

---

## Rapport Analyse de données

---



MICHELE BONA  
TRISTAN MICHEL

ENCADRÉ PAR :  
FANNY VILLIERS

## Table des matières

<b>1</b>	<b>Présentation du jeu de données</b>	<b>2</b>
<b>2</b>	<b>Importation du jeu de données</b>	<b>2</b>
<b>3</b>	<b>Analyses statistiques</b>	<b>3</b>
3.1	ACP . . . . .	4
3.2	MCA . . . . .	5
<b>4</b>	<b>Classification</b>	<b>5</b>
4.1	LDA-QDA . . . . .	5
4.2	K-Means . . . . .	6
<b>5</b>	<b>Prédiction</b>	<b>7</b>
5.1	LDA - QDA . . . . .	7
5.2	Stepwise LDA . . . . .	8
5.3	CART . . . . .	8
5.4	Random Forest . . . . .	9
5.5	Régression Logistique Lasso . . . . .	9
<b>6</b>	<b>Meilleur modèle pour nos données</b>	<b>11</b>
<b>7</b>	<b>Conclusion</b>	<b>12</b>

## 1 Présentation du jeu de données

Le jeu de données que nous avons analysé est un jeu de données artificiel créé par Prashant Patel et nous l'avons téléchargé sur *kaggle* à cette adresse :

<https://www.kaggle.com/colearninglounge/employee-attrition>

Ce jeu de données a pour variable cible l'*attrition*, il nous dit si les employés ont démissionné ou s'ils sont restés en fonction de nombreuses variables telles que la satisfaction au travail, le secteur d'activité (qualitatives) ou le salaire, les années passées dans l'entreprise (quantitatives).

L'objectif de ce jeu de données est de prévoir si un employé va partir ou non de l'entreprise, pour cela on a déjà un dataset d'entraînement pour créer notre modèle et ensuite un dataset de test (sans la variable cible) pour tester notre modèle, on fera donc une classification supervisée.

## 2 Importation du jeu de données

Ce jeu de données est déjà assez propre mais nous avons identifié quelques variables qui avaient la même valeur dans tout le dataset, nous les avons donc enlevé car elles n'auraient pas influencé le modèle. Il s'agit de *CountEmployee*, *standardHours* et *over18*. Nous avons mit en annexe les jeux de données originaux et les nouveaux.

De plus nous avons créé des sous-jeux de données pour faire l'ACP et la MCA. Le premier avec seulement les données quantitatives (qu'on a appelé dans notre code *data\_train\_num*). Mais étant donné qu'on a gardé des variables qui allaient de 1 à 5, on a remis tout à la même échelle dans un autre dataset nommé *data\_train\_log*. Enfin dans un dernier jeu de données appelé *data\_train\_fact* on a gardé seulement les variables qualitatives.

### 3 Analyses statistiques

Nous avons effectué des tests  $\chi^2$  pour voir si les différentes variables qualitatives étaient liées à notre variable cibles, par exemple :

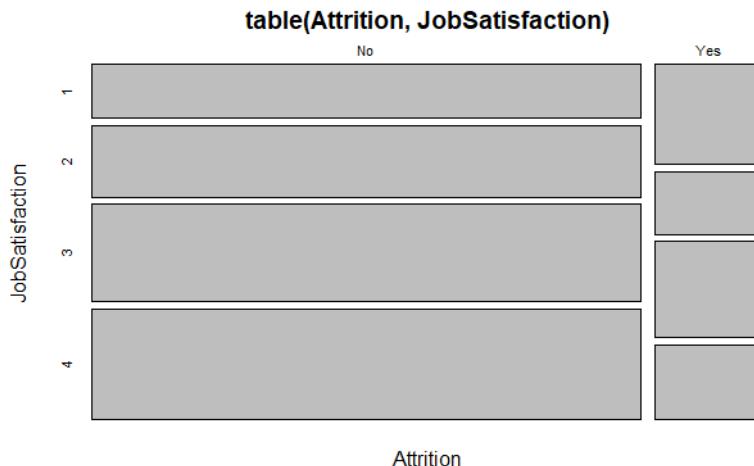


FIGURE 1 – Représentation de l'attrition en fonction de la satisfaction au travail

```
Pearson's Chi-squared test
data: table(Attrition, jobsatisfaction)
X-squared = 15.475, df = 3, p-value = 0.001453
```

FIGURE 2 – Résultats du test sur ces variables

La p-value du test est inférieure à 0.05 donc les variables sont bien liées, ce qui est cohérent avec la représentation où il semble que les employés les moins satisfait de leur travail ont plus tendance à partir (ce qui semble aussi cohérent avec la réalité).

### 3.1 ACP

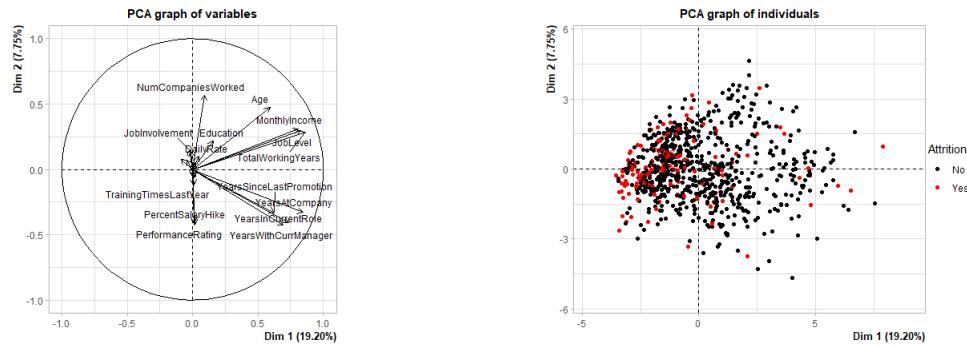


FIGURE 3 – ACP des valeurs numériques non échelonnées

On remarque un effet de bord soit qu'il y a des variables mal projetées. On construit un nouveau jeu de données avec une la transformation "double centrage" sur les données log-transformées nommé `data_train_log` puis on fait l'ACP sur ces nouvelles données :

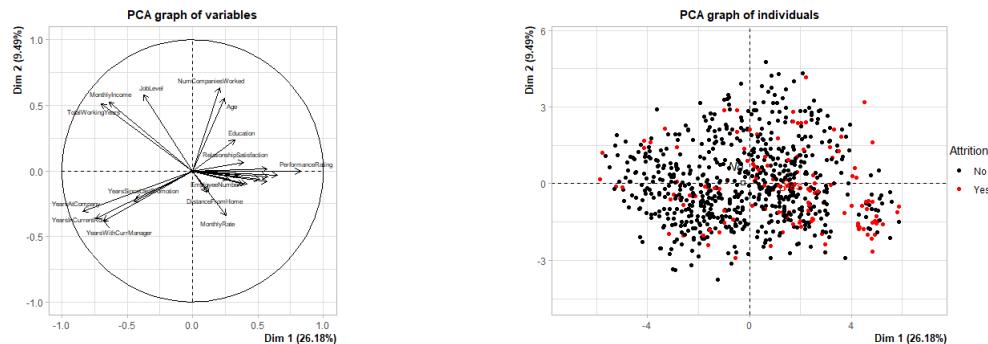


FIGURE 4 – ACP des valeurs numériques échelonnées

On remarque avec cette ACP que les employés qui partent (les points rouges) sont majoritairement regroupés en bas à droite, et donc on peut en tirer que les chances qu'un employé parte sont inversement proportionnelles à l'ancienneté, au salaire mensuel et à l'importance de son poste (resp. `TotalWorkingYears`, `MonthlyIncome`, `JobLevel`). Au contraire plus l'employé à de distance à parcourir pour aller au travail et plus il a de dépenses mensuelles élevées (resp. `DistanceFromHome`, `MonthlyRate`) plus il a de chances de partir. Ceci semble assez cohérent, même si la représentation n'est pas parfaite (la séparation entre les salariés qui partent et ceux qui restent n'est pas évidente, et beaucoup de facteur entre en jeu pour qu'un employé prenne cette décision).

### 3.2 MCA

On a appliqué un MCA sur les données qualitatives (on a gardé 2 axes car ils représentent bien les données) :

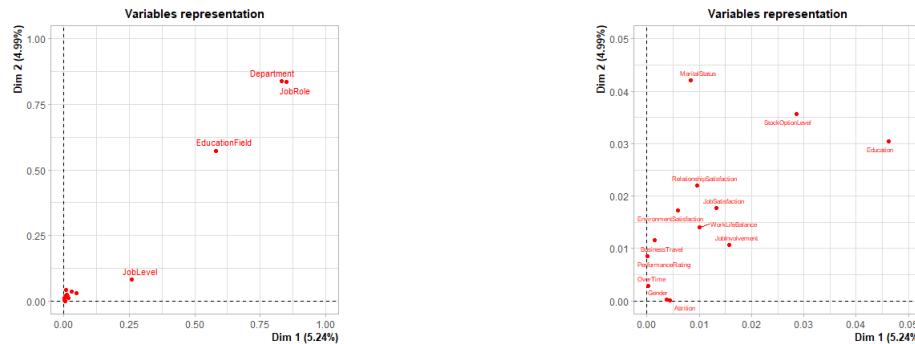


FIGURE 5 – ACP des valeurs numériques échelonnées

On voit sur ces représentations qu'il y a un lien entre **Department**, **JobRole** et **EducationField** par exemple, de la même manière les points sur le mêmes axe sont liées. Cependant sur le modèle il semblerait que **Gender** soit lié à **Attrition** ce qui est une erreur de notre modèle après vérification avec un test du  $\chi^2$ .

## 4 Classification

### 4.1 LDA-QDA

Nous avons pris les coordonnées des valeurs dans l'ACP visible dans la figure 4 pour notre classification supervisée avec les méthodes LDA et QDA. Celle-ci possèdent enormément d'erreur car les Yes sont très souvent prédit avec No.

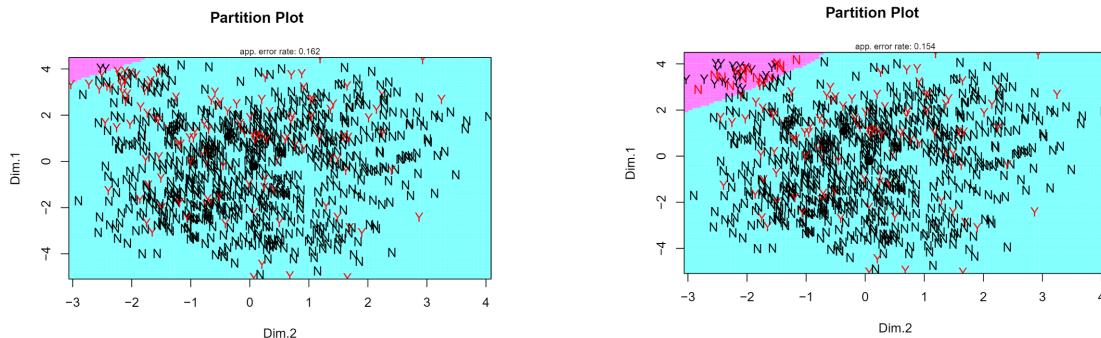


FIGURE 6 – Résultats LDA à gauche - QDA à droite. En rose la classe des employés qui vont probablement partir, en bleu ceux qui vont rester

Le résultat n'était pas concluant, en effet la séparation entre les deux groupes est une petite zone en haut à gauche dans les deux cas et nous avons beaucoup d'erreurs (des réponses Yes dans la classe des No et inversement, soit les points en rouge). Ceci est dû au fait que la représentation de l'ACP n'est pas parfaite et il n'y a pas de séparation évidentes entre les données une fois que nous les affichons sur 2 dimensions, or cela était nécessaire pour la classification LDA.

## 4.2 K-Means

Nous avons essayé la classification non supervisé au vu des mauvais résultats obtenus précédent. Mais les résultats contenaient encore beaucoup d'erreurs, montrant que les facteurs que nous avons n'indiquent pas uniquement la volonté de quitter l'entreprise. Ci-après en rouge les données misent dans la mauvaise classe :

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

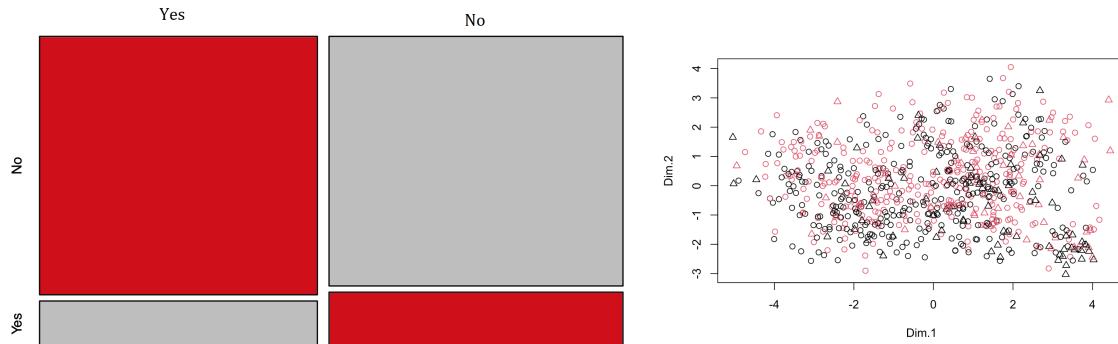


FIGURE 7 – Erreurs de la classification Supervisée et résultat

## 5 Prédiction

Finalement nous arrivons dans la partie prédiction. L'objectif premier est de prédire l'attrition du jeu de données test. Pour cela nous construisons le meilleur modèle possible en comparant les méthodes de prédiction. Or en faisant une première méthode d'estimation nous nous sommes rendu compte que nos données ne sont pas équilibré. Nous avions un total de 655 cas négatif pour seulement 129 cas positifs. Grâce à la fonction SMOTE de la librairie DMwR nous avons pu créer des nouveaux cas pour équilibrer notre jeu de données. Ainsi dans notre nouveau jeu de données nous avons 516 négatifs pour 387 positifs.

### 5.1 LDA - QDA

Le premier modèle que nous avons construit est le modèle LDA ainsi que le QDA. Nous l'avons entraîné avec nos données sur le facteur Attrition, puis pour mesurer son efficacité nous avons fait une prédiction sur les mêmes données en supprimant l'attribut Attrition. Comme ça nous avions un moyen simple de mesurer la précision du modèle obtenu.

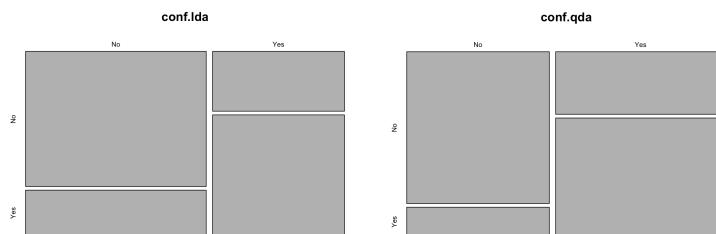


FIGURE 8 – Taux d'erreurs LDA à gauche - QDA à droite

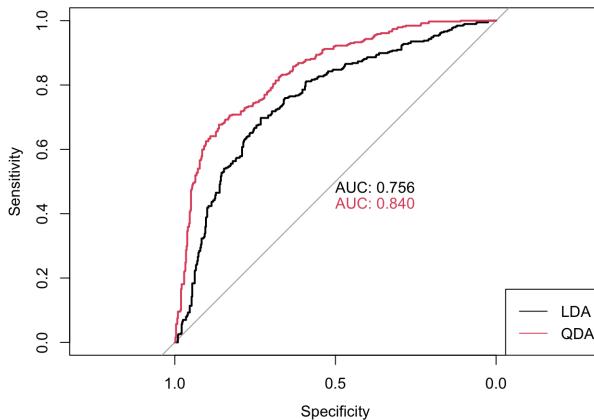


FIGURE 9 – Précision de la courbe ROC LDA à gauche - QDA à droite

## 5.2 Stepwise LDA

Ensuite nous avons utilisé la méthode Stepwise LDA. Avec cela nous espérions réduire la formule associé au modèle pour avoir un système plus précis. Heureusement cette réduction est plutôt rapide ( $\approx 4$  sec). Ce nouveau modèles plus efficace que la LDA classique mais reste moins efficace que la QDA. De même nous avons mesuré la précision sur le modèle de départ sans la valeur d'Attrition.

