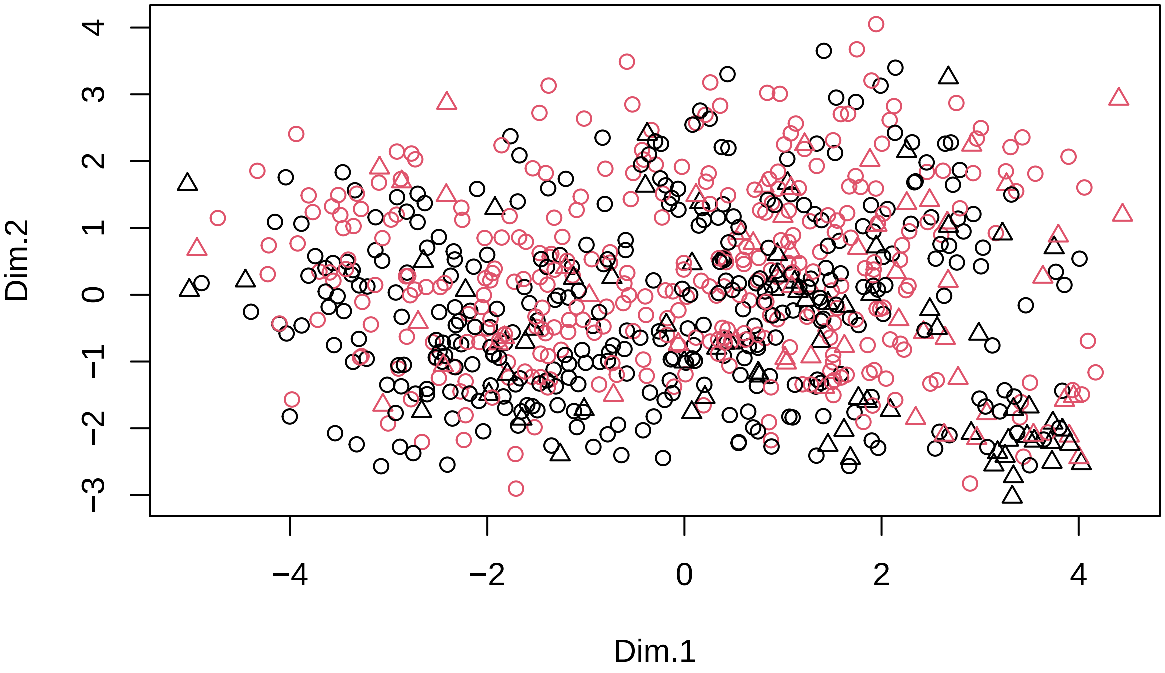


```
partimat(coord_data_train, grouping = data_train_num$Attrition, method = "qda")
```

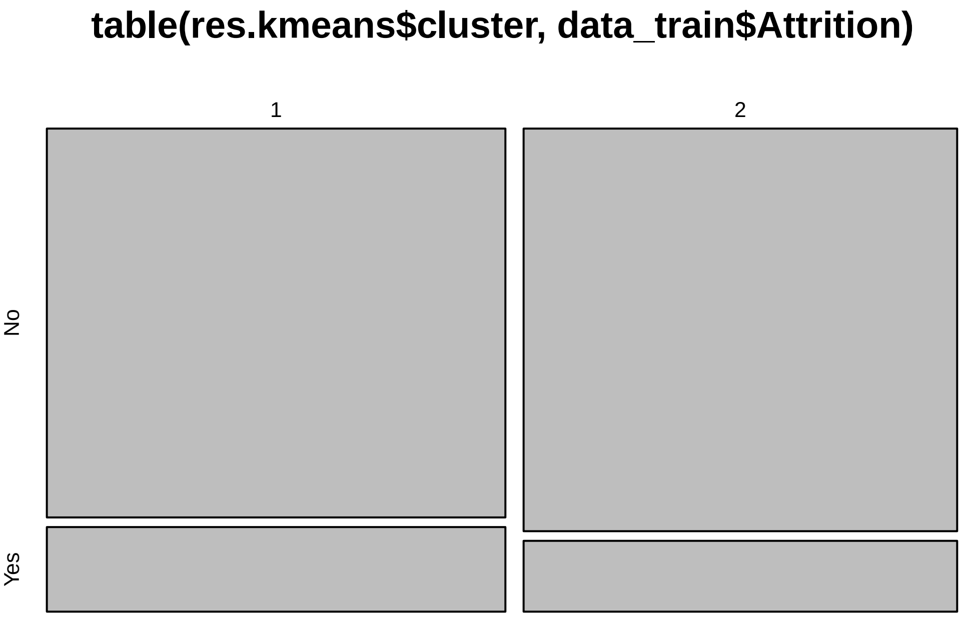


KMeans

```
res.kmeans <- kmeans(data_train_num[-16], centers = 2, nstart = 1000)
plot(coord_data_train, col = res.kmeans$cluster, pch = as.numeric(data_train$Attrition))
```

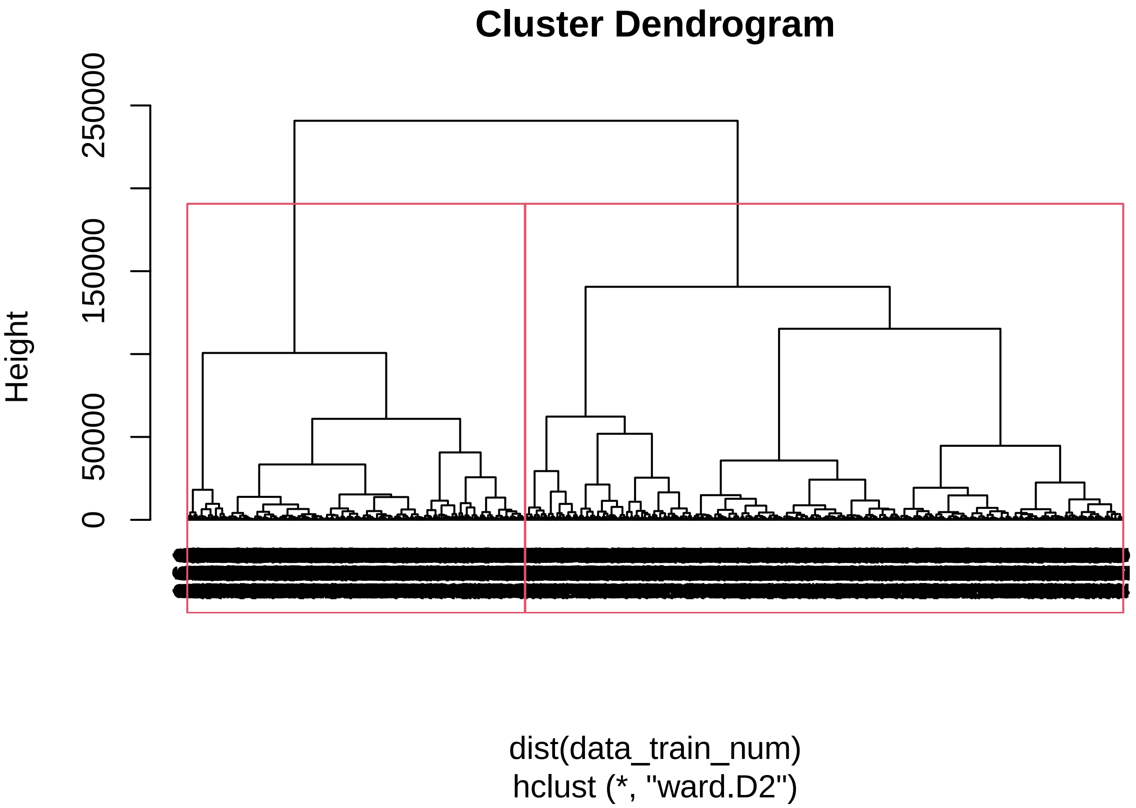


```
plot(table(res.kmeans$cluster, data_train$Attrition))
```

