3. Information Gain

2022-06-19

Contents


```r
# install.packages("FSelector")

# Load FSelector package for Feature Selection
library(FSelector)

# Load "caTools" package for data partitioning
library(caTools)

# Load tidyverse package
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
# Import data set and save it as empdata
empdata <- read.csv("EmployeeData.csv", stringsAsFactors = TRUE) #convert string variables to factor va
```

check structure

```r
# Check the summary of the dataset
summary(empdata)
```

```
##       Age        Attrition           BusinessTravel   DailyRate
##  Min.   :18.00   No :1202   Non-Travel       : 150   Min.   : 102.0
##  1st Qu.:30.00   Yes: 270   Travel_Frequently: 278   1st Qu.: 465.0
##  Median :36.00              Travel_Rarely    :1044   Median : 802.0
##  Mean   :36.79                                       Mean   : 802.6
##  3rd Qu.:42.00                                       3rd Qu.:1157.0
##  Max.   :60.00                                       Max.   :1499.0
##
##                   Department  DistanceFromHome   Education
##  Human Resources       : 63   Min.   : 1.000   Min.   :1.000
##  Research & Development:962   1st Qu.: 2.000   1st Qu.:2.000
##  Sales                 :447   Median : 7.000   Median :3.000
##                               Mean   : 9.183   Mean   :2.913
##                               3rd Qu.:14.000   3rd Qu.:4.000
##                               Max.   :29.000   Max.   :5.000
##
##           EducationField EmployeeCount EmployeeNumber   EnvironmentSatisfaction
##  Human Resources : 27    Min.   :1     Min.   :   1.0   L1:286
##  Life Sciences   :607    1st Qu.:1     1st Qu.: 491.8   L2:287
##  Marketing       :159    Median :1     Median :1023.0   L3:453
##  Medical         :464    Mean   :1     Mean   :1026.3   L4:446
##  Other           : 83    3rd Qu.:1     3rd Qu.:1557.2
##  Technical Degree:132    Max.   :1     Max.   :2070.0
##
##     Gender      HourlyRate       JobLevel                         JobRole
##  Female:589   Min.   : 30.00   Level1:544   Sales Executive          :326
##  Male  :883   1st Qu.: 48.00   Level2:534   Research Scientist       :292
##               Median : 66.00   Level3:218   Laboratory Technician    :260
##               Mean   : 65.91   Level4:107   Manufacturing Director   :145
##               3rd Qu.: 83.25   Level5: 69   Healthcare Representative:131
##               Max.   :100.00                Manager                  :103
##                                             (Other)                  :215
##  JobSatisfaction  MaritalStatus MonthlyIncome    MonthlyRate
##  L1:289          Divorced:327   Min.   : 1009   Min.   : 2094
##  L2:280          Married :674   1st Qu.: 2911   1st Qu.: 8044
##  L3:442          Single  :471   Median : 4933   Median :14236
##  L4:461                         Mean   : 6512   Mean   :14311
##                                 3rd Qu.: 8384   3rd Qu.:20463
##                                 Max.   :19999   Max.   :26999
##
##  NumCompaniesWorked Over18   OverTime   PercentSalaryHike PerformanceRating
##  Min.   :0.000      Y:1472   No :1045   Min.   :11.00     L3:1246
##  1st Qu.:1.000               Yes: 427   1st Qu.:12.00     L4: 226
```

```
