Classification in depth

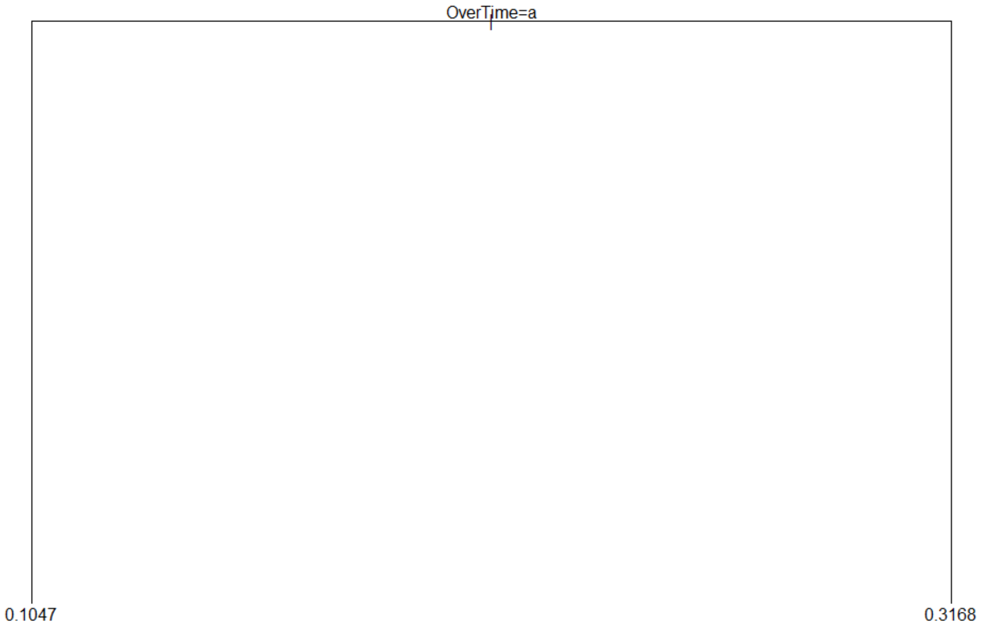
Subramani.M

21 May 2018

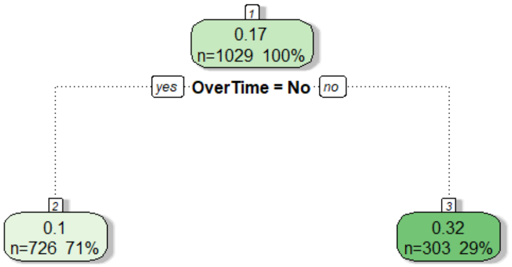
classification in depth

```
library(tree)

## Warning: package 'tree' was built under R version 3.4.4

setwd("C:/Users/Administrator/Desktop/Machine Learning/DATA SETS")
HR <- read.csv("C:/Users/Administrator/Desktop/Machine Learning/DATA SETS/HR Analytics.csv",header = T,sep = ",")
View(HR)
colnames(HR)

##  [1] "Age"                      "Attrition"               
##  [3] "BusinessTravel"           "DailyRate"               
##  [5] "Department"               "DistanceFromHome"        
##  [7] "Education"                "EducationField"          
##  [9] "EmployeeCount"            "EmployeeNumber"          
## [11] "EnvironmentSatisfaction"  "Gender"                  
## [13] "HourlyRate"               "JobInvolvement"          
## [15] "JobLevel"                 "JobRole"                 
## [17] "JobSatisfaction"          "MaritalStatus"           
## [19] "MonthlyIncome"            "MonthlyRate"             
## [21] "NumCompaniesWorked"       "Over18"                  
## [23] "OverTime"                 "PercentSalaryHike"       
## [25] "PerformanceRating"        "RelationshipSatisfaction"
## [27] "StandardHours"            "StockOptionLevel"        
## [29] "TotalWorkingYears"        "TrainingTimesLastYear"   
## [31] "WorkLifeBalance"          "YearsAtCompany"          
## [33] "YearsInCurrentRole"       "YearsSinceLastPromotion" 
## [35] "YearsWithCurrManager"

hr_training <- HR[1:(0.7*nrow(HR)),]
hr_testing <- HR[(0.7*nrow(HR) + 1):nrow(HR),]
nrow(hr_training)

## [1] 1029

nrow(hr_testing)

## [1] 441

# 2 catagorical classes
model = rpart::rpart(Attrition ~ OverTime + Gender,data = hr_training)
```

```
{{plot(model)
  text(model)}}
```



```
#install.packages("rattle")
library(rattle)

## Warning: package 'rattle' was built under R version 3.4.4

## Rattle: A free graphical interface for data science with R.
## Version 5.1.0 Copyright (c) 2006-2017 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.

fancyRpartPlot(model)
```



