HR Analytics - Why are employees leaving?

Ingrid Brizotti

The goal of this study is to investigate the causes that make employees leave their jobs. Also compare logistic regression and decision tree to solve this puzzle.

Dataset: Kaggle (https://www.kaggle.com/ludobenistant/hr-analytics)

Approach: find out what are the most relevant characteristics that make employees leave the company

Techniques used: Logistic regression and decision tree

Keywords: logistic regression, decision tree, supervised machine learning, HR analytics

Steps:

1) Prepare the data

2) Exploratory analysis

3) Data transformation

4) Divide between train and test set

5) Logistic regression

6) Decision tree

7) Conclusions:

- What are the most important aspects that are decisive to employees leave their jobs?
- What is more accurate to predict these aspects: logistic regression or decision tree?

1) Prepare the data

```
library(ggplot2)
library(scales)
library(gmodels) #logistic regression
library(rpart) #decision tree
#vizualise tree
#library(rattle)
#library(rpart.plot)
#library(RColorBrewer)
library(ROCR)
```

```
## Loading required package: gplots
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```r
library(rafalib)

# Load the data
setwd("/Users/ingridbrizotti/Desktop/GitHub/HR_Analytics_Kaggle/")

hr = read.csv("HR_comma_sep.csv")
dim(hr)
```

```
## [1] 14999    10
```

```r
# [1] 14999    10

attach(hr)

summary(hr)
```

```
##  satisfaction_level last_evaluation  number_project  average_montly_hours
##  Min.   :0.0900     Min.   :0.3600   Min.   :2.000   Min.   : 96.0
##  1st Qu.:0.4400     1st Qu.:0.5600   1st Qu.:3.000   1st Qu.:156.0
##  Median :0.6400     Median :0.7200   Median :4.000   Median :200.0
##  Mean   :0.6128     Mean   :0.7161   Mean   :3.803   Mean   :201.1
##  3rd Qu.:0.8200     3rd Qu.:0.8700   3rd Qu.:5.000   3rd Qu.:245.0
##  Max.   :1.0000     Max.   :1.0000   Max.   :7.000   Max.   :310.0
##
##  time_spend_company Work_accident         left
##  Min.   : 2.000     Min.   :0.0000   Min.   :0.0000
##  1st Qu.: 3.000     1st Qu.:0.0000   1st Qu.:0.0000
##  Median : 3.000     Median :0.0000   Median :0.0000
##  Mean   : 3.498     Mean   :0.1446   Mean   :0.2381
##  3rd Qu.: 4.000     3rd Qu.:0.0000   3rd Qu.:0.0000
##  Max.   :10.000     Max.   :1.0000   Max.   :1.0000
##
##  promotion_last_5years         sales          salary
##  Min.   :0.00000       sales      :4140   high  :1237
##  1st Qu.:0.00000       technical  :2720   low   :7316
##  Median :0.00000       support    :2229   medium:6446
##  Mean   :0.02127       IT         :1227
##  3rd Qu.:0.00000       product_mng: 902
##  Max.   :1.00000       marketing  : 858
##                        (Other)    :2923
```

```r
# Checking missing in all variables
propmiss <- function(dataframe) {
  m <- sapply(dataframe, function(x) {
    data.frame(
      nmiss=sum(is.na(x)),  # number of missing
      n=length(x),
      propmiss=sum(is.na(x))/length(x) # proportion of missing inside the variable
```

```r
    )
  })
  d <- data.frame(t(m))
  d <- sapply(d, unlist)
  d <- as.data.frame(d)
  d$variable <- row.names(d)
  row.names(d) <- NULL
  d <- cbind(d[ncol(d)],d[-ncol(d)])
  return(d[order(d$propmiss), ])
}
propmiss(hr)
```

```
##                  variable nmiss     n propmiss
## 1      satisfaction_level     0 14999        0
## 2         last_evaluation     0 14999        0
## 3          number_project     0 14999        0
## 4    average_montly_hours     0 14999        0
## 5      time_spend_company     0 14999        0
## 6           Work_accident     0 14999        0
## 7                    left     0 14999        0
## 8   promotion_last_5years     0 14999        0
## 9                   sales     0 14999        0
## 10                 salary     0 14999        0
```

The data has no missing values.

2) Exploratory analysis

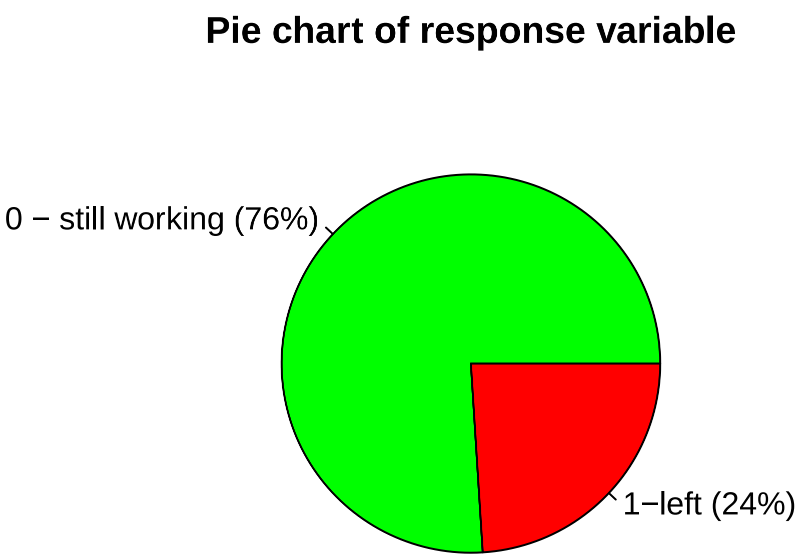
```r
str(hr)
```

```
## 'data.frame':    14999 obs. of  10 variables:
##  $ satisfaction_level   : num  0.38 0.8 0.11 0.72 0.37 0.41 0.1 0.92 0.89 0.42 ...
##  $ last_evaluation      : num  0.53 0.86 0.88 0.87 0.52 0.5 0.77 0.85 1 0.53 ...
##  $ number_project       : int  2 5 7 5 2 2 6 5 5 2 ...
##  $ average_montly_hours : int  157 262 272 223 159 153 247 259 224 142 ...
##  $ time_spend_company   : int  3 6 4 5 3 3 4 5 5 3 ...
##  $ Work_accident        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ left                 : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ promotion_last_5years: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ sales                : Factor w/ 10 levels "accounting","hr",..: 8 8 8 8 8 8 8 8 8 8 ...
##  $ salary               : Factor w/ 3 levels "high","low","medium": 2 3 3 2 2 2 2 2 2 2 ...
```

```r
# frequency of response variable
cbind( Freq=table(left),
       Cumul=cumsum(table(left)),
       relative=round((prop.table(table(left))*100),2))
```

```
##    Freq Cumul relative
## 0 11428 11428    76.19
## 1  3571 14999    23.81
```

```r
# pie chart of response variable
slices <- c(76, 24)
lbls <- c("0 - still working (76%)", "1-left (24%)")
pie(slices, labels = lbls, main="Pie chart of response variable",
  col=c("green", "red"))
```




```r
# frequency of work accident
cbind( Freq=table(Work_accident),
       Cumul=cumsum(table(Work_accident)),
       relative=round((prop.table(table(Work_accident))*100),2))
```

```
##    Freq Cumul relative
## 0 12830 12830    85.54
## 1  2169 14999    14.46
```