##  Median :2.000                       Median :14.00
##  Mean   :2.692                       Mean   :15.21
##  3rd Qu.:4.000                       3rd Qu.:18.00
##  Max.   :9.000                       Max.   :25.00
##
##  RelationshipSatisfaction StandardHours AvailableStocks  TotalWorkingYears
##  L1:277                   Min.   :80    Min.   :0.0000   Min.   : 0.0
##  L2:303                   1st Qu.:80    1st Qu.:0.0000   1st Qu.: 6.0
##  L3:460                   Median :80    Median :1.0000   Median :10.0
##  L4:432                   Mean   :80    Mean   :0.7928   Mean   :11.3
##                           3rd Qu.:80    3rd Qu.:1.0000   3rd Qu.:15.0
##                           Max.   :80    Max.   :3.0000   Max.   :40.0
##
##  TrainingTimesLastYear YearsAtCompany   YearsInCurrentRole
##  Min.   :0.0           Min.   : 0.000   Min.   : 0.000
##  1st Qu.:2.0           1st Qu.: 3.000   1st Qu.: 2.000
##  Median :3.0           Median : 5.000   Median : 3.000
##  Mean   :2.8           Mean   : 7.026   Mean   : 4.233
##  3rd Qu.:3.0           3rd Qu.: 9.250   3rd Qu.: 7.000
##  Max.   :6.0           Max.   :40.000   Max.   :18.000
##
##  YearsSinceLastPromotion YearsWithCurrManager
##  Min.   : 0.000          Min.   : 0.000
##  1st Qu.: 0.000          1st Qu.: 2.000
##  Median : 1.000          Median : 3.000
##  Mean   : 2.189          Mean   : 4.122
##  3rd Qu.: 3.000          3rd Qu.: 7.000
##  Max.   :15.000          Max.   :17.000
##
```

```r
# Check the structure of the dataset
str(empdata)
```

```
## 'data.frame':    1472 obs. of  33 variables:
##  $ Age                     : int  38 49 37 33 37 32 59 30 38 36 ...
##  $ Attrition               : Factor w/ 2 levels "No","Yes": 1 1 2 1 2 1 1 1 2 1 ...
##  $ BusinessTravel          : Factor w/ 3 levels "Non-Travel","Travel_Frequently",..: 3 2 3 2 3 2 3 3
##  $ DailyRate               : int  1102 279 1373 1392 591 1005 1324 1358 216 1299 ...
##  $ Department              : Factor w/ 3 levels "Human Resources",..: 3 2 2 2 2 2 2 2 2 2 ...
##  $ DistanceFromHome        : int  1 8 2 3 2 2 3 24 23 27 ...
##  $ Education               : int  2 1 2 4 1 2 3 1 3 3 ...
##  $ EducationField          : Factor w/ 6 levels "Human Resources",..: 2 2 5 2 4 2 4 2 2 4 ...
##  $ EmployeeCount           : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ EmployeeNumber          : int  1 2 4 5 7 8 10 11 12 13 ...
##  $ EnvironmentSatisfaction : Factor w/ 4 levels "L1","L2","L3",..: 2 3 4 4 1 4 3 4 4 3 ...
##  $ Gender                  : Factor w/ 2 levels "Female","Male": 1 2 2 1 2 2 1 2 2 2 ...
##  $ HourlyRate              : int  94 61 92 56 40 79 81 67 44 94 ...
##  $ JobLevel                : Factor w/ 5 levels "Level1","Level2",..: 2 2 1 1 1 1 1 1 3 2 ...
##  $ JobRole                 : Factor w/ 9 levels "Healthcare Representative",..: 8 7 3 7 3 3 3 3 5 1
##  $ JobSatisfaction         : Factor w/ 4 levels "L1","L2","L3",..: 4 2 3 3 2 4 1 3 3 3 ...
##  $ MaritalStatus           : Factor w/ 3 levels "Divorced","Married",..: 3 2 3 2 2 3 2 1 3 2 ...
##  $ MonthlyIncome           : int  5993 5130 2090 2909 3468 3068 2670 2693 9526 5237 ...
##  $ MonthlyRate             : int  19479 24907 2396 23159 16632 11864 9964 13335 8787 16577 ...
##  $ NumCompaniesWorked      : int  8 1 6 1 9 0 4 1 0 6 ...
```

```
##  $ Over18                  : Factor w/ 1 level "Y": 1 1 1 1 1 1 1 1 1 1 ...
##  $ OverTime                : Factor w/ 2 levels "No","Yes": 2 1 2 2 1 1 2 1 2 1 ...
##  $ PercentSalaryHike       : int  11 23 15 11 12 13 20 22 21 13 ...
##  $ PerformanceRating       : Factor w/ 2 levels "L3","L4": 1 2 1 1 1 1 2 2 2 1 ...
##  $ RelationshipSatisfaction: Factor w/ 4 levels "L1","L2","L3",..: 1 4 2 3 4 3 1 2 2 2 ...
##  $ StandardHours           : int  80 80 80 80 80 80 80 80 80 80 ...
##  $ AvailableStocks         : int  0 1 0 0 1 0 3 1 0 2 ...
##  $ TotalWorkingYears       : int  8 10 7 8 6 8 12 1 10 17 ...
##  $ TrainingTimesLastYear   : int  0 3 3 3 3 2 3 2 2 3 ...
##  $ YearsAtCompany          : int  6 10 0 8 2 7 1 1 9 7 ...
##  $ YearsInCurrentRole      : int  4 7 0 7 2 7 0 0 7 7 ...
##  $ YearsSinceLastPromotion : int  0 1 0 3 2 3 0 0 1 7 ...
##  $ YearsWithCurrManager    : int  5 7 0 0 2 6 0 0 8 7 ...
```