Rattle 2018-May-21 16:54:55 Administrator

```
#perform tests to see the relatedness. because there are many columns in the
dataset.
```

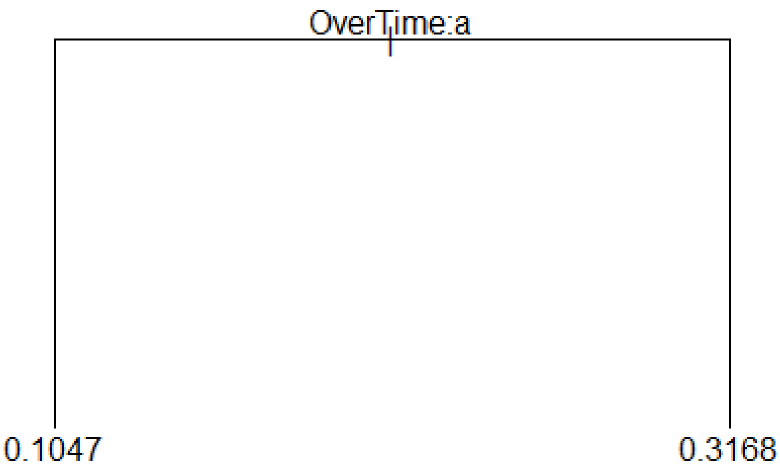
Gini impurity

input variable : catagorical with 2 classes

```
library(dplyr)

## Warning: package 'dplyr' was built under R version 3.4.3

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

nrow(hr_training)

## [1] 1029
```

```r
left_overtime = hr_training %>% filter(OverTime =='Yes')

## Warning: package 'bindrcpp' was built under R version 3.4.3

right_overtime = hr_training %>% filter(OverTime == 'No')
nrow(left_overtime)

## [1] 303

nrow(right_overtime)

## [1] 726

table(left_overtime$Attrition)

##
##   0   1
## 207  96

1 - (96/303)^2 - (207/303)^2

## [1] 0.4328987

table(right_overtime$Attrition)

##
##   0   1
## 650  76

1 - (76/726)^2 - (650/726)^2

## [1] 0.1874492

left_gender = hr_training %>% filter(Gender == 'Female')
right_gender = hr_training %>% filter(Gender == 'Male')
nrow(left_gender)

## [1] 431

nrow(right_gender)

## [1] 598

table(left_gender$Attrition)

##
##   0   1
## 364  67

gi_left = 1 - (364/nrow(left_gender))^2 - (67/nrow(left_gender))^2

table(right_gender$Attrition)
```

```
##
##   0   1
## 493 105

gi_right = 1 - (105/nrow(right_gender))^2 - (493/nrow(right_gender))^2

gi_gender = nrow(left_gender) / nrow(hr_training) * gi_left +
nrow(right_gender) / nrow(hr_training) * gi_right

# giny impurity for whole dataset
table(hr_training$Attrition)

##
##   0   1
## 857 172

1 -(857/nrow(hr_training))^2 - (172/nrow(hr_training))^2

## [1] 0.2784252

model = tree(Attrition ~ OverTime +Gender,data = hr_training)
{{plot(model)
  text(model)}}
```



# giny impurity

when input is catagorical

```
#single on one side and married and divorced on other side
marital_status_uniq = unique(hr_training$MaritalStatus)
```

```r
for(status in marital_status_uniq){
  samples_left = hr_training %>% filter(MaritalStatus == status)
  samples_right = hr_training %>% filter(MaritalStatus != status )
  p0_left = nrow(samples_left %>% filter(Attrition == 0 ))/nrow(samples_left)
  p1_left = nrow(samples_left %>% filter(Attrition == 1 ))/nrow(samples_left)
  gi_left = 1 - p0_left^2 - p1_left^2

  p0_right = nrow(samples_right %>% filter(Attrition == 0
))/nrow(samples_right)
  p1_right = nrow(samples_right %>% filter(Attrition == 1
))/nrow(samples_right)
  gi_right = 1 - p0_right^2 - p1_right^2

  gi_status = nrow(samples_left)/nrow(hr_training) * gi_left +
nrow(samples_right)/nrow(hr_training)*gi_right

temp = marital_status_uniq[marital_status_uniq != status]

print('left node')
print(status)
print('right node')
print(temp)
print(gi_status)
print('--------------------------')
}
```

```
## [1] "left node"
## [1] "Single"
## [1] "right node"
## [1] Married  Divorced
## Levels: Divorced Married Single
## [1] 0.2686809
## [1] "--------------------------"
## [1] "left node"
## [1] "Married"
## [1] "right node"
## [1] Single   Divorced
## Levels: Divorced Married Single
## [1] 0.2761979
## [1] "--------------------------"
## [1] "left node"
## [1] "Divorced"
## [1] "right node"
## [1] Single  Married
## Levels: Divorced Married Single
## [1] 0.2753798
## [1] "--------------------------"
```