## 5.3 CART

La nouvelle méthode que nous avons mis en place est la construction d'arbre CART. Cette méthode va construire un arbre qui nous allons par la suite réduire. Ainsi on se retrouve avec l'arbre suivant :

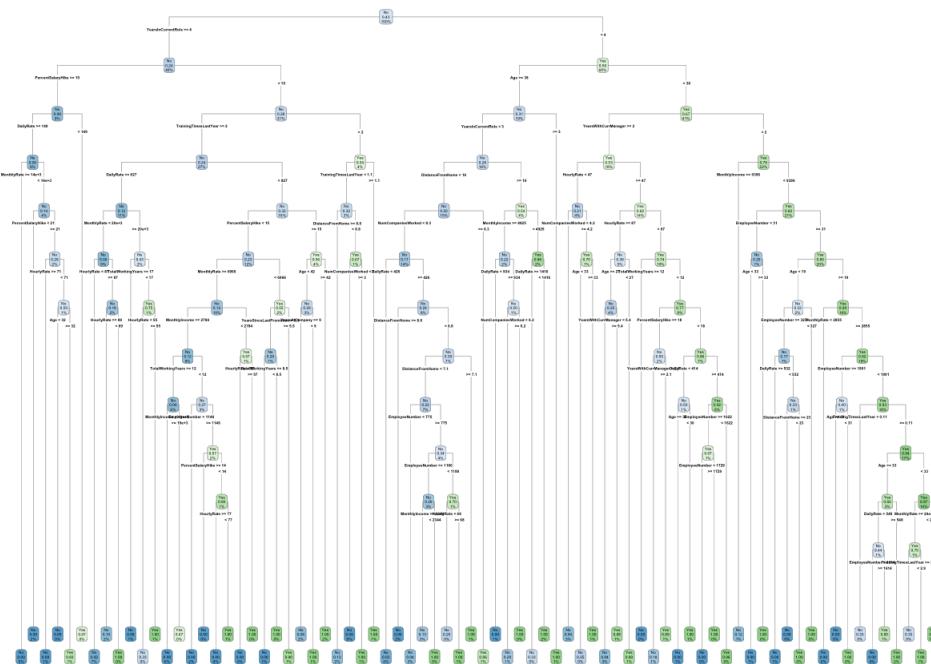


FIGURE 10 – L'arbre CART que nous avons construit puis réduit.

De même nous avons mesuré la précision comme dans les cas précédent, cela nous a donné des résultats plutôt prometteurs.

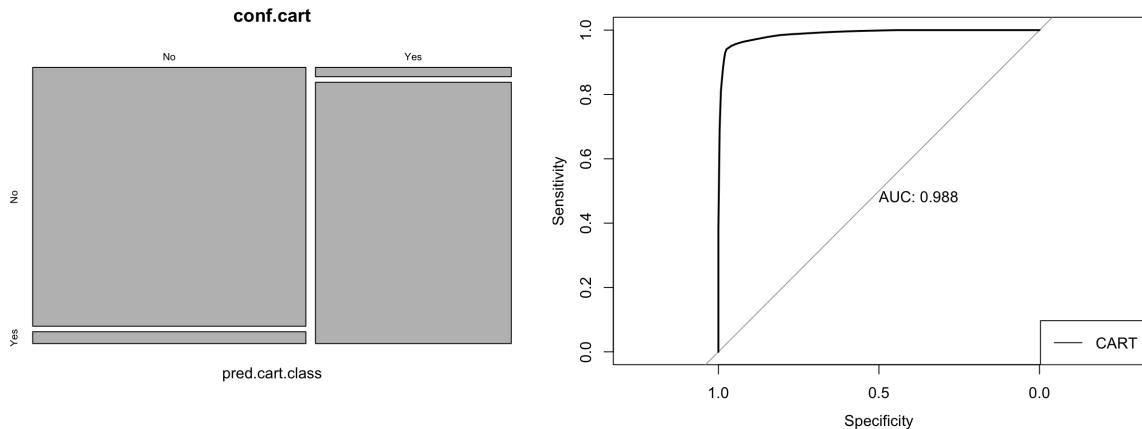


FIGURE 11 – Analyse de la prédiction de notre modèle CART

## 5.4 Random Forest

Après avoir construit un arbre, la solution suivante est de construire une forêt. La méthode de Random Forest nous a donné des résultats parfait suivant la méthode de mesure que nous avons appliquée jusqu'à présent.

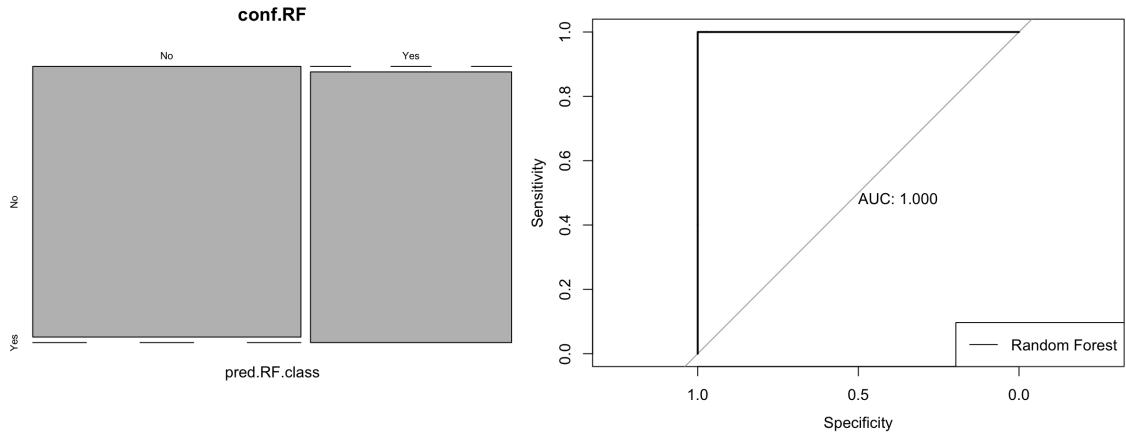


FIGURE 12 – Analyse de la prédiction de notre modèle CART

## 5.5 Régression Logistique Lasso

La dernière technique que nous avons utilisé est la Régression Logistique Lasso. Pour cela nous avons cherché le  $\lambda$  minimum pour réduire l'erreur de classe.

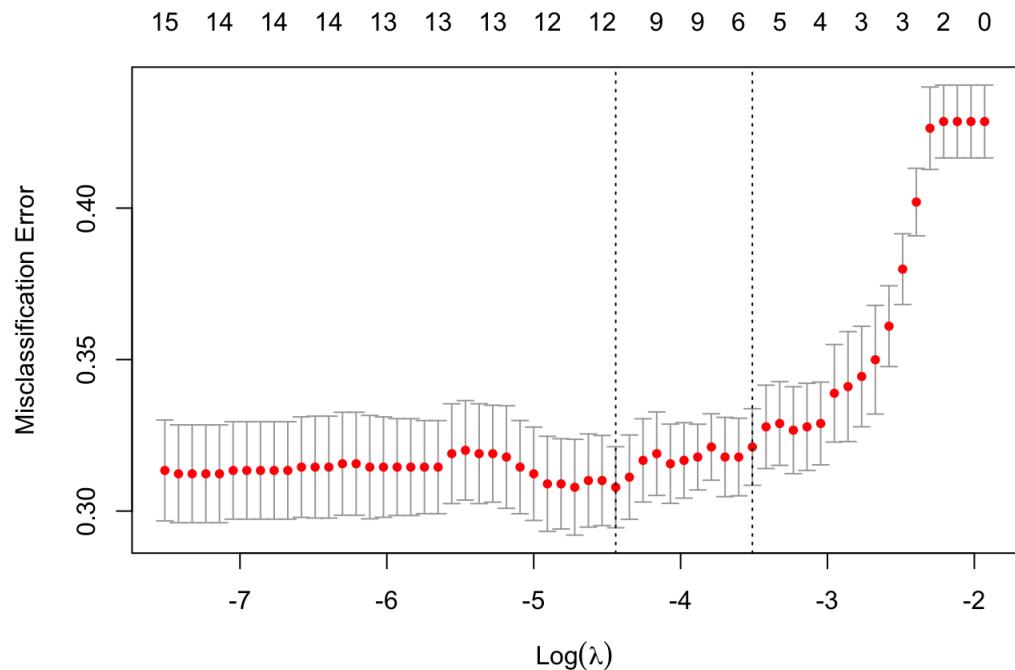


FIGURE 13 – Analyse de la prédiction de notre modèle CART

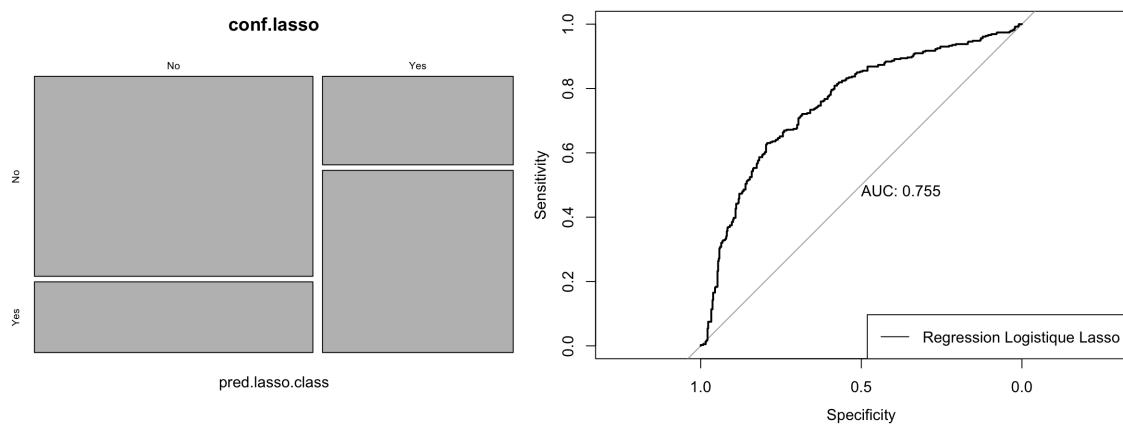


FIGURE 14 – Analyse de la prédiction de notre modèle Lasso

## 6 Meilleur modèle pour nos données

Le but de notre analyse étant de prédire le départ d'un jeu d'individu nous avons choisi le modèle de Random Forest qui nous donne un résultat parfait quand on mesure sa précision. En effet nous avons un ROC de 1, voici ci-dessus une étude comparative de la précision pour chaque méthode :

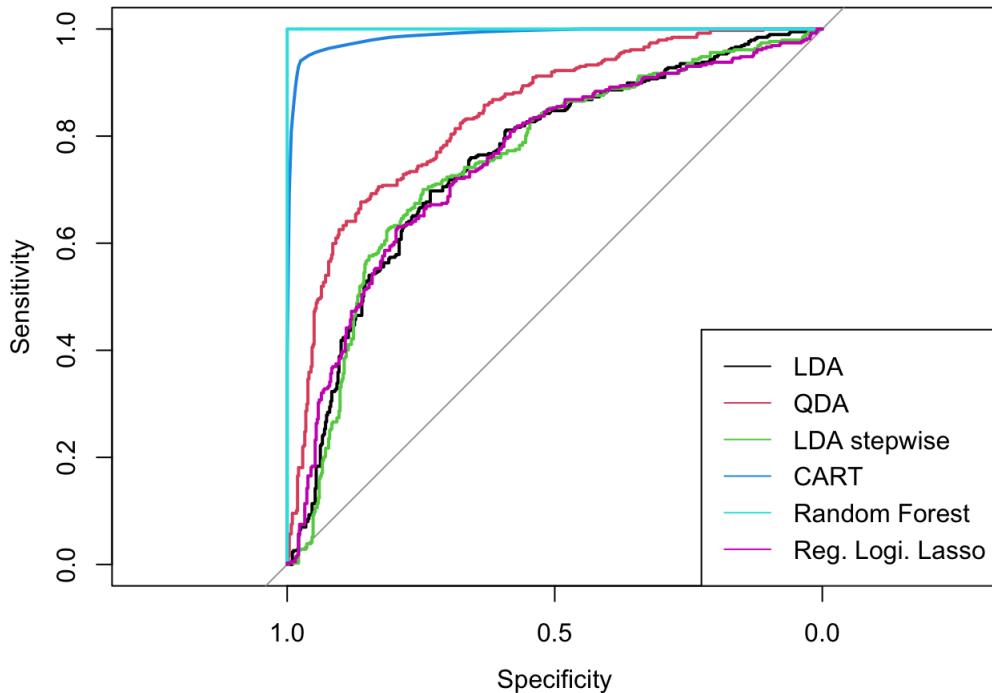


FIGURE 15 – Analyse de la prédiction de notre modèle CART

Nous avons donc utilisé Random Forest pour prédire les employés qui vont partir dans notre jeu de données test, pour ainsi répondre à la problématique du jeu de données. Nous trouvons le résultats suivant qui possède bien un regroupement de points rouge en bas à droite comme sur notre ACP de données d'entraînement, montrant que notre prédiction semble cohérente.

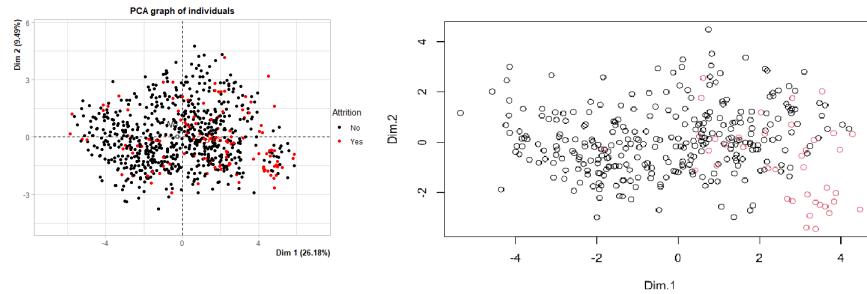


FIGURE 16 – Application de notre modèle Random Forest sur le test et comparaison avec le train

## 7 Conclusion

Ainsi nous avons pu prédire le départ de certains employés, mais ce n'est pas du tout une certitude. En effet notre analyse nous permet de connaître les salariés "à risque" de départ mais on ne pourra jamais être sûr, le départ d'un employé dépend de beaucoup de facteurs (dont des facteurs personnels impossible à quantifier) et est dur à prévoir. Des employés peuvent avoir une très bonne position et partir quand même pour avoir mieux ailleurs et d'autres peuvent avoir des conditions mauvaises mais ne pas avoir le choix de rester... Néanmoins nos résultats sont intéressants pour comprendre les facteurs importants pour faire rester les employés et surtout pour connaître les employés avec qui les ressources humaines doivent s'entretenir pour qu'ils ne risquent pas de partir (ce qui était la principale problématique du jeu de données).

Nous aurions pu pour préciser notre analyse et notre modèle regarder par exemple le niveau de vie ou la faciliter de trouver un emploi dans la région où se trouve l'entreprise, en effet ces facteurs pourraient beaucoup influer sur le départ des salariés. Mais l'entreprise étant fictive ça ne sera pas possible, à voir pour un cas réel...