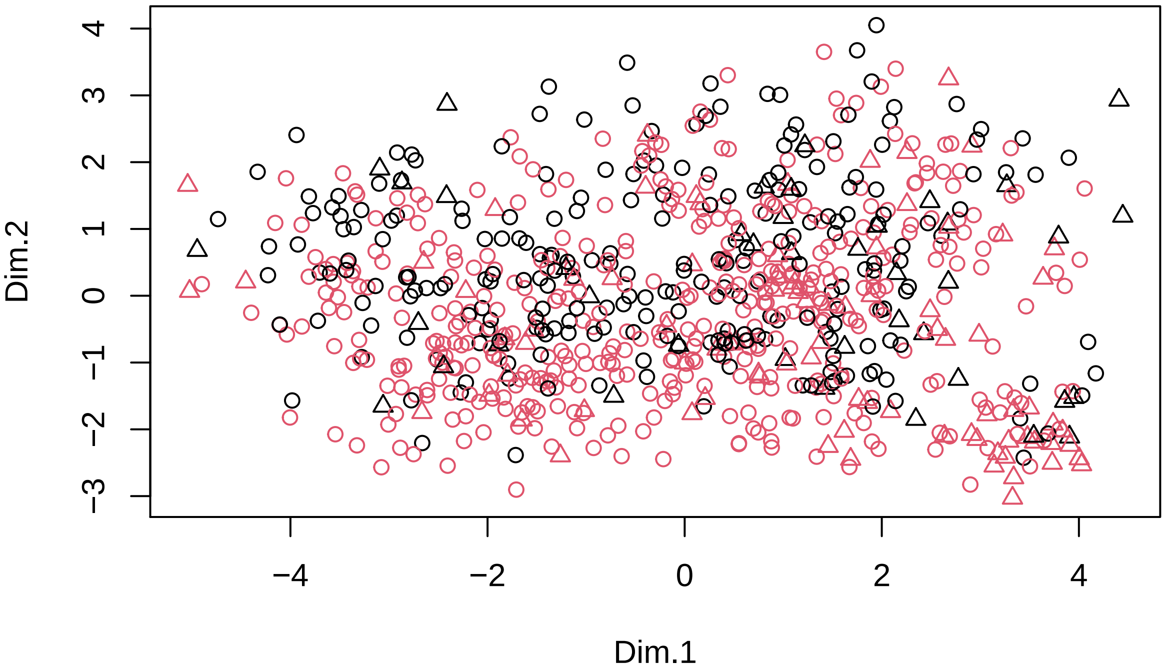


CAH

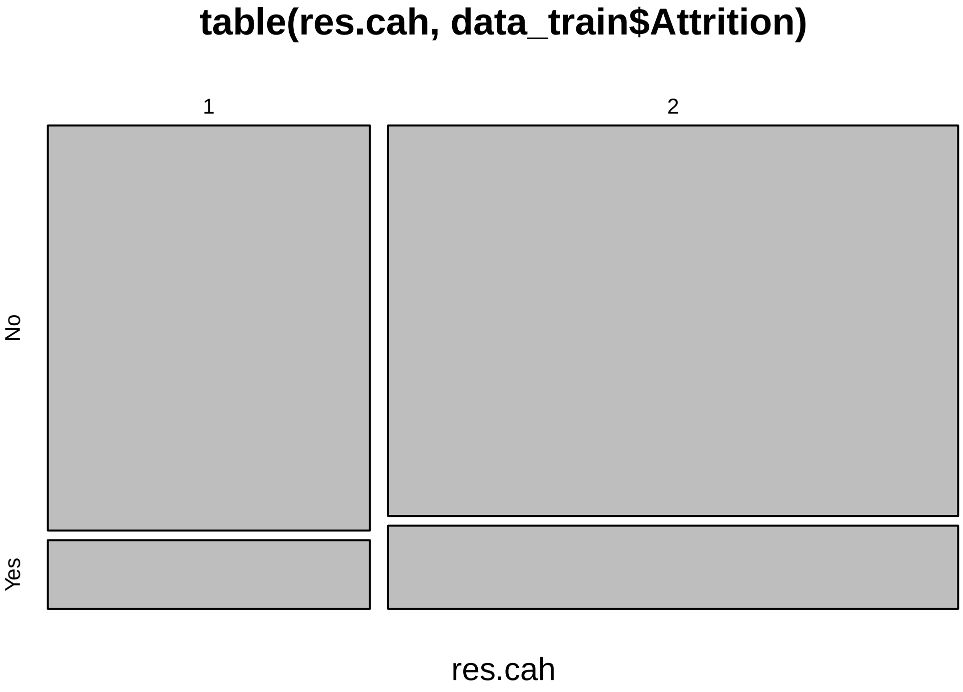
```
## Modèle
cah.ward <- hclust(dist(data_train_num), method = "ward.D2")
## Selection de 2 cluster (choix binaire)
plot(cah.ward, hang = -1)
rect.hclust(cah.ward, 2)
```



```
res.cah <- cutree(cah.ward, 2)

plot(coord_data_train, col = res.cah, pch = as.numeric(data_train$Attrition))
```



```
plot(table(res.cah, data_train$Attrition))
```



Équilibrage de la répartition des données

```
res.qda = qda(data_train_num[-16], grouping = data_train_num$Attrition)
res.qda
```

```
## Call:
```

```
## qda(data_train_num[-16], grouping = data_train_num$Attrition)
##
## Prior probabilities of groups:
##        No       Yes
## 0.8354592 0.1645408
##
## Group means:
##          Age DailyRate DistanceFromHome EmployeeNumber HourlyRate MonthlyIncome
## No  38.77099  792.5939         9.503817       1023.669   66.86260      7162.046
## Yes 34.42636  756.1938        10.449612       1039.922   67.96899      4947.279
##     MonthlyRate NumCompaniesWorked PercentSalaryHike TotalWorkingYears
## No     14124.12           2.708397          15.32672         12.670229
## Yes    14534.25           3.038760          15.16279          8.387597
##     TrainingTimesLastYear YearsAtCompany YearsInCurrentRole
## No               2.781679       7.767939           4.687023
## Yes              2.604651       5.240310           2.798450
##     YearsSinceLastPromotion YearsWithCurrManager
## No                 2.343511             4.465649
## Yes                1.837209             2.821705
```

```
pred.qda = predict(res.qda, data_train_num[-16])$class
table(data_train_num$Attrition, pred.qda)
```

```
##      pred.qda
##        No Yes
##   No  553 102
##   Yes  59  70
```

Sur les Yes prédits on a plus d'erreurs que de cas juste alors que ce n'est pas le cas avec les prédiction sur No.

```
library(DMwR)
table(data_train_num$Attrition)
```

```
##
##  No Yes
## 655 129
```

```
data_train_bal <- SMOTE(Attrition ~ ., data_train_num)
table(data_train_bal$Attrition)
```

```
##
##  No Yes
## 516 387
```

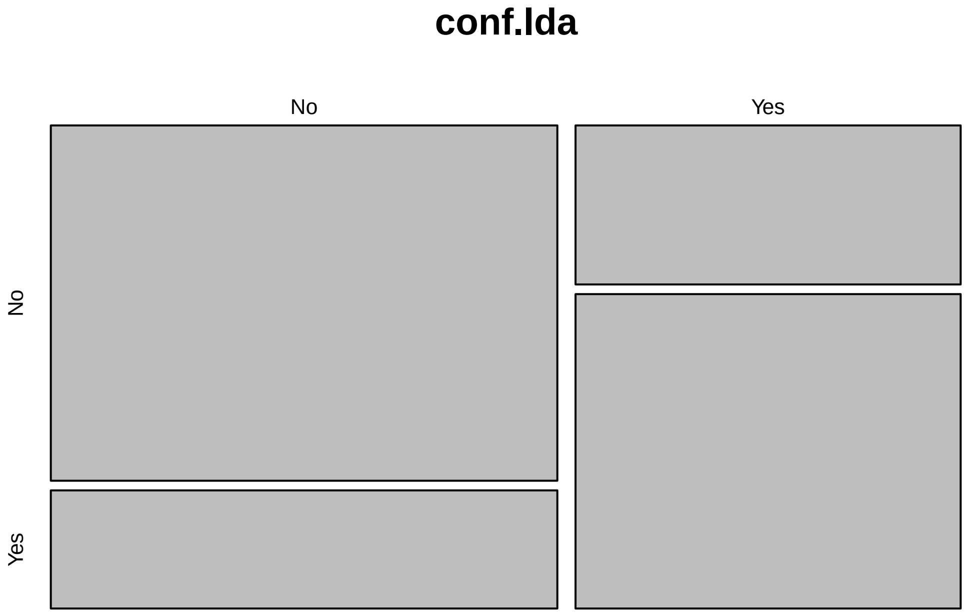
Détermination du meilleur modèle de Prédiction