for all columns combinations for job

```
x = c('a','b','c','d')
combn(x , 2 , simplify = FALSE)
```

```
## [[1]]
## [1] "a" "b"
##
## [[2]]
## [1] "a" "c"
##
## [[3]]
## [1] "a" "d"
##
## [[4]]
## [1] "b" "c"
##
## [[5]]
## [1] "b" "d"
##
## [[6]]
## [1] "c" "d"
```

for 2 combination of job roles

```
jobs_uniq = unique(hr_training$JobRole)
combinations_left=c()
combinations_right = c()
gi_all = c()

for(n in c(1,2,3,4)){
  comb_n = combn(jobs_uniq, n, simplify = FALSE)
for(i in seq(1,length(comb_n))){
  comb_left = comb_n[[i]]
  comb_right = jobs_uniq[!jobs_uniq %in% comb_left]

  samples_left = hr_training %>% filter(JobRole %in% comb_left)
  samples_right = hr_training %>% filter(JobRole %in% comb_right )
  p0_left = nrow(samples_left %>% filter(Attrition == 0 ))/nrow(samples_left)
  p1_left = nrow(samples_left %>% filter(Attrition == 1 ))/nrow(samples_left)
  gi_left = 1 - p0_left^2 - p1_left^2

  p0_right = nrow(samples_right %>% filter(Attrition == 0
))/nrow(samples_right)
  p1_right = nrow(samples_right %>% filter(Attrition == 1
))/nrow(samples_right)
  gi_right = 1 - p0_right^2 - p1_right^2

  gi_status = nrow(samples_left)/nrow(hr_training) * gi_left +
nrow(samples_right)/nrow(hr_training)*gi_right
#temp = jobs_uniq[jobs_uniq != status]
library(dplyr)
#print('left node')
#print(status)
#print('right node')
#print(temp)
#print(gi_status)
#print('-------------------------')
  combinations_left = c(combinations_left,paste0(comb_left,collapse=','))
  combinations_right = c(combinations_right,paste0(comb_right,collapse =
','))
  gi_all = c(gi_all,gi_status)
}
}

result = data.frame(left = combinations_left, right = combinations_right,gi =
gi_all )
View(result)
nrow(result)
```

```
## [1] 255
```

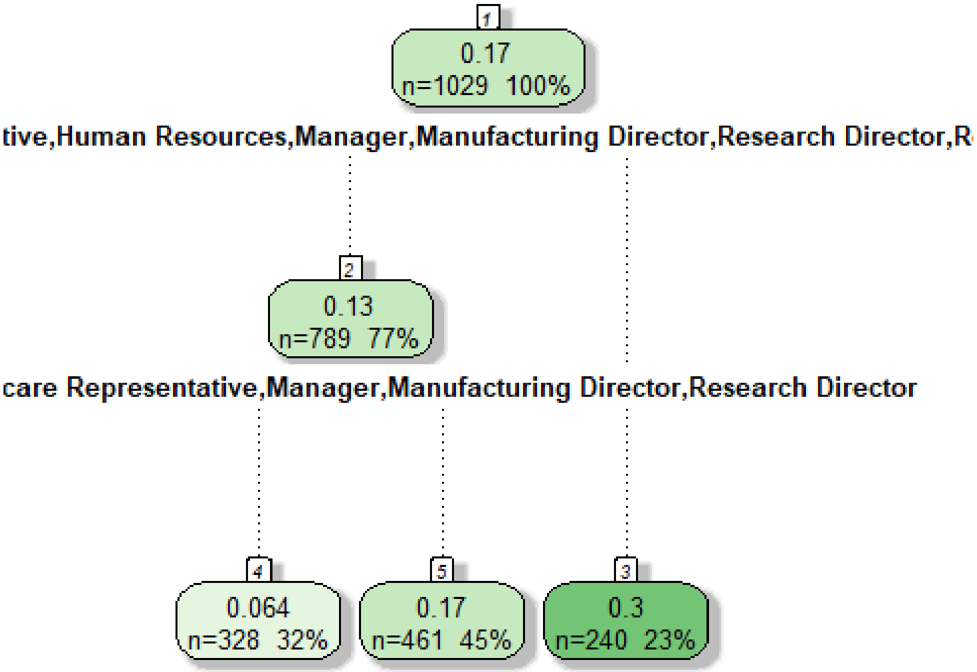
```
result %>% arrange(gi) %>% head(1)
```

```
##                                          left
## 1 Laboratory Technician,Sales Representative
##
right
## 1 Sales Executive,Research Scientist,Manufacturing Director,Healthcare
Representative,Manager,Research Director,Human Resources
##          gi
## 1 0.2683514
```

```
model = rpart::rpart(Attrition ~ JobRole, data = hr_training)
model
```