## Annexes

### R Markdown - Analyse

# Projet - Analyse de Données

## Projet KikiCkisenVa - Analyse

```
fact.data <- function(data) {  
  if (!is.null(data$Attrition))  
    data$Attrition <- as.factor(data$Attrition)  
  data$BusinessTravel <- as.factor(data$BusinessTravel)  
  data$Department <- as.factor(data$Department)  
  data$Education <- as.factor(data$Education)  
  data$EducationField <- as.factor(data$EducationField)  
  data$EnvironmentSatisfaction <- as.factor(data$EnvironmentSatisfaction)  
  data$Gender <- as.factor(data$Gender)  
  data$JobInvolvement <- as.factor(data$JobInvolvement)  
  data$JobLevel <- as.factor(data$JobLevel)  
  data$JobRole <- as.factor(data$JobRole)  
  data$JobSatisfaction <- as.factor(data$JobSatisfaction)  
  data$MaritalStatus <- as.factor(data$MaritalStatus)  
  data$OverTime <- as.factor(data$OverTime)  
  data$PerformanceRating <- as.factor(data$PerformanceRating)  
  data$RelationshipSatisfaction <- as.factor(data$RelationshipSatisfaction)  
  data$StockOptionLevel <- as.factor(data$StockOptionLevel)  
  data$WorkLifeBalance <- as.factor(data$WorkLifeBalance)  
  return(data)  
}
```

## 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_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

```

## Stats descriptives

```

chisq.test(data_train_num[-16])

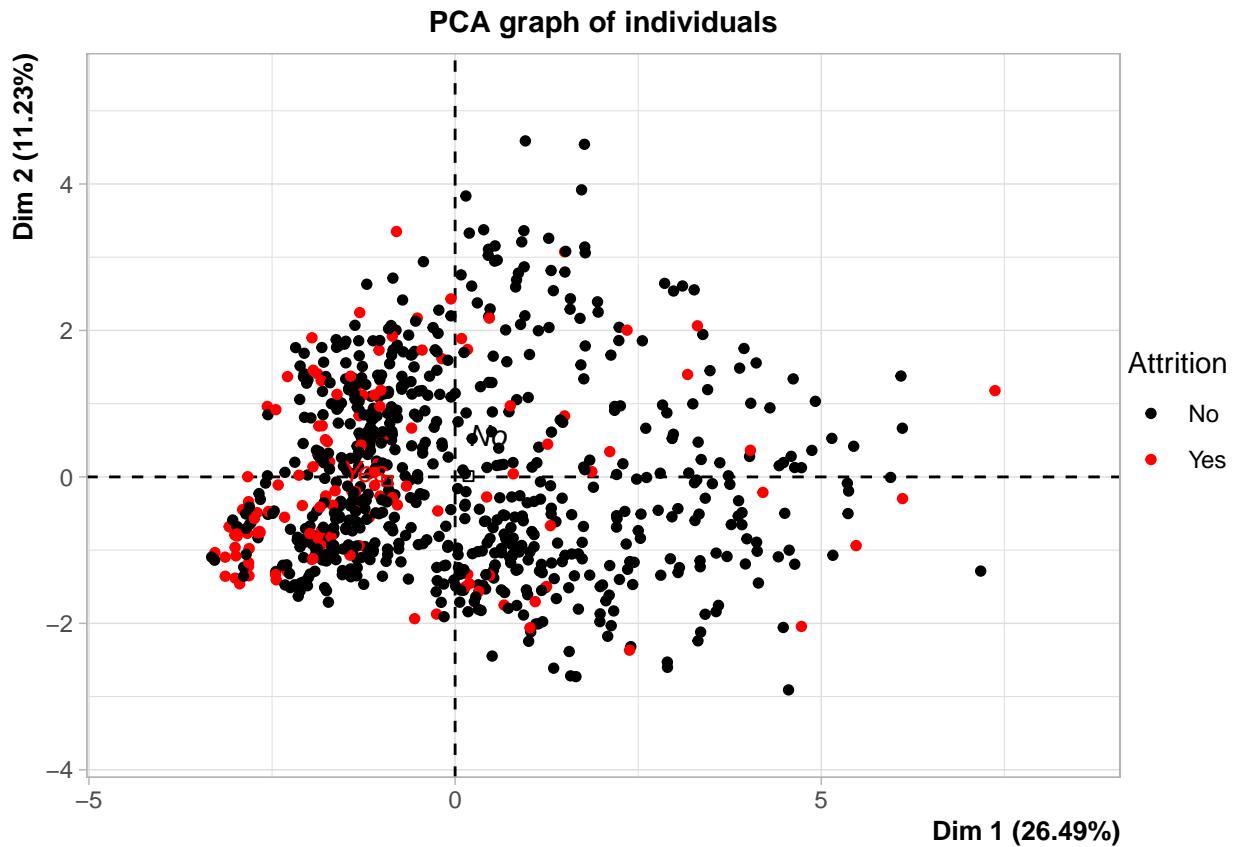
##
##  Pearson's Chi-squared test
##
## data: data_train_num[-16]
## X-squared = 3415673, df = 10962, p-value < 2.2e-16

```

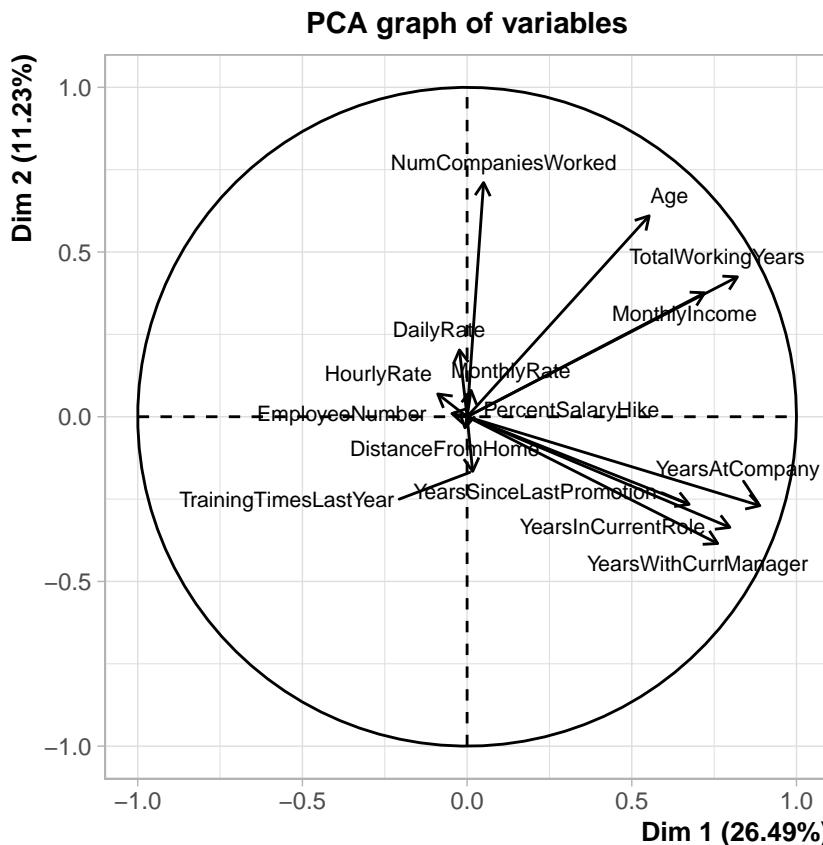
Toutes les variables ne semblent pas indépendantes entre elles.

## ACP

```
library(FactoMineR)
res.pca <- PCA(data_train_num, scale.unit = TRUE, graph = FALSE, quali.sup = 16)
plot(res.pca, choix = "ind", habillage = 16, select = FALSE, unselect = 0)
```



```
plot(res.pca, choix = "var", cex = 0.7)
```



On voit apparaître un effet taille.

Pour contrer cela nous allons transformer les données en appliquant

## Équilibrage des Données

```

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"]

head(data_train_log)

##          Age DailyRate DistanceFromHome EmployeeNumber HourlyRate
## 1  0.10875235  1.0787024      -1.1096083     1.0408076  0.19521799
## 2  0.20244958  0.8490462      -0.4441471     0.7792193  0.50821605
## 3  0.08343912  0.9616526      -0.2641722     1.4882483  0.20666944
## 4  0.03712663  1.0215512      -0.6319477     1.0607746  0.20884308
## 5 -0.14446568  1.0948206      -0.4117491     0.8284386  0.03481005
## 6 -0.12447781  0.9637158      -1.0010694     1.1777804  0.11447912
##          MonthlyIncome MonthlyRate NumCompaniesWorked PercentSalaryHike
## 1       1.931345    1.630159      -0.4253045     -0.146933864
## 2       1.978593    1.776077      -0.4441471     -0.009666449
## 3       1.617606    1.786383      -0.8798765     -0.068302317
## 4       1.806337    1.934180      -0.6319477     -0.298791173
## 5       1.896983    2.097910      -0.6701111     -0.487393757
## 6       1.794230    2.015974      -0.7544610     -0.610204272

```

```

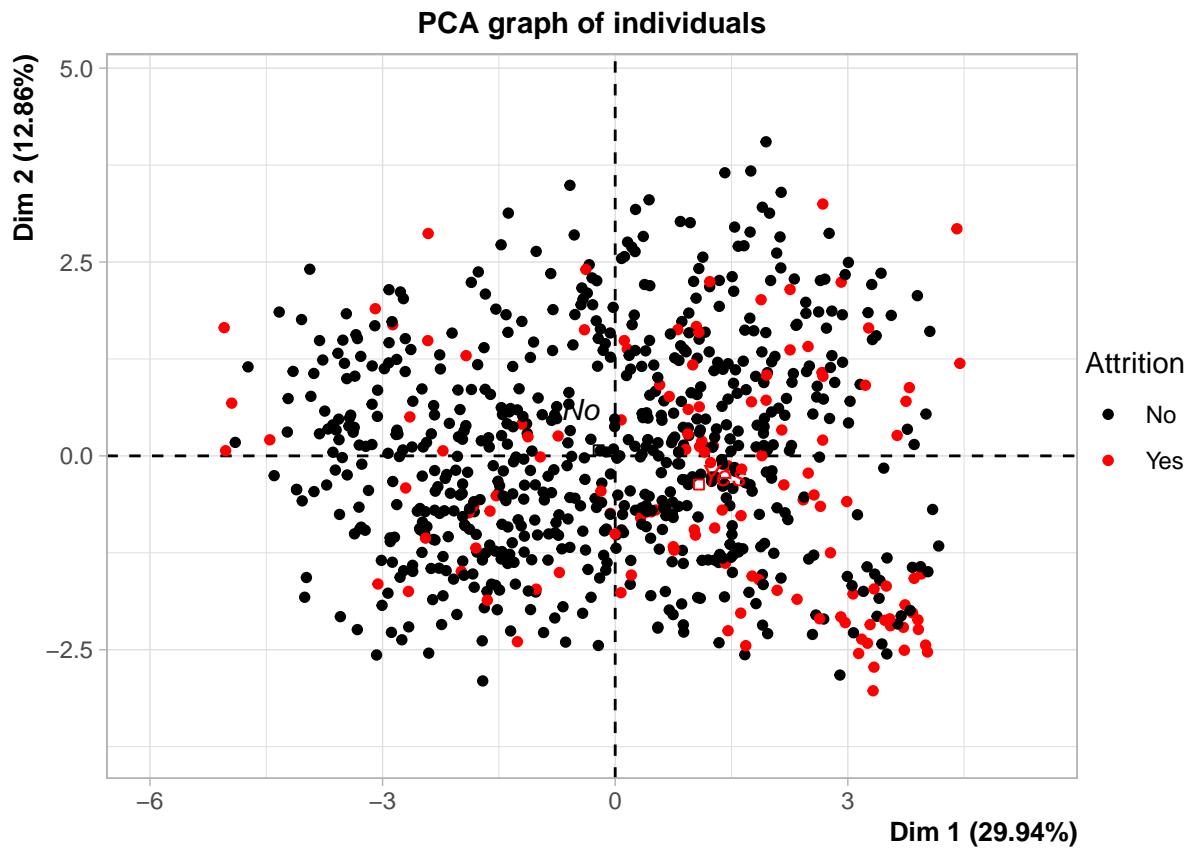
##   TotalWorkingYears TrainingTimesLastYear YearsAtCompany YearsInCurrentRole
## 1      -0.03023932      -1.1096083     -0.6083648      -0.6778607
## 2      -0.38851835      -1.0907437     -0.6942848      -0.8406060
## 3      -0.87987646      -0.5322652     -0.8798765      -0.8798765
## 4      -0.31683056      -0.8186621     -0.5873953      -0.6833797
## 5      -0.41174912      -1.0947920     -0.6701111      -0.7223675
## 6      -0.30875200      -1.0010694     -0.3945528      -0.6411612
##   YearsSinceLastPromotion YearsWithCurrManager Attrition
## 1      -1.1096083      -0.7674564        No
## 2      -1.0907437      -1.0907437        No
## 3      -0.8798765      -0.8798765       Yes
## 4      -0.8186621      -1.2811956        No
## 5      -0.6701111      -0.6701111       Yes
## 6      -0.5553604      -0.6750708        No

```

```

res.pca.log <- PCA(data_train_log, scale.unit = TRUE, graph = FALSE, quali.sup = 16)
plot(res.pca.log, choix = "ind", habillage = 16, select = FALSE, unselect = 0)