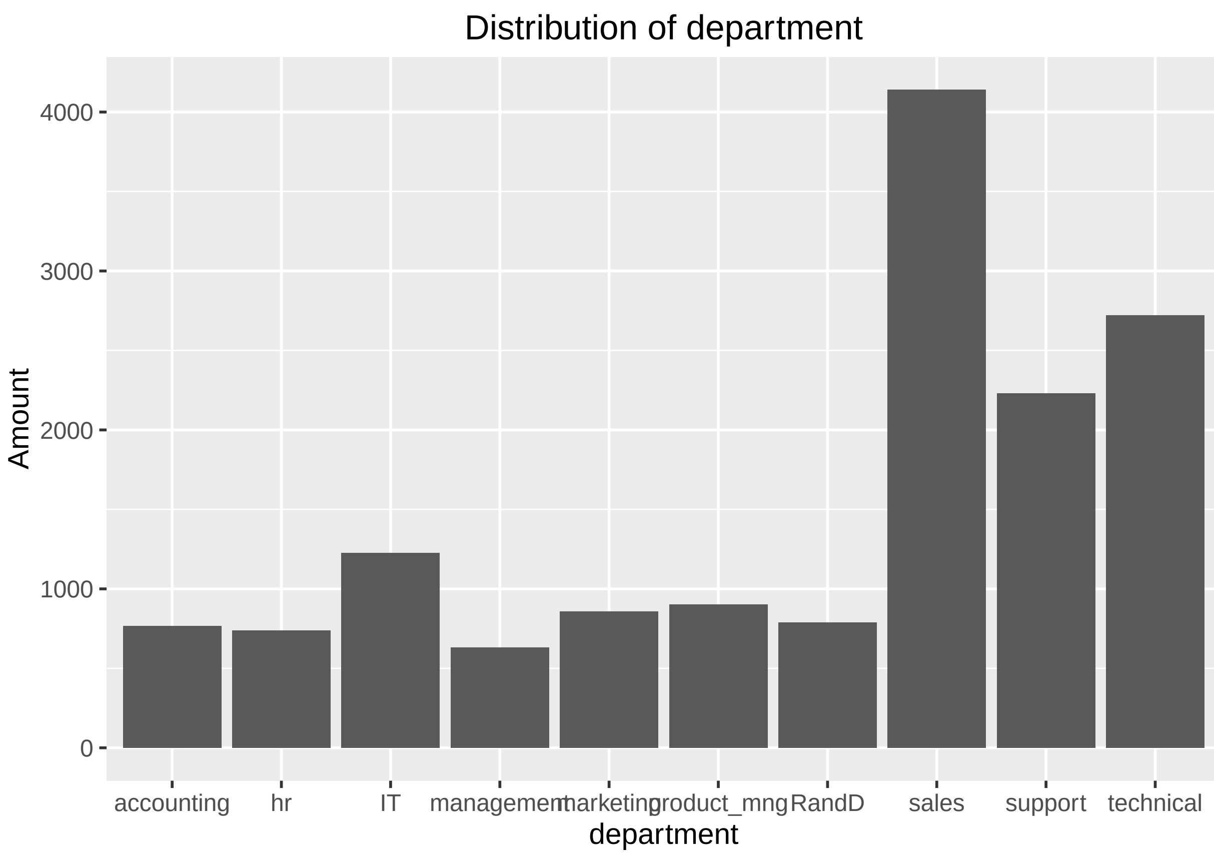
```r
# frequency of promotion_last_5years
cbind( Freq=table(promotion_last_5years),
       Cumul=cumsum(table(promotion_last_5years)),
       relative=round((prop.table(table(promotion_last_5years))*100),2))
```

```
##    Freq Cumul relative
## 0 14680 14680    97.87
## 1   319 14999     2.13
```

```r
# analyzing variable sales
vec_sales <- as.vector(sales)
unique(vec_sales)
```

```
##  [1] "sales"       "accounting"  "hr"          "technical"   "support"
##  [6] "management"  "IT"          "product_mng" "marketing"   "RandD"
```

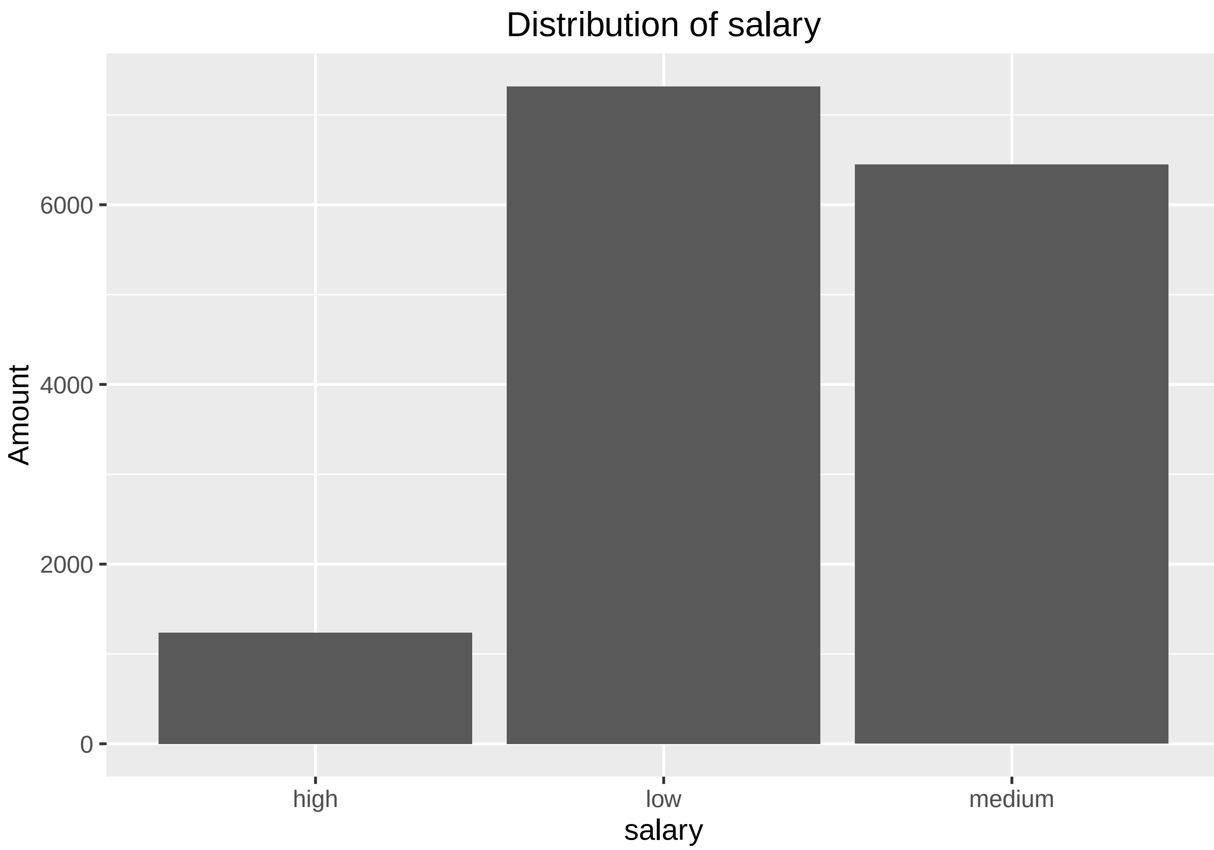
```r
vec_sales <- factor(vec_sales)
qplot(vec_sales, xlab="department", ylab="Amount") + ggtitle("Distribution of department")
```



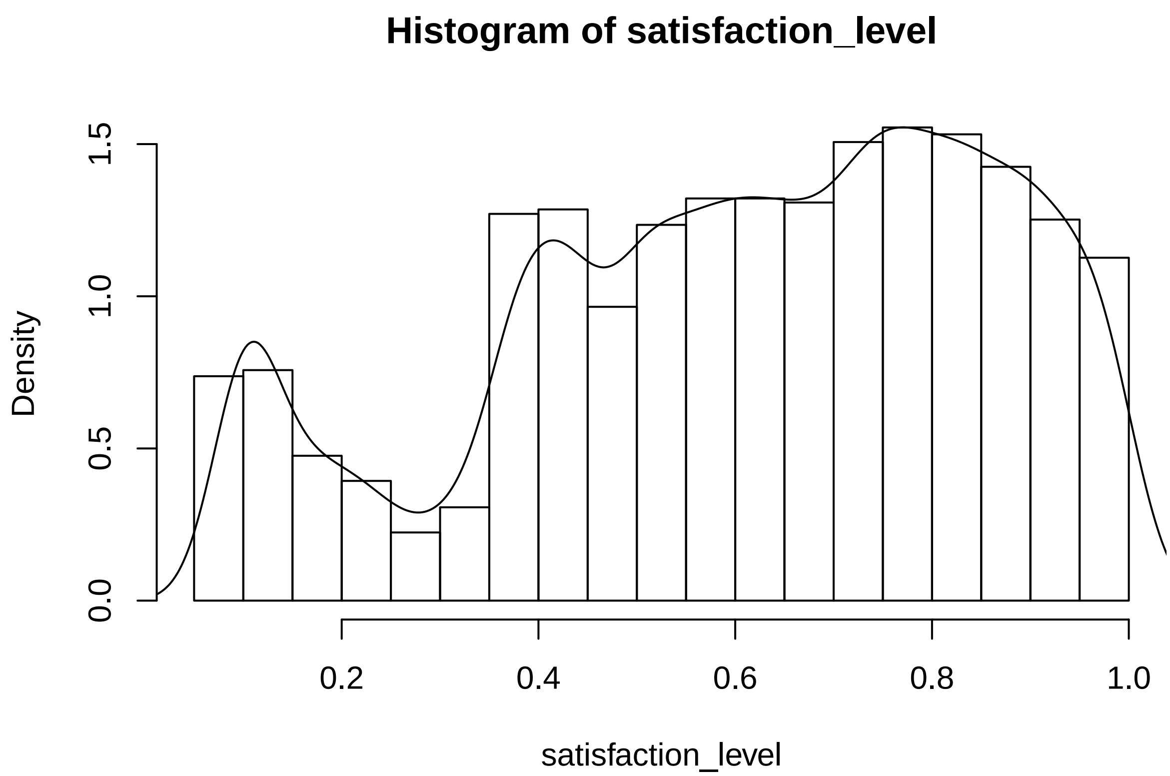

```r
# analyzing variable salary
vec_salary <- as.vector(salary)
unique(vec_salary)
```

```
## [1] "low"    "medium" "high"
```