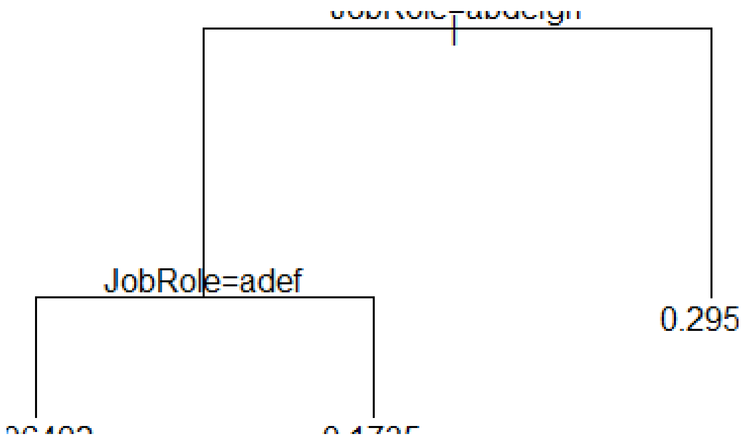
```
## n= 1029
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
## 1) root 1029 143.24980 0.16715260
##   2) JobRole=Healthcare Representative,Human
Resources,Manager,Manufacturing Director,Research Director,Research
Scientist,Sales Executive 789  88.07098 0.12801010
##     4) JobRole=Healthcare Representative,Manager,Manufacturing
Director,Research Director 328  19.65549 0.06402439 *
##     5) JobRole=Human Resources,Research Scientist,Sales Executive 461
66.11714 0.17353580 *
```

```
##   3) JobRole=Laboratory Technician,Sales Representative 240  49.99583
0.29583330 *

library(rattle)
fancyRpartPlot(model)
```


1
0.17
n=1029 100%
tive,Human Resources,Manager,Manufacturing Director,Research Director,R
2
0.13
n=789 77%
care Representative,Manager,Manufacturing Director,Research Director
4
0.064
n=328 32%
5
0.17
n=461 45%
3
0.3
n=240 23%

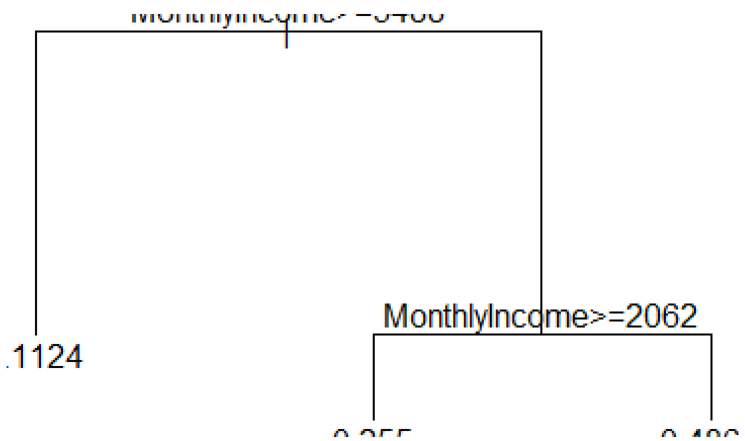
Rattle 2018-May-21 16:55:10 Administrator

```
{{plot(model)
  text(model)}}
```