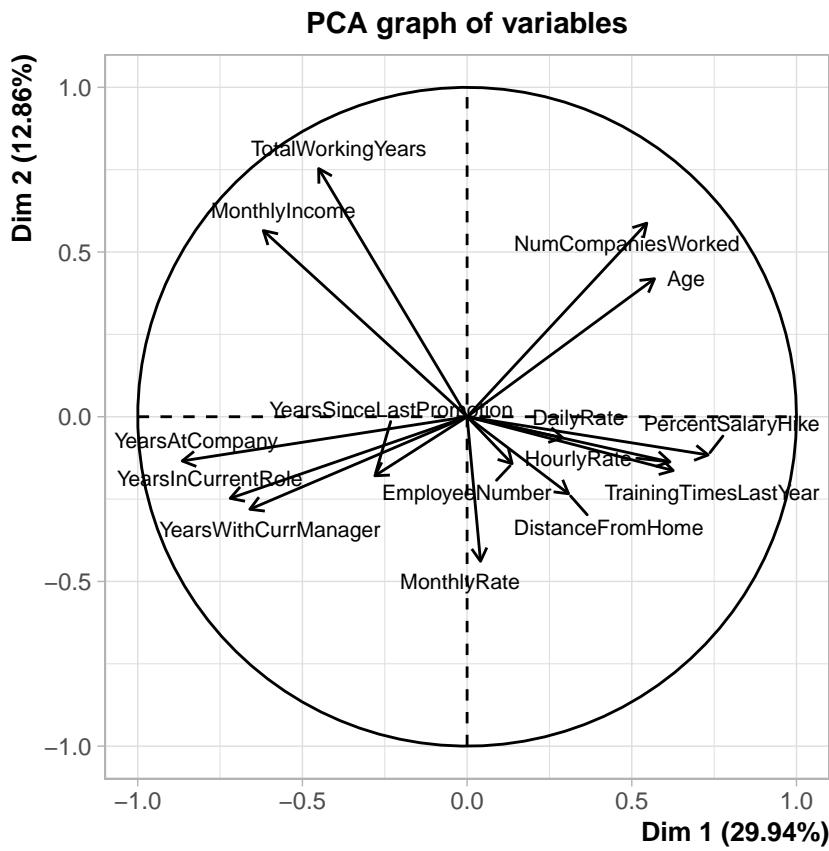
```



```

plot(res.pca.log, choix = "var", cex = 0.7)

```

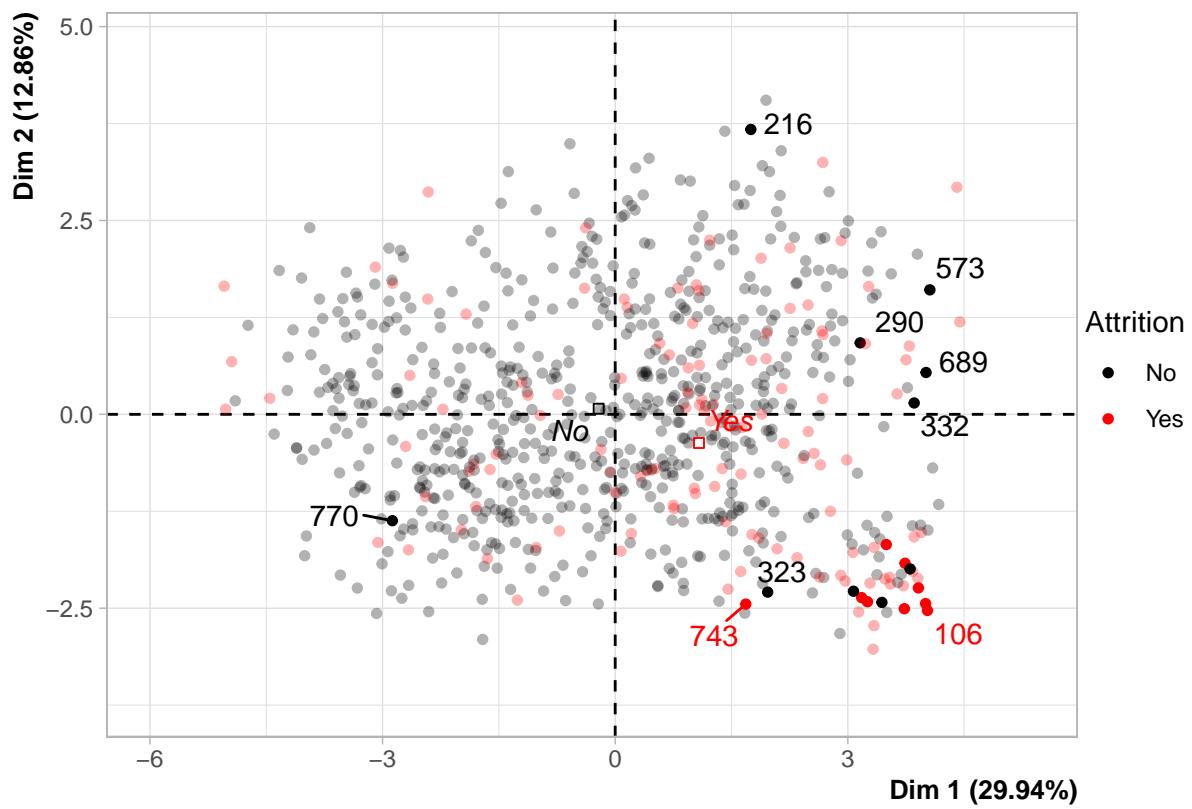


### Contribution et représentation des données

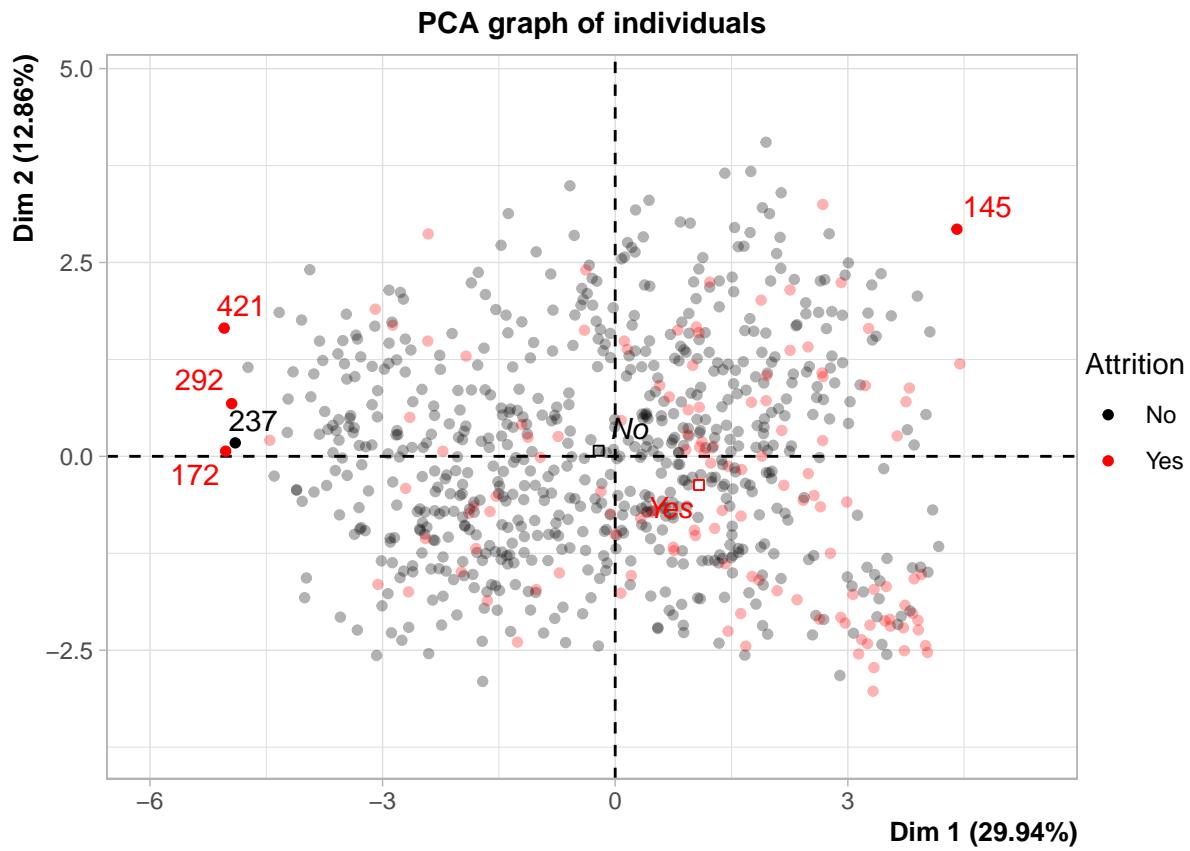
```
plot(res.pca.log, select="cos2 0.82", choix="ind", habillage = 16)
```

```
## Warning: ggrepel: 10 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

PCA graph of individuals



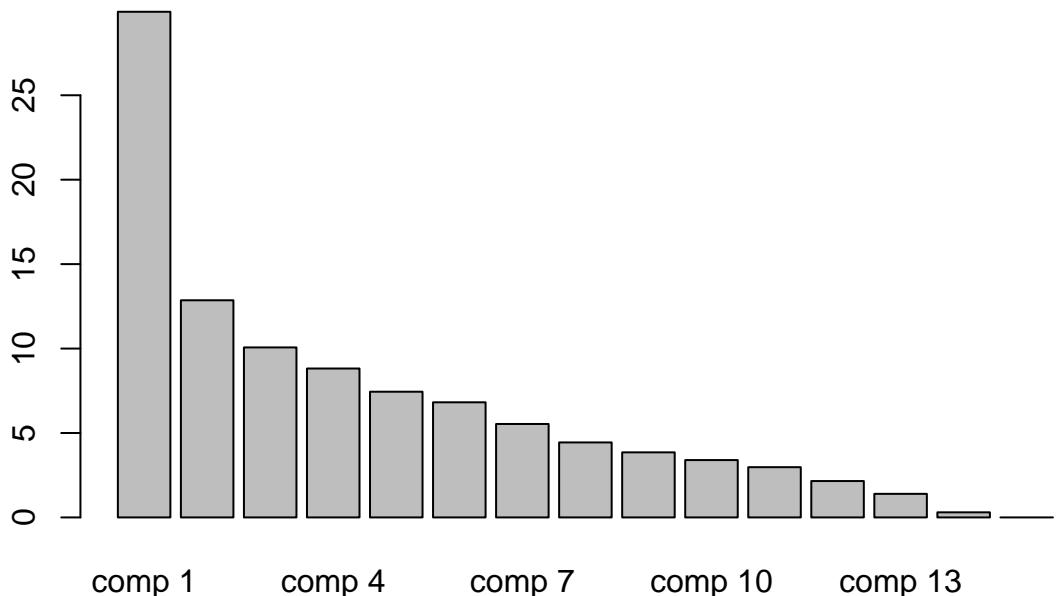
```
plot(res.pca.log, select="contrib 5", choix="ind", habillage = 16)
```



```
summary(res.pca.log$eig)
```

```
##      eigenvalue    percentage of variance cumulative percentage of variance
##  Min.   :0.0000   Min.   : 0.000   Min.   : 29.94
##  1st Qu.:0.3847   1st Qu.: 2.565   1st Qu.: 65.41
##  Median :0.6661   Median : 4.441   Median : 85.92
##  Mean   :1.0000   Mean   : 6.667   Mean   : 78.46
##  3rd Qu.:1.2197   3rd Qu.: 8.131   3rd Qu.: 97.23
##  Max.   :4.4915   Max.   :29.943   Max.   :100.00
```

```
barplot(res.pca.log$eig[,2])
```



chaque composante en pourcentage. On remarque que les 2 premiers axes suffisent car les autres apportent moins de 10%...

```
usefull_col <- (res.pca.log$var$contrib[,1] > median(res.pca.log$var$contrib[,1])) | (res.pca.log$var$contrib[,1] == 1)
```

```
##          Age           DailyRate      DistanceFromHome
##        TRUE           FALSE           FALSE
## EmployeeNumber       HourlyRate      MonthlyIncome
##      FALSE           TRUE            TRUE
## MonthlyRate       NumCompaniesWorked PercentSalaryHike
##      TRUE           TRUE            TRUE
## TotalWorkingYears TrainingTimesLastYear YearsAtCompany
##      TRUE           TRUE            TRUE
## YearsInCurrentRole YearsSinceLastPromotion YearsWithCurrManager
##      TRUE           FALSE            TRUE
```

## AFC-MCA

```
data_train_fact <- data_train[, unlist(lapply(data_train, is.factor))]
dim(data_train_fact)
```

```
## [1] 784 17
```

```
head(data_train_fact)
```

```
##   Attrition BusinessTravel          Department Education EducationField
## 1     No    Travel_Rarely Research & Development      2      Medical
## 2     No    Travel_Rarely Research & Development      2      Medical
## 3    Yes    Travel_Rarely             Sales          1 Marketing
## 4     No    Travel_Rarely Research & Development      4 Life Sciences
```

```

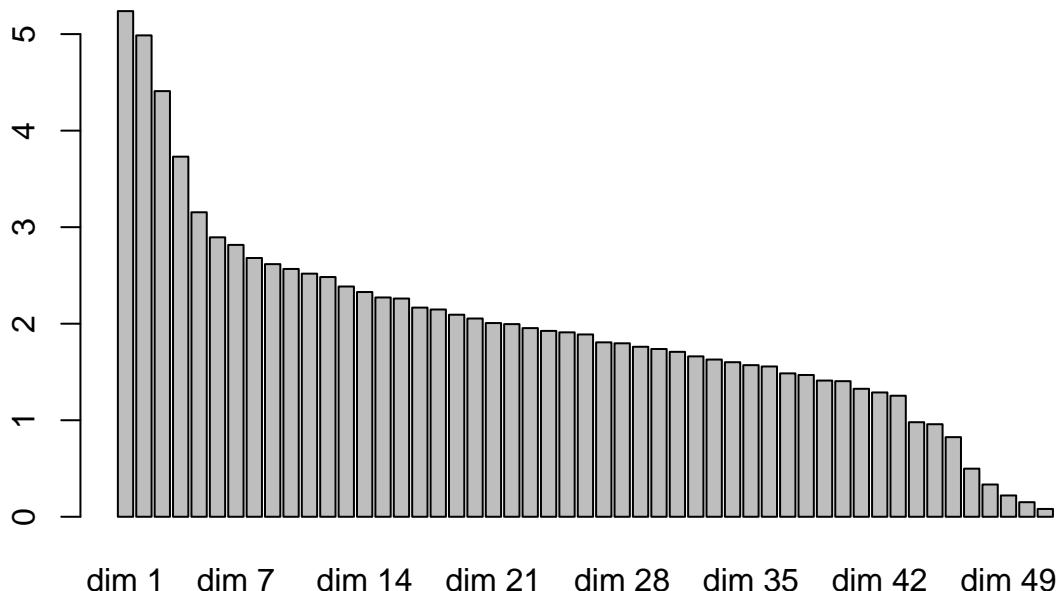
## 5      Yes Travel_Rarely Research & Development      1      Medical
## 6      No  Travel_Rarely Research & Development      3 Life Sciences
## EnvironmentSatisfaction Gender JobInvolvement JobLevel
## 1                  4   Male            3        4
## 2                  2   Male            3        2
## 3                  2   Male            3        1
## 4                  2   Male            3        3
## 5                  2   Male            3        3
## 6                  3 Female          2        3
## JobRole JobSatisfaction MaritalStatus OverTime
## 1     Research Director          4 Divorced    No
## 2 Manufacturing Director         2 Divorced    No
## 3     Sales Representative       2 Single     No
## 4 Healthcare Representative     2 Single     No
## 5           Manager             3 Married    Yes
## 6 Manufacturing Director        2 Divorced    Yes
## PerformanceRating RelationshipSatisfaction StockOptionLevel WorkLifeBalance
## 1             4                   3           1        2
## 2             4                   4           2        3
## 3             3                   2           0        3
## 4             3                   4           0        3
## 5             3                   4           0        4
## 6             3                   3           1        3

```

```

library(FactoMineR)
res.mca = MCA(data_train_fact, graph = FALSE)
barplot(res.mca$eig[,2])

```

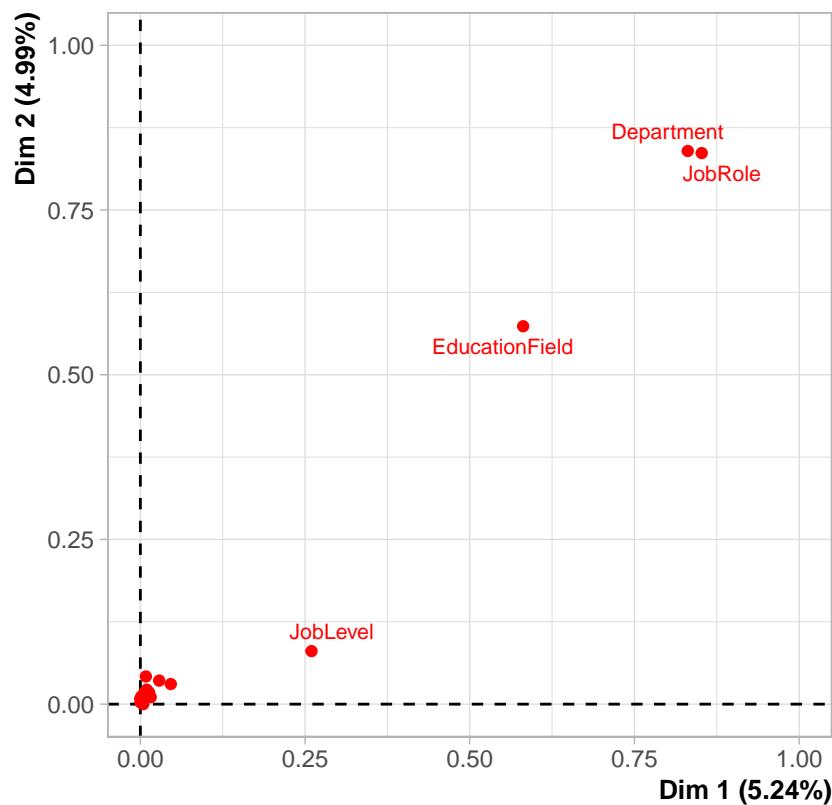


```

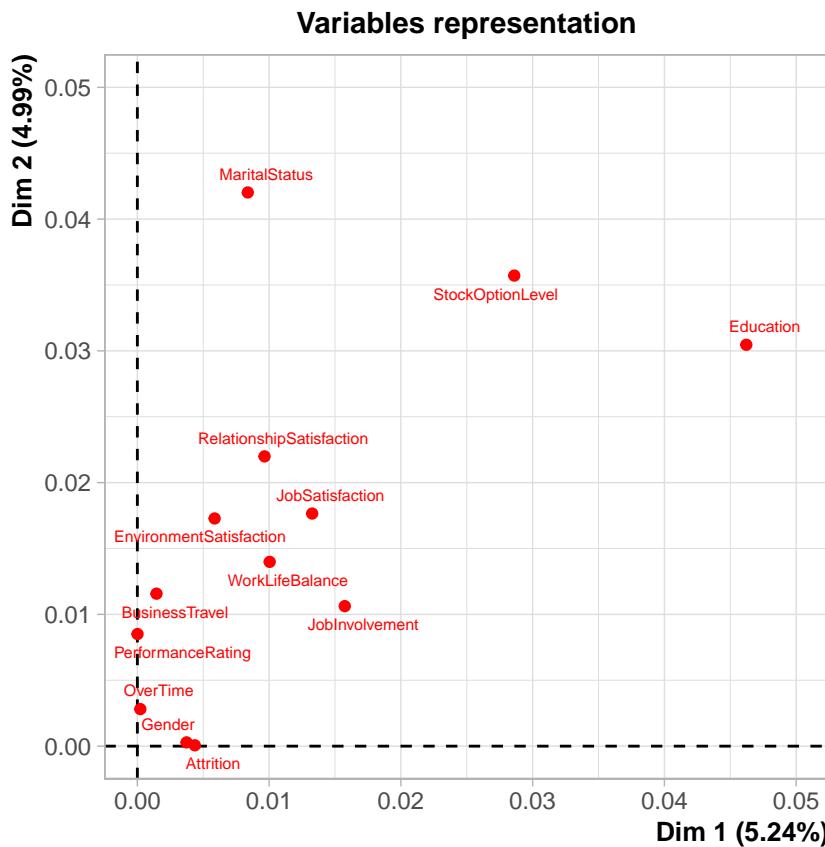
plot(res.mca, choix = "var", cex = 0.7)

```

### Variables representation



```
plot(res.mca, choix = "var", xlim = c(0, 0.05), ylim = c(0, 0.05), cex = 0.5)
```



```
attach(data_train)
chisq.test(table(EducationField, JobRole))
```

```
##
## Pearson's Chi-squared test
##
## data: table(EducationField, JobRole)
## X-squared = 506.77, df = 40, p-value < 2.2e-16
```

```
chisq.test(table(EducationField, Department))
```

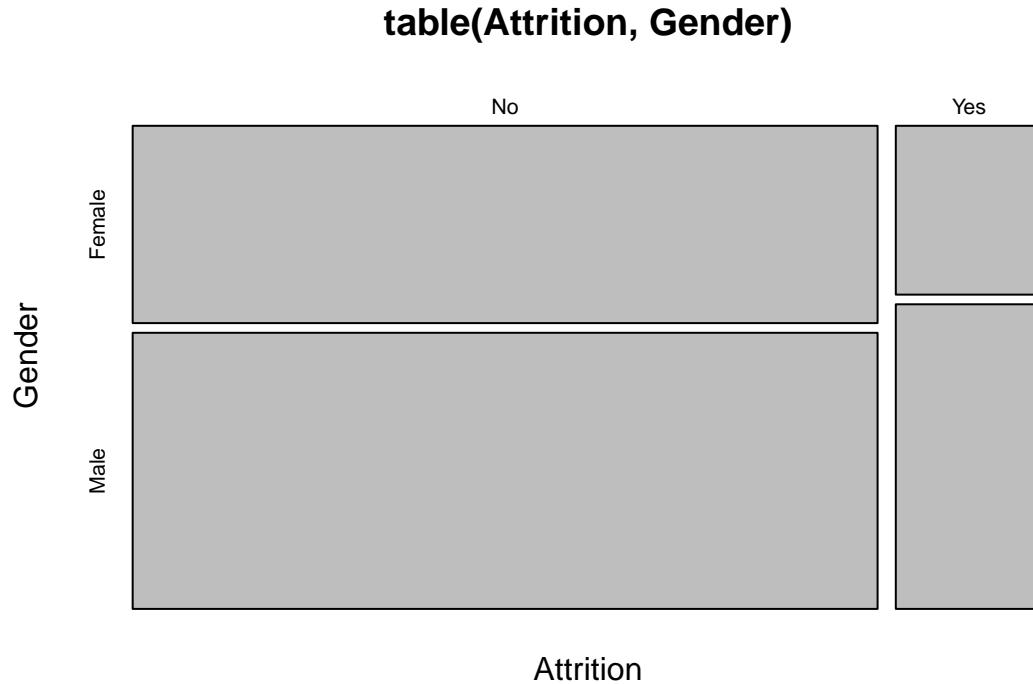
```
##
## Pearson's Chi-squared test
##
## data: table(EducationField, Department)
## X-squared = 620.76, df = 10, p-value < 2.2e-16
```

```
chisq.test(table(JobRole, Department))
```

```
##
## Pearson's Chi-squared test
##
## data: table(JobRole, Department)
## X-squared = 1351.6, df = 16, p-value < 2.2e-16
```

On a une p-value < 0.05, les variables sont donc effectivement liées.

```
attach(data_train)
plot(table(Attrition,Gender))
```



```
chisq.test(table(Attrition,Gender))
```

```
##  
## Pearson's Chi-squared test with Yates' continuity correction  
##  
## data: table(Attrition, Gender)  
## X-squared = 1.3788, df = 1, p-value = 0.2403
```

Finalement le test chi 2 nous montre l'indépendance, démontrant que notre modèle n'est pas parfait.