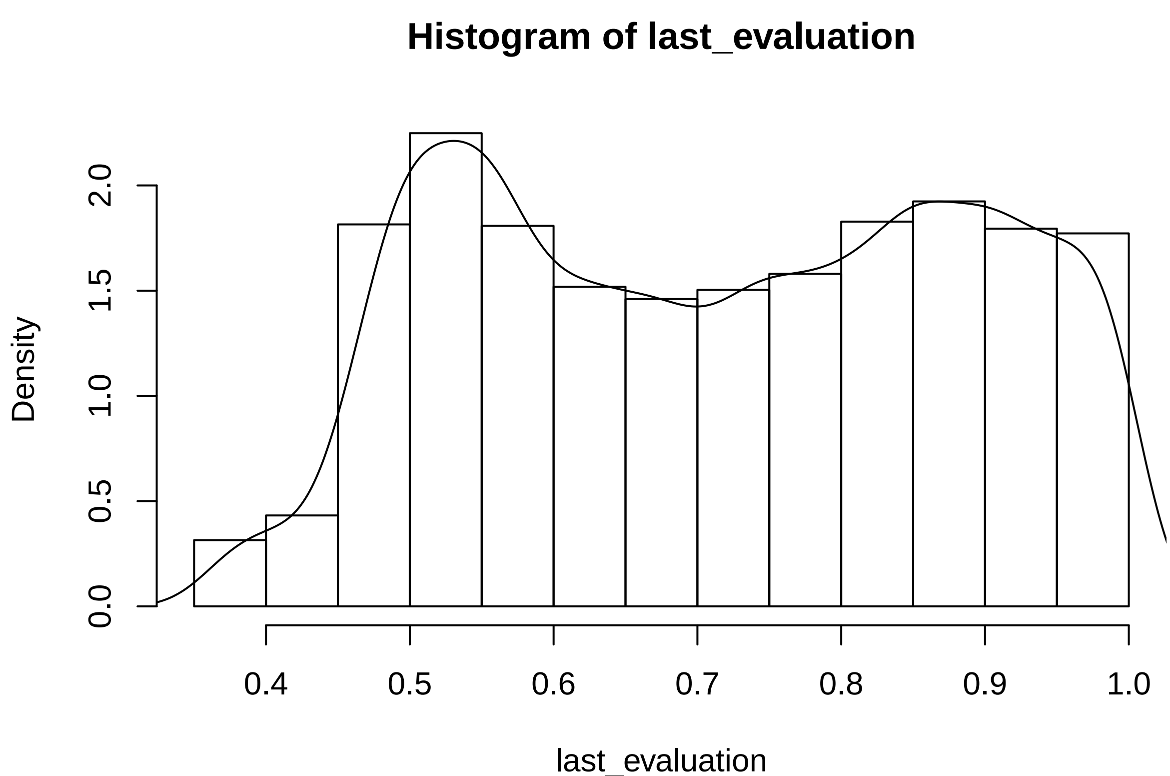
```r
vec_salary <- factor(vec_salary)
qplot(vec_salary, xlab="salary", ylab="Amount") + ggtitle("Distribution of salary")
```



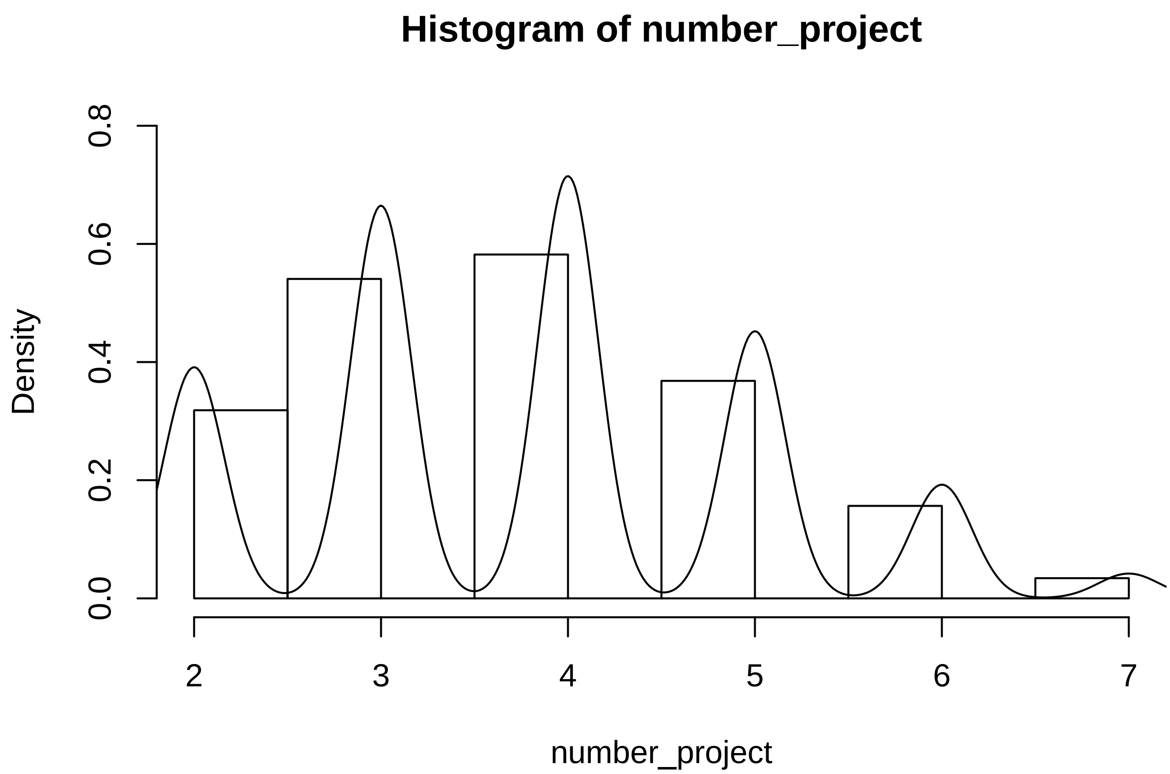

```
# analyze sastifaction level
hist(satisfaction_level, freq=F)
lines(density(satisfaction_level))
```