LDA - QDA

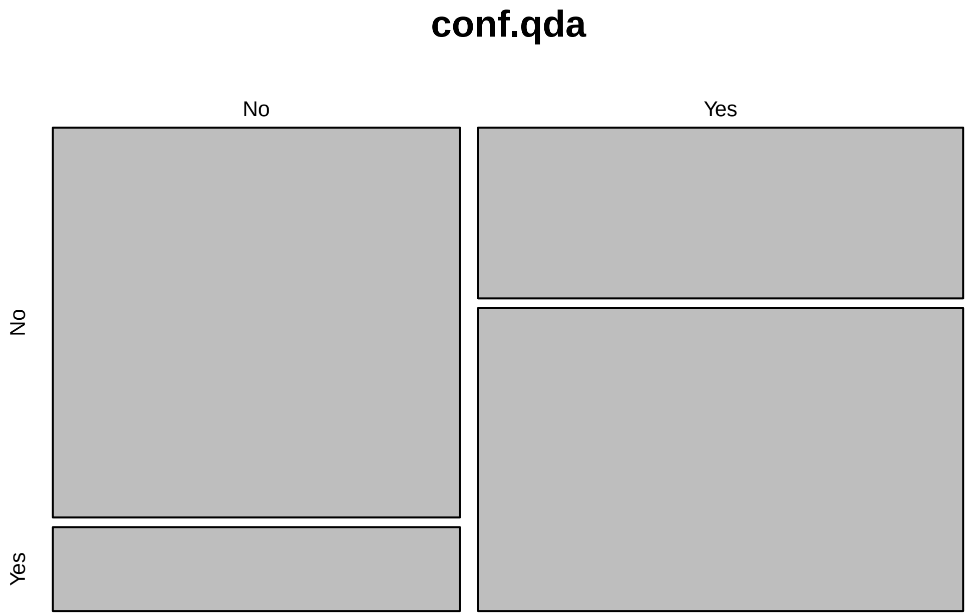
```r
library(MASS)

## Modèle
res.lda <- lda(data_train_bal[-16], grouping = data_train_bal$Attrition)
res.qda <- qda(data_train_bal[-16], grouping = data_train_bal$Attrition)
```

```r
## Prédiction
pred.lda <- predict(res.lda, newdata = data_train_bal[-16])
pred.qda <- predict(res.qda, newdata = data_train_bal[-16])
```

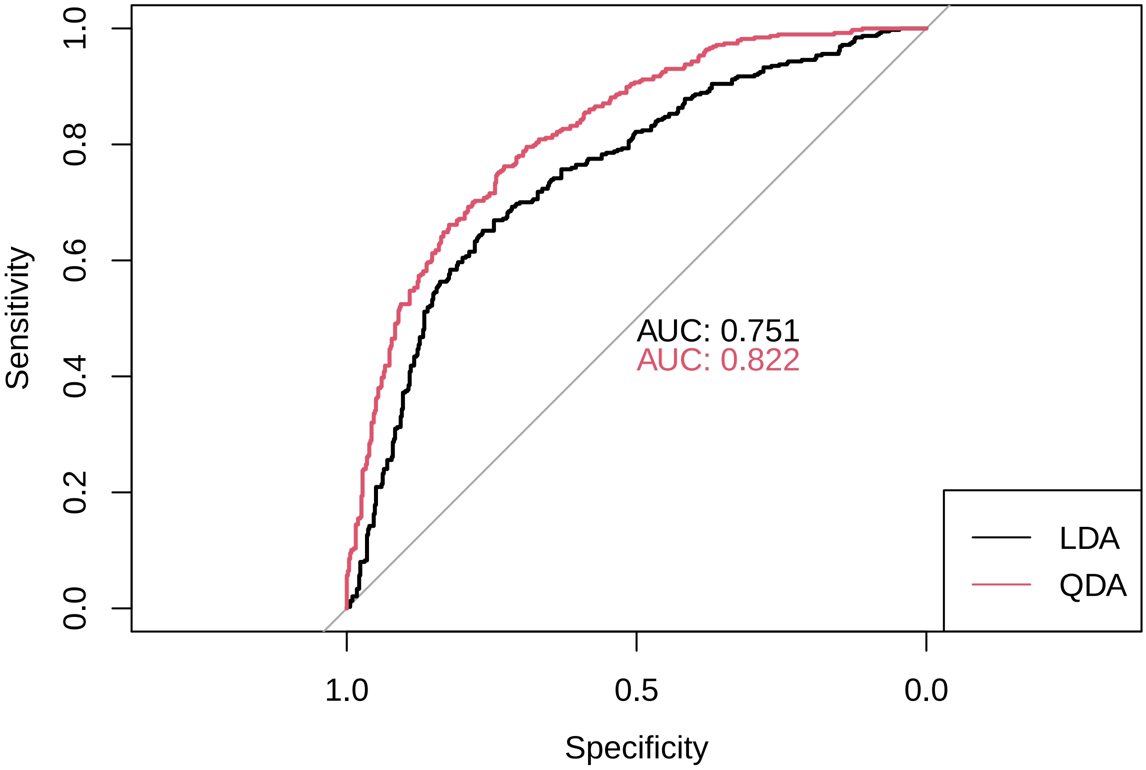
```r
## Table de confusion
conf.lda <- table(pred.lda$class, data_train_bal$Attrition)
accuracy.lda <- (conf.lda[1,1] + conf.lda[2,2]) / sum(conf.lda)
plot(conf.lda)
```



```r
conf.qda <- table(pred.qda$class, data_train_bal$Attrition)
accuracy.qda <- (conf.qda[1,1] + conf.qda[2,2]) / sum(conf.qda)
plot(conf.qda)
```



```
## courbe ROC
library(pROC)
ROC.lda <- roc(data_train_bal$Attrition, pred.lda$posterior[,2])
ROC.qda <- roc(data_train_bal$Attrition, pred.qda$posterior[,2])
plot(ROC.lda, print.auc=TRUE, print.auc.y = 0.5, col = 1)
plot(ROC.qda, add = TRUE, print.auc=TRUE,  print.auc.y = 0.45, col = 2)
legend("bottomright", lwd = 1, col = 1:2, c("LDA", "QDA"))
```



LDA avec selection de modèle

```
library(klaR)

## Modèle
stepwise.lda = stepclass(data_train_bal[-16], grouping = data_train_bal$Attrition, method = "lda", direc
```

```
## correctness rate: 0.69093;  starting variables (15): Age, DailyRate, DistanceFromHome, EmployeeNumber
## correctness rate: 0.70204;  out: "TotalWorkingYears";  variables (14): Age, DailyRate, DistanceFromHo
##
##  hr.elapsed min.elapsed sec.elapsed
##       0.000       0.000       1.992
```