## R Markdown - Classification & Prédiction

# 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

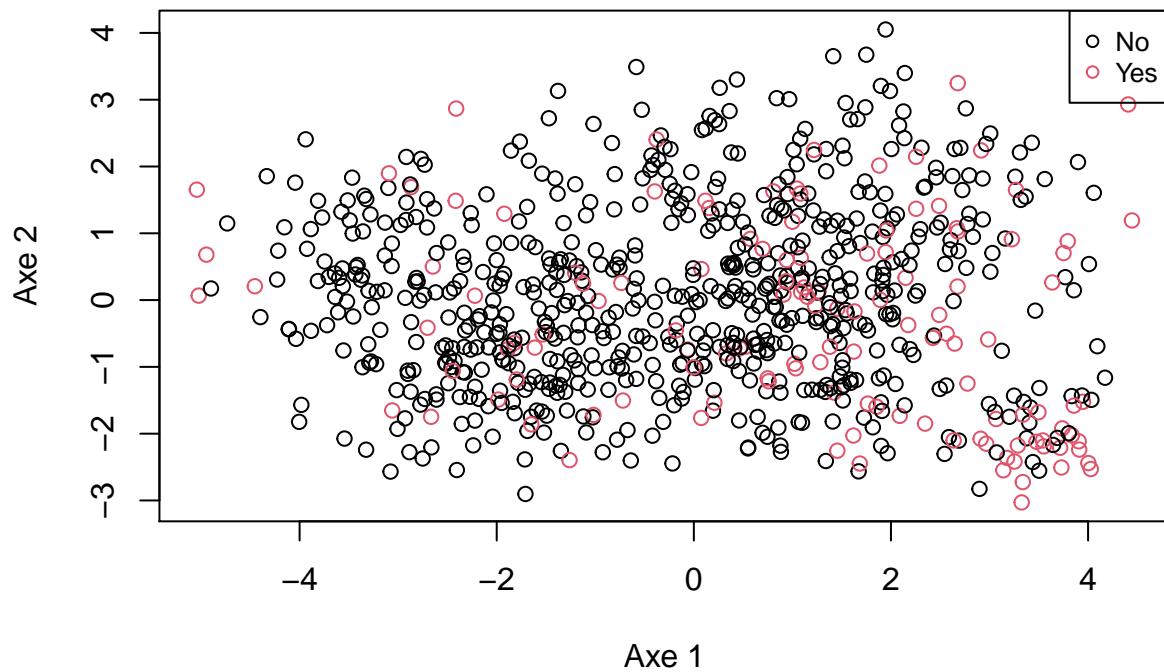
```

## 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$Attrition
coord_data_train <- PCA(data_train_log, scale.unit = TRUE, graph = FALSE)$ind$coord[,1:2]
plot(coord_data_train[,1], coord_data_train[,2], col = data_train$Attrition, xlab = "Axe 1", ylab = "Axe 2",
legend('topright', legend = levels(data_train$Attrition), col = 1:2, cex = 0.8, pch = 1)

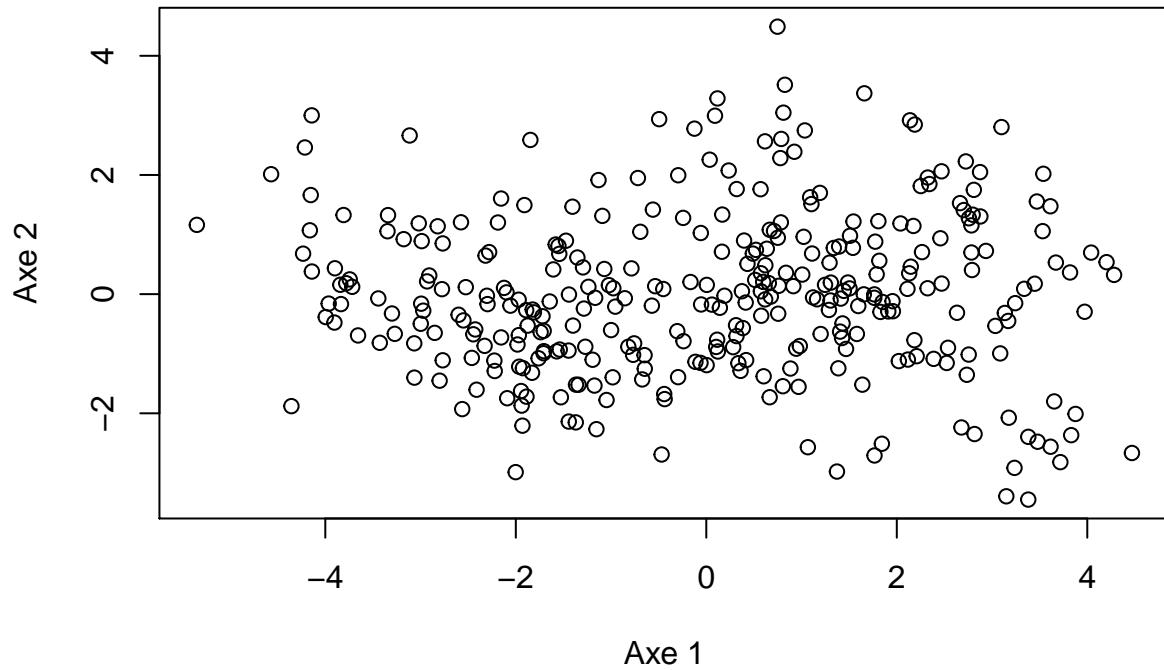
```



```

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")

```

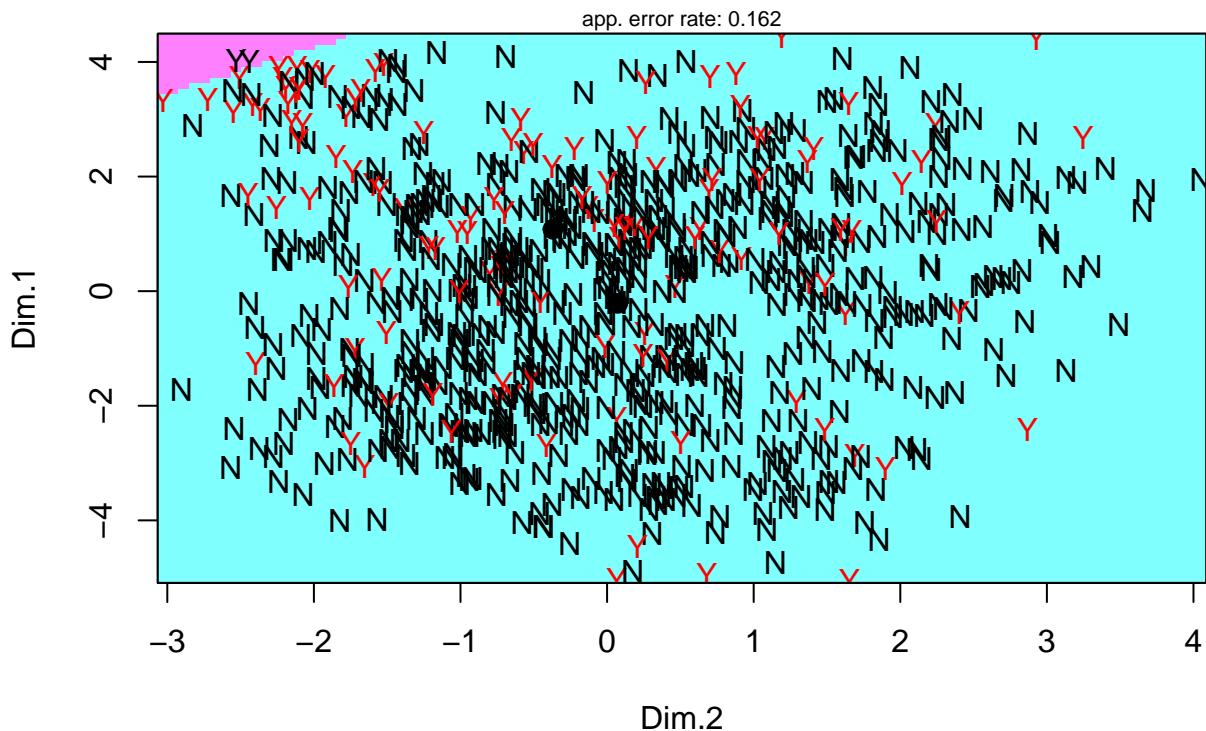


## Classification

### LDA - QDA

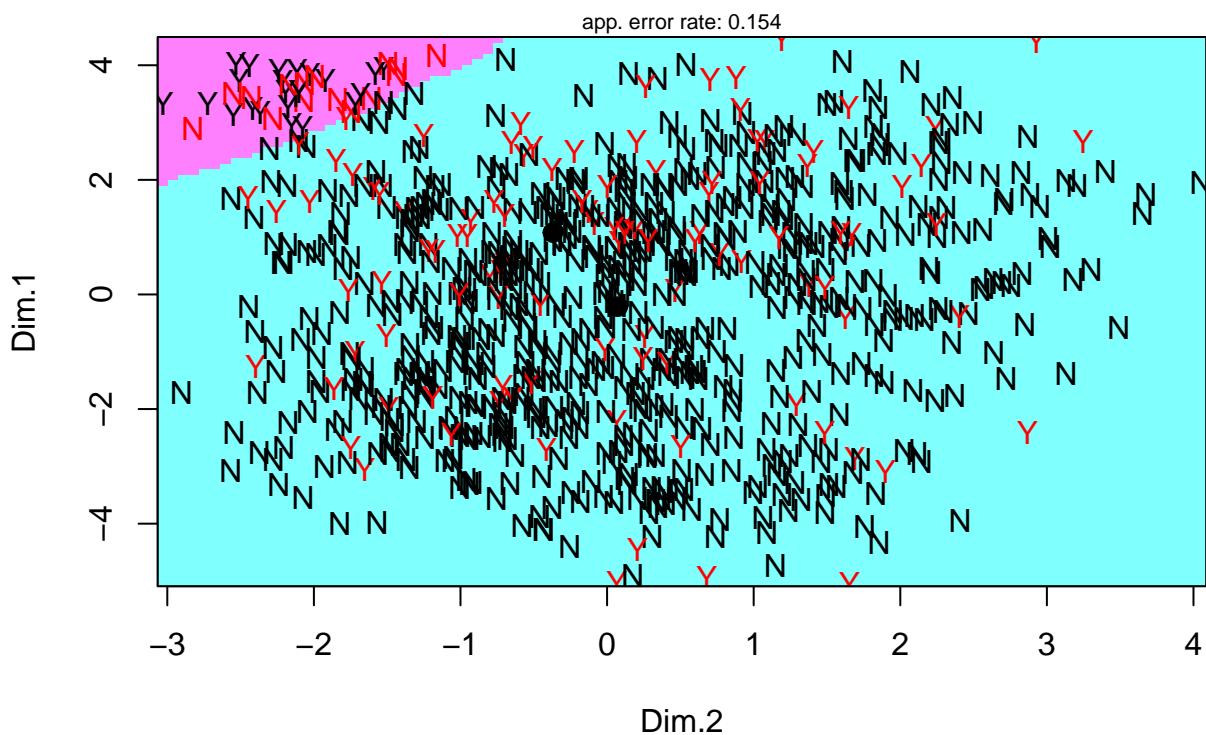
```
library(klaR)
partimat(coord_data_train, grouping = data_train_num$Attrition, method = "lda")
```

## Partition Plot



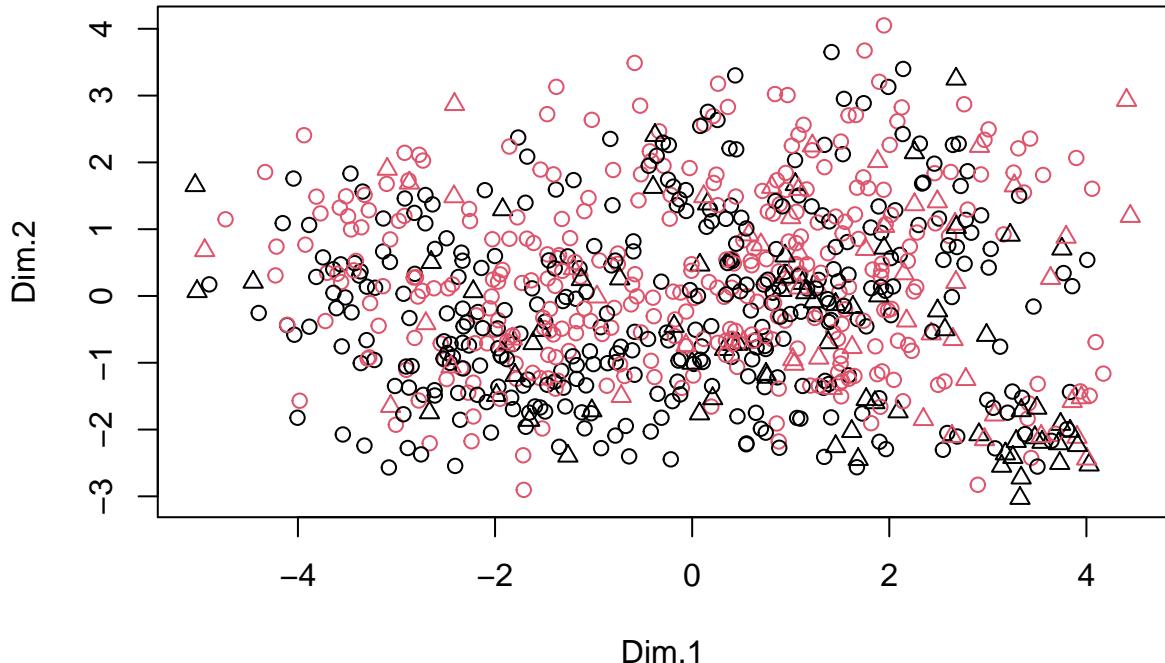
```
partimat(coord_data_train, grouping = data_train_num$Attrition, method = "qda")
```