redundant variables

```
# Remove redundant variables
empdata[c("EmployeeCount", "EmployeeNumber", "Over18", "StandardHours")] <- NULL
```

split data

```
# Set a seed
set.seed(10)

# Generate a vector named partition for data partitioning
partition = sample.split(empdata$Attrition, SplitRatio = 0.8)

# Create training set: training
training = subset(empdata, partition == TRUE)

# Create test set: test
test = subset(empdata, partition == FALSE)
```

feature selection

information.gain()

```
# information.gain(target~.,dataset)
```

```
# Use function information.gain to compute information gain values of the attributes
attr_weights <- information.gain(Attrition~. , empdata)

# Print weights
print(attr_weights)
```

```
##                           attr_importance
## Age                          2.915686e-02
```

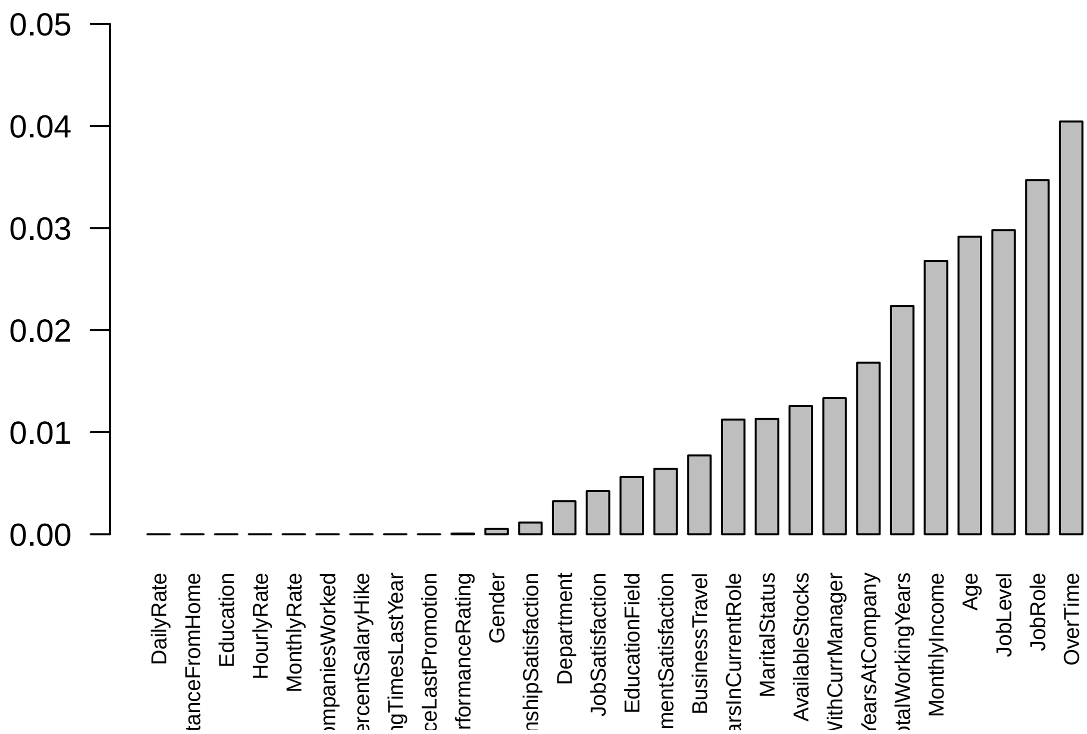
```
## BusinessTravel              7.728928e-03
## DailyRate                   0.000000e+00
## Department                  3.237261e-03
## DistanceFromHome            0.000000e+00
## Education                   0.000000e+00
## EducationField              5.607006e-03
## EnvironmentSatisfaction     6.421067e-03
## Gender                      5.283335e-04
## HourlyRate                  0.000000e+00
## JobLevel                    2.979136e-02
## JobRole                     3.470475e-02
## JobSatisfaction             4.224659e-03
## MaritalStatus               1.131948e-02
## MonthlyIncome               2.678926e-02
## MonthlyRate                 0.000000e+00
## NumCompaniesWorked          0.000000e+00
## OverTime                    4.043795e-02
## PercentSalaryHike           0.000000e+00
## PerformanceRating           7.588943e-05
## RelationshipSatisfaction    1.156400e-03
## AvailableStocks             1.255604e-02
## TotalWorkingYears           2.236349e-02
## TrainingTimesLastYear       0.000000e+00
## YearsAtCompany              1.681804e-02
## YearsInCurrentRole          1.124040e-02
## YearsSinceLastPromotion     0.000000e+00
## YearsWithCurrManager        1.333016e-02
```