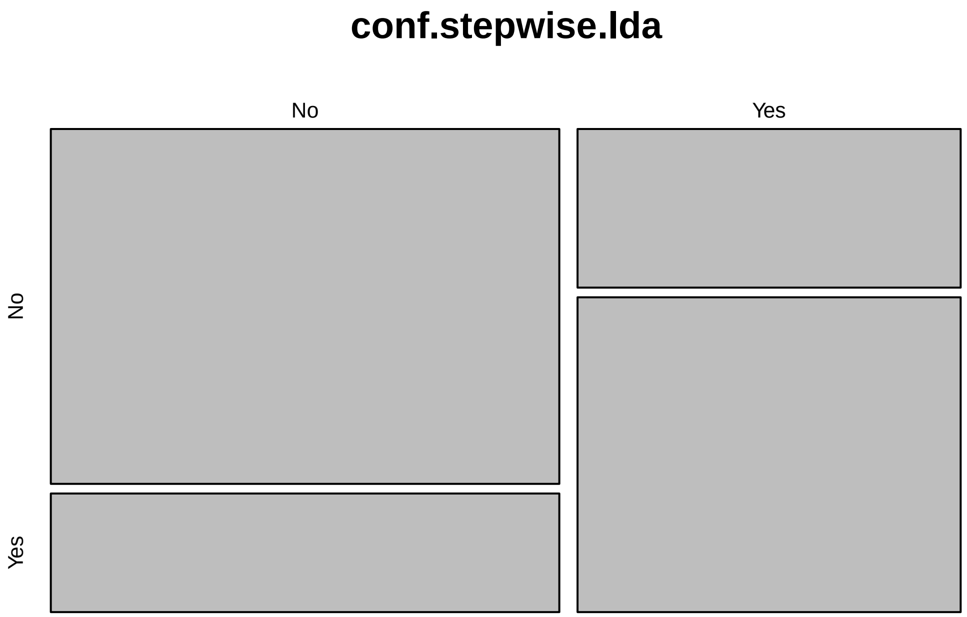
```
stepwise.lda
```

```
## method      : lda
## final model : data_train_bal$Attrition ~ Age + DailyRate + DistanceFromHome +
##     EmployeeNumber + HourlyRate + MonthlyIncome + MonthlyRate +
##     NumCompaniesWorked + PercentSalaryHike + TrainingTimesLastYear +
##     YearsAtCompany + YearsInCurrentRole + YearsSinceLastPromotion +
##     YearsWithCurrManager
## <environment: 0x7fe68091f008>
##
## correctness rate = 0.702
```

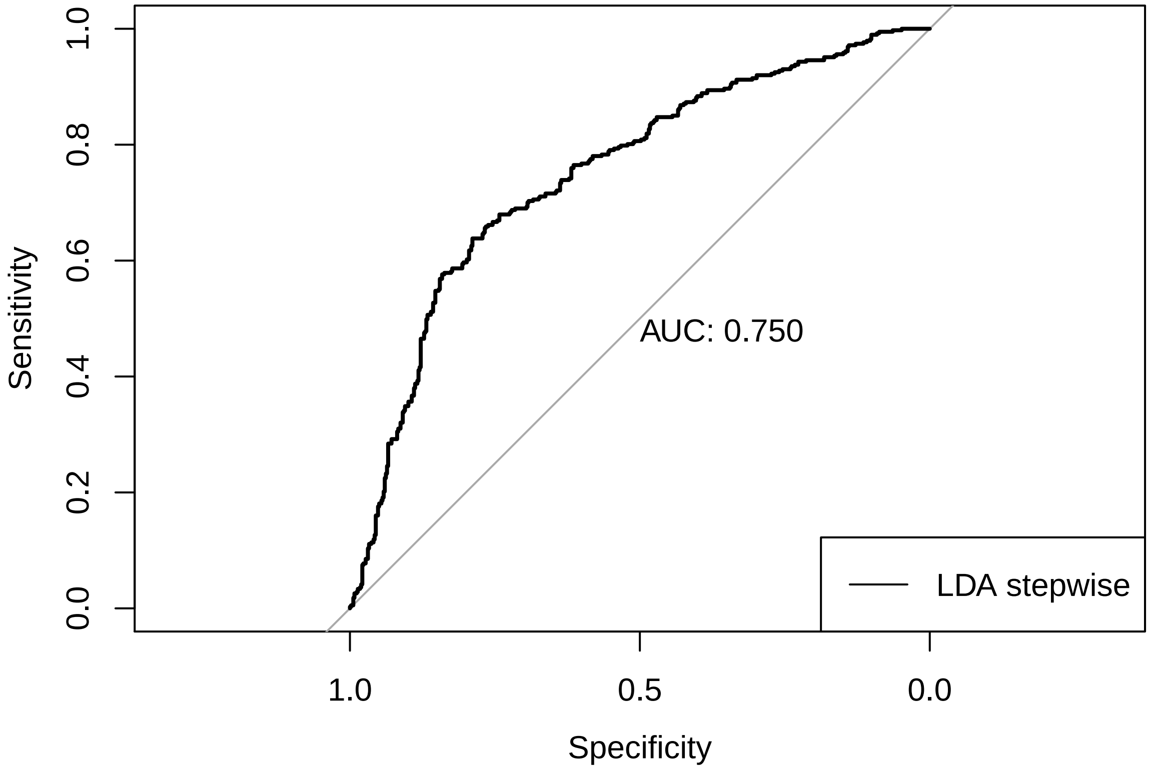
```
res.stepwise.lda = lda(stepwise.lda$formula, data = data_train_bal[-16])
```

```
## Prédiction
pred.stepwise.lda <- predict(res.stepwise.lda, newdata = data_train_bal[-16])
```

```
## Table de confusion
conf.stepwise.lda <- table(pred.stepwise.lda$class, data_train_bal$Attrition)
accuracy.stepwise.lda <- (conf.stepwise.lda[1,1] + conf.stepwise.lda[2,2]) / sum(conf.stepwise.lda)
plot(conf.stepwise.lda)
```

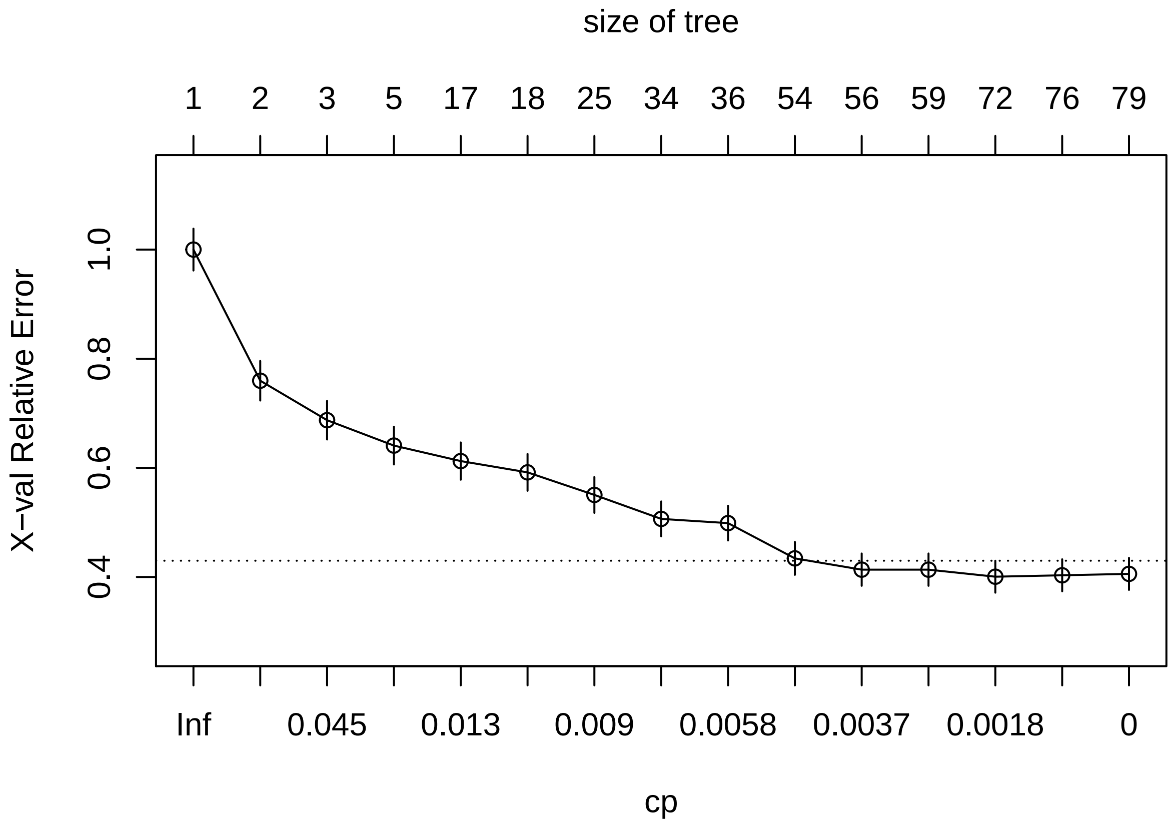


```
## courbe ROC
ROC.stepwise.lda <- roc(data_train_bal$Attrition, pred.stepwise.lda$posterior[,2])
plot(ROC.stepwise.lda, print.auc=TRUE, print.auc.y = 0.5)
legend("bottomright", lwd = 1, col = 1, "LDA stepwise")
```