## Partition Plot



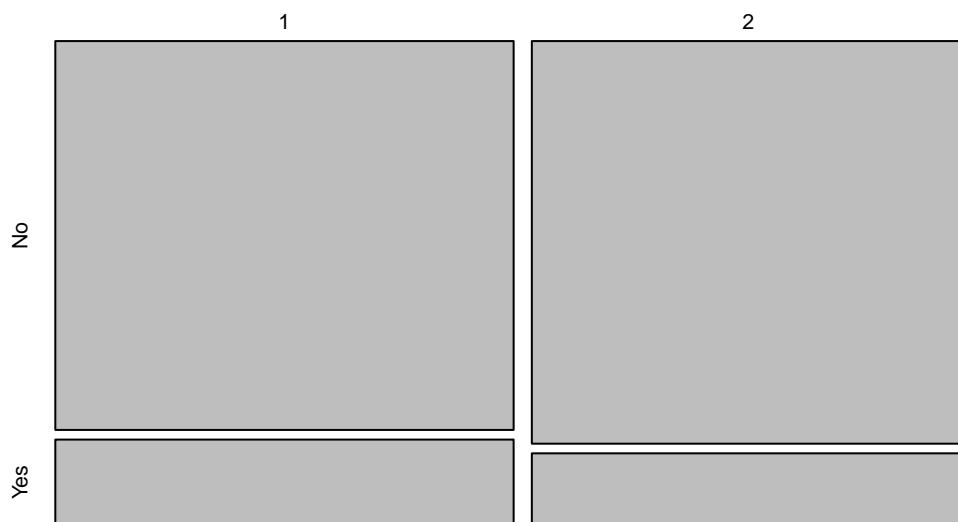
## KMeans

```
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))
```

**table(res.kmeans\$cluster, data\_train\$Attrition)**

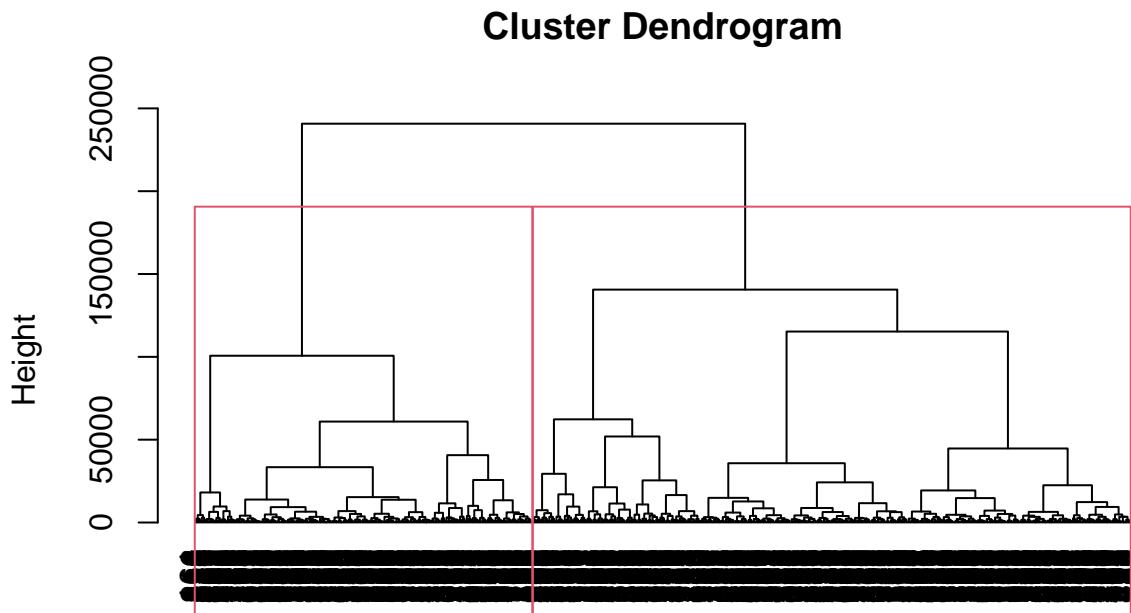


## CAH

```

## 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)

```



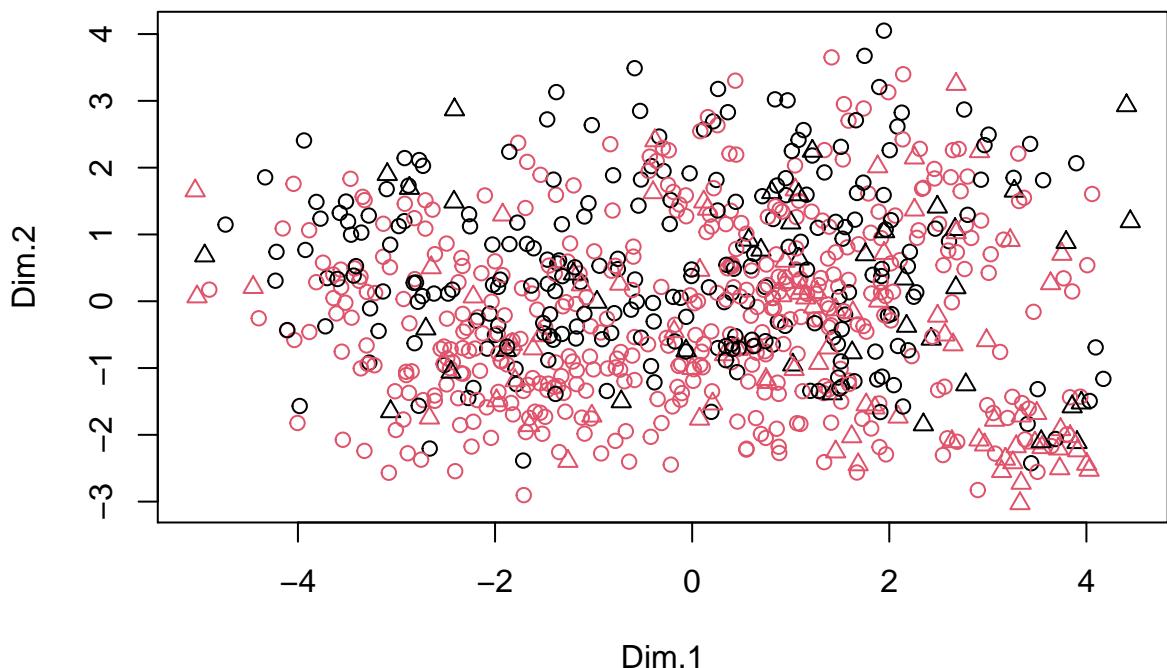
$\text{dist}(\text{data\_train\_num})$   
 $\text{hclust}(*, \text{"ward.D2"})$

```

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

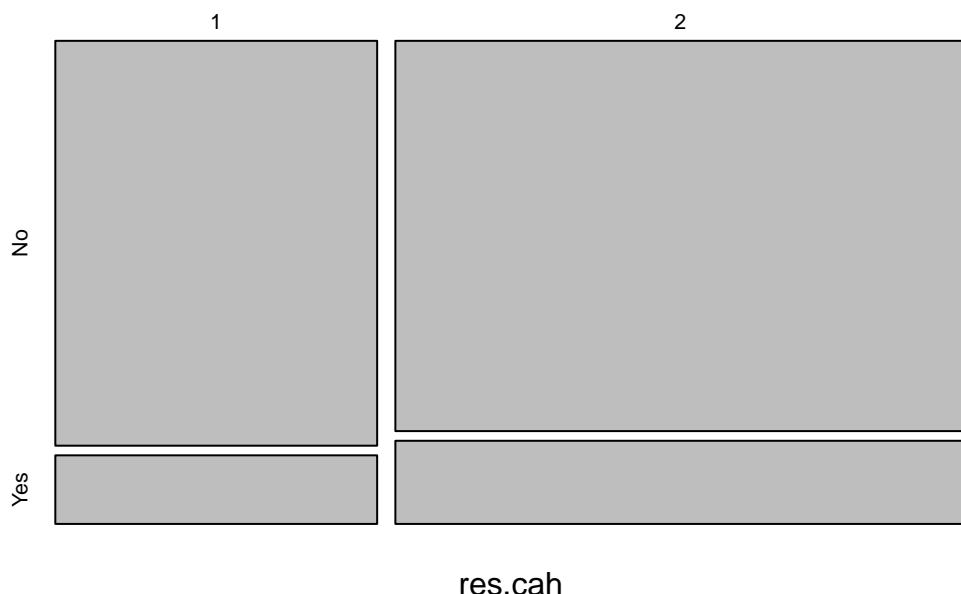
plot(coord_data_train, col = res.cah, pch = as.numeric(data_train$Attrition))

```



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

**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 prédition 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

```

library(MASS)

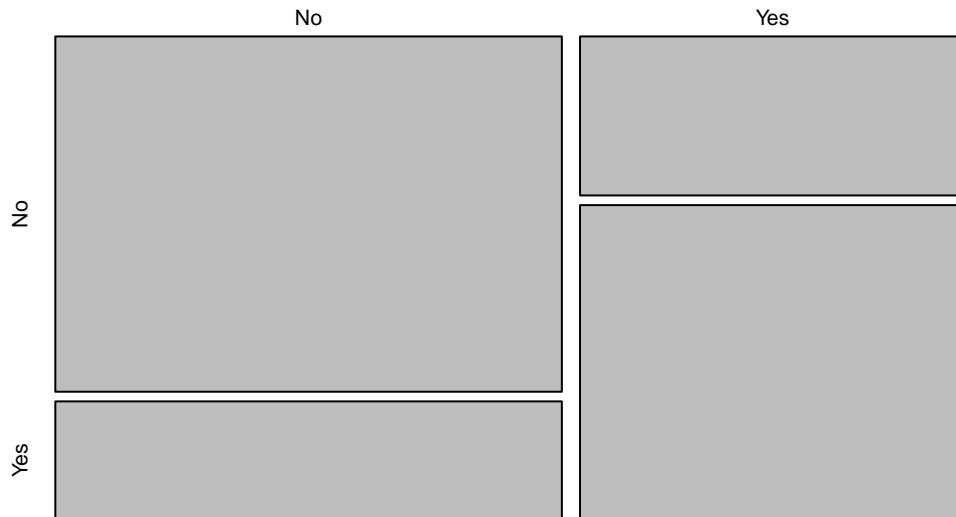
## Modèle
res.lda <- lda(data_train_bal[-16], grouping = data_train_bal$Attrition)
res.qda <- qda(data_train_bal[-16], grouping = data_train_bal$Attrition)

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

## 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)

```

## conf.lda

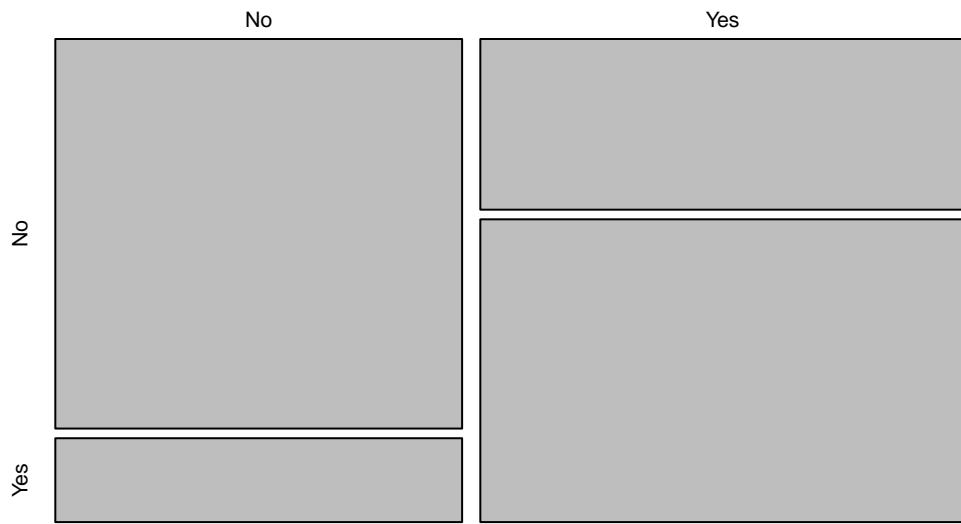


```

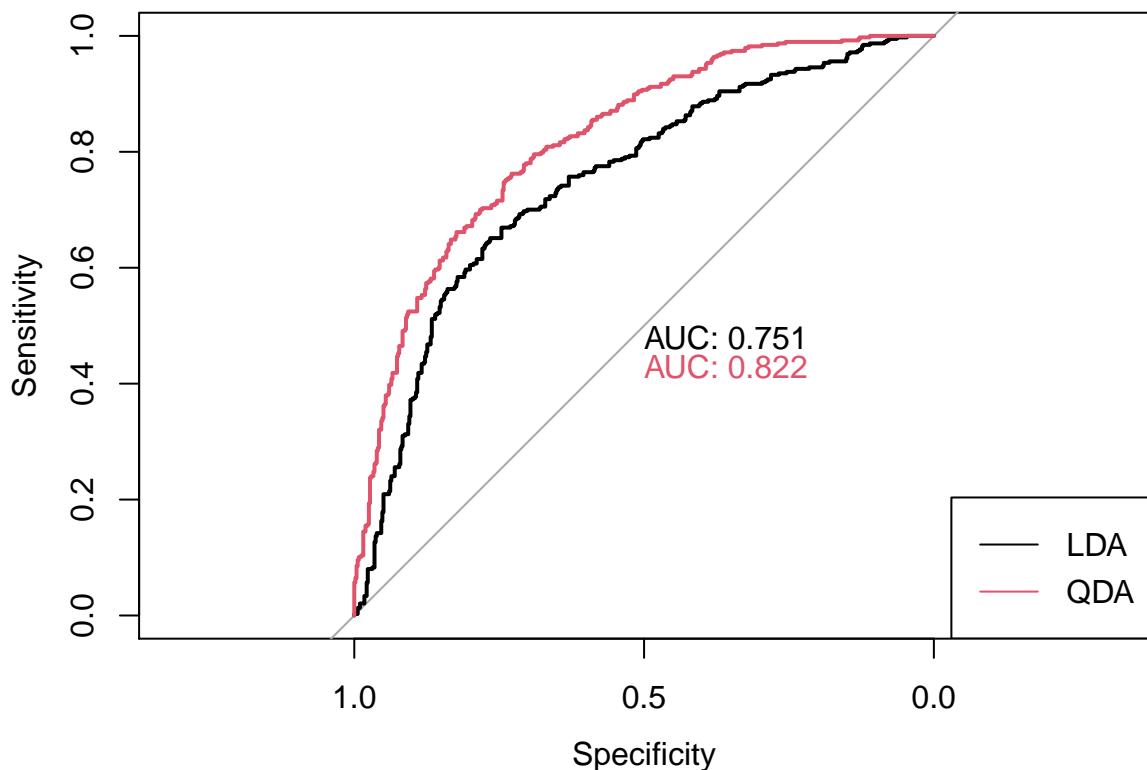
conf.qda <- table(pred.qda$class, data_train_bal$Attrition)
accuracy.qda <- (conf.qda[1,1] + conf.qda[2,2]) / sum(conf.qda)
plot(conf.qda)

```

## 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=1)

## correctness rate: 0.69093; starting variables (15): Age, DailyRate, DistanceFromHome, EmployeeNumber, HourlyRate, MonthlyIncome, MonthlyRate, NumCompaniesWorked, PercentSalaryHike, TrainingTimesLastYear, YearsAtCompany, YearsInCurrentRole, YearsSinceLastPromotion, YearsWithCurrManager
## correctness rate: 0.70204; out: "TotalWorkingYears"; variables (14): Age, DailyRate, DistanceFromHome, EmployeeNumber, HourlyRate, MonthlyIncome, MonthlyRate, NumCompaniesWorked, PercentSalaryHike, TrainingTimesLastYear, YearsAtCompany, YearsInCurrentRole, YearsSinceLastPromotion, YearsWithCurrManager
## hr.elapsed min.elapsed sec.elapsed
##      0.000      0.000     1.992

stepwise.lda

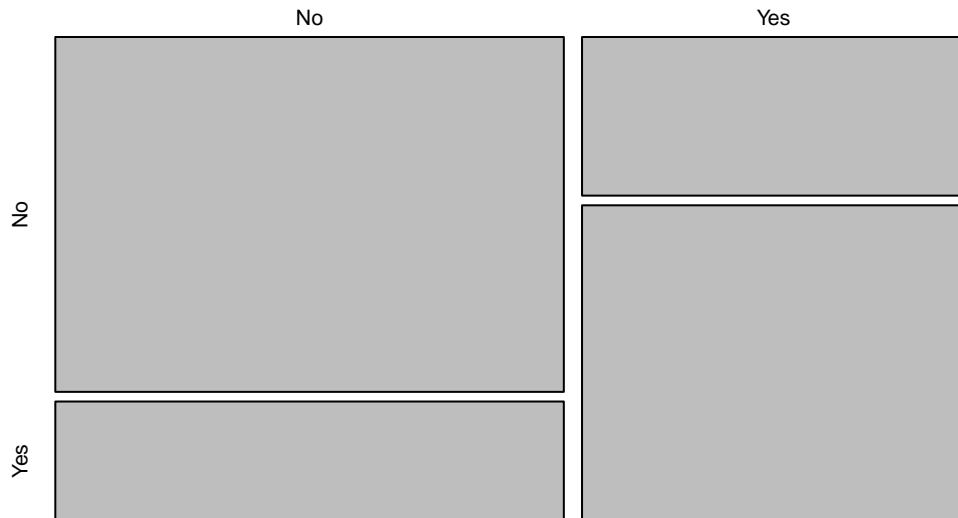
## method      : lda
## final model : data_train_bal$Attrition ~ Age + DailyRate + DistanceFromHome +
##                 EmployeeNumber + HourlyRate + MonthlyIncome + MonthlyRate +
##                 NumCompaniesWorked + PercentSalaryHike + TrainingTimesLastYear +
##                 YearsAtCompany +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