sorting the result

order()

```
# Sort the weights. Use order() function
sorted_weights <- attr_weights[order(attr_weights$attr_importance), ,drop = FALSE]

# Plot the sorted weights
barplot(unlist(sorted_weights),
        names.arg = rownames(sorted_weights), las = "2", cex.names=0.7,
        ylim = c(0,0.05), space = 0.5)
```

Use order() function to sort the attributes with respect to their information gain values. Then, use barplot() function to illustrate the result.




```
# Filter features where the information gain is not zero
library(dplyr)
attr_weights %>% filter(attr_importance > 0)
```

```
##                          attr_importance
## Age                        2.915686e-02
## BusinessTravel             7.728928e-03
## Department                 3.237261e-03
## EducationField             5.607006e-03
## EnvironmentSatisfaction    6.421067e-03
## Gender                     5.283335e-04
## JobLevel                   2.979136e-02
## JobRole                    3.470475e-02
## JobSatisfaction            4.224659e-03
## MaritalStatus              1.131948e-02
## MonthlyIncome              2.678926e-02
## OverTime                   4.043795e-02
## PerformanceRating          7.588943e-05
## RelationshipSatisfaction   1.156400e-03
## AvailableStocks            1.255604e-02
## TotalWorkingYears          2.236349e-02
## YearsAtCompany             1.681804e-02
## YearsInCurrentRole         1.124040e-02
## YearsWithCurrManager       1.333016e-02
```

cutoff.k()

filter the most informative k attributes

cutoff.k() orders the attributes according to their information gain and returns the first k.

cutoff.k.percent(weights, k) selects k* 100% of attributes.

```
# cutoff.k(weights,k)
```

```
# Use cutoff.k() to find the most informative 19 attributes
filtered_attributes <- cutoff.k(attr_weights, 19)

# Print filtered attributes
print(filtered_attributes)
```

cutoff.biggest.diff(weights) selects a subset of attributes which are significantly better than others.

```
##  [1] "OverTime"                 "JobRole"
##  [3] "JobLevel"                 "Age"
##  [5] "MonthlyIncome"            "TotalWorkingYears"
##  [7] "YearsAtCompany"           "YearsWithCurrManager"
##  [9] "AvailableStocks"          "MaritalStatus"
## [11] "YearsInCurrentRole"       "BusinessTravel"
## [13] "EnvironmentSatisfaction"  "EducationField"
## [15] "JobSatisfaction"          "Department"
## [17] "RelationshipSatisfaction" "Gender"
## [19] "PerformanceRating"
```

```
# Use cutoff.biggest.diff() to a subset of attributes which are significantly better than other
cutoff.biggest.diff(attr_weights)
```