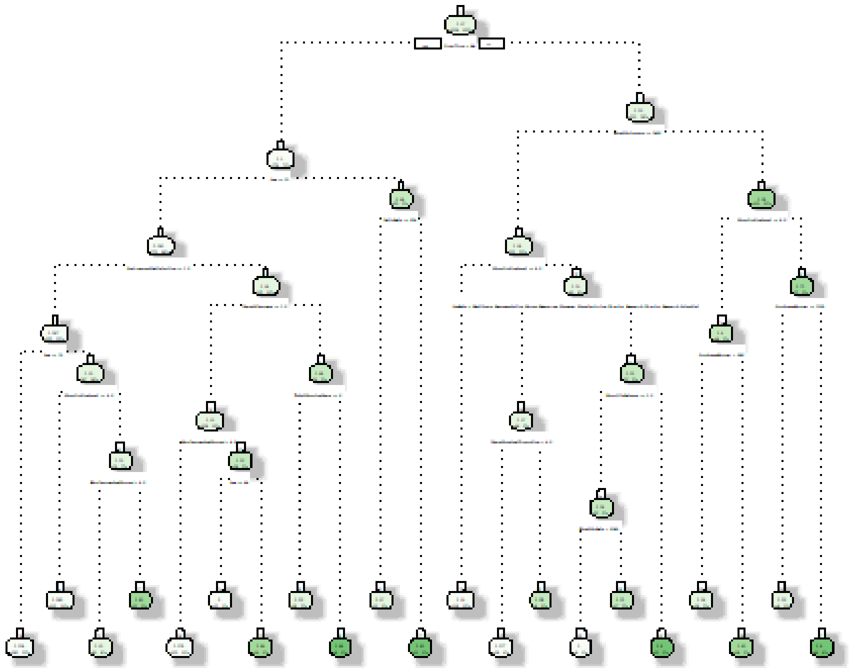


```r
MI_Uniqs=sort(unique(hr_training$MonthlyIncome))
cuts_MI=(MI_Uniqs[1:length(MI_Uniqs)-1] + MI_Uniqs[2:length(MI_Uniqs)])/2
temp=hr_training

gi_status=c()
gi_all=c()
for (cut in cuts_MI) {
  sample_left=temp%>%filter(MonthlyIncome>cut)
  sample_right=temp%>%filter(MonthlyIncome<cut)

  p0_left=nrow(sample_left%>%filter(Attrition==0))/nrow(sample_left)
  p1_left=nrow(sample_left%>%filter(Attrition==1))/nrow(sample_left)
  gi_left=1-p0_left^2-p1_left^2

  p0_right=nrow(sample_right%>%filter(Attrition==0))/nrow(sample_right)
  p1_right=nrow(sample_right%>%filter(Attrition==1))/nrow(sample_right)
  gi_right=1-p0_right^2-p1_right^2

  gi_status=nrow(sample_left)/nrow(hr_training)*gi_left +
                            nrow(sample_right)/nrow(hr_training)*gi_right
  gi_all=c(gi_all,gi_status)
}


model=rpart::rpart(Attrition~MonthlyIncome,data=hr_training)
{{plot(model)
```

```
  text(model)
}}
```



```
result_MI=data.frame(cuts=cuts_MI,gi=gi_all)
result_MI%>%arrange(gi)%>%head(1)

##   cuts        gi
## 1 3488 0.2660006

model = rpart::rpart(Attrition ~. , data =hr_training)
fancyRpartPlot(model)
```



Rattle 2018-May-21 16:56:00 Administrator

```
model = rpart::rpart(Attrition~OverTime, data =hr_training)

# overtime == no is towards left
table(hr_training$Attrition)

##
##   0   1
## 857 172

samples_left = hr_training %>% filter(OverTime =='No')
samples_right = hr_training %>% filter(OverTime == 'Yes')
nrow(samples_left)

## [1] 726

nrow(samples_left)/nrow(hr_training)

## [1] 0.7055394

nrow(samples_left %>% filter(Attrition == 1))/nrow(samples_left)

## [1] 0.1046832

nrow(samples_right %>% filter(Attrition == 0)) / nrow(samples_right)

## [1] 0.6831683

nrow(samples_left %>% filter(Attrition == 0))/nrow(samples_left)
```

```
## [1] 0.8953168
```

```
table(hr_training$Attrition)
```

```
## 
##   0   1 
## 857 172
```

```
model = rpart::rpart(Attrition~OverTime,data = HR)
```

```
237/1470
```

```
## [1] 0.1612245
```

```
samples_left = HR %>% filter(OverTime == 'No')
samples_right = HR %>% filter(OverTime == 'Yes')
nrow(samples_left)/nrow(HR)
```

```
## [1] 0.7170068
```

```
nrow(samples_right)/nrow(HR)
```

```
## [1] 0.2829932
```