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

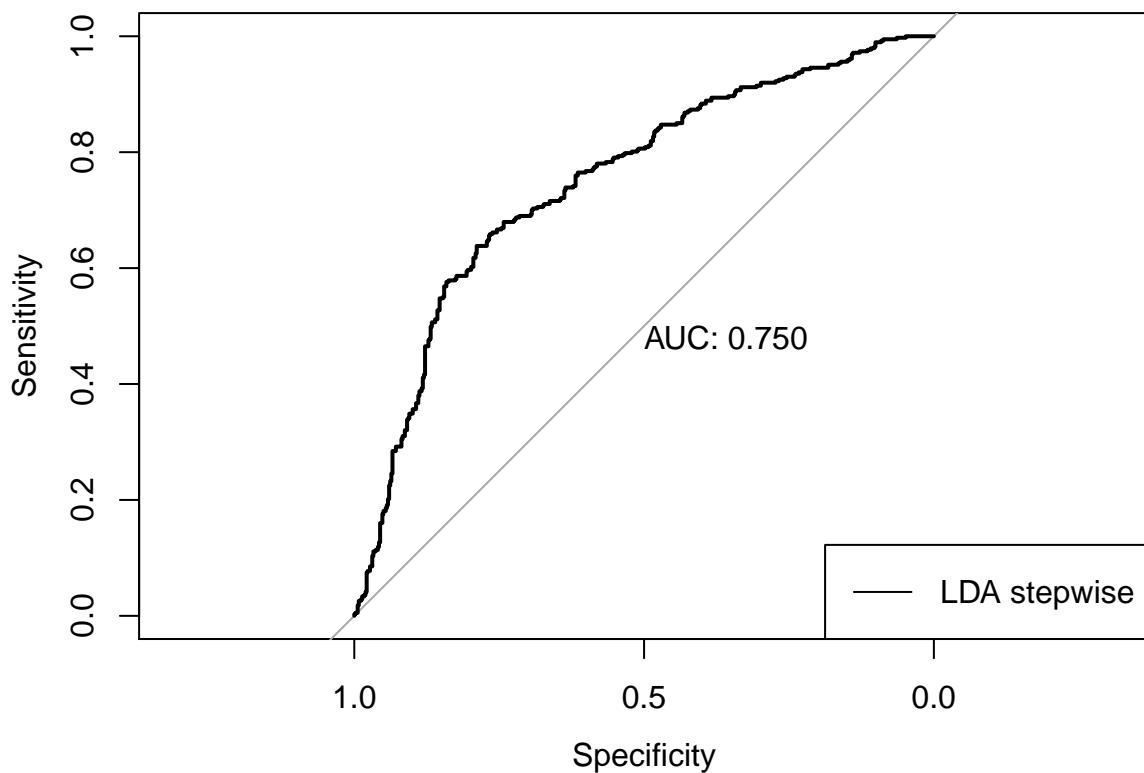
## Prédiction
pred.stepwise.lda <- predict(res.stepwise.lda, newdata = data_train_bal[-16])

## 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)
```

## 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")
```

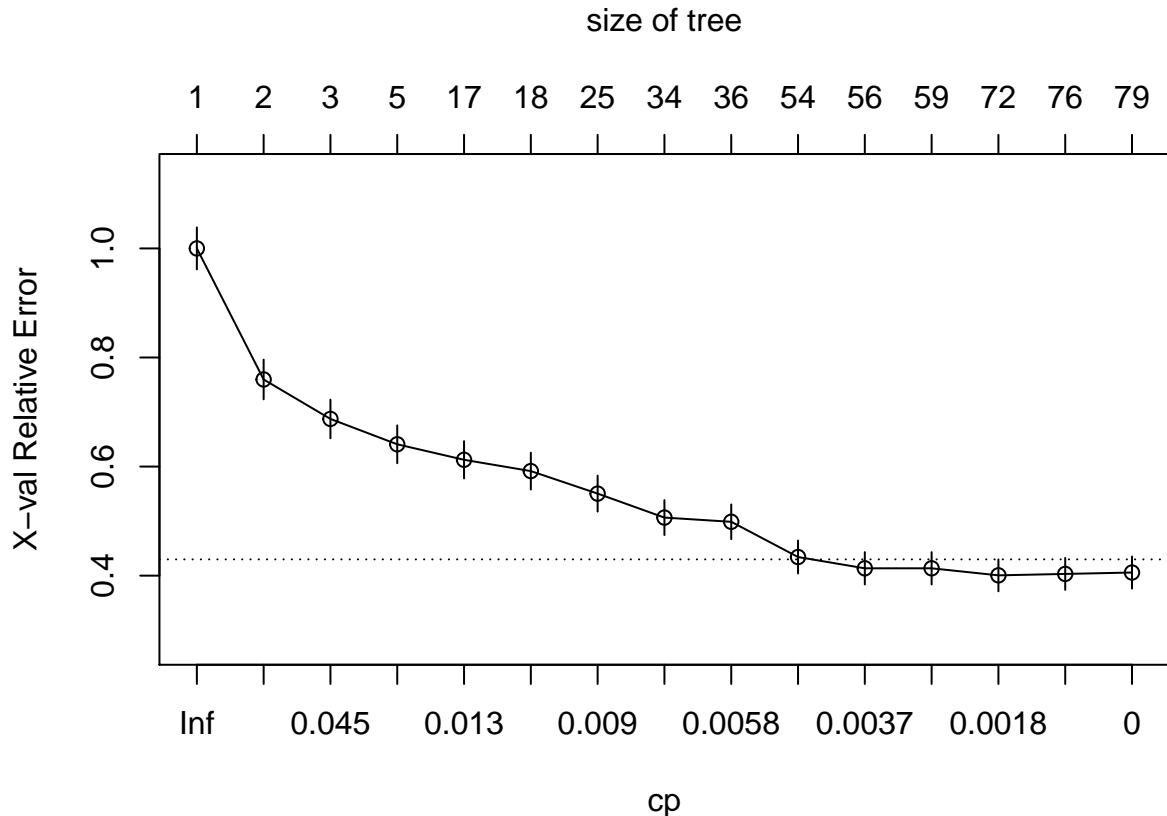


## CART

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

### *## Modèle*

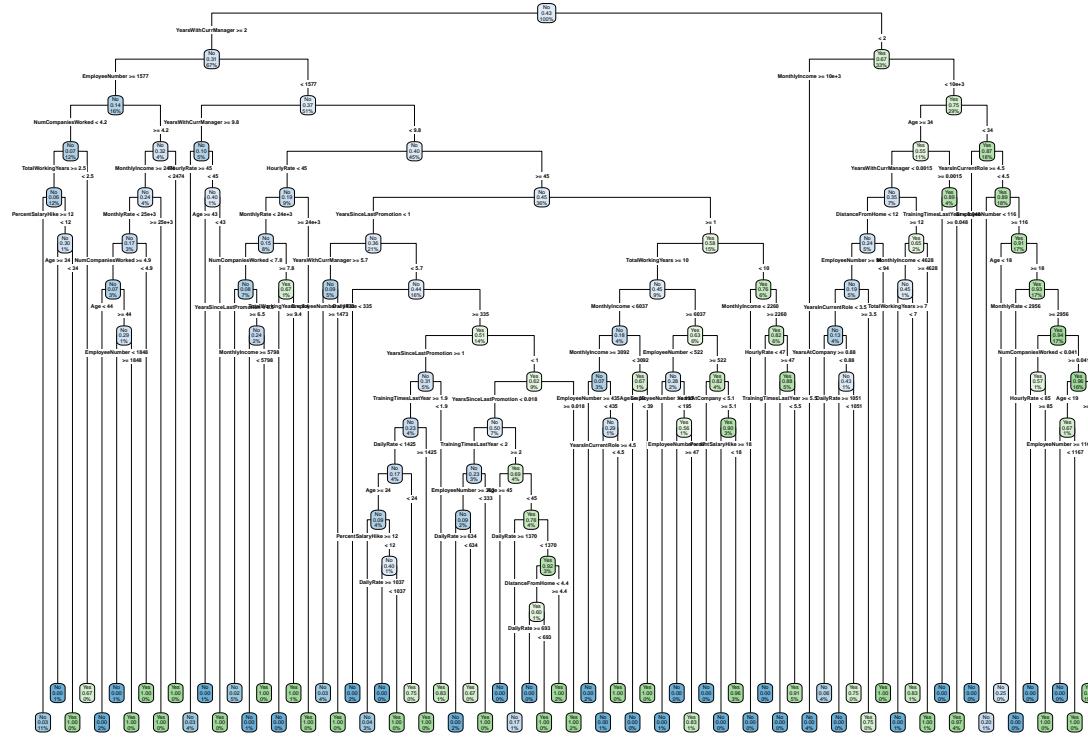
```
arbre.cart = rpart(data_train_bal$Attrition ~ ., data = data_train_bal[-16], control = rpart.control(minsplit=1))
plotcp(arbre.cart)
```



### *## 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
```



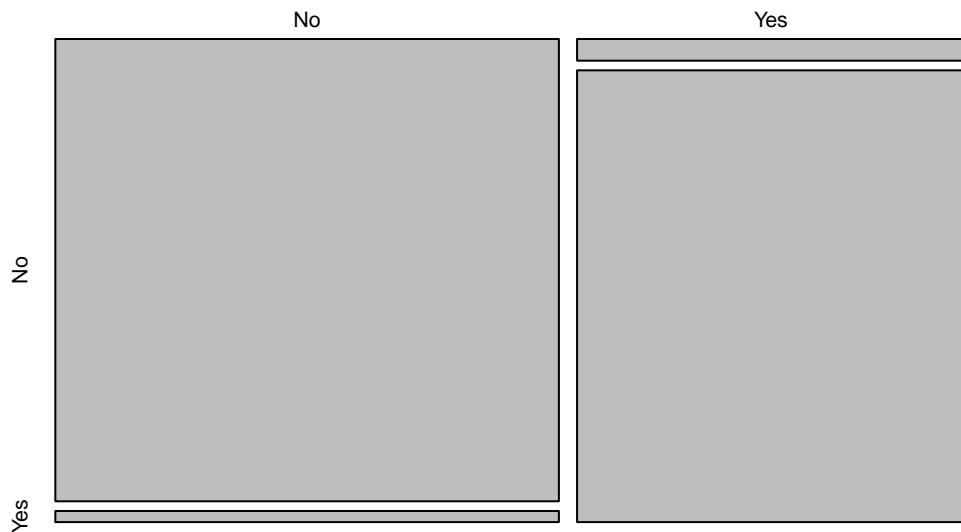
### **## Prédiction**

```
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]
```

### **## Table de confusion**

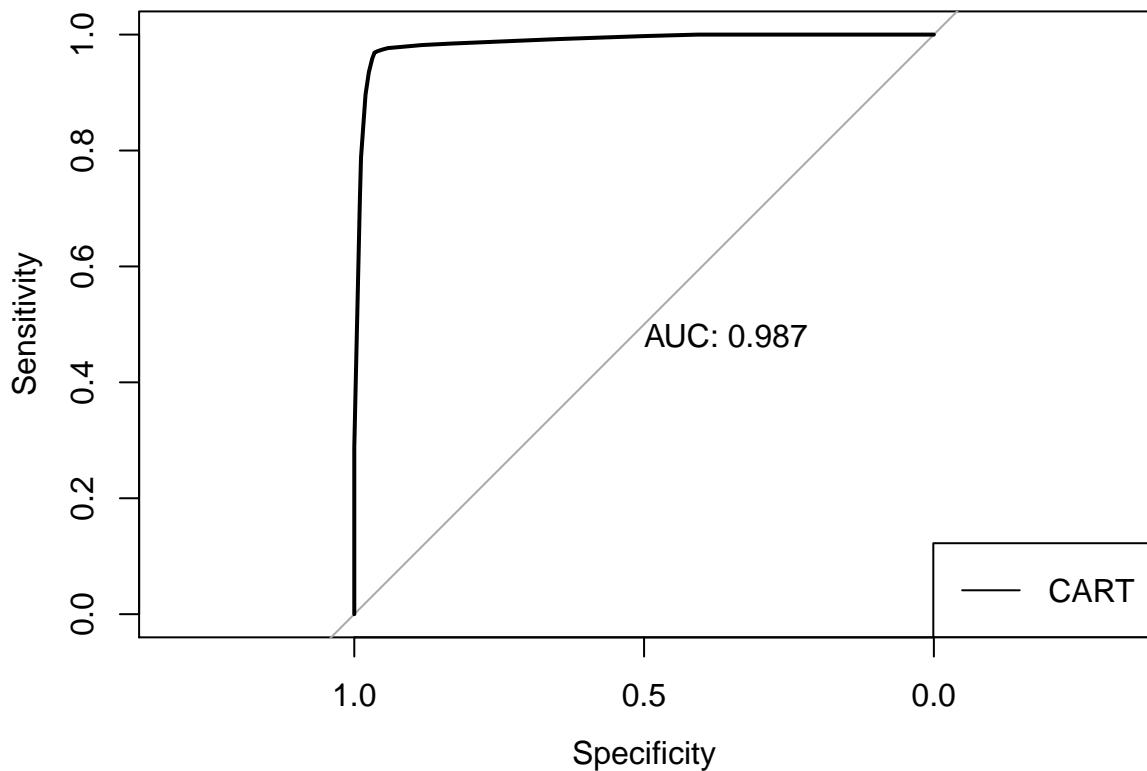
```
conf.cart <- table(pred.cart.class, data_train_bal$Attrition)
accuracy.cart <- (conf.cart[1,1] + conf.cart[2,2]) / sum(conf.cart)
plot(conf.cart)
```

## conf.cart



## pred.cart.class

```
## 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)

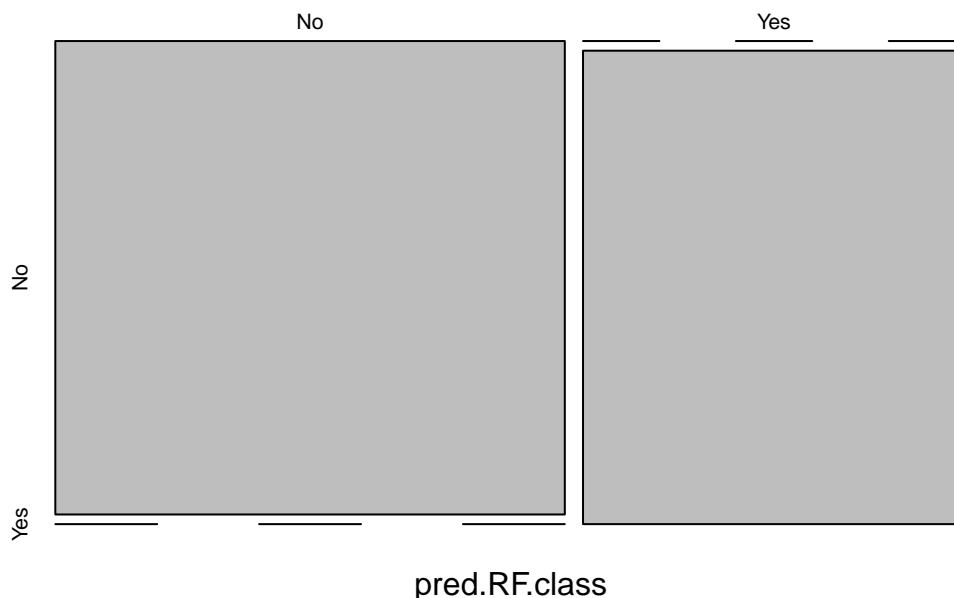
## Modèle
res.RF <- randomForest(data_train_bal$Attrition ~ ., data_train_bal[-16])
res.RF