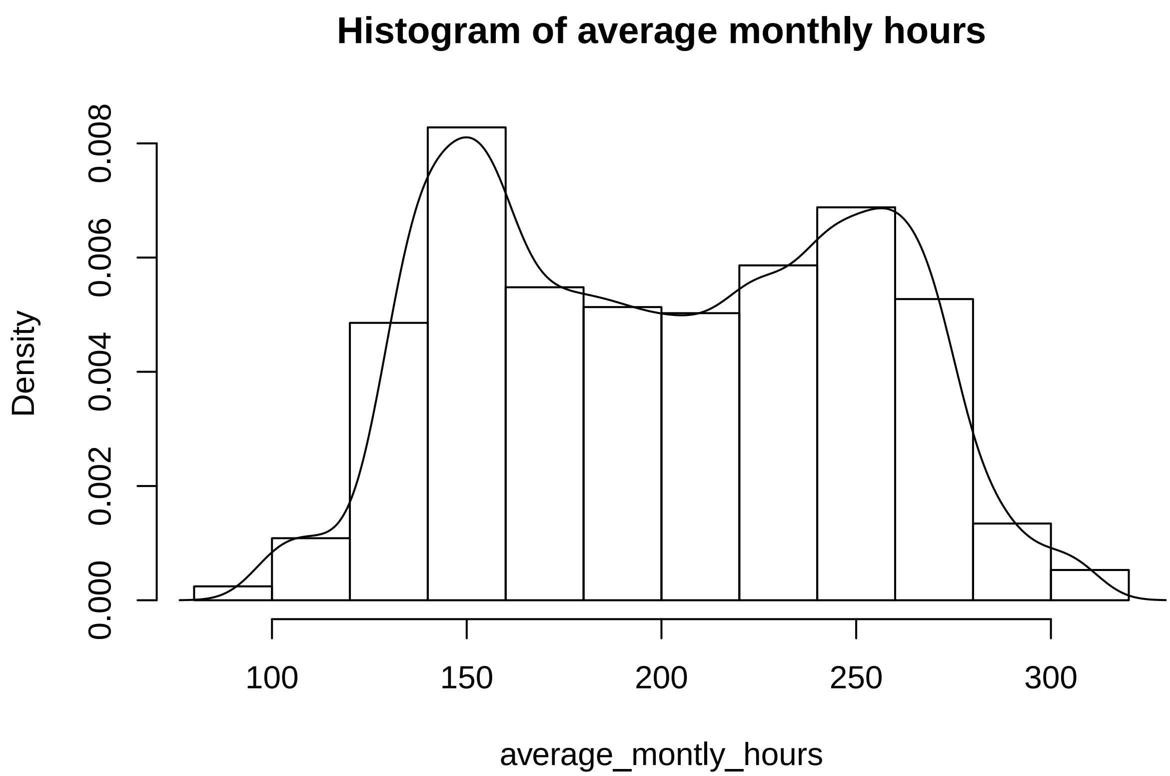

```r
# last evaluation
hist(last_evaluation, freq=F)
lines(density(last_evaluation))
```



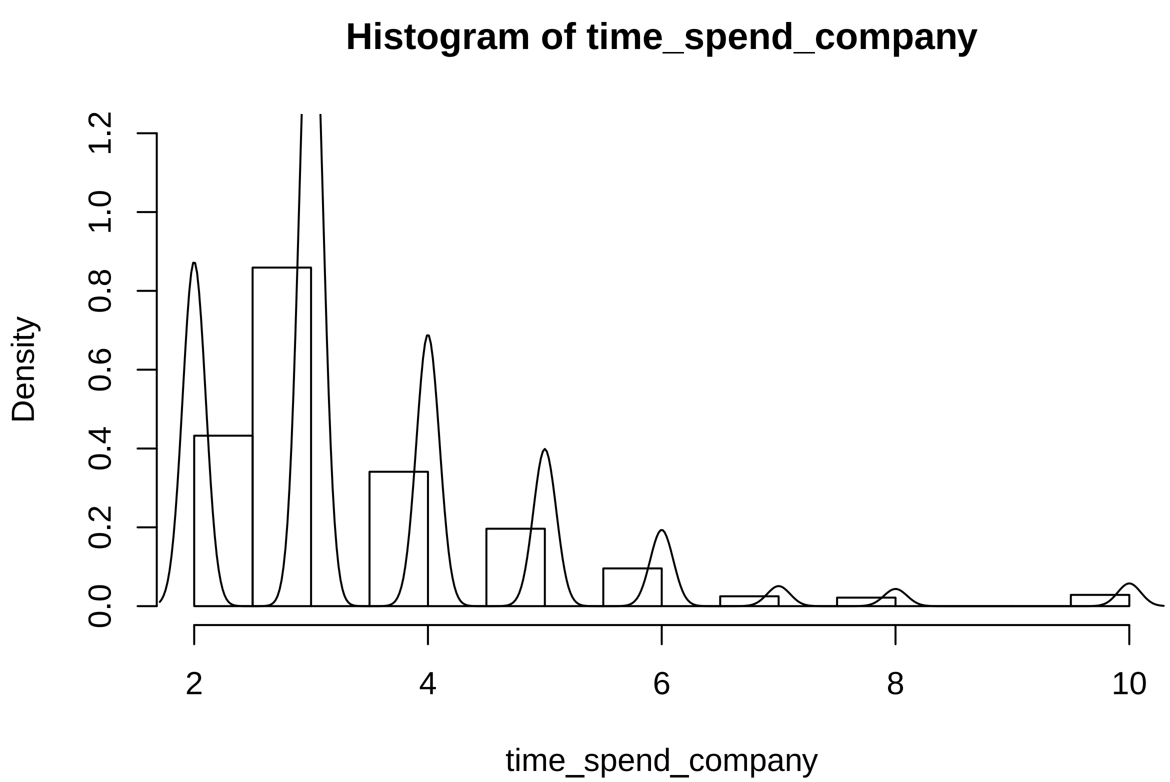
```r
# number of projects
hist(number_project, ylim = c(0,0.8), freq=F)
lines(density(number_project))
```



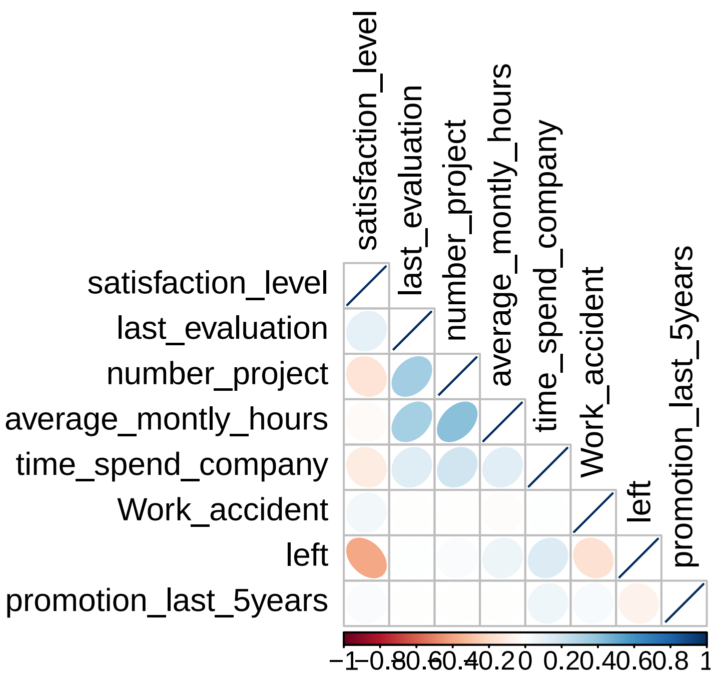

```r
# average_montly_hours
hist(average_montly_hours, freq=F, main="Histogram of average monthly hours")
lines(density(average_montly_hours))
```




```r
# time spend inside company
hist(time_spend_company,  ylim = c(0,1.2), freq=F)
lines(density(time_spend_company))
```



```r
### Calculate correlation ###
library(corrplot)
par(mar=c(4,3,2,2))
par(oma=c(1,1,2,2))
corrplot(cor(hr[,c(1,2,3,4,5,6,7,8)]),type="lower", tl.col="black",method="ellipse")
```




```r
# correlation
cor(hr[sapply(hr, is.numeric)])
```

```
##                       satisfaction_level last_evaluation number_project
## satisfaction_level            1.00000000     0.105021214   -0.142969586
## last_evaluation               0.10502121     1.000000000    0.349332589
## number_project               -0.14296959     0.349332589    1.000000000
## average_montly_hours         -0.02004811     0.339741800    0.417210634
## time_spend_company           -0.10086607     0.131590722    0.196785891
## Work_accident                 0.05869724    -0.007104289   -0.004740548
## left                         -0.38837498     0.006567120    0.023787185
## promotion_last_5years         0.02560519    -0.008683768   -0.006063958
##                       average_montly_hours time_spend_company
## satisfaction_level            -0.020048113       -0.100866073
## last_evaluation                0.339741800        0.131590722
## number_project                 0.417210634        0.196785891
## average_montly_hours           1.000000000        0.127754910
## time_spend_company             0.127754910        1.000000000
## Work_accident                 -0.010142888        0.002120418
## left                           0.071287179        0.144822175
## promotion_last_5years         -0.003544414        0.067432925
##                       Work_accident        left promotion_last_5years
## satisfaction_level      0.058697241 -0.38837498           0.025605186
## last_evaluation        -0.007104289  0.00656712          -0.008683768
## number_project         -0.004740548  0.02378719          -0.006063958
```

```
## average_montly_hours   -0.010142888  0.07128718          -0.003544414
## time_spend_company      0.002120418  0.14482217           0.067432925
## Work_accident           1.000000000 -0.15462163           0.039245435
## left                   -0.154621634  1.00000000          -0.061788107
## promotion_last_5years   0.039245435 -0.06178811           1.000000000
```