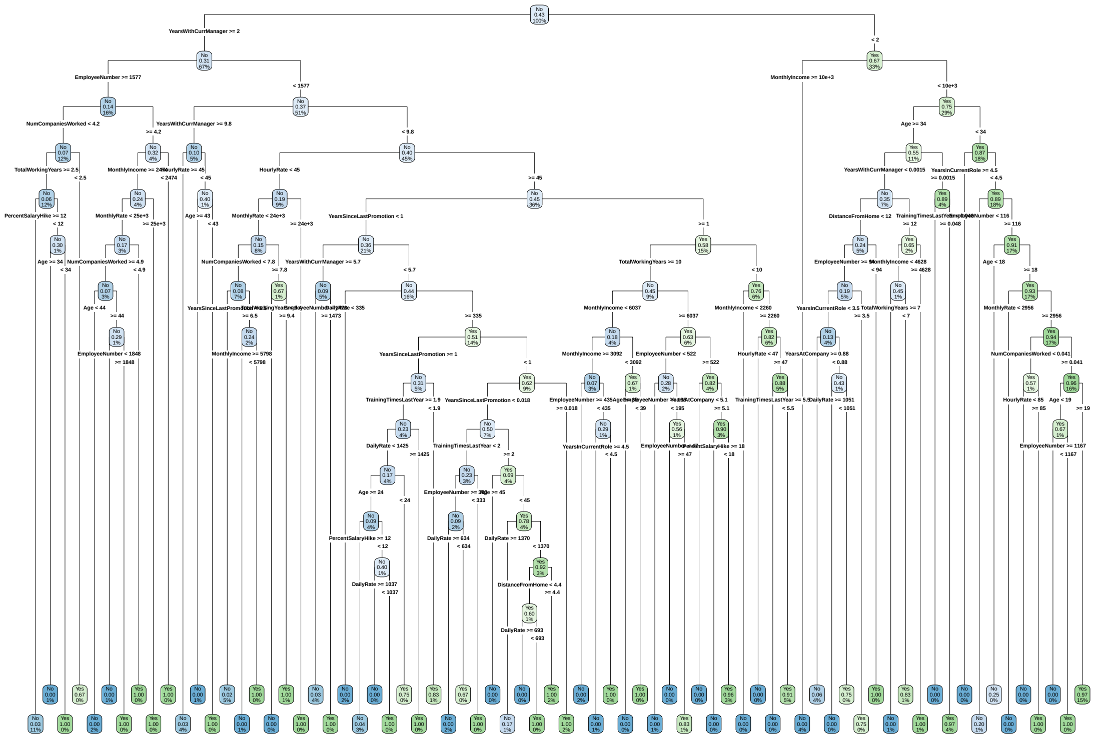
CART

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

## Modèle
arbre.cart = rpart(data_train_bal$Attrition ~ ., data = data_train_bal[-16], control = rpart.control(mi
plotcp(arbre.cart)
```



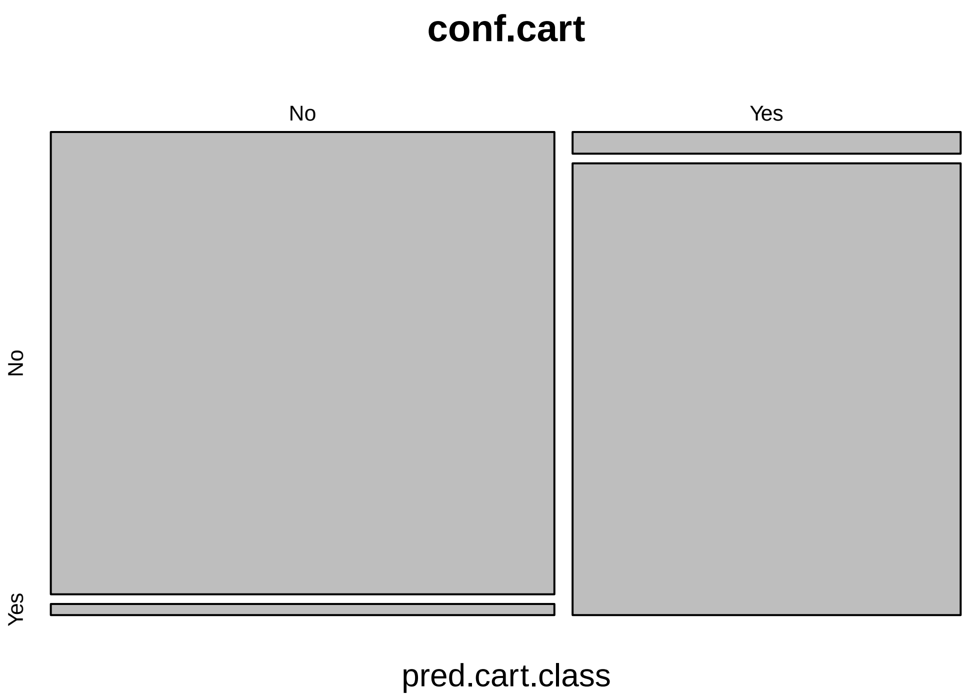
```
## Optimisation de l'arbre
cp.opt <- arbre.cart$cptable[which.min(arbre.cart$cptable[, "xerror"]), "CP"]
arbre.opt <- prune(arbre.cart, cp = cp.opt)
rpart.plot(arbre.opt, type=4, digits=2)
```

```
## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```

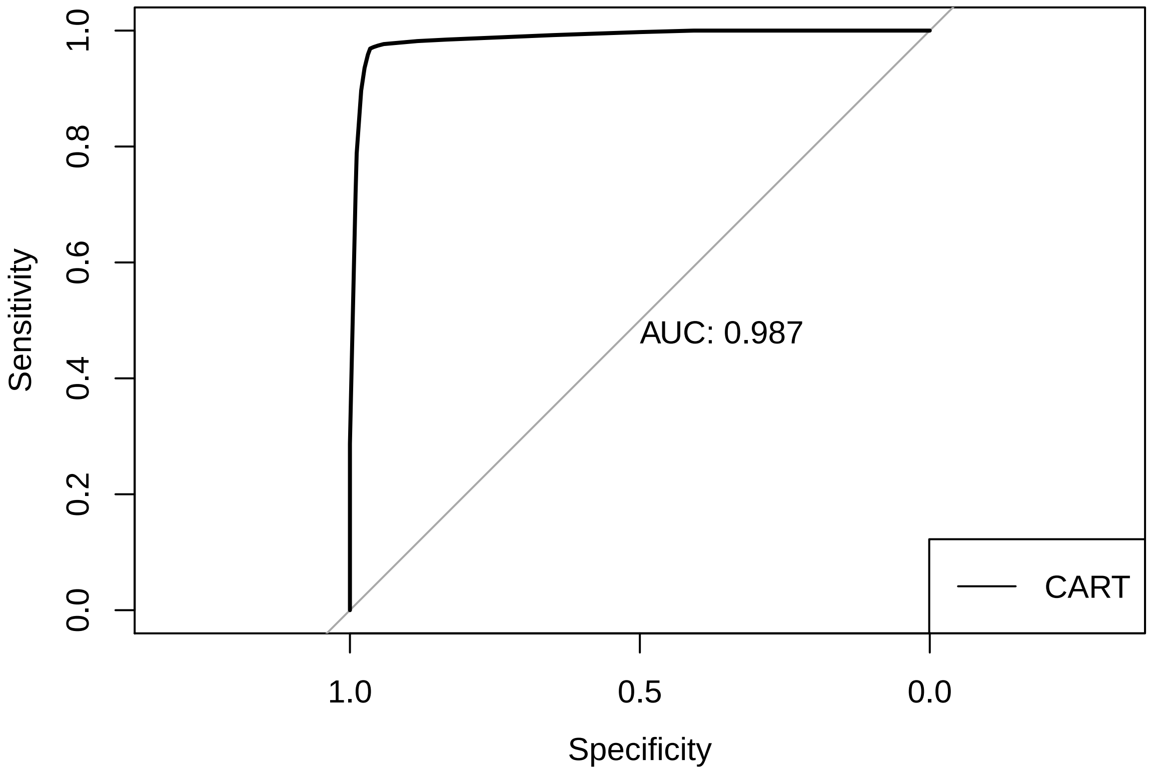


```r
## Prédiction
pred.cart.class <- predict(arbre.opt, newdata = data_train_bal[-16], type = "class")
pred.cart.prob <- predict(arbre.opt, newdata = data_train_bal[-16], type = "prob")[,2]
```

```r
## Table de confusion
conf.cart <- table(pred.cart.class, data_train_bal$Attrition)
accuracy.cart <- (conf.cart[1,1] + conf.cart[2,2]) / sum(conf.cart)
plot(conf.cart)
```



```
## courbe ROC
ROC.cart <- roc(data_train_bal$Attrition, pred.cart.prob)
plot(ROC.cart, print.auc=TRUE, print.auc.y = 0.5, col = 1)
legend("bottomright", lwd = 1, col = 1, "CART")
```