##
## 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

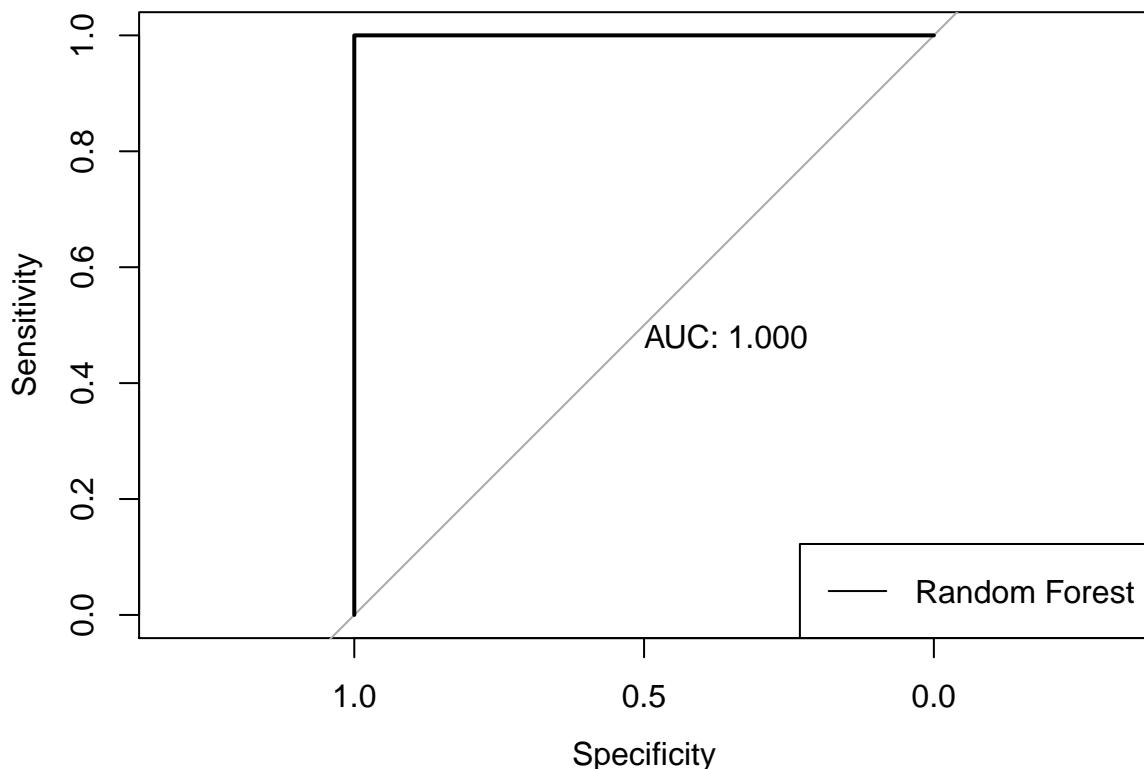
## Prédiction
pred.RF.class <- predict(res.RF, newdata = data_train_bal[-16], type="class")
pred.RF.prob <- predict(res.RF, newdata = data_train_bal[-16], type = "prob")[,2]

## 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)
```

**conf.RF**



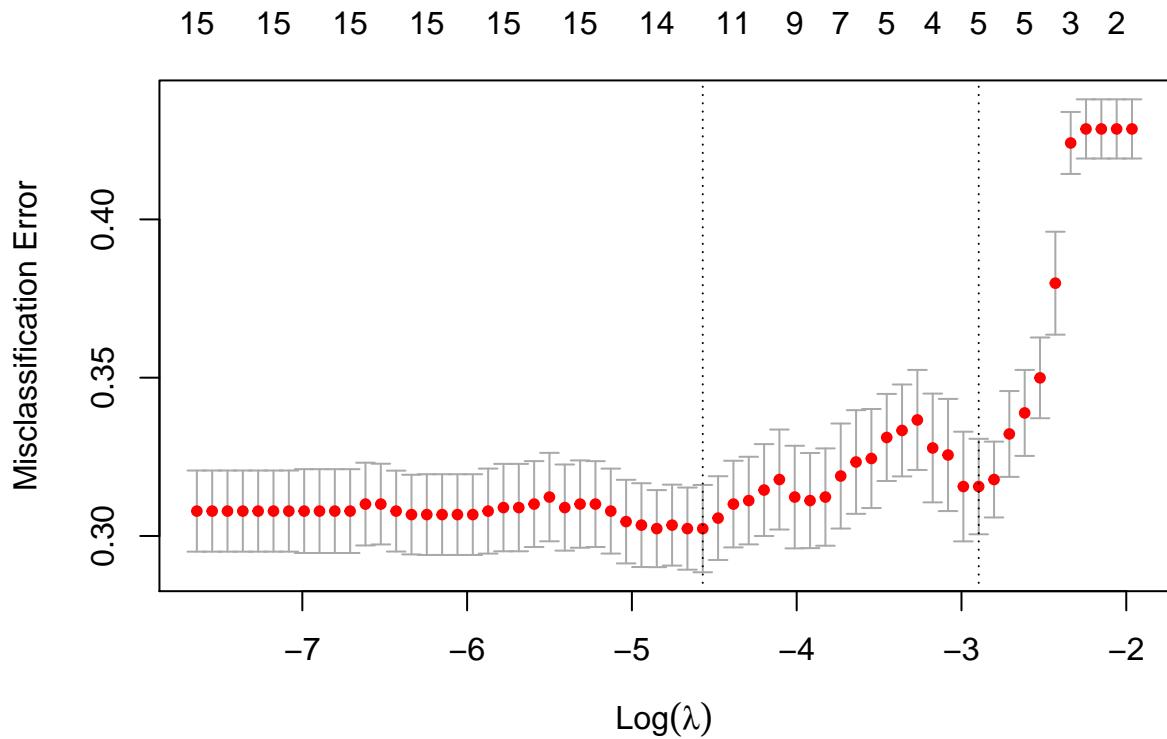
```
## courbe ROC
ROC.RF <- roc(data_train_bal$Attrition, pred.RF.prob)
plot(ROC.RF, print.auc=TRUE, print.auc.y = 0.5, col = 1)
legend("bottomright", lwd = 1, col = 1, "Random Forest")
```



### Regression Logistique Lasso

```
library(glmnet)

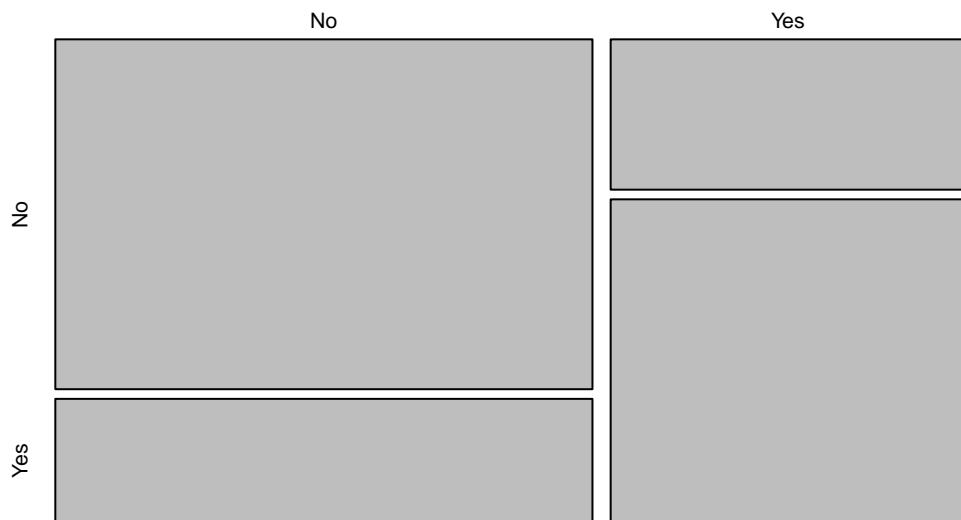
## Modèle
res.Lasso <- glmnet(as.matrix(data_train_bal[-16]), data_train_bal$Attrition, family='binomial')
cv.Lasso <- cv.glmnet(as.matrix(data_train_bal[-16]), data_train_bal$Attrition, family="binomial", type="cv")
plot(cv.Lasso)
```



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

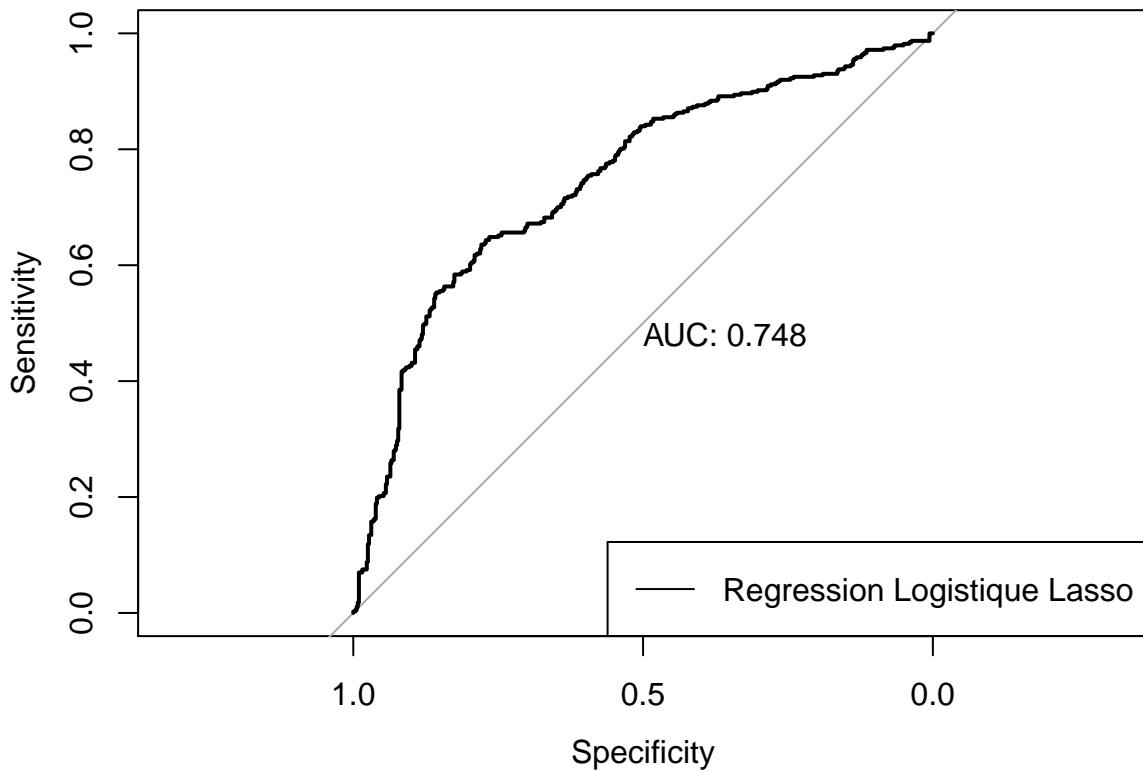
## Table de confusion
conf.lasso <- table(pred.lasso.class, data_train_bal$Attrition)
accuracy.lasso <- (conf.lasso[1,1] + conf.lasso[2,2]) / sum(conf.lasso)
plot(conf.lasso)
```

## conf.lasso



pred.lasso.class

```
## 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.RL)
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

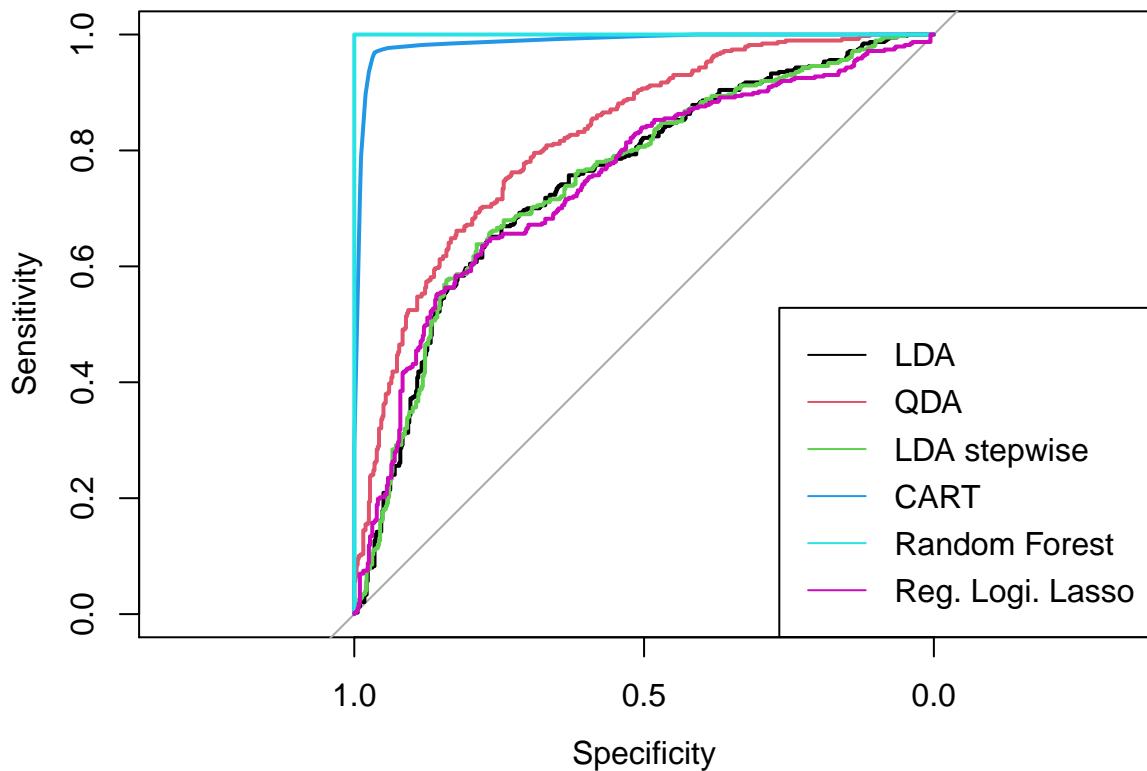
apply(result, 1, which.max )

## 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.

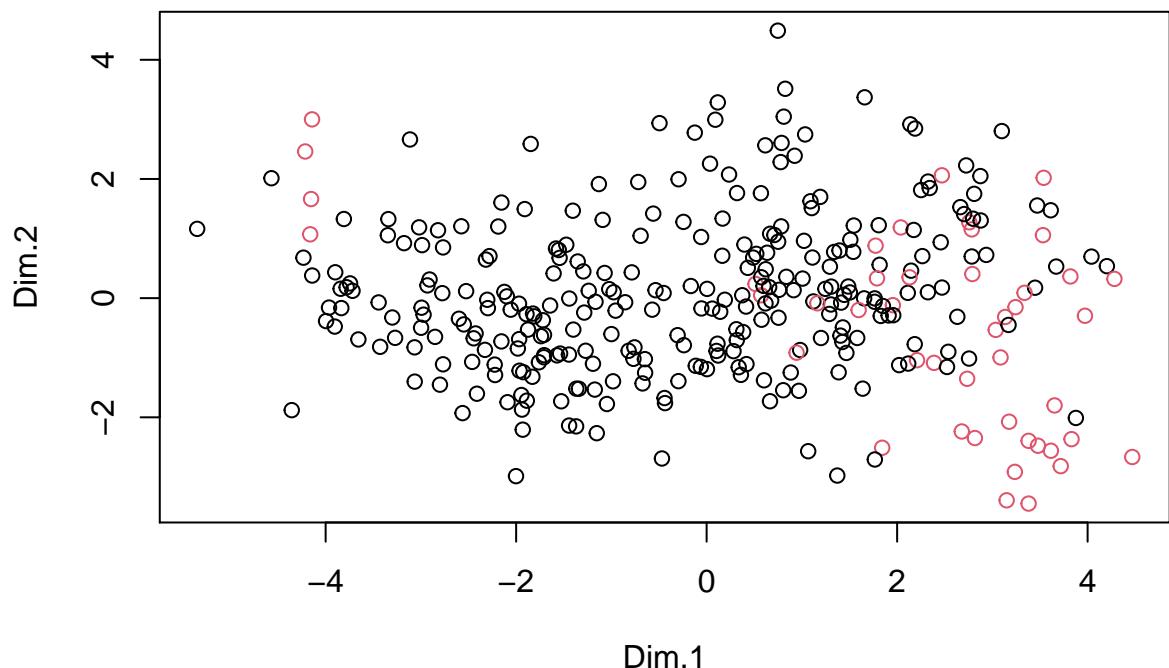
## Resolution de notre problème avec Random Forest

```

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 = ',')
```