Satisfaction level has the highest correlation, that has a negative relationship with left (response variable).

```r
### Bivariate analysis ###

# left vs work_accident
t <- table(hr$left, hr$Work_accident)
barplot(prop.table(t,2), legend=paste(unique(hr$left), "left"),
        ylab="Cumulative Probability", xlab="work accident")
```


```r
# or
CrossTable(hr$left, hr$Work_accident, prop.r=TRUE, prop.c=FALSE,
                                      prop.t=TRUE, prop.chisq=FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  14999
```

```
##
##
##              | hr$Work_accident
##      hr$left |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##            0 |      9428 |      2000 |     11428 |
##              |     0.825 |     0.175 |     0.762 |
##              |     0.629 |     0.133 |           |
## -------------|-----------|-----------|-----------|
##            1 |      3402 |       169 |      3571 |
##              |     0.953 |     0.047 |     0.238 |
##              |     0.227 |     0.011 |           |
## -------------|-----------|-----------|-----------|
## Column Total |     12830 |      2169 |     14999 |
## -------------|-----------|-----------|-----------|
##
##
```

```
aggregate(left ~ Work_accident, FUN=mean)
```

```
##   Work_accident       left
## 1             0 0.26515978
## 2             1 0.07791609
```

```
# left vs promotion_last_5years
CrossTable(hr$left, hr$promotion_last_5years, prop.r=TRUE, prop.c=FALSE,
           prop.t=TRUE, prop.chisq=FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  14999
##
##
##              | hr$promotion_last_5years
##      hr$left |         0 |         1 | Row Total |
## -------------|-----------|-----------|-----------|
##            0 |     11128 |       300 |     11428 |
##              |     0.974 |     0.026 |     0.762 |
##              |     0.742 |     0.020 |           |
## -------------|-----------|-----------|-----------|
##            1 |      3552 |        19 |      3571 |
##              |     0.995 |     0.005 |     0.238 |
##              |     0.237 |     0.001 |           |
## -------------|-----------|-----------|-----------|
## Column Total |     14680 |       319 |     14999 |
```

```
## -------------|-----------|-----------|-----------|
##
##
```

```r
aggregate(left ~ promotion_last_5years, FUN=mean)
```

```
##   promotion_last_5years       left
## 1                     0 0.24196185
## 2                     1 0.05956113
```

People that didn't have a promotion in the last 5 years left more than those who have it.

```r
# left vs sales
aggregate(left ~ sales, FUN=mean)
```

```
##          sales      left
## 1   accounting 0.2659713
## 2           hr 0.2909337
## 3           IT 0.2224939
## 4   management 0.1444444
## 5    marketing 0.2365967
## 6  product_mng 0.2195122
## 7        RandD 0.1537484
## 8        sales 0.2449275
## 9      support 0.2489906
## 10   technical 0.2562500
```

People from Management have the lowest average left and HR have the highest average.

```r
# left vs salary
aggregate(left ~ salary, FUN=mean)
```

```
##   salary       left
## 1   high 0.06628941
## 2    low 0.29688354
## 3 medium 0.20431275
```

Low salary have higher average left compared to other categories.

3) Data transformation

Categorize sales variable accordingly to left average rate.

```r
group1 <- c('hr')
group2 <- c('accounting','sales','support','technical')
group3 <- c('marketing', 'IT','product_mng')
group4 <- c('management','RandD')

hr$new_sales <- ifelse(sales %in% group1, 1,
                       ifelse(sales %in% group2, 2,
```

```r
                               ifelse(sales %in% group3, 3,4)))


aggregate(hr$left ~ hr$new_sales, FUN=mean)
```

```
##   hr$new_sales   hr$left
## 1            1 0.2909337
## 2            2 0.2506088
## 3            3 0.2256445
## 4            4 0.1496119
```

4) Divide between train and test set

Divide 70% to train and 30% to test

```r
set.seed(4)
hr_train <- sample(nrow(hr), floor(nrow(hr)*0.7))
train <- hr[hr_train,]
test <- hr[-hr_train,]
```

5) Logistic regression

Test 1: all variables

```r
names(hr)
```

```
##  [1] "satisfaction_level"    "last_evaluation"
##  [3] "number_project"        "average_montly_hours"
##  [5] "time_spend_company"    "Work_accident"
##  [7] "left"                  "promotion_last_5years"
##  [9] "sales"                 "salary"
## [11] "new_sales"
```

```r
model <- glm(formula = (left) ~  satisfaction_level
                               + last_evaluation
                               + number_project
                               + average_montly_hours
                               + time_spend_company
                               + Work_accident
                               + promotion_last_5years
                               + sales
                               + salary,
              family=binomial(logit), data=train)

summary(model)
```

```
##
## Call:
## glm(formula = (left) ~ satisfaction_level + last_evaluation +
```

```
##     number_project + average_montly_hours + time_spend_company +
##     Work_accident + promotion_last_5years + sales + salary, family = binomial(logit),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2877  -0.6592  -0.3965  -0.1171   2.9829
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -1.4478477  0.2325236  -6.227 4.76e-10 ***
## satisfaction_level    -4.1972542  0.1188674 -35.310  < 2e-16 ***
## last_evaluation        0.7350559  0.1791609   4.103 4.08e-05 ***
## number_project        -0.3363727  0.0256602 -13.109  < 2e-16 ***
## average_montly_hours   0.0045586  0.0006202   7.350 1.98e-13 ***
## time_spend_company     0.2850199  0.0187809  15.176  < 2e-16 ***
## Work_accident         -1.5368898  0.1074476 -14.304  < 2e-16 ***
## promotion_last_5years -1.2648892  0.2971023  -4.257 2.07e-05 ***
## saleshr                0.1362426  0.1586537   0.859  0.39048
## salesIT               -0.0598005  0.1451233  -0.412  0.68029
## salesmanagement       -0.6016577  0.1942614  -3.097  0.00195 **
## salesmarketing        -0.0570740  0.1568685  -0.364  0.71598
## salesproduct_mng      -0.2485619  0.1562315  -1.591  0.11161
## salesRandD            -0.7347178  0.1757300  -4.181 2.90e-05 ***
## salessales            -0.0083367  0.1216410  -0.069  0.94536
## salessupport           0.0498260  0.1299577   0.383  0.70142
## salestechnical         0.0668976  0.1267655   0.528  0.59769
## salarylow              1.9598589  0.1558854  12.572  < 2e-16 ***
## salarymedium           1.3999522  0.1567490   8.931  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 11495.5  on 10498  degrees of freedom
## Residual deviance:  8939.2  on 10480  degrees of freedom
## AIC: 8977.2
##
## Number of Fisher Scoring iterations: 5
```