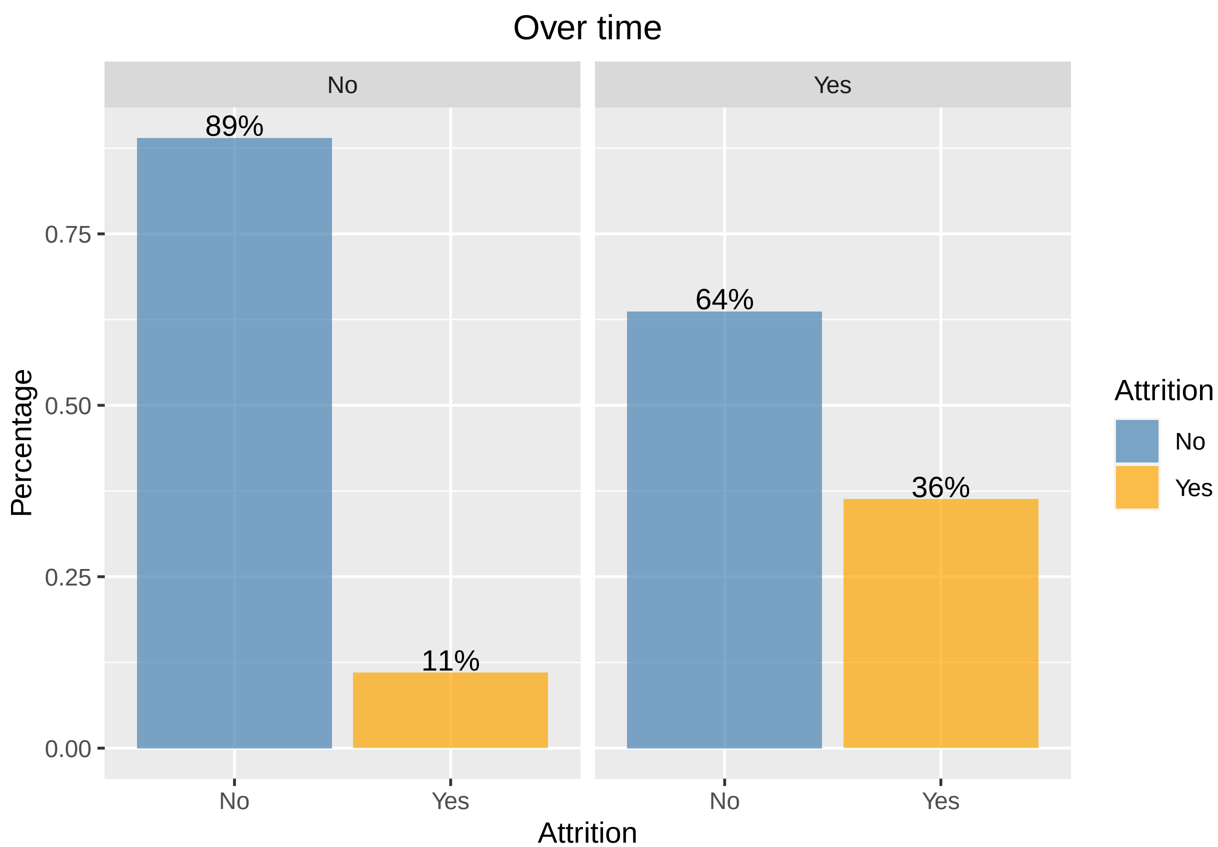
```
## [1] "OverTime"
```

ggplot

```
library(tidyverse)
ggplot(empdata,
      aes(x = Attrition, group = OverTime)) +
      geom_bar(aes(y = ..prop.., fill = factor(..x..)),
                   stat="count",
                   alpha = 0.7) +
      geom_text(aes(label = scales::percent(..prop..), y = ..prop.. ),
```

```
                  stat= "count",
                  vjust = -.1) +
      labs(y = "Percentage") +
      facet_grid(~OverTime) +
      scale_fill_manual("Attrition" ,values = c("steelblue","orange"), labels=c("No", "Yes")) +
      theme(plot.title = element_text(hjust = 0.5)) +
      ggtitle("Over time")
```

plot "Attrition" vs "OverTime"