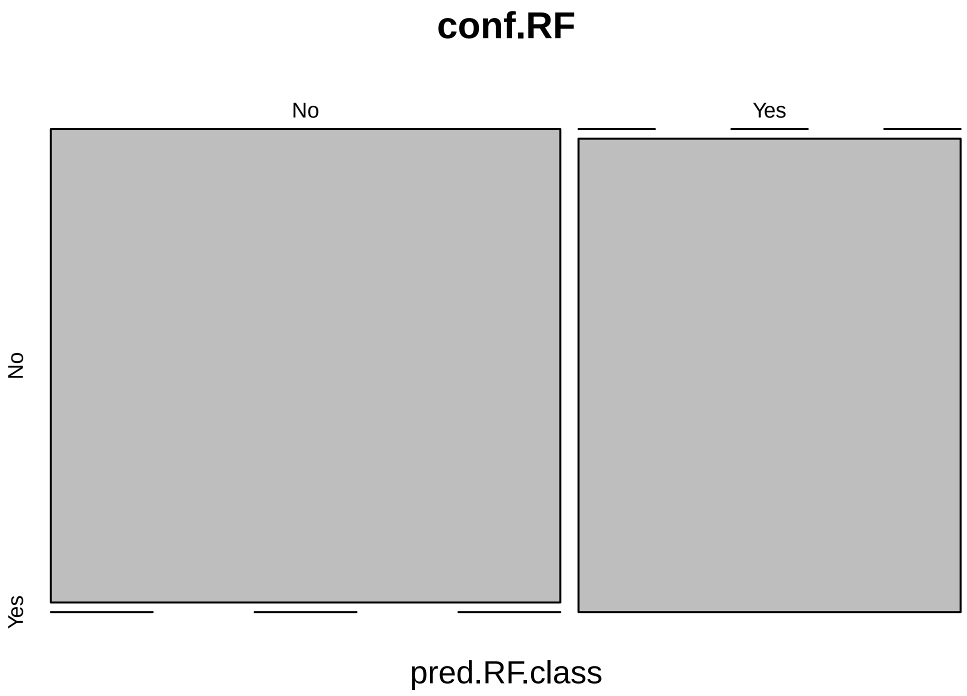
Random Forest

```
library(randomForest)

## Modèle
res.RF <- randomForest(data_train_bal$Attrition ~ ., data_train_bal[-16])
res.RF
```

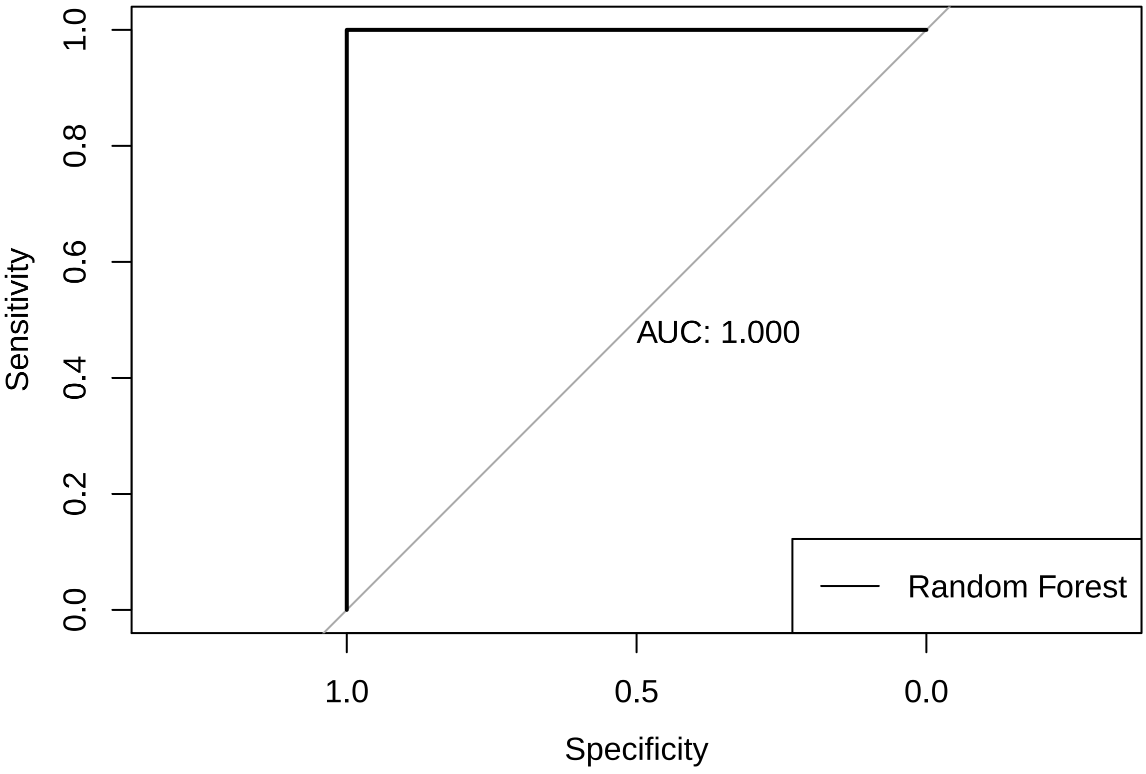
```
##
## Call:
##  randomForest(formula = data_train_bal$Attrition ~ ., data = data_train_bal[-16])
##               Type of random forest: classification
##                     Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 10.08%
## Confusion matrix:
##      No Yes class.error
## No  485  31  0.06007752
## Yes  60 327  0.15503876
```

```
## Prédiction
pred.RF.class <- predict(res.RF, newdata = data_train_bal[-16], type="class")
pred.RF.prob <- predict(res.RF, newdata = data_train_bal[-16], type = "prob")[,2]
```

```
## Table de confusion
conf.RF <- table(pred.RF.class, data_train_bal$Attrition)
accuracy.RF <- (conf.RF[1,1] + conf.RF[2,2]) / sum(conf.RF)
plot(conf.RF)
```