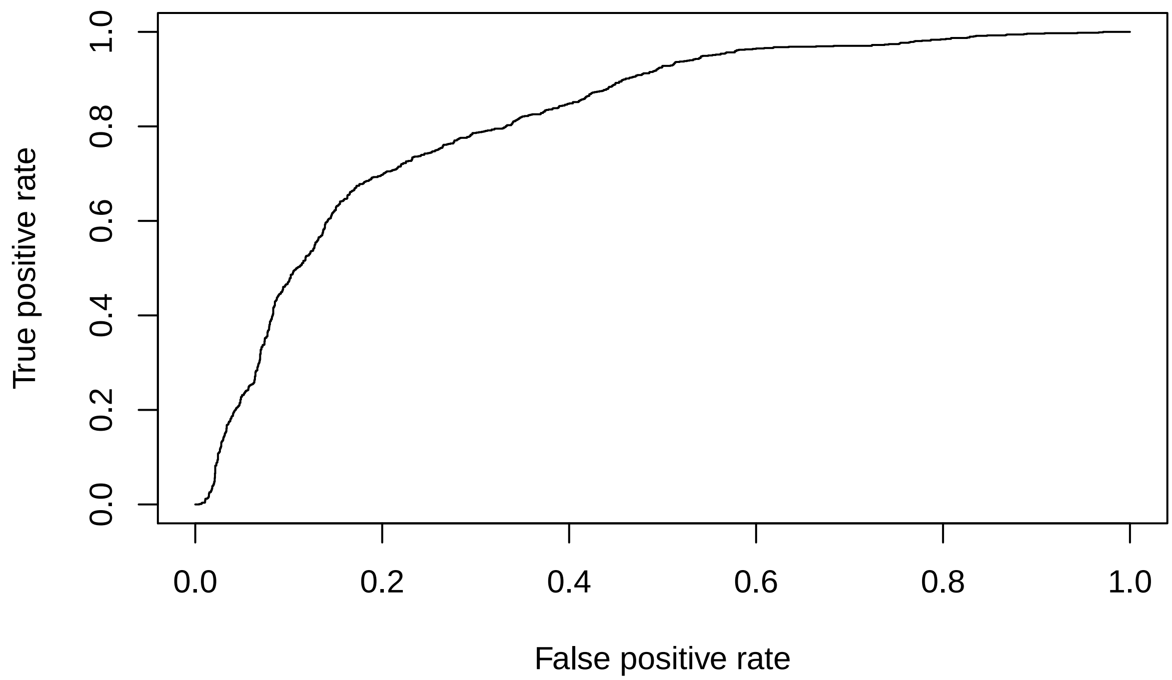
```r
anova(model, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: (left)
##
## Terms added sequentially (first to last)
##
##
##                        Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                   10498    11495.5
## satisfaction_level      1  1560.57     10497     9934.9 < 2.2e-16 ***
```

```
## last_evaluation        1    12.56     10496     9922.4 0.0003941 ***
## number_project         1    90.44     10495     9831.9 < 2.2e-16 ***
## average_montly_hours   1    59.26     10494     9772.7 1.384e-14 ***
## time_spend_company     1   148.33     10493     9624.3 < 2.2e-16 ***
## Work_accident          1   279.84     10492     9344.5 < 2.2e-16 ***
## promotion_last_5years  1    42.08     10491     9302.4 8.744e-11 ***
## sales                  9    84.81     10482     9217.6 1.777e-14 ***
## salary                 2   278.43     10480     8939.2 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

All the variables are relevant, and the most important ones are satisfaction level, work accident, and salary, in this order.

```
# test data set #
library(ROCR)
p <- predict(model, test, type="response")
pr <- prediction(p, test$left)

# calculate the true positive rate and false positive rate
prf <- performance(pr, measure = "tpr", x.measure = "fpr")
plot(prf)
```




```
# Area Under the Curve
auc <- performance(pr, measure = "auc")
auc <- auc@y.values[[1]]
auc
```

```
## [1] 0.8161495
```

```r
# KS is the maximum difference between the cumulative true positive and cumulative false positive rate.
max(attr(prf,'y.values')[[1]]-attr(prf,'x.values')[[1]])
```

```
## [1] 0.5027718
```

Test 2: put the categorized variable new_sales

```r
model2 <- glm(formula = (left) ~  satisfaction_level
             + last_evaluation
             + number_project
             + average_montly_hours
             + time_spend_company
             + Work_accident
             + promotion_last_5years
             + new_sales
             + salary,
             family=binomial(logit), data=train)

summary(model2)
```

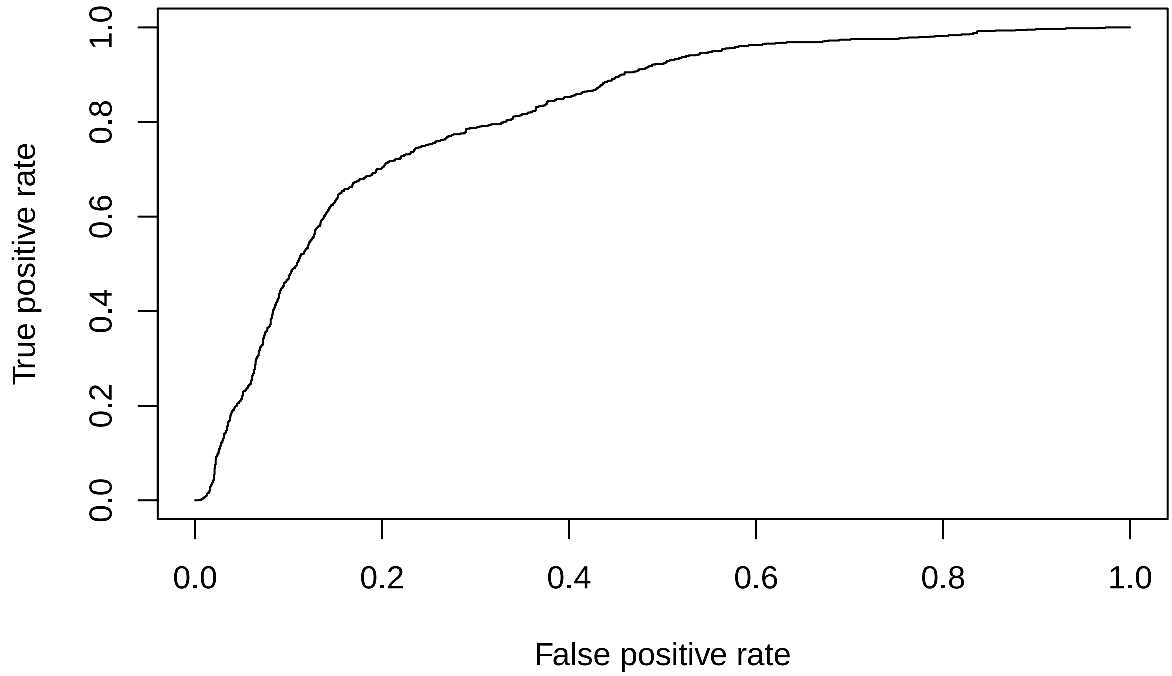
```
##
## Call:
## glm(formula = (left) ~ satisfaction_level + last_evaluation +
##     number_project + average_montly_hours + time_spend_company +
##     Work_accident + promotion_last_5years + new_sales + salary,
##     family = binomial(logit), data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2967  -0.6596  -0.3998  -0.1183   2.9379
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -0.9386982  0.2275844  -4.125 3.71e-05 ***
## satisfaction_level    -4.1896341  0.1186159 -35.321  < 2e-16 ***
## last_evaluation        0.7337754  0.1788736   4.102 4.09e-05 ***
## number_project        -0.3357623  0.0256253 -13.103  < 2e-16 ***
## average_montly_hours   0.0045522  0.0006196   7.347 2.03e-13 ***
## time_spend_company     0.2841716  0.0186652  15.225  < 2e-16 ***
## Work_accident         -1.5371170  0.1074154 -14.310  < 2e-16 ***
## promotion_last_5years -1.2873699  0.2957053  -4.354 1.34e-05 ***
## new_sales             -0.2484242  0.0392734  -6.326 2.52e-10 ***
## salarylow              1.9731726  0.1546876  12.756  < 2e-16 ***
## salarymedium           1.4090771  0.1556685   9.052  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 11495.5  on 10498  degrees of freedom
## Residual deviance:  8951.6  on 10488  degrees of freedom
## AIC: 8973.6
```