rename categories

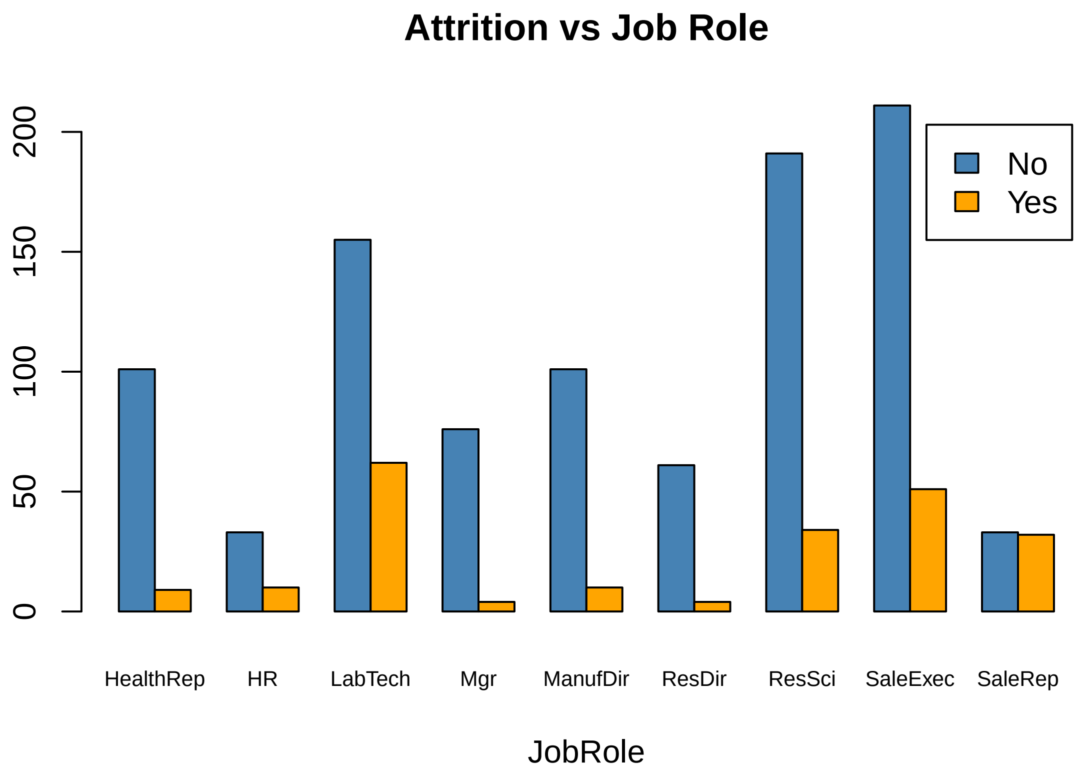
revalue()

```
# Revalue categories for the plot. Load 'plyr' package
library(plyr)
```

```
## ------------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## ------------------------------------------------------------------------------

##
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following object is masked from 'package:purrr':
##
##     compact
```

```
# Rename categories for illustration
training$JobRole <- revalue(training$JobRole,
                            c("Healthcare Representative" = "HealthRep",
                              "Human Resources" = "HR",
                              "Laboratory Technician" = "LabTech",
                              "Manager" = "Mgr",
                              "Manufacturing Director" = "ManufDir",
                              "Research Director" = "ResDir",
                              "Research Scientist" = "ResSci",
                              "Sales Executive" = "SaleExec",
                              "Sales Representative" = "SaleRep"))

barplotdata = table(training$Attrition, training$JobRole)

# Use barplot function to plot Attrition vs JobRole
barplot(barplotdata, main = "Attrition vs Job Role",
        xlab="JobRole",col=c("steelblue","orange"),
        legend=rownames(barplotdata), cex.names = 0.70, beside = TRUE)
```