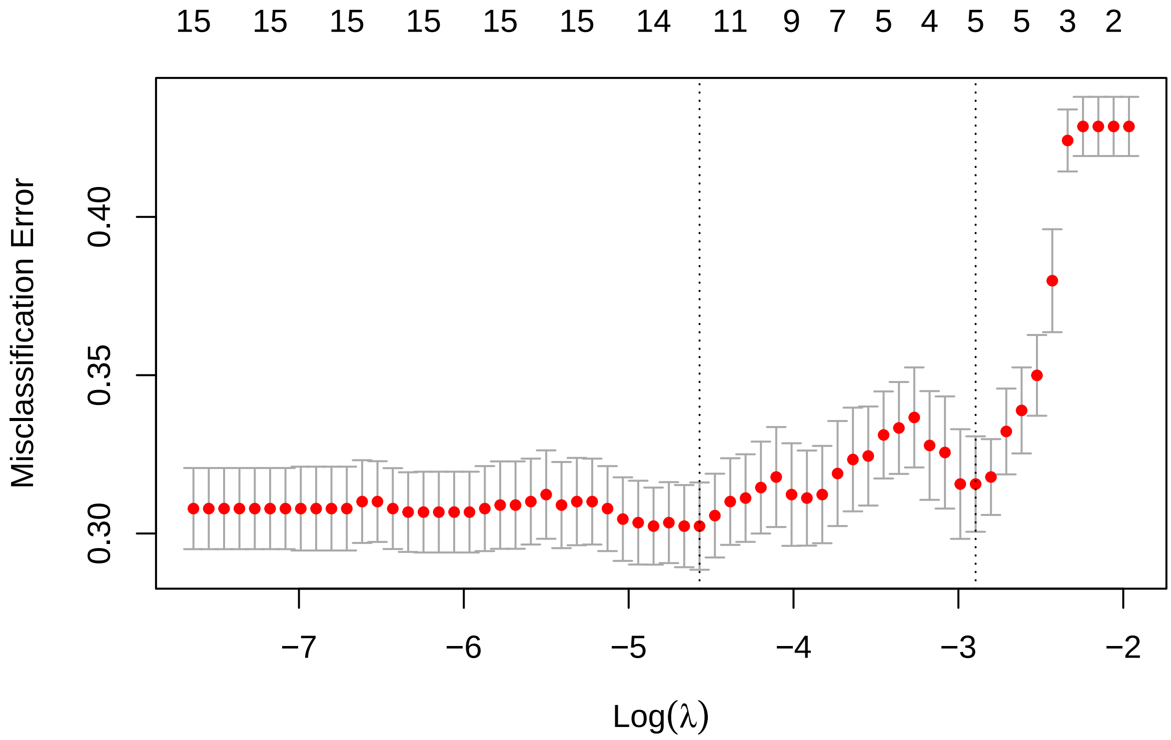


```r
## courbe ROC
ROC.RF <- roc(data_train_bal$Attrition, pred.RF.prob)
plot(ROC.RF, print.auc=TRUE, print.auc.y = 0.5, col = 1)
legend("bottomright", lwd = 1, col = 1, "Random Forest")
```



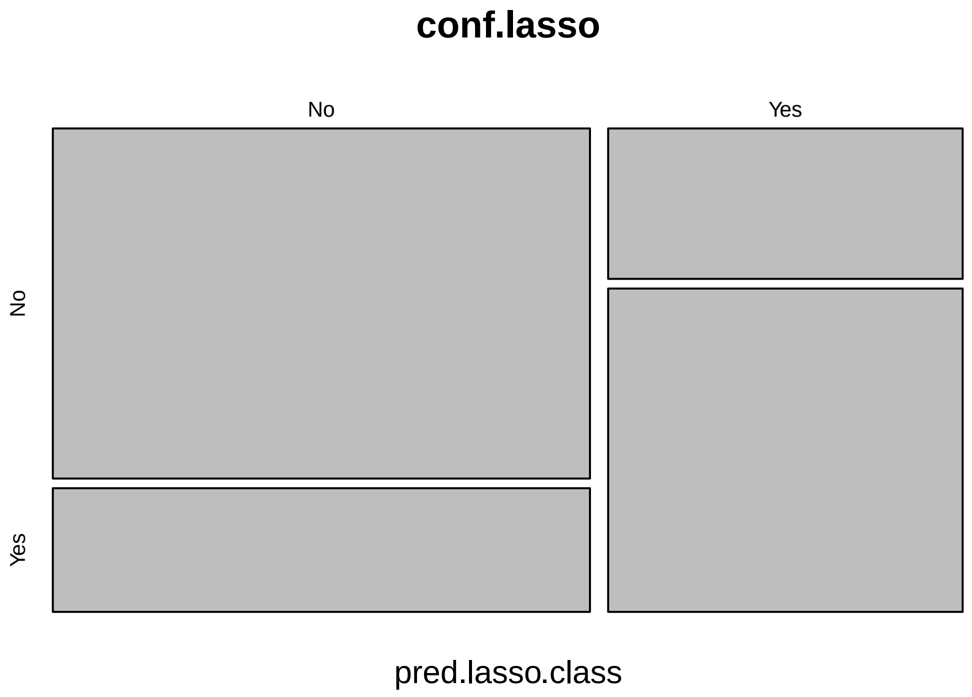
Regression Logistique Lasso

```r
library(glmnet)

## Modèle
res.Lasso <- glmnet(as.matrix(data_train_bal[-16]), data_train_bal$Attrition, family='binomial')
cv.Lasso <- cv.glmnet(as.matrix(data_train_bal[-16]), data_train_bal$Attrition, family="binomial", type
plot(cv.Lasso)
```

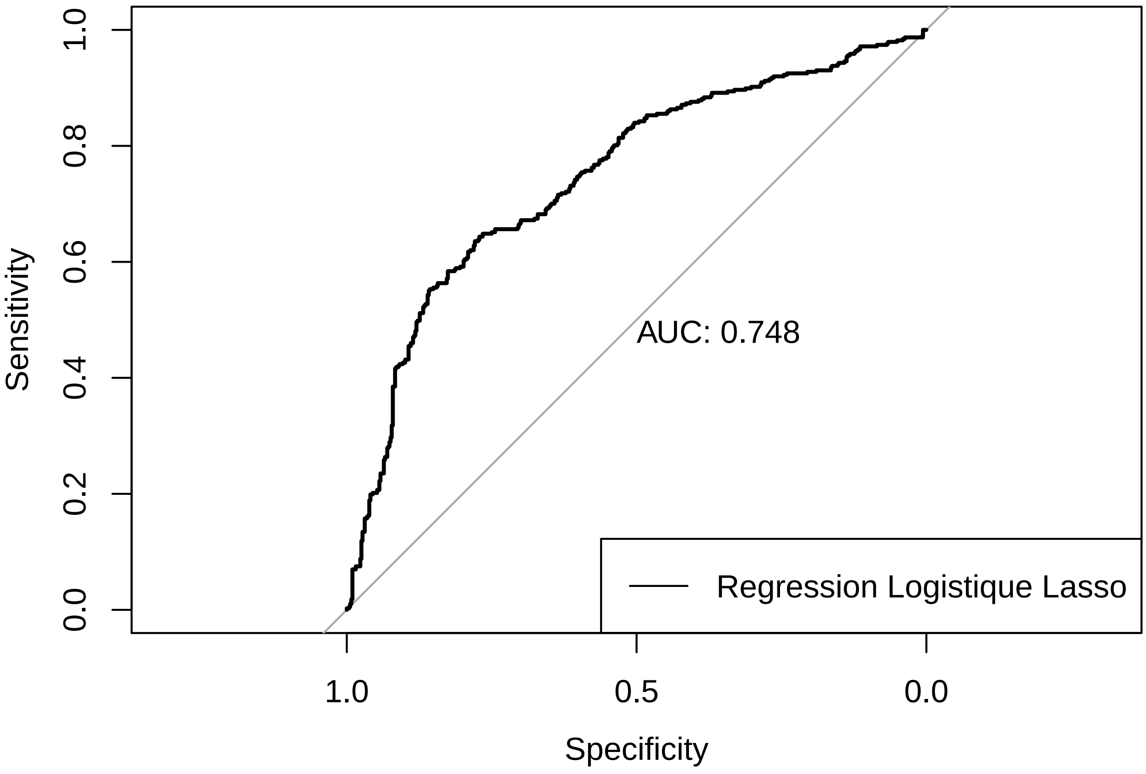


```
## Prédiction
pred.lasso.class <- predict(cv.Lasso, newx = as.matrix(data_train_bal[-16]), s = 'lambda.min', type = "c
pred.lasso.prob <- predict(cv.Lasso, newx = as.matrix(data_train_bal[-16]), s = 'lambda.min', type = "re
```

```
## Table de confusion
conf.lasso <- table(pred.lasso.class, data_train_bal$Attrition)
accuracy.lasso <- (conf.lasso[1,1] + conf.lasso[2,2]) / sum(conf.lasso)
plot(conf.lasso)
```



```
## courbe ROC
ROC.lasso <- roc(data_train_bal$Attrition, pred.lasso.prob)
plot(ROC.lasso, print.auc=TRUE, print.auc.y = 0.5, col = 1)
legend("bottomright", lwd = 1, col = 1, "Regression Logistique Lasso")
```