```
##
## Number of Fisher Scoring iterations: 5
```

```r
anova(model2, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: (left)
##
## Terms added sequentially (first to last)
##
##
##                       Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                  10498    11495.5
## satisfaction_level     1  1560.57     10497     9934.9 < 2.2e-16 ***
## last_evaluation        1    12.56     10496     9922.4 0.0003941 ***
## number_project         1    90.44     10495     9831.9 < 2.2e-16 ***
## average_montly_hours   1    59.26     10494     9772.7 1.384e-14 ***
## time_spend_company     1   148.33     10493     9624.3 < 2.2e-16 ***
## Work_accident          1   279.84     10492     9344.5 < 2.2e-16 ***
## promotion_last_5years  1    42.08     10491     9302.4 8.744e-11 ***
## new_sales              1    62.78     10490     9239.6 2.313e-15 ***
## salary                 2   288.00     10488     8951.6 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# test data set #
library(ROCR)
p2 <- predict(model2, test, type="response")
pr2 <- prediction(p2, test$left)
prf2 <- performance(pr2, measure = "tpr", x.measure = "fpr")
plot(prf2)
```




```
# Area Under the Curve
auc2 <- performance(pr2, measure = "auc")
auc2 <- auc2@y.values[[1]]
auc2
```

```
## [1] 0.8177612
```

```
# KS
max(attr(prf2,'y.values')[[1]]-attr(prf2,'x.values')[[1]])
```

```
## [1] 0.5095297
```

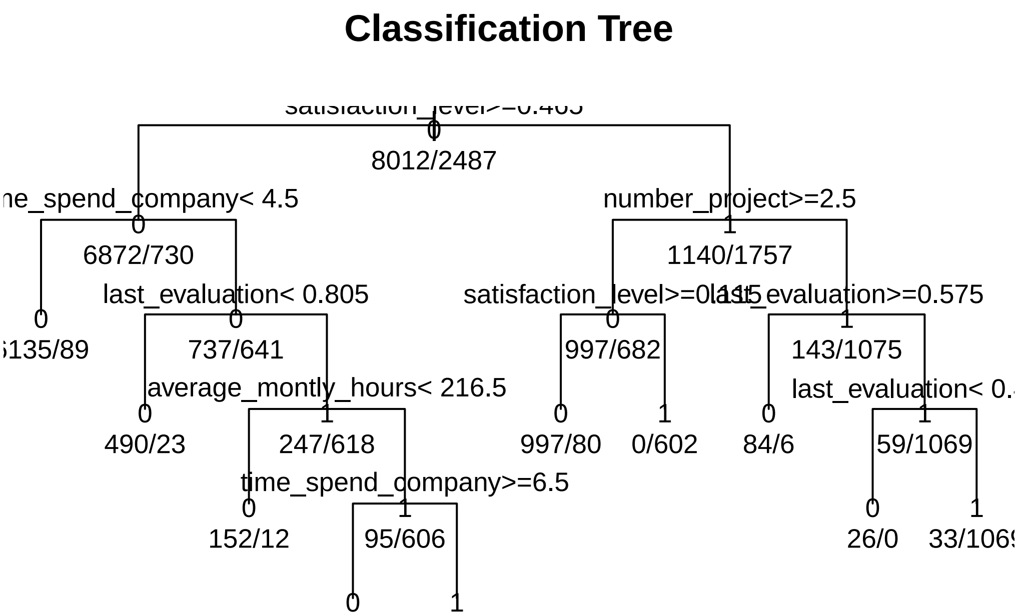
Slightly improvement using the categorized sales variable.

6) Decision tree

Let's use the best variables combination got on logistic regression

```
tree1 <- rpart(formula = left ~  satisfaction_level
                  + last_evaluation
                  + number_project
                  + average_montly_hours
                  + time_spend_company
                  + Work_accident
                  + promotion_last_5years
                  + new_sales
                  + salary,
                    data = train,
                    method = "class")
```

```r
# plot tree
plot(tree1, uniform=TRUE, main="Classification Tree")
text(tree1, use.n=TRUE, all=TRUE, cex=.8)
```




```r
# fancy plot tree using package rpart.plot is not possible on R Markdown
# fancyRpartPlot(tree1)
```

To validate the model I used the printcp and plotcp functions. Where 'CP' stands for Complexity Parameter of the tree. Also it's possibel to prune the tree to avoid any overfitting of the data.

```r
# Validation
# get the optimal prunings based on the cp value.
printcp(tree1)
```

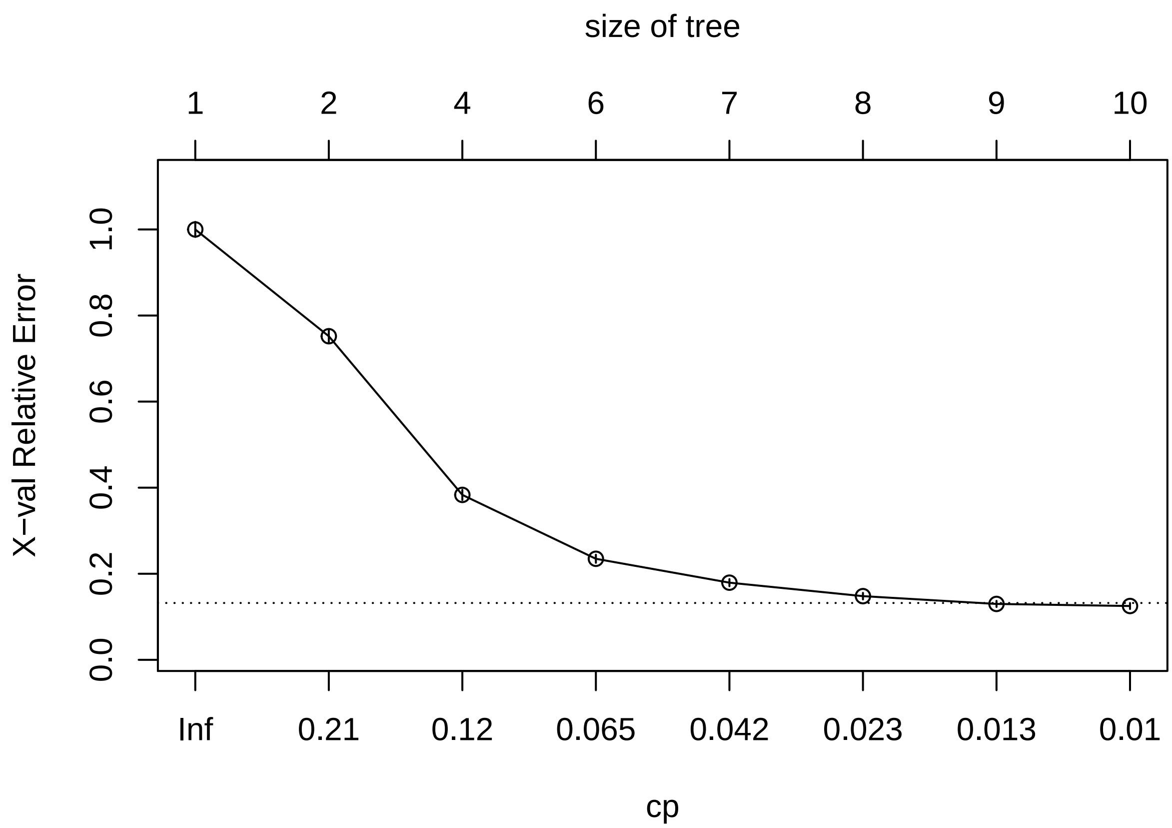
```
##
## Classification tree:
## rpart(formula = left ~ satisfaction_level + last_evaluation +
##     number_project + average_montly_hours + time_spend_company +
##     Work_accident + promotion_last_5years + new_sales + salary,
##     data = train, method = "class")
##
## Variables actually used in tree construction:
## [1] average_montly_hours last_evaluation      number_project
## [4] satisfaction_level   time_spend_company
##
## Root node error: 2487/10499 = 0.23688
##
## n= 10499
##
##         CP nsplit rel error  xerror      xstd
## 1 0.248090      0   1.00000 1.00000 0.0175170
## 2 0.184359      1   0.75191 0.75191 0.0157635
## 3 0.074588      3   0.38319 0.38319 0.0118361
```

```
## 4 0.056293      5   0.23402 0.23482 0.0094428
## 5 0.031363      6   0.17772 0.17933 0.0083093
## 6 0.017290      7   0.14636 0.14797 0.0075771
## 7 0.010454      8   0.12907 0.12988 0.0071144
## 8 0.010000      9   0.11862 0.12505 0.0069851
```

```
# The value of cp should be least, so that the cross-validated error rate is minimum.
tree1$cptable[which.min(tree1$cptable[,"xerror"]),"CP"]
```