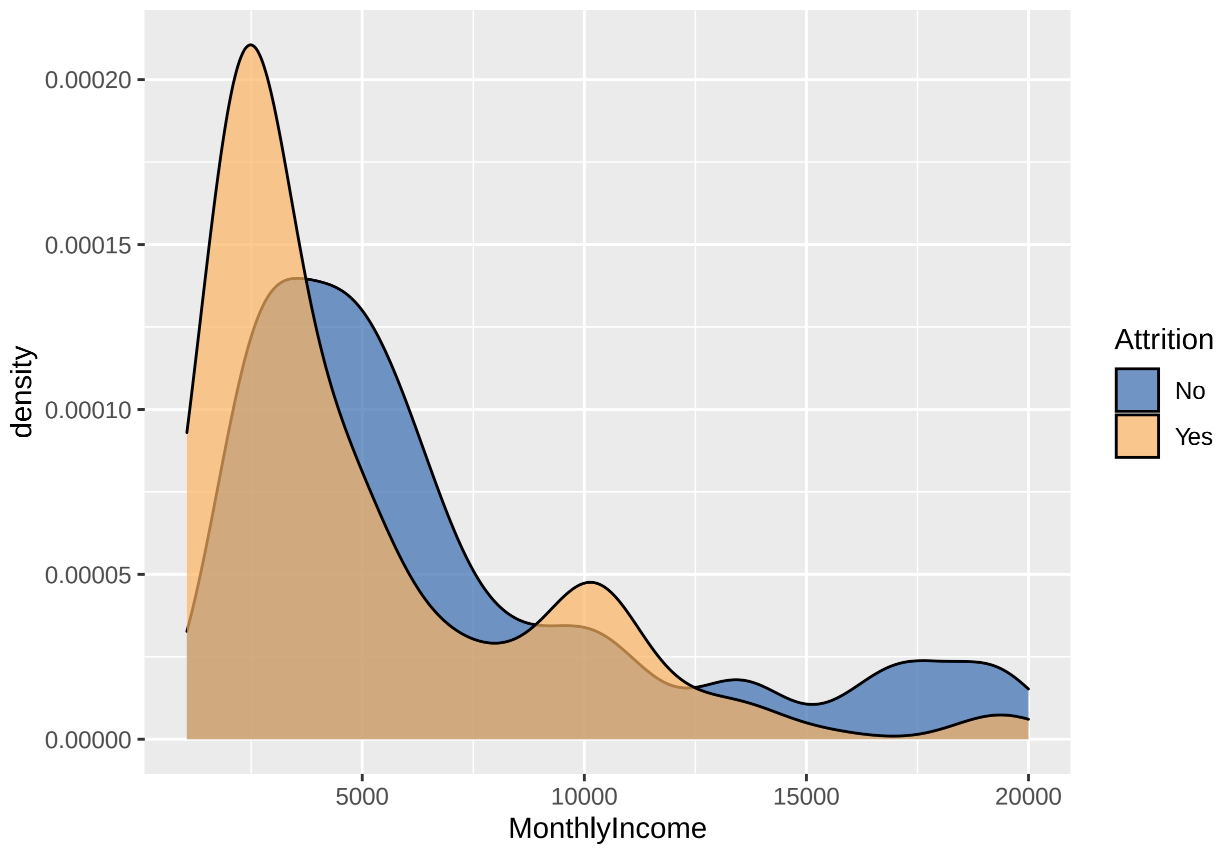


plot Attrition vs JobLevel

```
# Plot Attrition vs JobLevel
barplotdata2 = table(training$Attrition, training$JobLevel)


barplot(barplotdata2,
        main="Attrition vs JobLevel",
        xlab="Job Level", col=c("#386cb0","#fdb462"),
        legend=rownames(barplotdata), cex.names = 0.75, beside = TRUE)
```

plot Attrition vs Monthly Income

```
# Plot Attrition vs Monthly Income
ggplot(training, aes(x = MonthlyIncome, fill = Attrition)) +
  geom_density(alpha = 0.7) +
  scale_fill_manual(values = c("#386cb0","#fdb462"))
```



subset training set

```
# Select a subset of the dataset by using filtered_attributes
datamodelling <- training[filtered_attributes]
```

```
datamodelling["target"] <- training["Attrition"]
# or
datamodelling$target <- training$Attrition
```

Since filtered__attributes does not include `the target variable`, `Attrition` column is not present in our constructed data file. Adding it to the data file is needed for model building .