Comparaison des méthodes

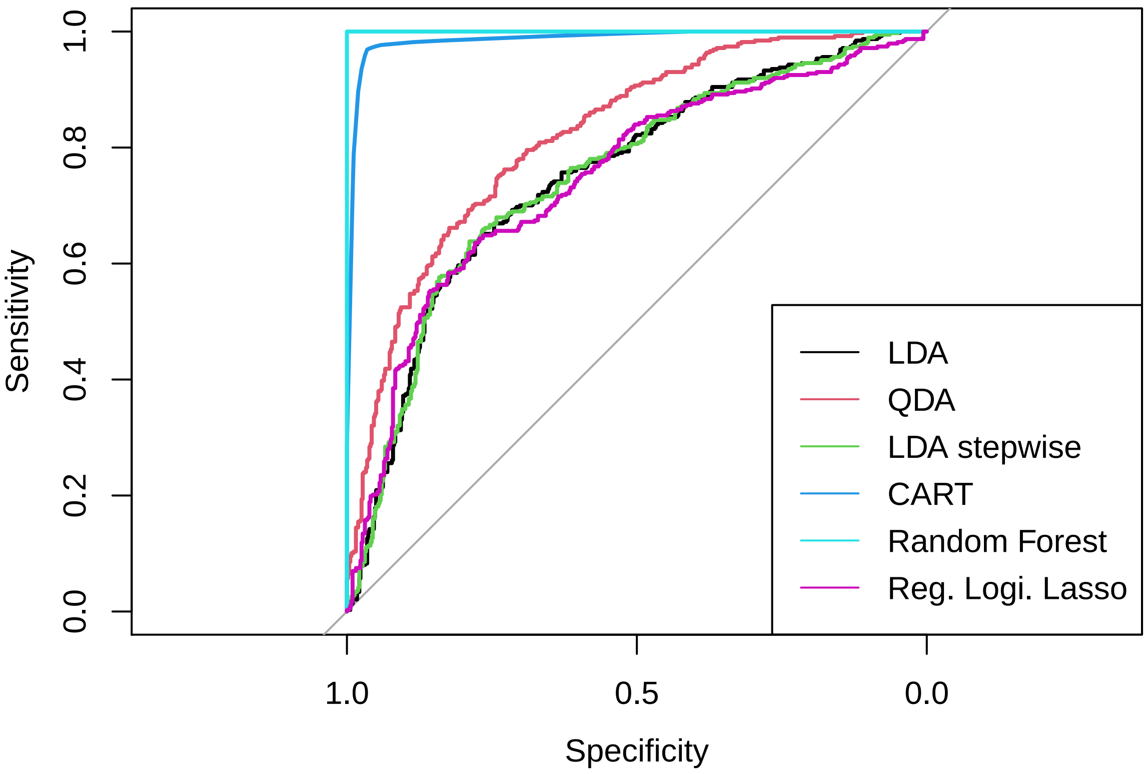
```
result = matrix(NA, ncol = 6, nrow = 2)
rownames(result) = c('accuracy', 'AUC')
colnames(result) = c('LDA', 'QDA',  'LDA stepwise', 'CART', 'Random Forest',  'Reg. Logi. Lasso')
result[1,] = c(accuracy.lda, accuracy.qda, accuracy.stepwise.lda, accuracy.cart, accuracy.RF, accuracy.l
result[2,] = c(ROC.lda$auc, ROC.qda$auc, ROC.stepwise.lda$auc, ROC.cart$auc, ROC.RF$auc,  ROC.lasso$auc
result
```

```
##                LDA       QDA LDA stepwise      CART Random Forest
## accuracy 0.7131783 0.7231451    0.7131783 0.9667774             1
## AUC      0.7506260 0.8215652    0.7502854 0.9868523             1
##          Reg. Logi. Lasso
## accuracy        0.7165006
## AUC             0.7476063
```

```
apply(result, 1, which.max )
```

```
## accuracy      AUC
##        5        5
```

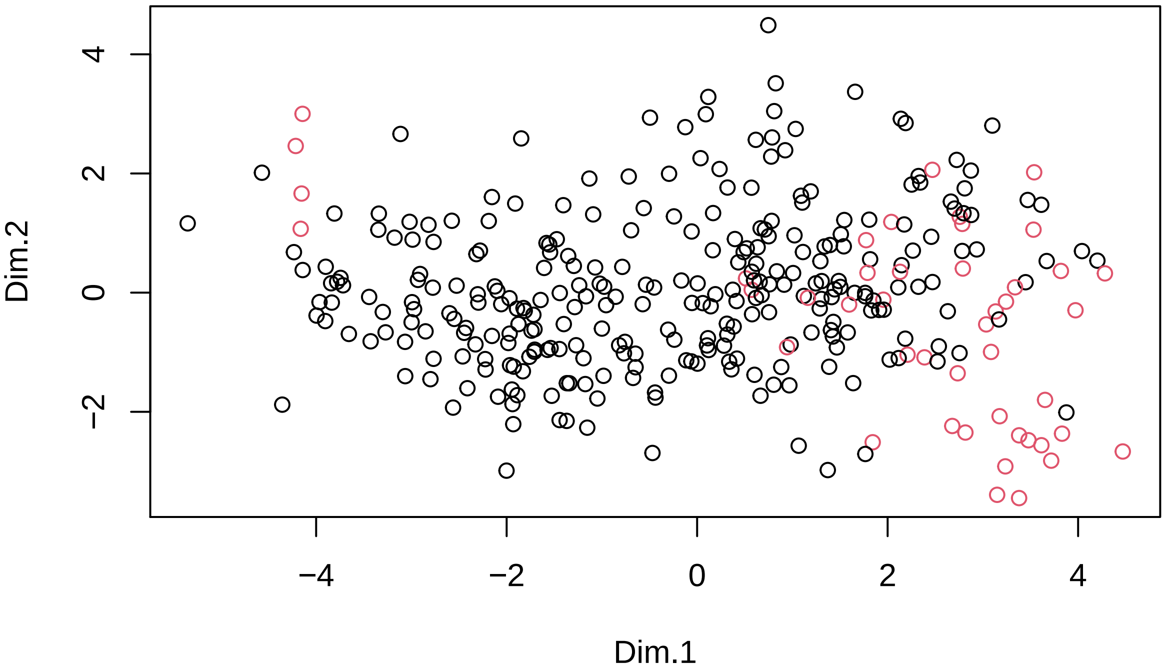
```
plot(ROC.lda, xlim = c(1,0))
plot(ROC.qda, add = TRUE, col = 2)
plot(ROC.stepwise.lda, add = TRUE, col = 3)
plot(ROC.cart, add = TRUE, col = 4)
plot(ROC.RF, add = TRUE, col = 5)
plot(ROC.lasso, add = TRUE, col = 6)
legend('bottomright', col = 1:6, paste(colnames(result)),  lwd = 1)
```



La meilleure méthode de prédiction en tout point est le random Forest.

Resolution de notre problème avec Random Forest

```
pred.Attrition <- predict(res.RF, newdata = data_test_num, type="class")

plot(coord_data_test, col = pred.Attrition)
```



```
data_test_pred <- data.frame(pred.Attrition, data_test)
write.csv(data_test_pred, file = "prediction.csv", quote = FALSE, sep = ',')
```