```
## [1] 0.01
```

```
plotcp(tree1)
```



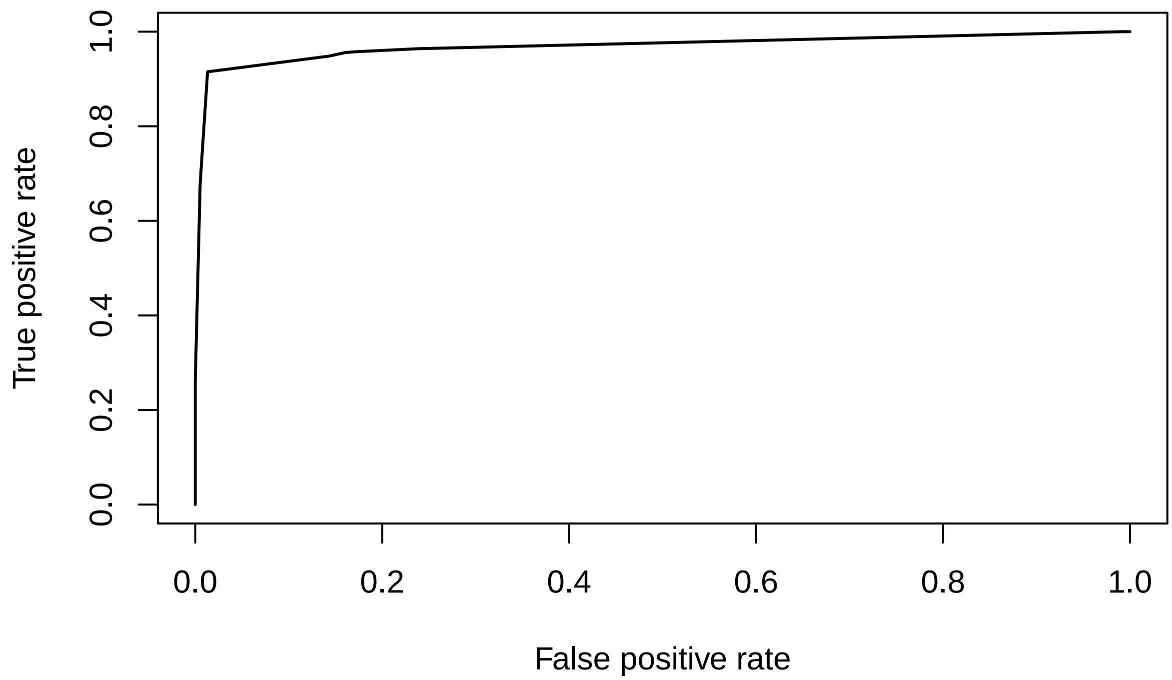

```
# This graph shows it's not necessary prune the tree

# confusion matrix (training data)
conf_matrix_tree <- table(train$left, predict(tree1, type="class"))
rownames(conf_matrix_tree) <- paste("Actual", rownames(conf_matrix_tree), sep = ":")
colnames(conf_matrix_tree) <- paste("Pred", colnames(conf_matrix_tree), sep = ":")
print(conf_matrix_tree)
```

```
##
##             Pred:0 Pred:1
##   Actual:0    7927     85
##   Actual:1     210   2277
```

```r
# On test set
test_tree = predict(tree1, test, type = "prob")

#Storing Model Performance Scores
pred_tree <-prediction(test_tree[,2], test$left)

# Calculating Area under Curve
perf_tree <- performance(pred_tree,"auc")
perf_tree
```

```
## An object of class "performance"
## Slot "x.name":
## [1] "None"
##
## Slot "y.name":
## [1] "Area under the ROC curve"
##
## Slot "alpha.name":
## [1] "none"
##
## Slot "x.values":
## list()
##
## Slot "y.values":
## [[1]]
## [1] 0.9692407
##
##
## Slot "alpha.values":
## list()
```

```r
# Calculating True Positive and False Positive Rate
perf_tree <- performance(pred_tree, "tpr", "fpr")

# Plot the ROC curve
plot(perf_tree,  lwd = 1.5)
```



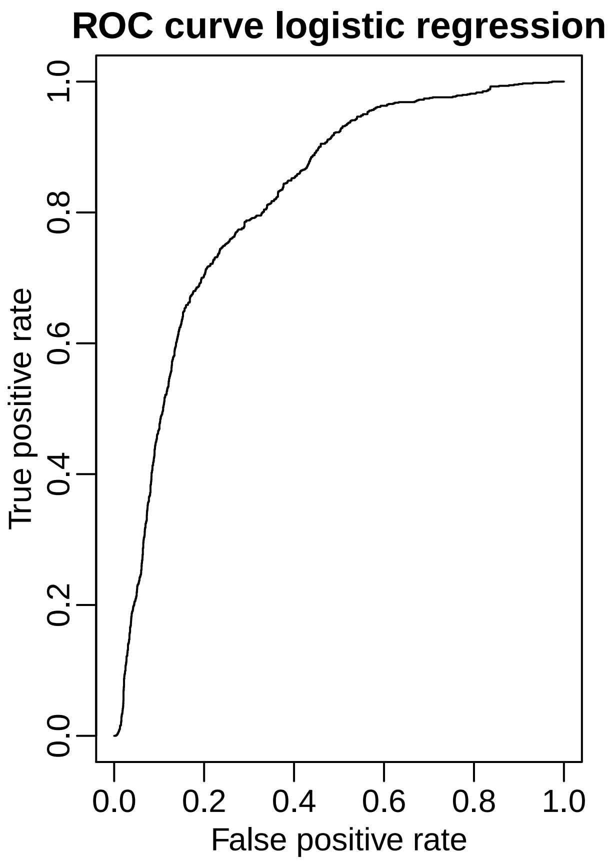

```r
#Calculating KS statistics
ks1.tree <- max(attr(perf_tree, "y.values")[[1]] - (attr(perf_tree, "x.values")[[1]]))
ks1.tree
```

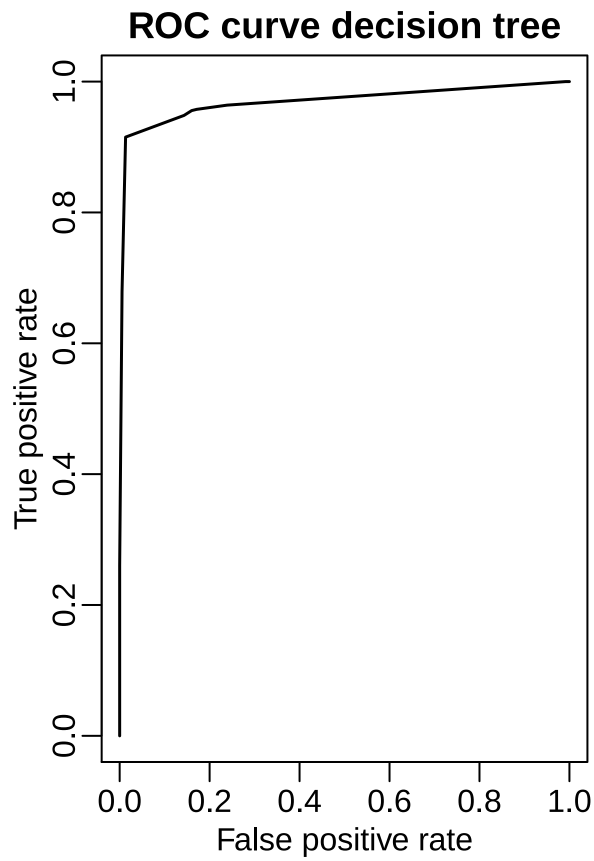
```
## [1] 0.9019558
```

7) Conclusions

- What are the most important aspects that are decisive to employees leave their jobs? In the logistic regression I found satisfaction level, work accident, and salary as the most relevant aspects. The decision tree, the most important are satisfaction level, time spend on company and number of project.
- What is more accurate to predict these aspects: logistic regression or decision tree? For this data set, decision tree got a better performance on test data set. We can observe this by the ROC curves below and comparing the K.S, for the decision tree is 0.90 and the logistic regression is 0.51.

```r
mypar(1,2)
plot(prf2)
title("ROC curve logistic regression")

plot(perf_tree,  lwd = 1.5)
title("ROC curve decision tree")
```


ROC curve logistic regression
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0


ROC curve decision tree
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0